## Table of contents

Overview of this manual ........................................................................................................ 9  
Safety ........................................................................................................................................ 13  

1 Welcome to PickMaster PowerPac ........................................................................................... 15  
   1.1 Introduction ................................................................................................................... 15  
   1.2 PickMaster PowerPac terms ......................................................................................... 17  

2 Installation .................................................................................................................................. 19  
   2.1 PickMaster package ......................................................................................................... 19  
   2.2 System requirements ....................................................................................................... 21  
      2.2.1 Hardware and software requirements ...................................................................... 21  
      2.2.2 Ethernet switch ........................................................................................................ 23  
      2.2.3 Vision system .......................................................................................................... 24  
      2.2.4 Camera requirements ............................................................................................. 27  
   2.3 PickMaster PowerPac license ............................................................................................ 30  
   2.4 Self-signing certificate ..................................................................................................... 33  
      2.4.1 Certificate handling .................................................................................................. 33  
      2.4.2 Replacing PickMaster® Runtime default certificate with self-signing certificate ... 34  
      2.4.3 Generating self-signing certificate with OpenSSL .................................................. 36  
      2.4.4 Installing self-signing certificate ............................................................................. 38  
   2.5 PickMaster time synchronization service .......................................................................... 40  
   2.6 Software installation ......................................................................................................... 44  
      2.6.1 Installing RobotStudio ............................................................................................. 45  
      2.6.2 Installing PickMaster Twin Client .......................................................................... 46  
   2.7 Electrical connection ......................................................................................................... 48  
      2.7.1 PickMaster® Twin Hardware connection illustration ............................................... 48  
      2.7.2 Connecting cameras ............................................................................................... 49  
      2.7.3 Connecting I/O signals ............................................................................................ 52  
      2.7.4 Configuring networks .............................................................................................. 54  
      2.7.5 Setting up robot controller .................................................................................... 60  
      2.7.6 Optional robot and process configuration ............................................................... 61  
      2.7.7 Six axes robot configuration .................................................................................... 62  

3 Navigating PickMaster PowerPac ............................................................................................. 63  
   3.1 Main window ................................................................................................................... 63  
   3.2 Ribbon tab ....................................................................................................................... 65  
   3.3 Tree view browser ........................................................................................................... 72  
      3.3.1 Layout ..................................................................................................................... 72  
      3.3.2 Process ................................................................................................................... 81  
   3.4 Log view .......................................................................................................................... 83  
   3.5 Status view ...................................................................................................................... 84  

4 Working with PickMaster PowerPac .......................................................................................... 85  
   4.1 Overview ......................................................................................................................... 85  
   4.2 Frame relationship ......................................................................................................... 87  
      4.2.1 What is a coordinate system? ................................................................................. 87  
      4.2.2 Frame relationship in PickMaster® Twin ............................................................... 94  
   4.3 Setting up Solution with Layout and Process in virtual Runtime (VRT) ......................... 102  
      4.3.1 Solution .................................................................................................................. 102  
      4.3.2 Layout .................................................................................................................... 103  
         4.3.2.1 Controller .......................................................................................................... 103  
         4.3.2.2 Gripper .............................................................................................................. 106  
         4.3.2.3 Conveyor ......................................................................................................... 109  
         4.3.2.4 Camera ............................................................................................................ 111  
         4.3.2.5 I/O sensor ......................................................................................................... 113  

Application manual - PickMaster® Twin - PowerPac  5  
3HAC080435-001 Revision: F  
© Copyright 2023 ABB. All rights reserved.
Table of contents

4.3.2.6 External sensor ........................................................................................................115
4.3.2.7 Work area ................................................................................................................116
4.3.2.8 Indexed work area ....................................................................................................119
4.3.2.9 Position generator ...................................................................................................121
4.3.2.10 Calibration .............................................................................................................126
4.3.3 Process ..........................................................................................................................127
4.3.3.1 Item ..........................................................................................................................127
4.3.3.2 Container ................................................................................................................132
4.3.3.3 Flow ........................................................................................................................138
4.3.3.4 Recipe .......................................................................................................................141
4.3.3.5 Simulation .................................................................................................................161

4.4 Configuration in real Runtime (RRT) ...........................................................................168
4.4.1 Switching to real Runtime ..........................................................................................168
4.4.2 Configuring camera ......................................................................................................177
4.4.3 Configuring external sensor .........................................................................................181
4.4.4 Calibrating robot .........................................................................................................195
4.4.5 Calibrating linear conveyor ..........................................................................................196
4.4.5.1 Calibrating linear conveyor with DSQC 2000 ........................................................197
4.4.5.2 Calibrating linear conveyor with DSQC 377 ..........................................................220
4.4.6 Calibrating circular conveyor .......................................................................................227
4.4.6.1 Calibrating circular conveyor with DSQC 2000 .......................................................228
4.4.6.2 Calibrating circular conveyor with DSQC 377 .......................................................252
4.4.6.3 Type configuration for circular conveyor .................................................................258
4.4.7 Calibrating indexed work area ....................................................................................260
4.4.8 Verifying conveyor calibrations ....................................................................................274
4.4.9 Calibrating camera .......................................................................................................276
4.4.9.1 Showing live images .................................................................................................282
4.4.9.2 Detailed vision information .......................................................................................283
4.4.9.3 The image windows ..................................................................................................285
4.4.10 Adding vision model ...................................................................................................286
4.4.10.1 Vision modeling ........................................................................................................286
4.4.10.2 Configuring a geometric model with PatMax .........................................................289
4.4.10.3 Configuring blob models .........................................................................................297
4.4.10.4 Configuring inspection models ...............................................................................304
4.4.11 Starting production .....................................................................................................313
4.4.12 Managing the robot in production ..............................................................................315

4.5 Using color vision ............................................................................................................317

4.6 Working with products of varying height (2.5D vision) ..................................................326

4.7 Production with flow (Ghost Picking) ............................................................................333

5 RAPID reference .............................................................................................................337

5.1 Instructions .....................................................................................................................337
5.1.1 AckltmTgt - Acknowledge an item target ....................................................................337
5.1.2 FlushItmSrc - Flush an item source ...........................................................................339
5.1.3 GetltmTgt - Get the next item target .........................................................................340
5.1.4 NextltmTgtType - Get the type of the next item target ..............................................346
5.1.5 QStartltmSrc - Start queue in item source .................................................................348
5.1.6 QStopItmSrc - Stop queue in item source ...................................................................349
5.1.7 ResetFlowCount - Reset flow counter .......................................................................350
5.1.8 ResetMaxUsageTime - Reset max measured usage time .........................................351
5.1.9 UseReachableTargets - Use reachable targets ...........................................................352

5.2 Functions ........................................................................................................................355
5.2.1 GetMaxUsageTime - Get max measured usage time .................................................355
5.2.2 GetQueueLevel - Get queue level ..............................................................................356
5.2.3 GetQueueTopLevel - Get queue top level .................................................................358
5.2.4 GetFlowCount - Get number of passed items ............................................................359

5.3 Data types .......................................................................................................................360
5.3.1 itmtgt - Item target data ............................................................................................360
5.3.2 selectiondata - Selection data ....................................................................................363
Table of contents

5.3.3 sortdata - Sort data ........................................................................................................ 366
5.4 RAPID program .................................................................................................................. 367
   5.4.1 RAPID programs ........................................................................................................ 367
   5.4.2 Variables .................................................................................................................... 372
   5.4.3 Routines .................................................................................................................... 375
5.5 Program examples ............................................................................................................. 379
   5.5.1 Example: Mixing one pick work area and two place work areas ................................ 379
   5.5.2 Example: Mixing two pick work areas and one place work area .............................. 380
   5.5.3 Example: Mixing with one pick and one place work area ....................................... 381
   5.5.4 Example: Double pick single place .......................................................................... 382
   5.5.5 Example: Placing a predefined pattern on indexed work area .................................. 385
   5.5.6 Example: Selecting item depending on clearance zone ........................................... 387
   5.5.7 Example: Sorting in negative y-direction ................................................................... 390
   5.5.8 Example: Indexed work area with predefined position ............................................ 393
   5.5.9 Example: Automatically generating new positions to indexed work area ................. 394
   5.5.10 Example: Item buffer ............................................................................................. 395

6 Troubleshooting ................................................................................................................ 399
   6.1 Introduction to troubleshooting ..................................................................................... 399
   6.2 Safety during troubleshooting ....................................................................................... 400
   6.3 Administering the log .................................................................................................... 401
   6.4 Fault symptoms or errors .............................................................................................. 402
       6.4.1 Warnings 4326 - 4329 .......................................................................................... 402
       6.4.2 The camera does not take pictures ........................................................................ 406
       6.4.3 Robot does not move ............................................................................................ 407
       6.4.4 Bad or varying position accuracy .......................................................................... 408
       6.4.5 Positions are used twice ....................................................................................... 409
       6.4.6 Problem with camera resolution in PickMaster .................................................... 410
       6.4.7 The Image Dialog cannot show ............................................................................ 411
       6.4.8 Robot fails to grip item when using camera on a circular conveyor ....................... 415
       6.4.9 Robot fails to start when clicking on Start button after Arm check point limit error for an indexed work area ........................................................................... 417
   6.5 Error codes .................................................................................................................. 418

7 Spare parts ....................................................................................................................... 431
   7.1 Licenses ......................................................................................................................... 432
   7.2 Camera parts ................................................................................................................ 433
   7.3 USB dongle parts ......................................................................................................... 437
   7.4 GigE Network card parts ............................................................................................. 438

8 Circuit diagram .................................................................................................................. 439
   8.1 Circuit diagrams .......................................................................................................... 439

Index ................................................................................................................................. 441
Overview of this manual

About this manual
This manual contains instructions for installation, configuration, and operation of PickMaster PowerPac.

Note
All safety information for working with the controller is described in the product manual for the controller.

Usage
This manual should be used during installation, configuration, and maintenance of a PickMaster Twin system.

PickMaster PowerPac is intended for use as an engineering tool on a portable laptop PC for offline use and online connection to a host computer in the installation for commissioning purposes. PickMaster PowerPac is not intended for use on the host computer under production conditions.

Who should read this manual?
This manual is intended for:
- Installation personnel
- Programmers
- Integrators
- Operators

Prerequisites
Any maintenance/repair/installation personnel working with an ABB robot must be trained by ABB and have the required knowledge of mechanical and electrical installation/repair/maintenance work.
Cybersecurity

This product is designed to be connected to and to communicate information and data via a network interface. It is your sole responsibility to provide, and continuously ensure, a secure connection between the product and to your network or any other network (as the case may be).

You shall establish and maintain any appropriate measures (such as, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of anti-virus programs, etc) to protect the product, the network, its system and the interface against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Ltd and its entities are not liable for damage and/or loss related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

Note

Only the qualified personnel can write or modify the script files.
It is the responsibility of the writer to make sure that the cell is safety when running with the script files.

The PickMaster PowerPac will use the following ports:

- 50000
- 6001

References

Tip

All documents can be found via myABB Business Portal, www.abb.com/myABB.

OmniCore

<table>
<thead>
<tr>
<th>Reference</th>
<th>Document ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application manual - PickMaster Twin - Operator</td>
<td>3HAC069977-001</td>
</tr>
<tr>
<td>Product specification - PickMaster® Twin</td>
<td>3HAC073650-001</td>
</tr>
<tr>
<td>Circuit diagram - PickMaster Twin</td>
<td>3HAC024480-020</td>
</tr>
<tr>
<td>Safety manual for robot - Manipulator and IRC5 or OmniCore controller</td>
<td>3HAC031045-001</td>
</tr>
<tr>
<td>Product manual - OmniCore C30</td>
<td>3HAC060860-001</td>
</tr>
<tr>
<td>Operating manual - OmniCore</td>
<td>3HAC065036-001</td>
</tr>
<tr>
<td>Application manual - Controller software OmniCore</td>
<td>3HAC066554-001</td>
</tr>
<tr>
<td>Technical reference manual - Lubrication in gearboxes</td>
<td>3HAC042927-001</td>
</tr>
<tr>
<td>Technical reference manual - System parameters</td>
<td>3HAC065041-001</td>
</tr>
</tbody>
</table>

\[This manual contains all safety instructions from the product manuals for the manipulators and the controllers.\]
Overview of this manual

Continued

IRC5

<table>
<thead>
<tr>
<th>Reference</th>
<th>Document ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application manual - PickMaster Twin - Operator</td>
<td>3HAC069977-001</td>
</tr>
<tr>
<td>Product specification - PickMaster® Twin</td>
<td>3HAC073650-001</td>
</tr>
<tr>
<td>Circuit diagram - PickMaster Twin</td>
<td>3HAC024480-020</td>
</tr>
<tr>
<td>Operating manual - RobotStudio</td>
<td>3HAC032104-001</td>
</tr>
<tr>
<td>Application manual - Conveyor tracking</td>
<td>3HAC050991-001</td>
</tr>
<tr>
<td>Product manual - IRC5</td>
<td>3HAC047136-001</td>
</tr>
<tr>
<td>Product manual - IRC5 Panel Mounted Controller</td>
<td>3HAC027707-001</td>
</tr>
<tr>
<td>Operating manual - IRC5 with FlexPendant</td>
<td>3HAC050941-001</td>
</tr>
<tr>
<td>Operating manual - Troubleshooting IRC5</td>
<td>3HAC020738-001</td>
</tr>
<tr>
<td>Technical reference manual - RAPID Instructions, Functions and Data types</td>
<td>3HAC050917-001</td>
</tr>
<tr>
<td>Technical reference manual - RAPID Overview</td>
<td>3HAC050947-001</td>
</tr>
<tr>
<td>Technical reference manual - System parameters</td>
<td>3HAC050948-001</td>
</tr>
</tbody>
</table>

External references

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognex Ethernet Camera Tool</td>
<td>For configuring camera networks.</td>
</tr>
<tr>
<td>Gigabit Ethernet Performance Driver</td>
<td>For camera communication.</td>
</tr>
<tr>
<td>aca1440-73gc</td>
<td>Information about Basler Ace Gigabit Ethernet cameras and the switch for Gigabit Ethernet cameras.</td>
</tr>
<tr>
<td></td>
<td>Note This camera uses ABB customized firmware, which needs to be purchased from ABB.</td>
</tr>
<tr>
<td>sca1300-32gc</td>
<td>Information about Basler Scout Gigabit Ethernet cameras and the switch for Gigabit Ethernet cameras.</td>
</tr>
<tr>
<td></td>
<td>Note This camera uses ABB customized firmware, which needs to be purchased from ABB.</td>
</tr>
<tr>
<td>CognexPCConfigGuide</td>
<td>Detailed information about PC requirements for the vision system.</td>
</tr>
</tbody>
</table>

Revisions

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>First edition.</td>
</tr>
</tbody>
</table>

Continues on next page
<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>
| B        | Released with PickMaster® Twin 2.0.1.  
  • Added Histogram and Caliper function.  
  • Added gripper related with TCP0 function.  
  • Added troubleshooting for Image Dialog cannot show in section The Image Dialog cannot show on page 411.  
  • Minor corrections. |
| C        | Released with PickMaster® Twin 2.1.  
  • Supported multiple languages.  
  • Updated trigger distance function.  
  • Added adjust base frame function.  
  • Added user script function.  
  • Added PMRT login function when connecting to PMRT.  
  • Updated information for circular conveyor calibration.  
  • Added copy function for Item, Container and Flow.  
  • Minor corrections. |
| D        | Released with PickMaster® Twin 2.1.1.  
  • Minor corrections. |
| E        | Released with PickMaster® Twin 2.2.  
  • Supported external sensor function.  
  • Added Self-signing certificate.  
  • Supported multiple languages.  
  • Minor corrections. |
| F        | Released with PickMaster® Twin 2.3.  
  • Supported Runtime file transfer function.  
  • Updated time synchronization service.  
  • Updated PickMaster PowerPac license.  
  • Added 2.5D vision.  
  • Minor corrections. |
Safety

Safety of personnel

A robot is heavy and extremely powerful regardless of its speed. A pause or long stop in movement can be followed by a fast hazardous movement. Even if a pattern of movement is predicted, a change in operation can be triggered by an external signal resulting in an unexpected movement. Therefore, it is important that all safety regulations are followed when entering safeguarded space.

Safety regulations

Before beginning work with the robot, make sure you are familiar with the safety regulations described in the manual *Safety manual for robot - Manipulator and IRC5 or OmniCore controller*.

When using PickMaster® Twin products

- When using with PickMaster® Twin products, it is the user’s responsibility to adhere to the relevant standards and safety directives. In addition, the application manuals for proper use must be observed.
- Only personnel with appropriate training and required knowledge are allowed to use PickMaster® Twin products.
- The integrator installing the PickMaster® Twin is responsible for the safety.
- Wherever possible, the auto mode of operation shall be performed with all persons outside the safeguarded space.
- An emergency stop must also be available to make sure the emergency stop function is enabled.
- PickMaster® Twin only provides Operational Stop (Program Stop). The integrator shall make sure that proper Normal Stop (machinery stop) is configured correctly in the system.
- Make sure the hazardous situation that resulted in the emergency stop condition no longer exists. Release the emergency stop button manually to remove the emergency stop condition.
- Stops for the machine is the responsibility of the integrator and must be addressed according to local legislation.
- The integrator is responsible to conduct a risk assessment of the final application.
- Sensitive body parts, such as the eyes and the larynx, must be protected by personal protective equipment (PPE).
- Protective measures should be the precondition when using PickMaster® Twin products. PickMaster® Twin does not guarantee the robot targets are always in safe zone. It is integrator’s responsibility to take protection measures, like using safe-move or setting proper robot work range etc.

Continues on next page
Safety

Continued

• Safety related status and operations shall be handled on the controller and by safety rated systems. PickMaster® Twin status information shall not be used as input for safety related information and operations.

• Protective measures should be the precondition when install/adjust/replace hardware parts, for example, the camera.

• The stop functions in PickMaster® Twin can never be used to replace A-stop/E-stop or any other safety related stops.
1 Welcome to PickMaster PowerPac

1.1 Introduction

About PickMaster® Twin

PickMaster® Twin is an application product designed for vision based high speed picking of random flow products on the fly. PickMaster® Twin supports ease-of-use configuration, simulation and operation of a big variation of smaller or larger line layouts composed of a multitude of robots, cameras, conveyors and fixed work areas. It is a production system that comprises all steps in the life cycle of a picking installation from proposal, engineering, commissioning, operation to maintenance and support.

PickMaster PowerPac can be customized for some of the following special needs:

- With the integrated vision system it can be used for full random operation on a continuously moving conveyors and for absolute accurate positioning on indexed feeders or trays.
- Without vision recognition it can be used as a tool for the efficient production with guided product flows on multiple conveyors.
- For efficient quality inspection and product categorization alone or together with the position recognition.

PickMaster® Twin is a modular product for controlling ABB robots in picking applications through the robot controller. It is configurable to perform pick and place operations of items. A vision system is used to find randomly placed items on conveyor belts or indexing static work areas. PickMaster PowerPac is the engineering software aimed at configuring and validating the application in offline simulation with a virtual system and in online mode directly connected to the real installation. It uses comprehensive graphical interfaces to configure powerful applications, where it can control multiple robots picking and placing sensor-detected items on different conveyor belts.

PickMaster® Twin comprises the following modules:

PickMaster® PowerPac

Ease of Use software for offline and online configuration and commissioning in a visual 3D environment, powered by RobotStudio™.
1 Welcome to PickMaster PowerPac

1.1 Introduction

Continued

PickMaster® Operator

State-of-the-art user interface for operating PickMaster on the shop floor, built on ABB’s Ability™ Zenon data management software.

PickMaster® Runtime

Efficient runtime operation software for orchestrating the coordination of the packaging process for a multitude of robots and conveyors including integrated vision software for precise robot guidance and quality inspection.

- Virtual Runtime: running the PickMaster process in a simulated virtual environment on a client system connected to virtual robot controllers.
- Real Runtime: running the PickMaster process in the real production installation on the Host computer connected to real robot controllers.

The following illustration is showing an installation example with 10 robots, 4 cameras and 3 conveyors.

Note

PickMaster® Twin is delivered with different hardware configurations. For more information, see Product specification - PickMaster® Twin.

About PickMaster PowerPac

This manual describes how to install and use PickMaster PowerPac as the engineering software for two modes:

- Configuring and validating the application in offline simulation with a virtual system.
- Commissioning in online mode directly connected to the real installation.

This manual also describes the components of the real system, their installation, configuration and calibration.
1.2 PickMaster PowerPac terms

About these terms

Some words have a specific meaning when used in this manual. Definitions of these words in this manual are listed below. Some of the terms are put in their context when describing a picking and placing process.

Term list

Words that have italic font style in the definition column are included in the term list and have their own definitions.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PickMaster PowerPac</td>
<td>The market name of PickMaster PC engineering software that is used for simulating and commissioning picking lines with virtual and real Runtime.</td>
</tr>
<tr>
<td>PickMaster Operator</td>
<td>The market name of PickMaster production operator interface software that is used for running PickMaster applications in production. PickMaster Operator can read and write to solutions generated by PickMaster PowerPac. It has access to real Runtime.</td>
</tr>
<tr>
<td>PickMaster Virtual Runtime</td>
<td>The core engine that orchestrates all the calculation of virtual pick and place operation in simulations.</td>
</tr>
<tr>
<td>PickMaster Real Runtime</td>
<td>The core engine that orchestrates all the calculation of pick and place operation in real product. Runtime communicates with cameras and the robots. It’s also called as Runtime.</td>
</tr>
<tr>
<td>PickMaster Twin Client</td>
<td>The installation package which contains PickMaster PowerPac, PickMaster virtual Runtime and PickMaster real Runtime.</td>
</tr>
<tr>
<td>PickMaster Twin Host</td>
<td>The installation package which contains PickMaster Operator and PickMaster Runtime.</td>
</tr>
<tr>
<td>Solution</td>
<td>Format for storing a PickMaster Twin configuration result.</td>
</tr>
<tr>
<td>Recipe</td>
<td>Format and a collection of parameters regarding the process of Pick and Place for storing the process to be executed in a station.</td>
</tr>
<tr>
<td>Layout</td>
<td>Description of static objects in a PickMaster installation, for example robots, work areas.</td>
</tr>
<tr>
<td>Process</td>
<td>Description of a PickMaster picking process and all items, containers, flow and recipes.</td>
</tr>
<tr>
<td>Work area</td>
<td>A defined picking and placing area for the robots.</td>
</tr>
<tr>
<td>Item</td>
<td>The generic term for a specific object to be picked or placed in a PickMaster PowerPac application.</td>
</tr>
<tr>
<td>Container</td>
<td>Defines a shape that can set specific patterns and what items to use for each position in the patterns.</td>
</tr>
<tr>
<td>Position generator</td>
<td>Defines the sensor configuration on the conveyor and indexed work area.</td>
</tr>
<tr>
<td>Emulation</td>
<td>An activity of imitating the behavior of real cell or line and display the activity on screen.</td>
</tr>
<tr>
<td>Ghost picking</td>
<td>A kind of dry run, when production uses recorded virtual items to pick, thus no real item to pick.</td>
</tr>
</tbody>
</table>

Continues on next page
## Welcome to PickMaster PowerPac

### 1.2 PickMaster PowerPac terms

*Continued*

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline Simulation</td>
<td>Simulation process when connected to the virtual robot.</td>
</tr>
</tbody>
</table>
2 Installation

2.1 PickMaster package

Concepts of using PickMaster Twin

PickMaster PowerPac is designed to be installed on a laptop computer that can host solutions for many different installations that can be connected for commissioning, new recipe introduction, maintenance and servicing purposes to several physical installations, where each one of those have their own permanent host computer.

There are two software installation packages: PickMaster Twin Client for the portable engineering system and PickMaster Twin Host for the permanent factory system.

The Client installation does not require any physical equipment installations. All physical component installations, configurations and calibrations are done on the Host system.

PickMaster Twin Client

The installations package for PickMaster Twin offline configuration, simulation and testing is named as PickMaster Twin Client. It installs the following softwares:

- PickMaster PowerPac
- PickMaster Virtual Runtime
- PickMaster Real Runtime

Note

This package is only intended for engineering and not for the final factory production installation. The ability to switch to real Runtime on the same computer is only intended for test purposes and it can be used for creating and editing vision models offline. A vision demo dongle can be used for this purpose.

Software Installation Package

Registered ABB customers can download the latest version of the PickMaster Twin Client and the user documentation for PickMaster PowerPac from the ABB download center.

Tip

The download center address is https://new.abb.com/products/robotics/application-software/pickmaster.

Note

The PickMaster software is available in 64-bit version.

Continues on next page
2 Installation

2.1 PickMaster package

Continued

When the PickMaster Twin Client is installed successfully, the user documentation for PickMaster PowerPac and the calibration papers are available in the installation folder *Documentation*.

**Note**

Any old version of PickMaster PowerPac must be uninstalled before installing a newer version of PickMaster PowerPac.
2.2 System requirements

2.2.1 Hardware and software requirements

Hardware and software requirements for PickMaster Twin Client

Hardware requirements

Following are the hardware requirements:

- A log on account with administrator rights on the computer.
- CPU: 2.0 GHz or faster processor. Multicore processor is recommended.
- Memory: 8 GB RAM is the minimum requirement if running Windows 64bit edition. 16 GB or more is recommended if working with vision or heavy CAD models.
- Free disk space: 10+ GB free space, solid state drive (SSD) recommended.
- Graphics card: High-performance, DirectX 11 compatible, gaming graphics card from any of the leading vendors. For the Advanced lightning mode Direct3D feature level 10_1 or higher is required.
- Display settings: 1920 x 1080 pixels or higher resolution is recommended.
- Mouse: Three-button mouse
- If robot movement can be initiated from an external control panel then an emergency stop must also be available.

Note

When running the software, close other software that consumes a lot of memory, otherwise it will affect the software normal use.

Software requirements

Following are the software requirements:

- Windows 10 (64 bit).
- Acrobat reader
- RobotStudio 2023.3
- Omnicore with RobotWare 7.12.
- IRC5 with RobotWare 6.15.04

Hardware and software requirements for PickMaster Twin Host

Recommended hardware

- Windows 10 (64 bit) IPC, 2GHz, 500 GBit SSD, 8 GBit RAM
- Memory: 8 GB RAM is the minimum requirement if running Windows 64bit edition. 16 GB or more is recommended if working with vision or heavy CAD models.
- Recommended 23 inches 1920x1080 multi-touch screen
- Minimum two USB slots, one Ethernet port and one free PCI Express slot for a 168 mm x 110 mm size PCIE card

Continues on next page
2 Installation

2.2.1 Hardware and software requirements

Continued

- Ethernet switch (robot network)

Software requirements

- Microsoft Windows 10, 64 bit (Home, Pro, Enterprise, Education, IoT, x64 versions) for touch panel
- RobotStudio 2023.3
- Omnicore with RobotWare 7.12.
- IRC5 with RobotWare 6.15.04
2.2.2 Ethernet switch

Overview

An Ethernet switch is used to connect the PC with multiple robot controllers. It is recommended to use an industrial switch with a communication speed of 1000 Mbit/s or higher.
2.2.3 Vision system

Overview

PickMaster PowerPac can acquire images and generate targets by using cameras that communicate over Ethernet. An Ethernet network (network interface card, cables, switches) is used for communication between the cameras and the Runtime PC. Trigger/Strobe and power voltage is connected to a Hirose 12-pin/6-pin connector on the camera housing. Preferably the power voltage to the Ethernet camera is supplied from a separate source that is independent of the robot controller.

Vision system requirements

The supported network card for Ethernet camera communication is GigE network card DSQC1083 (3HAC078753-001). Other network interface cards can work, but have not been tested.

A Cognex USB license is required for the Gigabit Ethernet vision system. The USB stick must be connected when Runtime is running.

The maximum number of cameras that can be used is ten.

Insert the vision network card in a free compatible PCI-express slot (PCI-express x4, x8, or x16).

Color vision

Color vision is available as a standard function and has the following features:

- connectivity for color cameras
- white balance calibration
- color filter configuration

Note

This allows you to define color filters that will run as a prestep to PatMax and Blob. The filter is available in Standalone, alignment, and sub inspection models.

For the validated cameras, see Validated cameras on page 49.

Trigger strobe cables

There are 4 types of cable used. Trigger strobe connections for those are as below.

<table>
<thead>
<tr>
<th>Cable</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New replacement cable</td>
<td>2000034085</td>
<td>Power-I/O PLC+ Cable HRS 12p/open, 10m</td>
</tr>
<tr>
<td>Old Scout cable (10 m)</td>
<td>2000026632</td>
<td>Power-I/O Cable, HRS 12p, open, 10 m</td>
</tr>
<tr>
<td>Old Scout cable (10 m)</td>
<td>2000022909</td>
<td>Power-I/O Cable, HRS 12p/open, 10 m</td>
</tr>
<tr>
<td>Ace camera cable</td>
<td>2000034084</td>
<td>Power-I/O PLC+ Cable 6p/open, , 10 m</td>
</tr>
</tbody>
</table>

Continues on next page
Trigger strobe connection for 2000034085 (Basler camera)

The following table describes the physical interface for trigger/strobe/power connection to the Basler Scout camera.

For further details about how to connect the camera, see the circuit diagram.

<table>
<thead>
<tr>
<th>Function Description</th>
<th>Wire Pair</th>
<th>Pin Number</th>
<th>Wire Color</th>
<th>Scout GigE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0V(CamPower-)</td>
<td>1</td>
<td>1</td>
<td>White</td>
<td>Camera ground</td>
<td>0V(CamPower-)</td>
</tr>
<tr>
<td>0V(CamPower-)</td>
<td>1</td>
<td>2</td>
<td>Green</td>
<td>Camera ground</td>
<td>0V(CamPower-)</td>
</tr>
<tr>
<td>Trigger</td>
<td>2</td>
<td>3</td>
<td>Pink</td>
<td>Opto in 1</td>
<td>Trigger</td>
</tr>
<tr>
<td>0V (Cam I/O-)</td>
<td>2</td>
<td>(5)</td>
<td>Grey</td>
<td>Opto in 1 ground</td>
<td>0V (Cam I/O-)</td>
</tr>
<tr>
<td>Not used</td>
<td>3</td>
<td>4</td>
<td>Red</td>
<td>Opto in 2</td>
<td>Not used</td>
</tr>
<tr>
<td>Not used</td>
<td>3</td>
<td>(5)</td>
<td>Blue</td>
<td>Opto in 2 ground</td>
<td>Not used</td>
</tr>
<tr>
<td>0V (CamPower-)</td>
<td>4</td>
<td>6</td>
<td>Violet</td>
<td>Opto out 1</td>
<td>Strobe</td>
</tr>
<tr>
<td>Opto out 1 VCC</td>
<td>4</td>
<td>(10)</td>
<td>Black</td>
<td>Opto out 1 VCC</td>
<td>24V (Cam I/O +)</td>
</tr>
<tr>
<td>Opto out 2</td>
<td>5</td>
<td>7</td>
<td>Red/Blue</td>
<td>Opto out 2</td>
<td>Not used</td>
</tr>
<tr>
<td>Opto out 2 VCC</td>
<td>5</td>
<td>(10)</td>
<td>Grey/Pink</td>
<td>Opto out 2 VCC</td>
<td>Not used</td>
</tr>
<tr>
<td>Camera VCC</td>
<td>6</td>
<td>8</td>
<td>Brown</td>
<td>Camera VCC</td>
<td>24V (CamPower +)</td>
</tr>
<tr>
<td>Camera VCC</td>
<td>6</td>
<td>9</td>
<td>Yellow</td>
<td>Camera VCC</td>
<td>24V (CamPower +)</td>
</tr>
<tr>
<td>Opto Out 3 VCC</td>
<td>7</td>
<td>(10)</td>
<td>White/Green</td>
<td>Opto Out 3 VCC</td>
<td>Not used</td>
</tr>
<tr>
<td>Opto Out 3 VCC</td>
<td>7</td>
<td>11</td>
<td>Brown/Green</td>
<td>Opto Out 3</td>
<td>Not used</td>
</tr>
<tr>
<td>Opto Out 4 VCC</td>
<td>8</td>
<td>(10)</td>
<td>White/Yellow</td>
<td>Opto Out 4 VCC</td>
<td>Not used</td>
</tr>
<tr>
<td>Opto Out 4 VCC</td>
<td>8</td>
<td>12</td>
<td>Yellow/Brown</td>
<td>Opto Out 4</td>
<td>Not used</td>
</tr>
</tbody>
</table>

i 0/24V for powering the camera. Preferably supplied from a source that remains turned on even if the robot controller is shut down.

ii Input signal that orders the camera to acquire an image.

iii Pin number inside parenthesis "(X)" means that the wire is connected internally to pin "X"

iv 0/24V for the I/O system of the camera.

v Output signal indicating that the camera has acquired an image.

Old Trigger strobe connection for 2000026632 and 2000022909 (Basler camera)

<table>
<thead>
<tr>
<th>Function Description</th>
<th>Pin Number</th>
<th>Wire Color</th>
<th>scout GigE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera Power Ground</td>
<td>1</td>
<td>White</td>
<td>Camera Power Ground</td>
<td>0V(CamPower-)</td>
</tr>
<tr>
<td>Camera Power Ground</td>
<td>2</td>
<td>Green</td>
<td>Camera Power Ground</td>
<td>0V(CamPower-)</td>
</tr>
<tr>
<td>I/O Input 1</td>
<td>3</td>
<td>Blue</td>
<td>I/O Input 1</td>
<td>Trigger</td>
</tr>
<tr>
<td>I/O Input 2</td>
<td>4</td>
<td>Red</td>
<td>I/O Input 2</td>
<td>Not used</td>
</tr>
<tr>
<td>I/O Input Ground</td>
<td>5</td>
<td>Gray</td>
<td>I/O Input Ground</td>
<td>0V (Cam I/O-)</td>
</tr>
<tr>
<td>I/O Output 1</td>
<td>6</td>
<td>Black</td>
<td>I/O Output 1</td>
<td>Strobe</td>
</tr>
<tr>
<td>I/O Output 2</td>
<td>7</td>
<td>Violet</td>
<td>I/O Output 2</td>
<td>Not used</td>
</tr>
<tr>
<td>Camera Power VCC</td>
<td>8</td>
<td>Brown</td>
<td>Camera Power VCC</td>
<td>24V (CamPower +)</td>
</tr>
<tr>
<td>Camera Power VCC</td>
<td>9</td>
<td>Yellow</td>
<td>Camera Power VCC</td>
<td>24V (CamPower +)</td>
</tr>
<tr>
<td>I/O Output VCC</td>
<td>10</td>
<td>Pink</td>
<td>I/O Output VCC</td>
<td>24V (Cam I/O +)</td>
</tr>
<tr>
<td>I/O Output 3</td>
<td>11</td>
<td>Gray/Pink</td>
<td>I/O Output 3</td>
<td>Not used</td>
</tr>
</tbody>
</table>
2 Installation

2.2.3 Vision system

Continued

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Wire Color</th>
<th>scout GigE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Red/Blue</td>
<td>I/O Output 4</td>
<td>Not used</td>
</tr>
</tbody>
</table>

Trigger strobe connection for 2000034084 (Ace camera)

The following table describes the physical interface for trigger/strobe/power connection to the ace camera.

Power-I/O Cable HRS 6p/open, twisted, 10 m - IOs / Power Cables Cable for power supply and trigger of opto coupled I/Os of Basler ace GigE cameras at a length of 10 meters.

The cable has an HRS 6-pin connector on the camera side. The other end is open so that the cable can be shortened to match individual requirements.

Wiring information:

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Wire Color</th>
<th>Ace GigE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Brown</td>
<td>Camera Power</td>
<td>24V (CamPower +)</td>
</tr>
<tr>
<td>2</td>
<td>Pink</td>
<td>Opto-isolated IN (Line1)</td>
<td>Trigger</td>
</tr>
<tr>
<td>3</td>
<td>Green</td>
<td>Not connected</td>
<td>0V (Cam I/O-)</td>
</tr>
<tr>
<td>4</td>
<td>Yellow</td>
<td>Opto-isolated OUT (Out1)</td>
<td>Not used</td>
</tr>
<tr>
<td>5</td>
<td>Gray</td>
<td>Opto-isolated I/O Ground</td>
<td>0V (CamPower-)</td>
</tr>
<tr>
<td>6</td>
<td>White</td>
<td>Camera Power Ground</td>
<td>0V (CamPower-)</td>
</tr>
</tbody>
</table>

Note

There is no strobe output on ace camera, so we need to have a jumper between TrigOut and SYNCIN
2.2.4 Camera requirements

Mounting
The cameras must be mounted in a very stable way to avoid vibration and other dynamic movement. The cameras can be mounted in any orientation to the image area.

Lighting
Even lighting of the image area is very important to obtain reliable results.

Other camera requirements
A PickMaster camera needs to be of type progressive scan (non-interlaced) as it is used to record images of objects on a moving belt.

A PickMaster camera also needs to support electronic shutter control. With this feature it is possible to set the exposure from PickMaster PowerPac, otherwise the exposure time must be manually set on the camera.

Camera configuration
Some cameras will need manual configuration to fulfill the above conditions. For detailed information about camera settings, see Cognex manual and PickMaster Release Notes.

For specific information about Basler Gigabit Ethernet cameras, see References on page 10.

Recommendation for lenses
When planning a cell it is important to choose a suitable camera/lens setup that gives an appropriate field of view (FOV).

The FOV of a camera is determined by three factors:

- The distance between the camera and the scene.
- The focal length of the lens.
- The size of the camera's sensor chip (normally specified as the distance of the diagonal of the chip, expressed in inches).
The graphic below shows the geometry of the optical setup.

To select a suitable lens, measure the distance between the camera and the items (D), and the size of the image area (W*H).

To calculate the appropriate focal length of the lens:
- If the height of the image area is most important: \( f = \frac{D}{W} \times C_w \)
- If the length of the image area is most important: \( f = \frac{D}{H} \times C_h \)

The table below lists the width and height of some common sensor chip sizes, expressed in millimeters.

<table>
<thead>
<tr>
<th>Sensor chip size (inch)</th>
<th>( C_h ) (mm)</th>
<th>( C_w ) (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4&quot;</td>
<td>2.4</td>
<td>3.2</td>
</tr>
<tr>
<td>1/3&quot;</td>
<td>3.6</td>
<td>4.8</td>
</tr>
<tr>
<td>1/2&quot;</td>
<td>4.8</td>
<td>6.4</td>
</tr>
<tr>
<td>2/3&quot;</td>
<td>6.6</td>
<td>8.8</td>
</tr>
</tbody>
</table>

A shorter focal length gives a wider field of view, that is the returned value is the maximum focal length to obtain the specified W and H.
Example: lens calculation

This example is based on a 1/2" sensor chip.

- The FOV should cover a conveyor belt with a width of 500 mm.
- The minimum height of the FOV is not restricted.
- The distance between the camera and the conveyor is 800 mm.
- The camera is mounted with the belly facing the robot (PickMaster default).

Because the width of the conveyor determines the minimum FOV the required focal length is calculated using:

\[ f = \left( \frac{D}{W} \right) \times C_w \]

Enter the known data, \( C_w \) is 6.4 mm (see graphic above).

\[ f = \left( \frac{800}{500} \right) \times 6.4 = 10.24 \text{ mm} \]

The resulting height \( H \) of the FOV is calculated as:

\[ H = \frac{D \times C_H}{f} = 800 \times 4.8 / 10.24 = 375 \text{ mm} \]

Alternative with increased height

To increase the height of the FOV (\( H \)), the camera can be rotated 90° so that the height dimension of the sensor chip (4.8 mm) is aligned with the width dimension of the conveyor. The width dimension (6.4 mm) is aligned with the x-axis of the conveyor.

\[ f = \left( \frac{800}{500} \right) \times 4.8 = 7.68 \text{ mm} \]

The resulting height \( H \) of the FOV is now:

\[ H = 800 \times 6.4 / 7.68 = 666 \text{ mm} \]

Normally lenses are available in some standard focal lengths. Choose a lens that has a focal length shorter than the calculated value to be sure to capture the entire scene.
2.3 PickMaster PowerPac license

Introduction to licensing

A license activation key provided by ABB must be installed and activated to run PickMaster PowerPac.

PickMaster PowerPac depends on the activation of RobotStudio. You can use PickMaster PowerPac normally only if you activate RS with a license that includes the PickMaster PowerPac option. It can also be activated separately from PickMaster PowerPac, but still invokes the RS activation procedure.

PickMaster PowerPac license options

Two license options are available for PickMaster PowerPac, Basic and Premium. Users can obtain the Basic option for free and work with limited functions. The Basic option only allows you to calibrate and simulate the existing solutions, and cannot add new components under the Layout, for example, the conveyors, controllers.

The Premium option provides more functions for professional integrators and commissioners.

Function comparison between license options

The following table lists the main application scenarios and differences between three license options.

<table>
<thead>
<tr>
<th>Function</th>
<th>Premium</th>
<th>Basic</th>
<th>Free</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Robotstudio License Activated</td>
<td>PickMaster PowerPac License Activated</td>
<td>PickMaster PowerPac License Unactivated</td>
</tr>
<tr>
<td>Open solution</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Layout Edit (Controller,Gripper,Conveyor...)</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Edit (Item/Container/Flow/Recipe)</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Pack&amp;Go</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Pack as Template</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Save</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Save as</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Operation-Start (Production/Simulation/Emulation)</td>
<td>Production</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Simulation</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Emulation</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Operation-Stop</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Create solution</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unpack</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Continues on next page
## 2 Installation

### 2.3 PickMaster PowerPac license

**Continued**

<table>
<thead>
<tr>
<th>Function</th>
<th>Premium Robotstudio License Activated</th>
<th>PickMaster PowerPac License Activated</th>
<th>Basic Robotstudio License Activated</th>
<th>Basic PickMaster PowerPac License Unactivated</th>
<th>Free Robotstudio License Activated</th>
<th>Free PickMaster PowerPac License Unactivated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Reset</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Recording</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Control-Start (Production/Simulation/Emulation)</td>
<td>Production</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Simulation</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Emulation</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Connect to RT</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language switch</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manual</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vision Configuration</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital Twin</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User script</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External sensor</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Information about the current license

Use the following procedure to get information about the current license.

<table>
<thead>
<tr>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start the PickMaster PowerPac.</td>
</tr>
<tr>
<td>2</td>
<td>Click the File tab.</td>
</tr>
<tr>
<td>3</td>
<td>Click Help.</td>
</tr>
<tr>
<td>4</td>
<td>Click Options.</td>
</tr>
</tbody>
</table>

![xx2300001444]

### Activating a license key

**Activating a license key automatically over the Internet**

Use this procedure to activate a license key automatically over the Internet.

<table>
<thead>
<tr>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
</table>
| 1      | To start the licencing application, either use:  
  • In the PickMaster PowerPac, on the Options menu, click Activate License. |
| 2      | Under Standalong License, choose I want to Activate a standalong license key and click Next. |

*Continues on next page*
2 Installation

2.3 PickMaster PowerPac license

Continued

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 Under Automatic Activation, choose Activate RobotStudio over the internet and click Next.</td>
</tr>
<tr>
<td>4 Enter your 25 character Activation Key (xxxxx-xxxxx-xxxxx-xxxxx-xxxxx) and click Next. Your activation request will be sent to ABB over the Internet. If you are using a valid Activation Key that has not expired or exceeded the number of activations allowed, your PickMaster PowerPac license will be activated immediately, and your PickMaster PowerPac is ready for use when started next time.</td>
</tr>
</tbody>
</table>

Activating a license key manually

If the computer with PickMaster PowerPac installed does not have an Internet connection, you must activate the license manually. This is done in three steps:

1 Create a license request file (*.licreqx).
2 Download a license file (*.bin) using an Internet connected computer.
3 Install the license file (*.bin).

Use this procedure to activate a PickMaster PowerPac license manually.

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 To start the licensing application either use: • In the PickMaster PowerPac, on the Options menu, click Verify License.</td>
</tr>
<tr>
<td>2 In the licensing application, click PickMaster License Activation Wizard....</td>
</tr>
<tr>
<td>3 Under Automatic Activation, select Step 1: Create a license request file and click Next.</td>
</tr>
<tr>
<td>4 Enter your 25 character Activation Key (xxxxx-xxxxx-xxxxx-xxxxx-xxxxx) and click Next.</td>
</tr>
<tr>
<td>5 Click Save Request.</td>
</tr>
<tr>
<td>6 Type a name for a license request file (*.licreqx), browse to a suitable folder, and click Save.</td>
</tr>
<tr>
<td>7 Click Finish.</td>
</tr>
<tr>
<td>8 Use a removable medium, such as a USB device, to transfer the license request file to a computer with an Internet connection.</td>
</tr>
<tr>
<td>9 On the computer with internet connection, start the internet browser, and go to the link <a href="http://www.manualactivation.e.abb.com/">http://www.manualactivation.e.abb.com/</a> and follow the instructions to activate your license manually. You are instructed to browse for the saved license request file. The result will be a license file (*.bin) that you must save.</td>
</tr>
<tr>
<td>10 Transfer the license file to the PickMaster PowerPac PC.</td>
</tr>
<tr>
<td>11 On the PickMaster PowerPac computer, start the licensing application.</td>
</tr>
<tr>
<td>12 Under Automatic Activation, select Step 3: Install a license file (*.bin) and click Next.</td>
</tr>
<tr>
<td>13 Follow the wizard instructions. The PickMaster license will now be activated for the PickMaster PowerPac and the Runtime, and the PickMaster installation ready to use.</td>
</tr>
</tbody>
</table>
2.4 Self-signing certificate

2.4.1 Certificate handling

Default self-signed certificates

PickMaster® Twin products support the use of X.509 certificates for secure communication over the network. The PickMaster® Runtime generates self-signed X.509 certificates by default for PickMaster® PowerPac and PickMaster® Operator. The generated self-signed certificate has an RSA key pair with a key length of 2048 bits.

Certificate replacement

To enhance the security of the system and to assure that data is being transmitted over a secure connection, it is recommended to replace the self-signed certificates on the PickMaster® Runtime with your own certificates. This provides added security and the ability to use your own trusted certificate chain.

To replace a self-signed certificate, export your desired certificate and private key and replace the certificate in PickMaster® Runtime. It is important to follow the proper procedures for certificate replacement in order to ensure seamless and secure communication.
2.4.2 Replacing PickMaster® Runtime default certificate with self-signing certificate

Procedure

Use the following procedure to replace the PickMaster® Runtime default certificate with the self-signing certificate.

1. Go to the installation path PickMaster® Runtime:
   - For virtual Runtime: C:\Program Files (x86)\ABB\PickMaster Twin 2\PickMaster Twin Client 2\PickMaster VirtualRuntime
   - For real Runtime: C:\Program Files (x86)\ABB\PickMaster Twin 2\PickMaster Twin Client 2\PickMaster Runtime

2. Add the self-signing certificate .key and .crt files to the installation path. For example, the file names in red are the self-signing certificate files.

3. Double click to open the appweb.conf file.
2.4.2 Replacing PickMaster® Runtime default certificate with self-signing certificate

Continued

4 Change the `SSLCertificateFile` and `SSLCertificateKeyFile` to the self-signing certificate files.

5 Save the changes as administrator.
2.4.3 Generating self-signing certificate with OpenSSL

Introduction

When the users need to generate a self-signing certificate file, it's recommended to generate with OpenSSL.

Procedure

Use the following procedure to generate self-signing certificate with OpenSSL.

1. Generate a private key.

```
C:\Program Files\OpenSSL-v1.1.1a\bin\openssl genrsa -des3 -out mypickmasterwithpassword.key 2048
Enter PEM pass phrase: [mypickmasterwithpassword key]
Verifying - Enter PEM pass phrase:
```

2. Enter a password for the .key file and verify it.

```
C:\Program Files\OpenSSL-v1.1.1a\bin\openssl genrsa -des3 -out mypickmasterwithpassword.key 2048
Enter PEM pass phrase: [mypickmasterwithpassword key]
Verifying - Enter PEM pass phrase:
```

3. Generate the certificate signing request (CSR).

Tip

Enter the name of the key to invoke the generated private key for the .csr file.

```
C:\Program Files\OpenSSL-v1.1.1a\bin\openssl req -new -key mypickmasterwithpassword.key -out mypickmaster.csr
```

4. Fill in the mandatory and optional information.

Tip

User could fill the information based on their own conditions.
5 Transform the generated private key file to a new private key file without password.

```
C:\Program Files\OpenSSL-Win64\bin\openssl rsa -in myprivatekeywithpassword.key -out myprivatekey2.key
Enter pass phrase for myprivatekeywithpassword.key
Writing RSA key
```

Note

Enter the name of the key to invoke the generated private key for the new .key file.

```
C:\Program Files\OpenSSL-Win64\bin\openssl rsa -in myprivatekeywithpassword.key -out myprivatekey2.key
Enter pass phrase for myprivatekeywithpassword.key
Writing RSA key
```

6 Generate the self signing certificate.

```
C:\Program Files\OpenSSL-Win64\bin\openssl x509 -req -days 365 -in myprivatekey2.key -out mycertificate.crt
```

The new private key file without password and the generated self signing certificate .crt file are the final output.
2 Installation

2.4.4 Installing self-signing certificate

Procedure

Use the following procedure to install a self-signing certificate.

1. When a PickMaster® Twin product is used for the first time, the following dialog box will pop up.

   ![Certificate Installation dialog box]

   - [Subject]
     - E=ABB@cn.abb.com, CN=ABB PickMaster Twin, OU=RD, O=ABB, L=Shanghai, S=Shanghai, C=CN
   - [Issuer]
     - E=ABB@cn.abb.com, CN=ABB PickMaster Twin, OU=RD, O=ABB, L=Shanghai, S=Shanghai, C=CN
   - [Serial Number]
     - 4236BDCDA841A8C0DA02570636F603D79D8B3531
   - [Not Before]
     - 11/16/2022 14:35:25
   - [Not After]
     - 10/23/2122 14:35:25
   - [Thumbprint]
     - 2F82EF5C2BF73045E516ADD5C23D5394A9883357

   ![Certificate installation dialog box]

   2. Click Yes to install the self-signing certificate.
3 Click OK to finish the installation.
2 Installation

2.5 PickMaster time synchronization service

### Time synchronization service

PickMaster Twin uses a time synchronization service to synchronize the time between the robot controllers and the Host PC running PickMaster. The synchronization is performed over the same network used for communication between PickMaster Runtime and the robot controllers.

PickMaster Twin PTP v1 is used for IRC5(RobotWare 6) controller.

PickMaster Twin PTP v2 is used for OmniCore(RobotWare 7) controller.

#### Note

Whenever the Grandmaster is changed, Runtime will take several seconds or minutes to re-synchronize the time in the local area network. During this time, the robots may stop.

#### Note

To enable the time synchronization service, the user should select the local IP address which is connected to the real controller during installing the PickMaster Twin Client.

If the computer is not yet connected to a real controller, the IP address could also be configured after the installation. For detailed information, see Configuring local IP address on page 47.

#### Tip

It is recommended to use PTPTrackHound to have an overview on the PTP status of all devices in current local area network.

To download the tool PTPTrackHound, see [www.ptptrackhound.com](http://www.ptptrackhound.com).

### Settings

The synchronization service is based on the precision time protocol (PTP), which in turn implements the IEEE 1588-2002 PTP v1(For RobotWare 6)/1588-2008 PTP v2 (For RobotWare 7) standard. This protocol uses multicast messages over UDP/IP and requires that UDP port 319 and 320 are available (for both incoming and outgoing traffic). It is therefore necessary that any firewall is not blocking these ports. Please contact your system administrator to make sure that the proper configurations are performed.

PTP was originally defined in the IEEE 1588-2002 standard, officially entitled "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems" and published in 2002. In 2008, IEEE 1588-2008 was released as a revised standard; also known as PTP Version 2, it improves accuracy, precision and robustness but is not backward compatible with the original 2002 version.

Continues on next page
The time synchronization service must be set to operate on the correct PC network interface port, that is, the network port which communicates with the robot controllers.

### Time Synchronization Settings interface

<table>
<thead>
<tr>
<th>NO.</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
</table>
| 1   | Start/stop time sync with application operation | When the checkbox is:  
  • Checked: The PTP server will start/stop automatically with the start/stop of the Runtime.  
  Tip  
  The automatically restarted PTP server is the one used last time.  
  The PTP server for IRC5(RobotWare 6) and OmniCore(RobotWare 7) exist independently.  
  • Unchecked: The PTP server will not start/stop automatically with the start/stop of the Runtime. |

Tip

The automatically restarted PTP server is the one used last time. The PTP server for IRC5(RobotWare 6) and OmniCore(RobotWare 7) exist independently.
2 Installation

2.5 PickMaster time synchronization service

Continued

<table>
<thead>
<tr>
<th>NO.</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
</table>
| 2   | Use low priority for IRC5 time sync | PickMaster Twin PTPv1 is used for IRC5(RobotWare 6) controller. There are two priorities in PTPv1 strategy: high priority and low priority. The time synchronization server will use high priority device as Grandmaster device. If no high priority device exists or several high priority devices exist in the local area network, the device with better hardware will set as Grandmaster automatically. When the checkbox is:  
  - Checked: The current device is set as low priority.  
  - Tip: The automatically restarted time synchronization server is the one used last time.  
  - Unchecked: The current device is set as high priority.  
  The priority value is saved in the registry. |
| 3   | Priority level for OmniCore time sync (0-255) | PickMaster Twin PTPv2 is used for OmniCore(RobotWare 7) controller.  
Use the priority1 value of PTPv2 to control the priority of the devices for the time synchronization service.  
The range is from 0 to 255. The value is smaller, the priority is higher. The highest priority device in the local area network will be the Grandmaster device.  
The default value is 128.  
If the priority1 value is set as the same one on multiple devices in the local area network, the device with better hardware will set as Grandmaster automatically.  
The priority value is saved in the registry. |
| 4   | Controller network adaptor    | This IP must be set as the IP address of the device which connected to the controller.                                               |

Setting PickMaster options

Use this procedure to set PickMaster options.

1 On the File menu, select Options.  
The PickMaster Options dialog is opened.
2 Select to start/stop the PTP server automatically with the start/stop of the Runtime from the Start/stop time sync with application operation box.

3 Select to change the priority of current device for PTP v1 server from the Use low priority for IRC5 time sync box.

4 Define the value of priority1 of current device for PTP v2 server in the Priority level for OmniCore time sync (0-255) text box.

**Tip**

The value is smaller, the priority is higher.

5 Select the IP-address of the network adapter that communicates with the robot controller(s) from the Controller network adapter box.

6 If needed, select Log RIS commands to show all RIS commands in the log area.

7 If needed, select Log IRC5 status messages to include showing status messages in the log area. As default, only warnings and error messages are shown in the log area.

8 Select Disable vision system if PickMaster’s internal vision system should not be used. PickMaster will then not connect to any attached cameras. This is useful to avoid conflicts when an external vision system is used through the external sensor interface.

9 Click OK.
2.6 Software installation

Note

Anyone working with installation of an ABB robot must be trained by ABB and have the required knowledge of mechanical and electrical installation work.
2.6.1 Installing RobotStudio

Instruction

For the detailed RobotStudio installation procedure, see Operating manual - RobotStudio, 3HAC032104-001.

Note

When set the unit in RobotStudio, Unspecified is recommended. The other unit may cause uncertain errors in PickMaster product.
2 Installation

2.6.2 Installing PickMaster Twin Client

Procedure

Installing PickMaster Twin Client

Note

The PickMaster 3 and PickMaster Twin Client are not recommended to be installed on a same PC. They may influence each other.

Note

The PickMaster Twin Client and PickMaster Twin Host are not recommended to be installed on a same PC.

Note

Make sure that you have installed RobotStudio on your computer before installing PickMaster Twin Client. For the installation procedure of RobotStudio, see *Operating manual - RobotStudio*.

Use the following procedure to install the PickMaster Twin Client:

2. Click Next.
3. Read the license agreement and accept the terms.
4. Click Next.
5. Choose to install the Congnex vision driver and click Next.
6. Click Next.
7. Choose an IP address for network adaptor configuration and click Next.

Note

To enable the time synchronization service, the user should select the local IP address which is connected to the real controller during installing the PickMaster Twin Client. If the computer is not yet connected to a real controller, the IP address could also be configured after the installation. For detailed information, see *Configuring local IP address on page 47*.

8. Click Next to start the installation.
9. When the installation is complete, choose to restart the computer now or later and click Finish.

Continues on next page
Configuring local IP address

The local IP address should be reconfigured in the PickMaster Runtime (RRT) in the following cases:

- Previous IP configuration during installation is wrong or no IP address was selected during installation.
- The network interface currently used for connecting the real controller has been changed.

Start Runtime, click File - Options to open a pop-up dialog. Select the corresponding interface in the list box and click OK.

Note that the network interface configured in Runtime must be the IP address of the local computer connected to the controller using WAN interface.

Use the following procedure to configure the local IP address in the PickMaster Runtime (RRT):

1. Start Runtime.
2. Click File - Options to open a pop-up dialog.
3. Select the corresponding IP address in the list box and click OK.
2.7 Electrical connection

2.7.1 PickMaster® Twin Hardware connection illustration
2.7.2 Connecting cameras

Introduction to camera connections

The camera does not receive power voltage through the Ethernet cable. A separate connection provides power and I/O functions, this is the power/trig/strobe cable. We recommend using an external power supply for the Gigabit Ethernet cameras. This way, they will receive power regardless if the robot controller is turned on or not. If the camera is supplied with power directly from the robot controller it will shut down when the controller is turned off. Runtime can not reconnect to a camera that has been shut down and restarted. This means that if Runtime is running when a controller that serves as a camera power supply is shut down, Runtime must be restarted after the controller has been switched on again. This problem is avoided by using an external power supply.

A 4-port Gigabit Ethernet board which is included in the GigE Ready option must be used for the Gigabit Ethernet cameras. And the cameras cannot use the same network card with the controller, or the captured images will be affected.

The jumbo packet function of the network card shall be activated when using with the camera.

The schematics of how the trigger strobe and power wires from the camera must be connected to the robot controller I/O board can be seen in the circuit diagrams, see Circuit diagram - PickMaster Twin, 3HAC024480-020. Detailed information about avoiding EMI/ESD problems is described in Avoid_EMI_ESD_in_camera_installations, see References on page 10.

Note

All safety information for working with the controller is described in the product manual for the controller.

Prerequisites

Make sure all power is switched off before connecting cameras.

Validated cameras

The following cameras are supported by the PickMaster® Twin:

- Basler Ace acA1440-73gc
- Basler Scout scA1300-32gc
- Basler Scout scA1390-17gm

Continues on next page
2 Installation

2.7.2 Connecting cameras

Continued

CAUTION

Personal injury hazard and risk of damage to camera in case of short circuits. Short circuits may cause an extreme rise in temperature of the camera's housing. This may damage the camera and may also lead to person injuries, for example, burns. In the worst case, the overheating may cause a fire.

In order to prevent that, limit the current flowing through each individual wire during a short circuit. The maximum current allowed is 2 A. Use a fuse or use a limited power supply.

Connecting the cameras

Use this procedure to connect the cameras.

1. Connect the Ethernet cable with screw connector to the camera.
2. Connect the other end of the Ethernet cable to the PC or the switch (if used).
3. If a switch is used, connect the switch to the PC.
4. Connect the power wires of the power/trig/strobe cable to the external power supply accordingly.
   In case no external power supply is used, connect to the controller.
5. Connect the trig/strobe wires of the power/trig/strobe cable to the robot controller.

Note

If Runtime is shut down and restarted quickly, and with several Gigabit Ethernet cameras, the Gigabit Ethernet performance driver may not be loaded properly for some cameras. The symptom is that the camera for which the driver is not loaded may occasionally fail to acquire an image, if the system is stressed. This can be avoided by waiting for 15 seconds between shutting down and restarting.

Continues on next page
Network configuration for the cameras

The following procedure is recommended to modify the computer network configuration which camera is connected to:

1. Click Configure and then choose the Advanced tab.

2. Modify the following properties as necessary:
   - Select the Jumbo Packet property and choose the highest possible value in the dialog box.
   - In the Networking tab, clear all the check boxes listed under This connection uses the following items except for eBUS Universal Pro Driver and Internet Protocol Version 4 (TCP/IPv4).

3. In addition, Cognex recommends you modify the following properties for this network connection, which may or may not be grouped together with the previous properties:
   - Change the Receive Buffers property and choose the highest possible value in its Value list.
   - Change the Interrupt Moderation Rate property to Extreme in its Value list.

4. Click OK.

Refer to the embedded Questions and Answers of the Gig Vision Configuration Tool for more details on what system properties you should modify as necessary.

Related information

Circuit diagram - PickMaster Twin, 3HAC024480-020.
2 Installation

2.7.3 Connecting I/O signals

Introduction to I/O connections

The Runtime concept consists of a number of I/O components that need to be connected physically.

Robot controller I/O board

At least one standard DI/DO board is required. Encoder boards are needed for conveyor tracking.

The encoder boards are delivered with a standard address that can differ from the I/O configuration. This address can be changed.

For further information about how to read the encoder board address, see the product manual for the controller, see References on page 10.

Prerequisites

Make sure all power has been switched off.

Connecting the I/O signals

Use this procedure to connect the I/O signals.

1. If conveyors are used, connect each conveyor controller to the standard DI/DO board for control from Runtime.

   The drawings in Circuit diagram - PickMaster Twin, 3HAC024480-020, uses ACS 301-1P6-3 as conveyor controller, but other conveyor controllers can be used.

2. Connect the trig/strobe wires of the power/trig/strobe cables from the cameras to the robot controller.

3. Connect the I/O cables from any external tool signals to the robot controller.

4. Connect the I/O cables for other external devices, such as sensors to the robot controller.

I/O connections

The trigger strobe loop enables very precise synchronization between the robot controller and the image acquired. The I/O port of the Gigabit Ethernet camera closes this loop.

To be able to use more than one connection in input number 9 (StartSig) on the encoder board we recommend using diodes, for example HER105/Taw diode 1A 400V DO41 (the diodes are not supported by ABB). This will eliminate any possibilities of reverse currents.

When connecting a camera to multiple robot controllers it is important to consider how the system should work if one of the controllers is turned off. We recommend using an external 24V power supply to power the cameras. This way the cameras will have both power and I/O regardless if the controllers are turned off.

Related information

Circuit diagram - PickMaster Twin, 3HAC024480-020

Continues on next page
2 Installation

2.7.3 Connecting I/O signals

Continued

I/O signals on page 172.
Conveyor work area default I/O signals on page 174.
2.7.4 Configuring networks

Introduction to the controller network

The PickMaster PowerPac and the robot controller communicate through Ethernet. If you have problems in connecting to the network, contact the local network administrator.

**Note**

The PickMaster PowerPac must be connected to the WAN port on the controller. Do not use the service port.

Configuring the controller network

If a new local area network (LAN) is created specifically for PickMaster PowerPac the following settings can be used.

- Use static IP numbering with different addresses for both the computer and the robot controller.
- IP addresses: 192.168.1.X (where X is between 1 and 253).
- Subnet mask: 255.255.255.0
- Gateway: 192.168.1.254
- DNS: N/A.
- Wins: N/A.

**Note**

The robot controller has a service Ethernet card configured with an IP address (192.168.125.1). Therefore, the same subnet (192.168.125.X) must not be used for the standard LAN Ethernet card.

For more information, see the Windows documentation and the product manual for the robot controller to set up the IP configuration.

**Note**

It's not allowed to use any of the following IP addresses which are allocated for other functions:

- 192.168.127.0 - 255

The IP address cannot be on a subnet which overlaps with any of the above reserved IP addresses. If a subnet mask in the class B range has to be used, then a private address of class B must be used to avoid any overlapping. Contact your local network administrator regarding network overlapping.

See the section Communication in Technical reference manual - System parameters.

*Continues on next page*
Prerequisites for vision networks

The vision network settings must be configured similar to the robot controller network settings.

Use a separate network for the vision system, that is controllers and cameras cannot be connected to the same network port on the PC.

To use more cameras than the number of available Ethernet ports on the PC, use one or two additional GigE cards.

The maximum number of cameras that can be used with one PC is 10. Distribute them evenly on the dedicated vision network ports on the PC. Use the supplied cables with fastening screws between GigE card and camera. For the example of network architecture, see *Example of suitable network architecture on page 58.*

Configuring the vision network

Use this procedure to configure the vision network.

1. Assign each camera with its own IP-address. The same rules apply as for other Ethernet networks, that is each camera and vision network card must have a unique IP address, and be located on the same subnet. The communication with cameras and controllers should be separated on different subnets. See *Example of suitable network architecture on page 58.*
2 Configure the IP addresses for the cameras using Cognex's *Ethernet Camera Tool* (available on the Windows Start menu in the PickMaster folder). It can be used to set IP addresses of both cameras and network interface cards.

3 When all cameras are configured, install the *Performance Driver* for Gigabit Ethernet vision for each port, see steps 4-6.

4 In the *Ethernet Camera Tool*, for each vision network port in the tree view, do the following settings:
   a In the *Properties* section set the value of MTU at around 9000. If the MTU value is around 1500, it means that the Jumbo frames is not set.
To set the Jumbo frames:

I Click ...

The Ethernet Properties window is displayed.

II Click the Networking tab.

III Click Configure.

The properties window is displayed.

IV Click the Advanced tab.

V Select Jumbo Frame from the Property list.

VI Select a value as high as possible from the Value drop-down list.

VII Click OK/Apply until you are back in the Ethernet Camera Configuration tool.

VIII Press F5 to refresh the values in the window.

IX Verify that the MTU value is about 9000.

b Select the eBus Universal Pro Driver check box. A warning about installing unsigned software appears.

c Click OK.

5 Reboot the PC when the installation is complete for all the vision ports.

Continues on next page
6 Start the **Ethernet Camera Tool** and verify that the performance driver has been successfully installed for each vision network port. Also verify that the Jumbo frames MTU value is set to about 9000.

**Note**

In case you face any issue during image capture, modify the following network configuration on the ethernet where the camera is connected:

- In the **Ethernet Camera Tool** for vision network port in the tree view Click ...
  
  The Ethernet Properties window is displayed.
  
  - In the **Networking** tab, clear all the check boxes listed under This connection uses the following items except eBUS Universal Pro Driver and Internet Protocol Version 4 (TCP/Ipv4).
  - Click **Configure** and then choose the **Advanced** tab.
    
    # Select the **Receive Buffers** property and choose the highest possible value in the **Value** list.
    
    # Select the **Interrupt Moderation Rate** property and choose the value as **Extreme**.

**Note**

Running the **Ethernet Camera Tool** and **Runtime** at the same time may result in unpredictable behavior. To avoid this, use only one of the programs at a time.

**CAUTION**

Running camera traffic and controller traffic on the same network can cause serious communication failure.

---

**Configuring the Runtime network**

If a new local area network (LAN) is created specifically for **Runtime** the following settings can be used.

- Use static IP numbering with different addresses for the PickMaster PowerPac and the robot controller.
- IP addresses: 192.168.1.X (where X is between 1 and 253).
- Select **Connect to RRT**, the **Sign in** window is displayed. How to connect to RRT, see **Runtime on page 73**.

**Example of suitable network architecture**

- Use static IP numbering with different addresses for both the computer and the camera(s).
- IP addresses of Port #1 and the cameras connected to it: 192.168.101.X (where X is between 1 and 253).
- IP addresses of Port #2 and the cameras connected to it: 192.168.102.X (where X is between 1 and 253).
2 Installation

2.7.4 Configuring networks

Continued

- Subnet mask: 255.255.255.0
- Gateway: Not Needed.
- DNS: N/A.
- Wins: N/A.

Note

Changes made to the camera settings outside Runtime will not be applied until Runtime is restarted. This means that if a camera is restarted (power on/off) or a camera’s IP address is changed, then Runtime must be restarted to function properly. Therefore, Runtime and the Ethernet camera tool program should not be run simultaneously, to avoid unpredictable behavior. Instead, shut down Runtime before making changes, then start Runtime after changes are saved.
2 Installation

2.7.5 Setting up robot controller

CAUTION

If robot movement can be initiated from an external control panel then an emergency stop must also be available.

RobotWare

PickMaster PowerPac supports IRC5 and OmniCore robot controller. RobotWare is installed on the robot controller. The option PickMaster Ready is required to run Runtime. For more details on option, see PickMaster Twin Product Specification.

For more information see the product manual for the controller, see References on page 10.

System parameters

The number of conveyors must be specified in the system parameters. Some other parameters must also be defined, such as motion, process, and encoder I/O parameters for the conveyors.

System parameters can be changed using the FlexPendant or RobotStudio.

I/O signals

How to configure I/O signals and boards is described in the section I/O signals on page 172.

The predefined I/O signals are described in the section Conveyor work area default I/O signals on page 174.

Related information

Product manual for the controller, see References on page 10.


Six axes robot configuration on page 62.
2.7.6 Optional robot and process configuration

Conveyor process modification
Modifications can be done on the system parameters.

Topic Process
The following parameter can be modified in the topic Process. It belongs to the type Conveyor systems.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum distance</td>
<td>Defines the standard tracking distance of a conveyor work object before it is switched to a new work object. This is by default set to 20000mm. The work object switch is done automatically and fast but may steal some process time for a high speed picking application. Increasing the value may improve the cycle time slightly.</td>
</tr>
</tbody>
</table>
2.7.7 Six axes robot configuration

Modifications for six axes robots

When using PickMaster with a six axes robot, some modifications must be done in the system parameters to optimize the robot motion with the conveyor tracking process.

Topic Process

The following three parameters can be modified in the topic Process. They belong to the type Conveyor systems.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start ramp</td>
<td>This is the correction start filter ramp that is used when connecting to a moving conveyor. This is by default set to 5 (steps). Tune this parameter if higher accuracy is needed. A lower value gives better accuracy but the manipulator may jerk when connecting to the moving object.</td>
</tr>
<tr>
<td>Stop ramp</td>
<td>This is the correction stop filter ramp that is used when disconnecting from a moving conveyor. This is by default set to ten (steps). Tune this parameter to eliminate manipulator jerks when leaving the moving object. A lower value gives better accuracy when leaving the conveyor.</td>
</tr>
<tr>
<td>Adjustment speed</td>
<td>The speed (in mm/s) at which the robot should catch up to the conveyor. The general recommended value is 130% of the conveyor speed. As minimum, the value should be more than 100% with some margin. If the robots speed is very fast compared to the conveyor speed, a further increase of the value is often necessary. If the value is set too low, robot movements may become jerky or the conveyor tracking accuracy may become reduced. On the other hand, if the value is set too high, the drive system may become overloaded, causing motion supervision errors. Generally, the maximum recommended value is 200%. For IRB360 in applications with high robot speed, the maximum recommended value is 500%.</td>
</tr>
</tbody>
</table>
3 Navigating PickMaster PowerPac

3.1 Main window

Overview

This chapter describes about the user interface of the PickMaster PowerPac. The following figure and table provides information regarding the major elements in the user interface.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ribbon tab: Contains the general functions for PickMaster PowerPac. When creating a new solution, the work flow is usually from left to right. For more details, see the section <em>Ribbon tab on page 65</em>.</td>
</tr>
<tr>
<td>2</td>
<td>Tree view browser: Organizes the programmable objects (for example, robots, sensors, and conveyors) of the picking application in a tree structure. It is separated into <strong>Layout</strong> and <strong>process</strong> tabs. For more details, see the section <em>Tree view browser on page 72</em>.</td>
</tr>
<tr>
<td>3</td>
<td>Station view: Realistic 3D display of the picking application. The objects in the station view are highlighted when selected or edited using the tree view browser.</td>
</tr>
<tr>
<td>4</td>
<td>Log view: Shows all the events happened to current station.</td>
</tr>
<tr>
<td></td>
<td>Tip: You can search with key words in the search-box for the specific event.</td>
</tr>
<tr>
<td>5</td>
<td>Status view: Shows the status of the controller and system at present.</td>
</tr>
</tbody>
</table>

*Continues on next page*
### 3 Navigating PickMaster PowerPac

3.1 Main window

Continued

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Additional operation view</td>
</tr>
</tbody>
</table>

**Tip**

All windows can be distributed and floating freely.
3.2 Ribbon tab

Overview

The PickMaster PowerPac ribbon contains elements arranged in various groups. The following figures and tables provide more information regarding the elements in the PickMaster PowerPac ribbon.

Following are the objects and configurations saved in the ribbon tab.

<table>
<thead>
<tr>
<th>File</th>
<th>Layout</th>
<th>Process</th>
<th>Operation</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Controller</td>
<td>Gripper</td>
<td>Conveyor</td>
<td>Camera</td>
</tr>
</tbody>
</table>

Note

If the solution will be used in the PickMaster Operator, it must have been connected to a real controller with the same configuration on PickMaster PowerPac.

Continues on next page
### 3 Navigating PickMaster PowerPac

#### 3.2 Ribbon tab

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Save as</strong></td>
<td>Save your present solution as a new solution in desired location.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>If the solution will be used in the PickMaster Operator, it must have been connected to a real controller with the same configuration on PickMaster PowerPac.</td>
</tr>
<tr>
<td><strong>Open</strong></td>
<td>Open other solutions or any solutions saved in your local folder.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>Only solutions or shared files which are created with PickMaster PowerPac 2.0 or later can be opened.</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Close your present solution.</td>
</tr>
<tr>
<td><strong>Recent</strong></td>
<td>Open the solutions which has been opened before.</td>
</tr>
<tr>
<td><strong>New</strong></td>
<td>Create a new empty solution.</td>
</tr>
<tr>
<td>Solution with Empty Station</td>
<td>Create a new solution with the template.</td>
</tr>
</tbody>
</table>

---

*Continues on next page*
### 3.2 Ribbon tab

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Share</strong></td>
<td><strong>Share data with other people</strong></td>
</tr>
<tr>
<td></td>
<td>Pack and GO &lt;br&gt;Pack all the information of current solution, controller used in the solution and 3D models into a file so that it makes sharing files between users.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> &lt;br&gt;It is not allowed to rename the packed file. Otherwise it may cause unpacking problem.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> &lt;br&gt;Python script files will not be included in the Pack&amp;Go file. Copy the Python script files to the desired destination.</td>
</tr>
<tr>
<td><strong>UnPack and Work</strong></td>
<td>Unpack the shared files which contains all the information of a solution, controller used in the solution and 3D models.</td>
</tr>
<tr>
<td><strong>Pack As Template</strong></td>
<td>Pack your present solution as a template in your local folder.</td>
</tr>
<tr>
<td><strong>Runtime file transfer</strong></td>
<td><strong>Upload to Runtime</strong> &lt;br&gt;Upload a desired .rpag file to the connected Host computer Runtime.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip</strong> &lt;br&gt;Connect to the real Runtime on the Host computer before the uploading.</td>
</tr>
<tr>
<td></td>
<td><strong>Download from Runtime</strong> &lt;br&gt;Download a desired .rpag file from the connected Host computer Runtime.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip</strong> &lt;br&gt;Connect to the real Runtime on the Host computer before the downloading.</td>
</tr>
</tbody>
</table>

*Continues on next page*
### 3.2 Ribbon tab

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help</td>
<td>Shows the basic version information.</td>
</tr>
</tbody>
</table>
| Options | **Language**: choose the applied language. Eight languages are supported:  
  - English  
  - Simplified Chinese  
  - German  
  - Italian  
  - Spanish  
  - Japanese  
  - French  
  - Korean  

  **Rapid Editor**: specify the editor to open Rapid.  
  **License**: show current license type.  
  **Disable License** check box: disable the premium license if checked.  
  **Activate License** icon: activate a premium license. |
| Exit | Close and exit the PickMaster PowerPac. |
| Manual | Open the PickMaster PowerPac application manual. |

#### Tip

When opening or creating a new solution with PickMaster PowerPac, the **Virtual Runtime** will start and be connected automatically.

---

*Continues on next page*
Tip

The PickMaster® Runtime (VRT and RRT) is defined to use 50000 port. If 50000 port is occupied by other program, you will have this warning and not be able to connect to Runtime,

Release the 50000 port and restart the PickMaster® Runtime.

Use this procedure to release the 50000 port:

1. Enter the command `netstat -aon | findstr "50000"` in the CMD window.
2. The process that occupies port 50000 will be listed in the window. Obtain the PID code of the process.
3. Find the process corresponding to this PID in the task manager and close the it (Make sure that this process is allowed to be closed on this computer).
4. Restart PickMaster® Runtime and connect.

### Layout

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>Add a controller with a robot system in the station view. More details about creating a controller is available in the section <strong>Controller on page 103</strong>.</td>
</tr>
<tr>
<td>Gripper</td>
<td>Add a gripper. More details about creating a gripper is available in the section <strong>Gripper on page 106</strong>.</td>
</tr>
<tr>
<td>Conveyor</td>
<td>Add a conveyor. More details about creating a conveyor is available in the section <strong>Conveyor on page 109</strong>.</td>
</tr>
<tr>
<td>Camera</td>
<td>Add a camera. More details about creating a camera is available in the section <strong>Camera on page 111</strong>.</td>
</tr>
<tr>
<td>I/O Sensor</td>
<td>Add an I/O sensor. More details about creating an I/O sensor is available in the section <strong>Adding an I/O sensor on page 113</strong>.</td>
</tr>
</tbody>
</table>
### 3 Navigating PickMaster PowerPac

#### 3.2 Ribbon tab

**Continued**

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Sensor</td>
<td>Add an external sensor. More details about creating an external sensor is available in the section <em>Adding an external sensor on page 115.</em></td>
</tr>
<tr>
<td>Conveyor Work Area</td>
<td>Add a conveyor work area. More details about creating a conveyor work area is available in the section <em>Work area on page 116.</em></td>
</tr>
<tr>
<td>Indexed Work Area</td>
<td>Add an indexed work area. More details about creating an indexed work area is available in the section <em>Adding an indexed work area on page 119.</em></td>
</tr>
<tr>
<td>Calibration</td>
<td>Calibrate the created solution in PickMaster PowerPac. More details about calibrating the created solution is available in the section <em>Calibration on page 126.</em></td>
</tr>
</tbody>
</table>

#### Process

- **File**
  - Layout
  - Process
  - Operation
  - Runtime

- **Item**
- **Container**
- **Flow**
- **Recipe**

**Continued on next page**
### 3 Navigating PickMaster PowerPac

#### 3.2 Ribbon tab

### Button Description

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Start a simulation. When click on the drop-down arrow, Start and Record shows up. Start and Record: start and record the simulation as an .exe file.</td>
</tr>
<tr>
<td>Stop</td>
<td>Stop the simulation.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>Stop will stop the solution and set the robot back to origin.</td>
</tr>
<tr>
<td>Reset</td>
<td>Reset the station view from objects temporarily created in the previously run simulation. Reset will clean the items and containers on the conveyor.</td>
</tr>
<tr>
<td>Control</td>
<td>Start a production. More details about how to run the production is available in the section Simulation on page 161 and Emulation on page 313.</td>
</tr>
<tr>
<td>Start Recording</td>
<td>Record the simulation including the curser and mouse-clicks as .mp4 file.</td>
</tr>
<tr>
<td>Stop Recording</td>
<td>Stop recording the simulation including the curser and mouse-clicks.</td>
</tr>
</tbody>
</table>

### Runtime

![Runtime Diagram]

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Local RRT</td>
<td>Start the Runtime on the computer.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Local RRT means the Runtime installed with PickMaster PowerPac. It can be used for test purposes.</td>
</tr>
<tr>
<td>Connect to RRT</td>
<td>Connect to the real Runtime.</td>
</tr>
<tr>
<td>Start and Connect to VRT</td>
<td>Start the virtual Runtime on the computer and connect to it.</td>
</tr>
</tbody>
</table>
3 Navigating PickMaster PowerPac

3.3 Tree view browser

3.3.1 Layout

Overview

The Layout tab displays the Runtime and the application hardware objects such as robots, cameras, conveyors, and so on.

Following are the objects and configurations saved in the Layout tab.

- Runtime
- Position generator
- Layout
  - Controllers
  - Grippers
  - Conveyors
  - Indexed Work Areas
  - Cameras
  - I/O Sensors
  - External Sensors
Runtime

Right-click on Runtime to set the connection to the virtual Runtime (VRT) in simulation mode or the real Runtime (RRT) for operating the real robots on the Host computer in emulation mode.

**Tip**

Before connecting to RRT, start the PickMaster Runtime on the Host computer.

When selecting Connect to RRT, the Sign in window is displayed.

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IP Address</strong></td>
<td>Enter the IP address of the Runtime computer.</td>
</tr>
<tr>
<td><strong>Credential</strong></td>
<td></td>
</tr>
<tr>
<td><strong>UserName</strong></td>
<td>The default user name is admin. And it CANNOT be changed.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Enter the password of your account in the Runtime.</td>
</tr>
</tbody>
</table>

**Note**

Loopback address is NOT allowed to use as the real PickMaster Runtime IP address, for example 127.0.0.1. Loopback address will cause errors in vision function.

**Local RRT** means the Runtime installed with PickMaster PowerPac. It can be used for test purposes.
Position generator

Right-click on Position Generator will allow you to define the relationship and source type of conveyors.

<table>
<thead>
<tr>
<th>Description</th>
<th>Source Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selects a conveyor or indexed work area in order to set the related relationships.</td>
<td><strong>Source Type</strong></td>
</tr>
<tr>
<td>Select the input signal source type:</td>
<td><strong>Vision</strong>: If the source type is set to Vision, a camera and vision models are used to find the object positions. The vision models are described in section <em>Adding vision model on page 286</em>.</td>
</tr>
<tr>
<td></td>
<td><strong>Predefined</strong>: If the source type is set to Predefined, the positions generated by the position source are statically defined and no camera is used.</td>
</tr>
<tr>
<td></td>
<td><strong>External</strong>: If the source type is set to External, an external sensor in the solution together with external position generators are used to define item positions.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>If the source type is set to Vision, all available cameras and related items will be listed in the Available Camera.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>If an indexed work area is used, external sensor function will be disabled.</td>
</tr>
</tbody>
</table>
### Trigger Setting

Select Trigger type to define when to generate new item positions.

**Note**

If the trigger type is set to Distance the trigger distance must be defined in the Trigger Distance box in Operation setting under Recipe.

A distance trigger can only be used with a conveyor work area and the entered value is the distance the conveyor should move between consecutive triggers.

**CAUTION**

If the Predefined and IO sensor are selected in the recipe, tune the pick location in the Tuning for a radial distance of the item to make up the offset.

**Tip**

If an indexed work area is used, Trigger Setting is not available.

### Base Frame Adjustment

Adjust the base frame for selected conveyor or indexed work areas.

For more information, see [Adjusting the base frame on page 122](#).

### Controllers

#### Managing controller

Right-click on a Controller in the tree view to select and define a controller.

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit controller</td>
<td>Change the settings for the selected controller. When you right-click on a controller and select Edit controller, the Edit controller window is displayed. See the following section for more details about managing a selected controller.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the selected controller.</td>
</tr>
<tr>
<td>Rename</td>
<td>Change the name of the selected controller.</td>
</tr>
</tbody>
</table>

The following table provides details about the **Edit controller** window.

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller Name</td>
<td>Displays the name of the selected controller.</td>
</tr>
<tr>
<td>System Name</td>
<td>Displays the name of the system.</td>
</tr>
<tr>
<td>IP Address</td>
<td>Displays the IP address of the selected controller.</td>
</tr>
<tr>
<td>Version</td>
<td>Displays the version of the system.</td>
</tr>
<tr>
<td>System ID</td>
<td>Displays the ID of the system.</td>
</tr>
<tr>
<td>Select Virtual controller icon</td>
<td>Start the selected virtual controller.</td>
</tr>
<tr>
<td>Select Real controller icon</td>
<td>Select a real controller when running production.</td>
</tr>
</tbody>
</table>

*Continues on next page*
Managing robot

Right-click on a **Robot** in the tree view to manage the robot.

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Jump Home</td>
<td>Move the robot to the home position.</td>
</tr>
</tbody>
</table>
| Set Position     | Set a position for the selected robot.  
  When you right-click on a robot and select *Set Position*, the *Set Robot Pose* window is displayed. See the following section for more details about managing the position of a selected robot. |
| Examine          | Examine the robot in the Station view. |
| Rename           | Change the name of the selected robot. |

Set Position

The following table provides details about the **Set Position** configuration window.

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Select a coordinate system.</td>
</tr>
<tr>
<td>Position X,Y,Z (mm)</td>
<td>Set a new position for the selected robot.</td>
</tr>
<tr>
<td>Orientation (deg)</td>
<td>Set a new orientation for the selected robot.</td>
</tr>
</tbody>
</table>

Grippers

Managing grippers

Right-click on a **Gripper** in the tree view to manage the gripper.

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
</table>
| Settings    | Manage the settings of the selected gripper.  
  When you select *Setting*, the *Robot_Gripper Setting* window is displayed. More details about managing a selected gripper is available in the section *Gripper on page 106*. |
| Delete      | Delete the selected gripper. |
| Rename      | Change the name of the selected gripper. |
| Examine     | Examine the selected gripper in the Station view. |

Conveyors

Managing conveyor

Right-click on a **Conveyor** in the tree view to manage the conveyor.

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
</table>
| Setting     | Manage the settings of the selected conveyor.  
  When you select *Setting*, the *Conveyor Setting* window is displayed. More details about managing a selected conveyor is available in the section *Conveyor on page 109*. |
| Delete      | Delete the selected conveyor. |
| Rename      | Change the name of the selected conveyor. |
Managing work area

Right-click on a **Conveyor WA** in the tree view to manage the work area.

<table>
<thead>
<tr>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage the settings of the selected work area.</td>
<td>When you right-click on a conveyor work area and select <strong>Setting</strong>, the <strong>Conveyor WA Setting</strong> window is displayed. More details about managing a conveyor work area is available in the section <em>Work area on page 116</em>. When you right-click on an indexed work area and select <strong>Settings</strong>, the <strong>Indexed WA Setting</strong> window is displayed. More details about managing an indexed work area is available in the section <em>Adding an indexed work area on page 119</em>.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the selected conveyor work area.</td>
</tr>
<tr>
<td>Rename</td>
<td>Change the name of the selected conveyor work area.</td>
</tr>
</tbody>
</table>

Indexed Work Areas

Managing indexed work area

Right-click on a **Indexed Work Areas** in the tree view to manage the indexed work area.

<table>
<thead>
<tr>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage the settings of the selected indexed work area.</td>
<td>When you select <strong>Setting</strong>, the <strong>Indexed Work Area Setting</strong> window is displayed. More details about managing a selected conveyor is available in the section <em>Indexed work area on page 119</em>.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the selected indexed work area.</td>
</tr>
<tr>
<td>Rename</td>
<td>Change the name of the selected indexed work area.</td>
</tr>
</tbody>
</table>
### 3.3.1 Layout

#### Hotspots

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hotspot</strong></td>
</tr>
<tr>
<td>Manage the hotspots. When you select Hotspot, the Set Indexed Work Area hotspots window is displayed. See the following section for more details about the Set Indexed Work Area hotspots window.</td>
</tr>
</tbody>
</table>

**Note**

The hotspot is a saved location on the indexed work area. A hotspot is used to define where on the indexed work area the flow shall be generated. There is always a default hotspot, Hotspot0, located at the beginning of the indexed work area. If the flow appears at a wrong location, modify the hotspot location to adjust it.

<table>
<thead>
<tr>
<th>Examine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examine the selected indexed work area in the Station view.</td>
</tr>
</tbody>
</table>

#### Cameras

Right-click on a **Camera** in the tree view to manage the camera.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configuration</strong></td>
</tr>
<tr>
<td>Configure the selected camera. When you right-click on a camera and select Configuration, the Camera Configuration window is displayed. More details about managing a camera is available in the section Configuring camera on page 177.</td>
</tr>
</tbody>
</table>

| **Calibration** |
| Calibrate the selected camera. When you right-click on a camera and select Calibration, the Camera Calibration window is displayed. More details about managing a camera is available in the section Calibrating camera on page 276. |

**Tip**

Calibration, Live Video and Setting are enabled only for real camera.
### Description

<table>
<thead>
<tr>
<th>Live Video</th>
<th>Shows the view of the scene of the real camera before production.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Image" alt="Calibration Dialog" /></td>
<td>When you started the live video, you can stop it by click the Stop Video after right click on the sensor.</td>
</tr>
<tr>
<td><img src="Image" alt="Tip" /></td>
<td>Tip Calibration, Live Video and Setting are enabled only for real camera.</td>
</tr>
<tr>
<td>Setting</td>
<td>Manage the settings of the selected camera. When you right-click on a camera and select Setting, the Camera Setting window is displayed. More details about managing a camera is available in the section Camera on page 111.</td>
</tr>
<tr>
<td><img src="Image" alt="Tip" /></td>
<td>Tip Calibration, Live Video and Setting are enabled only for real camera.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the selected camera.</td>
</tr>
<tr>
<td>Rename</td>
<td>Change the name of the selected camera.</td>
</tr>
<tr>
<td>Examine</td>
<td>Examine the selected camera in the Station view.</td>
</tr>
</tbody>
</table>
I/O Sensors

Right-click on an I/O Sensor in the tree view to manage the I/O sensor.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setting</strong></td>
</tr>
<tr>
<td>Manage the settings of the selected I/O sensor.</td>
</tr>
<tr>
<td>When you right-click on an I/O sensor and select Setting, the I/O Sensor Setting window is displayed. More details about managing an I/O sensor is available in the section Adding an I/O sensor on page 113.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
</tr>
<tr>
<td>Delete the selected I/O sensor.</td>
</tr>
<tr>
<td><strong>Rename</strong></td>
</tr>
<tr>
<td>Change the name of the selected I/O sensor.</td>
</tr>
<tr>
<td><strong>Examine</strong></td>
</tr>
<tr>
<td>Examine the selected I/O sensor in the Station view.</td>
</tr>
</tbody>
</table>

External Sensors

Right-click on an External Sensors in the tree view to manage the external sensor.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setting</strong></td>
</tr>
<tr>
<td>Manage the settings of the selected external sensor.</td>
</tr>
<tr>
<td>When you right-click on an external sensor and select Setting, the External Sensor Setting window is displayed. More details about managing a camera is available in the section Adding an external sensor on page 115.</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
</tr>
<tr>
<td>Configure the selected external sensor.</td>
</tr>
<tr>
<td>When you right-click on an external sensor and select Configuration, the External Sensor Configuration window is displayed. More details about managing a camera is available in the section Configuring external sensor on page 181.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
</tr>
<tr>
<td>Delete the selected external sensor.</td>
</tr>
<tr>
<td><strong>Rename</strong></td>
</tr>
<tr>
<td>Change the name of the selected external sensor.</td>
</tr>
<tr>
<td><strong>Examine</strong></td>
</tr>
<tr>
<td>Examine the selected external sensor in the Station view.</td>
</tr>
</tbody>
</table>
3.3.2 Process

Overview

The Process tab displays the configuration file and the application hardware objects such as items, containers, flows, and recipes.

Following are the objects and configurations saved in the Process tab.

- Items
- Containers
- Flow
- Recipes

Items

Managing item

Right-click on an Item in the tree view to manage the item.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting</td>
</tr>
<tr>
<td>Manage the settings of the selected item. When you select Setting, the Item Setting window is displayed. More details about managing a selected item is available in the section Item on page 127.</td>
</tr>
<tr>
<td>Delete</td>
</tr>
<tr>
<td>Delete the selected item.</td>
</tr>
<tr>
<td>Rename</td>
</tr>
<tr>
<td>Change the name of the selected item.</td>
</tr>
<tr>
<td>Copy</td>
</tr>
<tr>
<td>Create a copy of the selected item with all settings.</td>
</tr>
</tbody>
</table>

Containers

Managing container

Right-click on a Container in the tree view to manage the container.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting</td>
</tr>
<tr>
<td>Manage the settings of the selected container. When you select Setting, the Container Setting window is displayed. More details about managing a selected container is available in the section Container on page 132.</td>
</tr>
<tr>
<td>Delete</td>
</tr>
<tr>
<td>Delete the selected container.</td>
</tr>
<tr>
<td>Rename</td>
</tr>
<tr>
<td>Change the name of the selected container.</td>
</tr>
<tr>
<td>Copy</td>
</tr>
<tr>
<td>Create a copy of the selected container with all settings.</td>
</tr>
</tbody>
</table>
Flow

Managing flow

Right-click on a Flow in the tree view to manage the flow.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting</td>
<td>Manage the settings of the selected flow. When you right-click on a flow and select Setting, the Flow Setting window is displayed. More details about managing a flow is available in the section Flow on page 138.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the selected flow.</td>
</tr>
<tr>
<td>Rename</td>
<td>Change the name of the selected flow.</td>
</tr>
<tr>
<td>Copy</td>
<td>Create a copy of the selected flow with all settings.</td>
</tr>
</tbody>
</table>

Recipes

Managing recipe

Right-click on a Recipe in the tree view to manage the recipe.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting</td>
<td>Manage the settings of the selected recipe. When you select Setting, the Recipe Setting window is displayed. More details about managing a selected recipe is available in the section Recipe on page 141.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the selected recipe.</td>
</tr>
<tr>
<td>Rename</td>
<td>Change the name of the selected recipe.</td>
</tr>
<tr>
<td>Copy</td>
<td>Create a copy file of the selected recipe with all settings.</td>
</tr>
</tbody>
</table>
3.4 Log view

Log

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>Shows all logs.</td>
</tr>
<tr>
<td>Note</td>
<td>If right click on one log message, Save Log and Clear All are available.</td>
</tr>
<tr>
<td>Filter box</td>
<td>Filter the specific logs with key words.</td>
</tr>
<tr>
<td>Context menu</td>
<td>Expands more operation on the logs, for example export or clean up the current logs.</td>
</tr>
<tr>
<td>Picking Status</td>
<td>Shows an overview of the picking status in summary or detail.</td>
</tr>
</tbody>
</table>
3 Navigating PickMaster PowerPac

3.5 Status view

3.5 Status view

When the system starts, the status of the controller and the Runtime will show up on the top right corner as the illustration.

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime</td>
<td>Grey: No solution is opened.</td>
<td></td>
</tr>
<tr>
<td>VRT</td>
<td><strong>Red</strong>: The connection to the virtual Runtime fails.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Green</strong>: The connection to the virtual Runtime successes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Yellow</strong>: The connection to the virtual Runtime is progressing.</td>
<td></td>
</tr>
<tr>
<td>RRT</td>
<td><strong>Red</strong>: The connection to the real Runtime fails.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Green</strong>: The connection to the real Runtime succeeds.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Yellow</strong>: The connection to the real Runtime is progressing.</td>
<td></td>
</tr>
<tr>
<td>Controllers</td>
<td><strong>Red</strong>: There is at least one controller stopped.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Green</strong>: All controllers are started and auto-running.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Yellow</strong>: There is at least one controller started and under manual controlling or just connected.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grey: No controller is added in the existing solution.</td>
<td></td>
</tr>
<tr>
<td>Controller</td>
<td><strong>Red</strong>: Controller is stopped</td>
<td>Click on the <strong>Controllers</strong> button, the detailed status for each controller will show up.</td>
</tr>
<tr>
<td></td>
<td><strong>Green</strong>: Controller is started and auto-running.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Yellow</strong>: Controller is started and under manual controlling or just connected.</td>
<td></td>
</tr>
</tbody>
</table>

Click on the **Controllers** button, the detailed status for each controller will show up.
4 Working with PickMaster PowerPac

4.1 Overview

Overview

Working with PickMaster PowerPac in virtual Runtime is to fulfill the simulation function in a visual status.

Working with PickMaster PowerPac in real Runtime is to fulfill the emulation and production function in real stations with real robots and controllers.

Simulation is a previous debugging procedure to save cost and time when creating real stations.

The following is a recommended flow for working with PickMaster PowerPac. After you complete the workflow, you can perform these task in any order.

Note

The controller (contains at least one robot system) should be set up in RobotStudio or PickMaster PowerPac before adding controller to the solution in PickMaster PowerPac.

If multiple controllers is needed in the solution, you need to create multiple controllers in advance. The same controller cannot be imported into the same solution repeatedly in PickMaster PowerPac.

Note

If any firewall or antivirus software is installed, add pickmasteru.exe and visionclient.exe to the white list.

Otherwise the PickMaster PowerPac cannot connect Runtime and the vision function cannot work normally.

Workflow for PickMaster PowerPac

Use this procedure to work with PickMaster PowerPac:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRT 1</td>
<td>Create an empty solution. For detailed information, see Solution on page 102.</td>
</tr>
<tr>
<td>2</td>
<td>Add a controller. For detailed information, see Controller on page 103.</td>
</tr>
<tr>
<td>3</td>
<td>Add a gripper. For detailed information, see Gripper on page 106.</td>
</tr>
<tr>
<td>4</td>
<td>Add a conveyor. For detailed information, see Conveyor on page 109.</td>
</tr>
<tr>
<td>5</td>
<td>Add a camera. For detailed information, see Camera on page 111.</td>
</tr>
<tr>
<td>6</td>
<td>Add an I/O sensor. For detailed information, see Adding an I/O sensor on page 113.</td>
</tr>
<tr>
<td>7</td>
<td>Add an external sensor. For detailed information, see External sensor on page 115.</td>
</tr>
</tbody>
</table>

Continues on next page
## 4 Working with PickMaster PowerPac

### 4.1 Overview

**Continued**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Add a work area. For detailed information, see <em>Work area on page 116.</em></td>
</tr>
<tr>
<td>9</td>
<td>Add an indexed work area. For detailed information, see <em>Indexed work area on page 119.</em></td>
</tr>
<tr>
<td>10</td>
<td>Set position generator. For detailed information, see <em>Position generator on page 121.</em></td>
</tr>
<tr>
<td>11</td>
<td>Calibrate the solution. For detailed information, see <em>Calibration on page 126.</em></td>
</tr>
<tr>
<td>12</td>
<td>Add an items. For detailed information, see <em>Item on page 127.</em></td>
</tr>
<tr>
<td>13</td>
<td>Add a container. For detailed information, see <em>Container on page 132.</em></td>
</tr>
<tr>
<td>14</td>
<td>Add a recipe. For detailed information, see <em>Recipe on page 141.</em></td>
</tr>
<tr>
<td>15</td>
<td>Do simulation For detailed information, see <em>Simulation on page 161.</em></td>
</tr>
<tr>
<td>16</td>
<td>RRT Calibrate the robots. For detailed information, see <em>Calibrating robot on page 195.</em></td>
</tr>
<tr>
<td>17</td>
<td>Switch to real Runtime. For detailed information, see <em>Switching to real Runtime on page 168.</em></td>
</tr>
<tr>
<td>18</td>
<td>Configure the cameras. For detailed information, see <em>Configuring camera on page 177.</em></td>
</tr>
<tr>
<td>19</td>
<td>Configure the external sensors. For detailed information, see <em>Configuring external sensor on page 181.</em></td>
</tr>
<tr>
<td>20</td>
<td>Calibrate the cameras. For detailed information, see <em>Calibrating camera on page 276.</em></td>
</tr>
<tr>
<td>21</td>
<td>Calibrate the conveyors or indexed work area. For detailed information, see <em>Calibrating linear conveyor on page 196, Calibrating circular conveyor on page 227, Calibrating indexed work area on page 260.</em></td>
</tr>
<tr>
<td>22</td>
<td>Verify the calibrations. For detailed information, see <em>Verifying conveyor calibrations on page 274.</em></td>
</tr>
<tr>
<td>23</td>
<td>Add a vision model. For detailed information, see <em>Adding vision model on page 286.</em></td>
</tr>
<tr>
<td>24</td>
<td>Start the production. For detailed information, see <em>Starting production on page 313.</em></td>
</tr>
</tbody>
</table>
4.2 Frame relationship

4.2.1 What is a coordinate system?

Overview

A coordinate system defines a plane or space by axes from a fixed point called the origin. Robot targets and positions are located by measurements along the axes of coordinate systems.

A robot uses several coordinate systems, each suitable for specific types of jogging or programming.

- The base coordinate system is located at the base of the robot. It is the easiest one for just moving the robot from one position to another. See The base coordinate system on page 88 for more information.

- The world coordinate system that defines the robot cell, all other coordinate systems are related to the world coordinate system, either directly or indirectly. It is useful for jogging, general movements and for handling stations and cells with several robots or robots moved by external axes. See The world coordinate system on page 89 for more information.

- The user coordinate system is useful for representing equipment that holds other coordinate systems, like work objects. See The user coordinate system on page 90 for more information.

- The work object coordinate system is related to the work piece and is often the best one for programming the robot. See The work object coordinate system on page 91 for more information.

- The tool coordinate system defines the position of the tool the robot uses when reaching the programmed targets. See The tool coordinate system on page 92 for more information.
The base coordinate system has its zero point in the base of the robot, which makes movements predictable for fixed mounted robots. It is therefore useful for jogging a robot from one position to another. For programming a robot, other coordinate systems, like the work object coordinate system are often better choices. See the work object coordinate system on page 91 for more information.

When you are standing in front of the robot and jog in the base coordinate system, in a normally configured robot system, pulling the joystick towards you will move the robot along the X axis, while moving the joystick to the sides will move the robot along the Y axis. Twisting the joystick will move the robot along the Z axis.
The world coordinate system has its zero point on a fixed position in the cell or station. This makes it useful for handling several robots or robots moved by external axes.

By default the world coordinate system coincides with the base coordinate system.
The user coordinate system

The user coordinate system can be used for representing equipment like fixtures, workbenches. This gives an extra level in the chain of related coordinate systems, which might be useful for handling equipment that hold work objects or other coordinate systems.

For information on how to define the user coordinate system, see information about the data type *wobjdata* in *Technical reference manual - RAPID Instructions, Functions and Data types*.

---

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>User coordinate system</td>
</tr>
<tr>
<td>B</td>
<td>World coordinate system</td>
</tr>
<tr>
<td>C</td>
<td>Base object coordinate system</td>
</tr>
<tr>
<td>D</td>
<td>Moved user coordinate system</td>
</tr>
<tr>
<td>E</td>
<td>Work object coordinate system, moved with user coordinate system</td>
</tr>
</tbody>
</table>
The work object coordinate system

The work object coordinate system corresponds to the work piece: It defines the placement of the work piece in relation to the world coordinate system (or any other coordinate system).

A robot can have several work object coordinate systems, either for representing different work pieces or several copies of the same work piece at different locations.

It is in work object coordinate systems you create targets and paths when programming the robot. This gives a lot of advantages:

- When repositioning the work piece in the station you just change the position of the work object coordinate system and all paths are updated at once.
- Enables work on work pieces moved by external axes or conveyor tracks, since the entire work object with its paths can be moved.

For information on how to define the work object coordinate system, see information about the data type `wobjdata` in *Technical reference manual - RAPID Instructions, Functions and Data types*.
The displacement coordinate system

Sometimes, the same path is to be performed at several places on the same object, or on several work pieces located next to each other. To avoid having to reprogram all positions each time a displacement coordinate system can be defined.

This coordinate system can also be used in conjunction with searches, to compensate for differences in the positions of the individual parts.

The displacement coordinate system is defined based on the work object coordinate system.

The tool coordinate system

The tool coordinate system has its zero position at the center point of the tool. It thereby defines the position and orientation of the tool. The tool coordinate system
is often abbreviated TCPF (Tool Center Point Frame) and the center of the tool coordinate system is abbreviated TCP (Tool Center Point).

It is the TCP the robot moves to the programmed positions, when executing programs. This means that if you change the tool (and the tool coordinate system) the robot’s movements will be changed so that the new TCP will reach the target. All robots have a predefined tool coordinate system, called tool0, located at the wrist of the robot. One or many new tool coordinate systems can then defined as offsets from tool0.

When jogging a robot the tool coordinate system is useful when you don’t want to change the orientation of the tool during the movement, for instance moving a saw blade without bending it.

For information on how to define the tool coordinate system, see information about the data type tooldata in Technical reference manual - RAPID Instructions, Functions and Data types.
4.2.2 Frame relationship in PickMaster® Twin

Overview
The section describes the definition of the coordinate systems regarding conveyor in PickMaster PowerPac solution.

World frame
World frame is the fundamental frame in a PickMaster PowerPac solution. The location of all the other components like robot, conveyor etc. are expressed in this frame.

Local frame
All objects have coordinate systems of its own called the local coordinate system. Object dimensions are defined with respect to this coordinated system. When the object's position is referred from other coordinate systems like WCS, the local origin of the object is used as the point of reference.

TCP(0)
Tool center position (0) is the origin position of the tool coordinate system which is expressed in the wrist coordinate system (tool0).
Conveyor frame
For linear conveyor

A frame that is located at the bottom corner of a linear conveyor as conveyor frame. This frame is fixed relative to the conveyor. The location of a conveyor is defined as the distance (3 dimensional) between the conveyor frame and the world frame expressed in the world frame. The orientation of a conveyor is defined as the angles between the conveyor frame and the world frame expressed in the world frame. Conveyor frame is used to define where the conveyor is in a PickMaster PowerPac solution but is not directly used in robot controller system.
For circular conveyor

A frame that is attached to the bottom center of a circular conveyor as conveyor frame. This frame is fixed relative to the conveyor. The location of a conveyor is defined as the distance (3 dimensional) between the conveyor frame and the world frame expressed in the world frame. The orientation of a conveyor is defined as the angles between the conveyor frame and the world frame expressed in the world frame. Conveyor frame is used to define where the conveyor is in a PickMaster PowerPac solution but is not directly used in robot controller system.

For linear conveyor

Hotspots is a frame attached to a conveyor but can be configured by user and is expressed in the conveyor frame. It is where the item or container come out when camera or IO sensor is used. And it is also the frame in which the predefined item or container is expressed.

The predefined value of x, y, z and angle Z indicate where the items or containers come out in a hotspots frame.
The axes of the hotspots frame are always parallel to the axes of the conveyor frame and the location can be configured. The orientation of hotspots frame cannot be configured.

For circular conveyor

Hotspots is a frame attached to a conveyor but can be configured by user and is expressed in the conveyor frame. It is where the item or container come out when camera or IO sensor is used. And it is also the frame in which the predefined item or container is expressed.

The predefined value of x, y, z and angle Z indicate where the items or containers come out in a hotspots frame.

Continues on next page
The Y axis of the hotspots frame is always along a radius of the circular conveyor and points outwards. The Z axis of the hotspots frame is parallel to that of the conveyor frame.

Conveyor base frame

Conveyor base frame is to define a conveyor's location and orientation relative to a robot's base frame. The concept is from ABB conveyor tracking product. This frame is to tell the robot where the conveyor is and is used to express all the items on the conveyor. To let a robot “know” where an item is, first the conveyor base frame must be defined, and then the items location and orientation need to be detected by certain sensor and is expressed in the conveyor base frame. Conveyor base frame is directly used to calculate the location and orientation of items but not explicitly used in the PickMaster PowerPac solution. For simulation the conveyor base frame is decided by clicking the calibration button in the PickMaster PowerPac. For real system the conveyor base frame is decided by certain measurements in real world.

For linear conveyor

**Tip**

I/O sensor on a linear conveyor is always perpendicular with the x direction of conveyor.
No camera and I/O sensor

X, Y and Z axes are parallel to those of conveyor frame respectively.
The location of conveyor base frame is at Hotspot0.

---

Camera is used

X, Y and Z axes are parallel to those of conveyor frame respectively.
The location of conveyor base frame is at the intersection point of the camera center axis and the top surface of the conveyor.

---

I/O sensor and predefined source type

X, Y and Z axes are parallel to those of conveyor frame respectively.
Location along X of conveyor frame is determined by I/O sensor.
Locations along Y and Z of conveyor frame are determined by hotspot0.

Note

PickMaster PowerPac only support counter-clock rotation direction. The Z axis of conveyor base frame will define the direction of positive rotation using the right-hand-rule.

For circular conveyor

Tip

I/O sensor on a circular CNV is always on a radius and points inward.

No camera and I/O sensor

X points to hotspot 0.

Location along Z if conveyor frame is determined by hotspot0.
Camera is used

X points to the intersection point of the camera center axis and the top surface of the conveyor.
Location is on the top surface of the conveyor.

I/O sensor and predefined source type

X points to the direction of the I/O sensor.
Location along Z of conveyor frame is determined by hotspot0.

Note

The Z axis of conveyor base frame will be defined the direction of positive rotation using the right-hand-rule.

Note

PickMaster PowerPac only support counter-clock rotation direction temporarily.

For more information about base frame adjustment, see Position generator on page 121.
4.3 Setting up Solution with Layout and Process in virtual Runtime (VRT)

4.3.1 Solution

**Tip**

The PickMaster® Runtime (VRT and RRT) is defined to use 50000 port. If 50000 port is occupied by other program, you will have this warning and not be able to connect to Runtime,

Release the 50000 port and restart the PickMaster® Runtime.

![Error message](image)

Use this procedure to release the 50000 port:

1. Enter the command `netstat -aon|findstr "50000"` in the CMD window.
2. The process that occupies port 50000 will be listed in the window. Obtain the PID code of the process.
3. Find the process corresponding to this PID in the task manager and close the it (Make sure that this process is allowed to be closed on this computer).
4. Restart PickMaster® Runtime and connect.

**Note**

When the SSL dialog box pops up during the first operation of the PickMaster PowerPac, click Yes.
Otherwise the PickMaster PowerPac cannot work normally.

---

**Opening a solution**

The user can create a new solution or open an existing solution from the File ribbon tab.

For more information, see *File on page 65*
4.3.2 Layout

4.3.2.1 Controller

Overview

This section describes how to add and modify a controller.

Create a controller

For more information on how to create a controller, refer to Operating manual - RobotStudio.

**WARNING**

When creating an OmniCore controller, the **Remote Start and Stop in Auto** must be selected.

Otherwise, due to RobotWare authority restrictions, the controller cannot be started remotely through PMPP and PMOP. In this case, simulation and production **CANNOT** operate normally.

![Image](xx2100000051)

**Note**

The controller must be created before adding to solution.

Adding a controller

Click **Controller** on the ribbon to add a controller in the solution.
The following table provides details about the Controller adding dialog box.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Location</strong></td>
</tr>
<tr>
<td><strong>Manage...</strong></td>
</tr>
<tr>
<td><strong>Virtual Controllers</strong></td>
</tr>
</tbody>
</table>
| **Reset system(I-start)** | The controller will reset when this is selected.  
  **Note**  
  All parameters and configuration will be restored to factory values. |
| **Import new libraries** | Add the predefined robot to the PickMaster PowerPac. |
| **Use existing station libraries** | Open an existed robot from the RobotStudio. |
| **Sync RAPID program to station** | Sync the RAPID program to the solution. |

### Procedure

On the PickMaster PowerPac ribbon-tab, click **Layout**.

Use this procedure to add a controller:

**Note**

You can only add the existing controller in the system to the solution. For more information about create controller, see [Create a controller on page 103](#).

1. On the ribbon-tab, click **Controller**.  
   The Controller adding dialog box is opened.
2. To add a folder to the **Location** list, click **...** button and then browse and select the folder to be added.
3 The Virtual Controllers lists the virtual controller systems found in the selected system folder. Click a system to select it for the new solution.

4 Select the required check boxes in Options.

**Note**

A virtual controller system that has been modified using the Modify System function of the System Builder must be restarted with the Reset System option for the changes to take effect.

5 In the dialog box, click OK to add the selected controller to the solution. The selected controller is added into the solution. The new added controller shows up in the Layout window Controller list.

6 In the Robot window, enter numbers in the Position X Y Z (mm) text box and Orientation (deg) text box according to your requirements.

7 Click OK.

**Note**

The position can be adjusted when the other parts are added. Right click on the Robot1 in the Controller list.
4 Working with PickMaster PowerPac

4.3.2.2 Gripper

Overview

This section describes how to add a gripper.

Adding a gripper

Click Gripper on the ribbon to add a gripper in the solution.

The following table provides details about the Gripper setting dialog box.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import</td>
</tr>
<tr>
<td>Import a pre-defined gripper from the library or upload an user defined gripper to the library.</td>
</tr>
<tr>
<td>Note: To upload an user defined gripper, click Add to library, browse to the local folder and select the * .rslib file, the gripper will be added to the library automatically.</td>
</tr>
<tr>
<td>Gripper Name</td>
</tr>
<tr>
<td>Set the name of the gripper.</td>
</tr>
<tr>
<td>Controller</td>
</tr>
<tr>
<td>Select a controller from the Controller list.</td>
</tr>
<tr>
<td>Available Robot</td>
</tr>
<tr>
<td>Select a robot from the Robot list.</td>
</tr>
<tr>
<td>Reference Coordinate</td>
</tr>
<tr>
<td>Select the reference coordinate for the gripper.</td>
</tr>
<tr>
<td>Position XYZ(mm)</td>
</tr>
<tr>
<td>Set the position of the gripper.</td>
</tr>
<tr>
<td>Orientation XYZ(deg)</td>
</tr>
<tr>
<td>Set the orientation of the gripper.</td>
</tr>
<tr>
<td>Mass Setting</td>
</tr>
<tr>
<td>For more details, see following section.</td>
</tr>
<tr>
<td>Activator Setting</td>
</tr>
<tr>
<td>For more details, see following section.</td>
</tr>
</tbody>
</table>

Mass Setting

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Default</td>
</tr>
<tr>
<td>Use default setting for the mass setting.</td>
</tr>
<tr>
<td>Mass</td>
</tr>
<tr>
<td>Type the mass of the tool in the Mass (kg) field.</td>
</tr>
<tr>
<td>Center of gravity</td>
</tr>
<tr>
<td>Type the coordinates of the center of gravity.</td>
</tr>
<tr>
<td>Inertia</td>
</tr>
<tr>
<td>Type the values of the inertia in Inertia (kgm²).</td>
</tr>
</tbody>
</table>

Activator Setting

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activator Using</td>
</tr>
<tr>
<td>Select the activator to be used.</td>
</tr>
</tbody>
</table>
### Activator Setting

<table>
<thead>
<tr>
<th>Activator Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add button</td>
<td>Add a new activator.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> When you need to do multiple pick, you should add enough activators for each pick. For example, if you need to pick four items and then place them, you need to add another three activators besides the default one. To do multiple pick, the Multiple-Pick rapid file in the installation package should be imported to the recipe for the required robots.</td>
</tr>
<tr>
<td>Delete button</td>
<td>Delete a selected activator.</td>
</tr>
<tr>
<td>Rapid tool data</td>
<td>Select a RAPID tool data from Tool data. The selected tool data shall be used by the RAPID program when picking with this activator.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> The RAPID program needs to be updated if more than one activator is used. For more details see, <em>Example: Double pick single place on page 382.</em></td>
</tr>
<tr>
<td>TCP Position</td>
<td>Type the coordinates of the tool center point. The tool center point defines the location on the tool where an item is attached.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> The coordinates are applied to the selected tool data during the simulation.</td>
</tr>
<tr>
<td>TCP Orientation</td>
<td>Type the orientation of the tool center point. The TCP orientation defines the desired orientation of the tool while picking up an item. The orientation shall be specified as Euler XYZ angles (degrees).</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> The orientation is applied to the selected tool data during the simulation.</td>
</tr>
<tr>
<td>Activator Signal Type</td>
<td>Choose to use the default setting or customized setting for the signal.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> The activator signal setting in PickMaster PowerPac must be exactly same with the signal setting in the connected controller. Otherwise the gripper will not pick or place the items in PickMaster PowerPac.</td>
</tr>
<tr>
<td>Default Settings</td>
<td>Shows the detailed default setting of the signal.</td>
</tr>
<tr>
<td>Customized Settings</td>
<td>Shows the detailed customized setting of the signal and allows the user to change the signals.</td>
</tr>
</tbody>
</table>

### Procedure

Use this procedure to add grippers:

*Continues on next page*
4 Working with PickMaster PowerPac

4.3.2.2 Gripper

Continued

On the PickMaster PowerPac ribbon-tab, click **Layout**.

1. On the ribbon-tab, click **Gripper**.
   The **Gripper** window opens.

2. In the **Gripper** window, enter a name in the **Gripper Name** text box or use the default one.

3. In the **Gripper** window, use default for the **Mass Setting** and **Activator Setting**.

4. Click **OK**.
4.3.2.3 Conveyor

Overview
This section describes how to add a conveyor.

Adding a conveyor
Click Conveyor on the ribbon to add a conveyor in the solution. The following table provides details about the Conveyor setting dialog box.

<table>
<thead>
<tr>
<th>Description</th>
<th>Set the name of the conveyor.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip</td>
<td>Make sure the name is unique in the current task.</td>
</tr>
<tr>
<td>Conveyor Type</td>
<td>Select the a liner conveyor or a circular conveyor.</td>
</tr>
<tr>
<td>Size (x,y,z)[mm]</td>
<td>Define the size of the conveyor.</td>
</tr>
<tr>
<td>Reference Coordinate</td>
<td>Select the reference coordinate for the conveyor.</td>
</tr>
<tr>
<td>Position XYZ(mm)</td>
<td>Set the position of the conveyor.</td>
</tr>
<tr>
<td>Orientation XYZ(deg)</td>
<td>Set the orientation of the conveyor.</td>
</tr>
</tbody>
</table>

Note
If a circular conveyor and camera or I/O sensor are used at the same time, the camera or I/O sensor MUST be set between the conveyor's hotspots and the first robot in the rotation direction. Otherwise the robots may miss the items.

Procedure
On the PickMaster PowerPac ribbon-tab, click Layout. Use this procedure to add conveyors:

1. On the ribbon-tab, click Conveyor. The Conveyor window opens.
2. In the Conveyor window, enter a name in the Conveyor Name text box or use the default one.
3. In the Conveyor window, select a type as liner or circular in the Conveyor Type drop-down list.
4. If you select a liner conveyor, in the Conveyor window, enter numbers in the Size (x,y,z)[mm] text box to define the size of the conveyor according to your requirements.
5. If you select a circular conveyor, in the Conveyor window, enter numbers in the RH Size(mm) text box to define the size of the conveyor according to your requirements.

Continues on next page
6 In the Conveyor window, enter numbers in the Position X Y Z (mm) text box and Orientation (deg) text boxes to define the location of the conveyor according to your requirements.

7 Click OK.
4.3.2.4 Camera

Overview

This section describes how to add a camera.

Adding a camera

Click Camera on the ribbon to add a camera in the solution.

The following table provides details about the Camera setting dialog box.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Set the name of the camera.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>Make sure the name is unique in the current task.</td>
</tr>
</tbody>
</table>
| **Attached to Conveyor/In-

  dex**                                   | Choose the conveyor if the camera shall be attached to a conveyor.    |
| **Entry (mm)**                            | Type an entry limit for the visible area below the camera along a conveyor. A negative value is used if the visible area starts upstreams from the camera location. |
| **Exit (mm)**                             | Type an exit limit for the visible area below the camera along a conveyor. A positive value is used if the visible area ends downstreams from the camera location. |
| **Enable vision width**                   | Select this to enable a width limitation of the visible area.         |
| **Note**                                  | Only when the Enable vision width checkbox is selected, the Left (mm) and Right (mm) values would be implemented to the setting. |
| **Left (mm)**                             | Type a limit value for the left side of the visible area. A negative value is used if the visible area ends on the left side of the camera location (from an upstream viewpoint). |
| **Note**                                  | The robot may catch air or miss some items when the Left (mm) and Right (mm) are not correctly set. |
| **Right (mm)**                            | Type a limit value for the right side of the visible area. A positive value is used if the visible area ends on the right side of the camera location (from an upstream viewpoint). |
| **Reference Coordinate**                  | Select the reference coordinate for the camera.                       |
| **Position(X,Y,Z)[mm]**                   | Set the position of the camera.                                       |
| **Orientation[deg]**                      | Set the orientation of the camera.                                    |
4 Working with PickMaster PowerPac

4.3.2.4 Camera

Continued

Note

The visible area is not limited if the camera is used with an indexed work area.

Note

The camera will not detect any objects created or placed on the other conveyors or indexed work areas.

Note

If a circular conveyor and camera or I/O sensor are used at the same time, the camera or I/O sensor MUST be set between the conveyor's hotspots and the first robot in the rotation direction. Otherwise the robots may miss the items.

Procedure

On the PickMaster PowerPac ribbon-tab, click Layout.

Use this procedure to add cameras:

1. On the ribbon-tab, click Camera.
   The Camera window opens.
2. In the Camera window, enter a name in the Camera Name text box or use the default one.
3. In the Camera window, choose a conveyor in the Attached to Conveyor/Index box to define which conveyor the new camera is attached to according to your requirements.
4. In the Camera window, use default for the other settings.
5. Click OK.
4.3.2.5 I/O sensor

Overview

This section describes how to add an I/O sensor.

Adding an I/O sensor

Click I/O sensor on the ribbon to add an I/O sensor in the solution.

The following table provides details about the I/O sensor setting dialog box.

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Set the name of the I/O sensor.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>Make sure the name is unique in the current task.</td>
</tr>
<tr>
<td>LH Size[mm]</td>
<td>The height and length of the new I/O sensor.</td>
</tr>
<tr>
<td>Attached to Conveyor/In-</td>
<td>Choose the conveyor if the sensor shall be attached to a conveyor.</td>
</tr>
<tr>
<td>dex</td>
<td></td>
</tr>
<tr>
<td>Reference Coordinate</td>
<td>Select the reference coordinate for the I/O sensor.</td>
</tr>
<tr>
<td>Position(X,Y,Z)[mm]</td>
<td>Set the position for the I/O sensor.</td>
</tr>
<tr>
<td>Orientation[deg]</td>
<td>Set the orientation of the I/O sensor.</td>
</tr>
</tbody>
</table>

Note

To function correctly, an I/O sensor must not be in contact with other stationary objects, for example, the conveyor.

Note

If a circular conveyor and camera or I/O sensor are used at the same time, the camera or I/O sensor MUST be set between the conveyor's hotspots and the first robot in the rotation direction.
Otherwise the robots may miss the items.

Procedure

On the PickMaster PowerPac ribbon-tab, click Layout.

Use this procedure to add I/O sensors:

1. On the ribbon-tab, click I/O sensor.
   The I/O sensor window opens.

2. In the I/O sensor window, enter a name in the I/O sensor Name text box or use the default one.

3. In the I/O sensor window, enter numbers in the I/O sensor Height text box to define the height of the I/O sensor according to your requirements or use default settings.

Continues on next page
4 In the I/O sensor window, enter numbers in the I/O sensor Length text box to define the length of the I/O sensor according to your requirements or use default settings.

5 In the I/O sensor window, choose a conveyor in the Attached to Conveyor/Index box to define which conveyor the new camera is attached to according to your requirements.

6 Click OK.
4.3.2.6 External sensor

Overview

The External sensor is a function that allows the users to have the full control of generating the item positions. This section describes how to add an external sensor with using any kind of sensing device or a pure virtual software sensor.

Adding an external sensor

Click External Sensor on the ribbon to add an external sensor in the solution. The following table provides details about the External Sensor setting dialog box.

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the name of the external sensor.</td>
<td>Tip</td>
</tr>
<tr>
<td>Make sure the name is unique in the current task.</td>
<td></td>
</tr>
<tr>
<td>Choose the conveyor if the sensor shall be attached to a conveyor.</td>
<td>Attached to Conveyor/Index</td>
</tr>
<tr>
<td>Select the reference coordinate for the external sensor.</td>
<td>Reference Coordinate</td>
</tr>
<tr>
<td>Set the position for the external sensor.</td>
<td>Position(X,Y,Z)[mm]</td>
</tr>
<tr>
<td>Set the orientation of the external sensor.</td>
<td>Orientation[deg]</td>
</tr>
</tbody>
</table>

Procedure

On the PickMaster PowerPac ribbon-tab, click Layout. Use this procedure to add external sensors:

2. In the External Sensor window, enter a name in the External Sensor Name text box or use the default one.
3. In the External Sensor window, choose a conveyor in the Attached to Conveyor/Index box to define which conveyor the new camera is attached to according to your requirements.
4. Click OK.
4 Working with PickMaster PowerPac

4.3.2.7 Work area

Overview

This section describes how to add a work area.

Adding a work area

The conveyor work area is an area on the conveyor where the robot picks or places items. One conveyor board is required for each conveyor work area. A robot usually has only one conveyor work area on each related conveyor, but there is no restriction.

Click Conveyor Work Area on the ribbon to add a work area in the solution.

The following table provides details about the Conveyor Work Area setting dialog box.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Work Area Name</strong></td>
</tr>
<tr>
<td>Set the name of the conveyor work area.</td>
</tr>
<tr>
<td><strong>Controller</strong></td>
</tr>
<tr>
<td>Select a controller from the list.</td>
</tr>
<tr>
<td><strong>Robot</strong></td>
</tr>
<tr>
<td>Select a robot from the list.</td>
</tr>
<tr>
<td><strong>Conveyor Board</strong></td>
</tr>
<tr>
<td>Select a conveyor board from the list.</td>
</tr>
<tr>
<td><strong>Conveyor</strong></td>
</tr>
<tr>
<td>Select a conveyor from the list.</td>
</tr>
<tr>
<td><strong>Work Area Type</strong></td>
</tr>
<tr>
<td>Select work area type from the available options.</td>
</tr>
<tr>
<td>• Pick: Select this if the work area is a picking area.</td>
</tr>
<tr>
<td>• Place: Select this if the work area is a placing area.</td>
</tr>
<tr>
<td><strong>Selection Index</strong></td>
</tr>
<tr>
<td>Select an index to specify the pick or place order in the RAPID program when using more than one pick work area and one place work area with the selected robot.</td>
</tr>
<tr>
<td><strong>Signal Type</strong></td>
</tr>
<tr>
<td>Configure the signals.</td>
</tr>
<tr>
<td>Select the Default Settings check box to use the default signal configuration.</td>
</tr>
<tr>
<td>It's recommended to use the default setting when working with virtual Runtime.</td>
</tr>
<tr>
<td>Use the Customized Settings options to manage the signals. The signals should be setting as Customized Settings accordingly when working with real Runtime. For more information, see Configuring the I/O on page 172.</td>
</tr>
</tbody>
</table>

**Note**

All the signals with "*" is a required signal.

When any of Controller, Robot or Conveyor is changed in work area setting, the user must reopen the recipe setting page to enable the modification.

Continues on next page
Conveyor work area signals

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conveyor start/stop</strong></td>
</tr>
<tr>
<td>Digital output signal. This signal is used if an overflow shall be avoided by letting the conveyor movement be controlled by the work area. The signal goes high when the conveyor shall start moving and goes low when the conveyor shall stop to avoid an overflow.</td>
</tr>
<tr>
<td><strong>Queue idle</strong></td>
</tr>
<tr>
<td>Digital output signal. This signal is high when the queue for this work area is empty. The signal goes high when the last item is retrieved from the queue.</td>
</tr>
<tr>
<td><strong>Position available</strong></td>
</tr>
<tr>
<td>Digital output signal. This signal is high when there is one or more items between the enter and exit limits for the work area.</td>
</tr>
<tr>
<td><strong>Position generator</strong></td>
</tr>
<tr>
<td>Digital input signal that tells that it is time to generate a new vision image or generate new predefined positions. This signal is ignored if a distance triggered conveyor is used.</td>
</tr>
<tr>
<td><strong>Trig</strong></td>
</tr>
<tr>
<td>If vision is used this digital output signal must be connected to the trigger input on the I/O port on the camera. If predefined positions are used this output signal must be connected directly to the start input on the conveyor encoder board. This is best done using the doManSyncX signal. If predefined positions are distributed only to this work area (For instance, Runtime with a single robot), the encoder signal cXSoftSyncSig can be used instead of doManSyncX, that is, without the need of connecting a signal to the start input of the encoder board.</td>
</tr>
<tr>
<td><strong>Strobe</strong></td>
</tr>
<tr>
<td>This is the input signal name for the strobe signal and is the start signal for the encoder board for the conveyor. The signal name is set to cXNewObjStrobe. If vision is used the signal must be generated from the strobe output on the I/O port of the camera. When predefined positions are used, the strobe may be generated directly from the doManSyncX signal, which is directly connected to the start signal on the encoder board.</td>
</tr>
</tbody>
</table>

**Note**

Using distance triggered Positions Source with DSQC2000, camera or predefined source, configure cxTrigVis as Trig signal. From RW6.10 and later, the Strobe signal is automatically configured and can therefore be omitted in the work area signal configuration.

**Procedure**

On the PickMaster PowerPac ribbon-tab, click Layout.

Use this procedure to a work area:

1. On the ribbon-tab, click **Conveyor Work Area**.
   
The **Conveyor Work Area** window opens.

2. In the **Conveyor Work Area** window, enter a name in the **Work Area Name** text box or use the default one.

3. If you created several controllers, select the required controller in **Controller**.

4. If you created several robots, select the required robot in **Robot**.

5. Select the required conveyor board in **Conveyor Board**.

6. Select the desired conveyor, set required work area type and configure the settings.

Continues on next page
4 Working with PickMaster PowerPac

4.3.2.7 Work area

Continued

7 Click OK.
4.3.2.8 Indexed work area

Overview

This section describes how to add an index work area.

Adding an indexed work area

An indexed work area is a fixed area where a robot picks or places items without conveyor.

Click Indexed Work Area on the ribbon to add an indexed work area in the solution. The following table provides details about the Indexed Work Area setting dialog box.

### Table: Indexed Work Area Setting Dialog Box

<table>
<thead>
<tr>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Work Area Name</strong></td>
<td>Set the name of the indexed work area.</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>Define the zone of the indexed work area.</td>
</tr>
<tr>
<td><strong>Work Area Type</strong></td>
<td>Select work area type from the available options.</td>
</tr>
<tr>
<td>• <strong>Pick</strong></td>
<td>Select this if the indexed work area is a picking area.</td>
</tr>
<tr>
<td>• <strong>Place</strong></td>
<td>Select this if the indexed work area is a placing area.</td>
</tr>
<tr>
<td><strong>Controller</strong></td>
<td>Select a controller from the list.</td>
</tr>
<tr>
<td><strong>Robot</strong></td>
<td>Select a robot from the list.</td>
</tr>
<tr>
<td><strong>Work object</strong></td>
<td>Select a RAPID work object data (wobjdata). The associated wobjdata is automatically used with the indexed work area.</td>
</tr>
<tr>
<td><strong>Selection Index</strong></td>
<td>Select an index to specify the pick or place order in the RAPID program when using more than one pick work area and one place work area with the selected robot.</td>
</tr>
<tr>
<td><strong>Reference Coordinate</strong></td>
<td>Select the reference coordinate for the indexed work area.</td>
</tr>
<tr>
<td><strong>Position XYZ(mm)</strong></td>
<td>Set the position for the indexed work area.</td>
</tr>
<tr>
<td><strong>Orientation XYZ(deg)</strong></td>
<td>Set the orientation of the indexed work area.</td>
</tr>
<tr>
<td><strong>Signal Type</strong></td>
<td>Configure the signals. Use the Customized Settings options to manage the signals. For more information regarding indexed work area signals see the following section. Select the Default Settings check box to use the default signal configuration.</td>
</tr>
</tbody>
</table>

Note

All the signals with "**" is a required signal.
### Indexed work area signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot execution</td>
<td>This optional digital input I/O signal is used to indicate that it is allowed for the robot to execute an item target in the RAPID program. Execution starts when the signal is high and stops when the signal goes low. If the signal goes low, all remaining items in the currently executing scene is dropped, so when the signal goes high again, the item targets for the next scene is executed. The signal must also go low after one scene is finished and then go high again to start executing item targets for the next scene.</td>
</tr>
<tr>
<td>Queue idle</td>
<td>Digital output signal. This signal is high when the queue for this work area is empty. The signal goes high when the last item is retrieved from the queue.</td>
</tr>
<tr>
<td>Position available</td>
<td>This output signal is high when there are one or more items when the Robot execution signal is high for the work area. If no Robot Execution signal is used the Position Available signal will go high as soon as there are any items in the queue.</td>
</tr>
<tr>
<td>Position generator</td>
<td>Digital input signal that tells that it is time to generate a new vision image or generate new predefined positions. This signal is ignored if a distance triggered conveyor is used.</td>
</tr>
<tr>
<td>Trig</td>
<td>If vision is used this digital output signal must be connected to the trigger input on the I/O port on the camera.</td>
</tr>
<tr>
<td>Strobe</td>
<td>This is the input signal name for the strobe signal. If vision is used, the signal must be generated from the strobe output on the I/O port of the camera. If predefined positions are used, the strobe may be generated directly by the trigger output. This is best done using a simulated output signal for the trigger signal and a logic cross connection to a simulated strobe input signal.</td>
</tr>
</tbody>
</table>

### Procedure

On the PickMaster PowerPac ribbon-tab, click Layout.

Use this procedure to add a indexed work area:

1. On the ribbon-tab, click Indexed Work Area.
   The Indexed Work Area window opens.
2. In the Indexed Work Area window, enter a name in the Work Area Name text box or use the default one.
3. Select the required work area type.
4. Click OK.
4.3.2.9 Position generator

Overview

This section describes how to set the position generator of the created solution.

Setting the position generator

Click Position Generator to define where and how positions are generated in a solution. The Position Generator should be correctly defined before the station can be calibrated.

The following table provides details about the Position Generator setting dialog box.

<table>
<thead>
<tr>
<th>Source Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available conveyor and indexed work area list</td>
<td>Selects a conveyor or indexed work area in order to set the related relationships.</td>
</tr>
<tr>
<td>Source Type</td>
<td>Select the input signal source type:</td>
</tr>
<tr>
<td>Vision</td>
<td>If the source type is set to Vision, a camera and vision models are used to find the object positions. The vision models are described in section Adding vision model on page 286.</td>
</tr>
<tr>
<td>Predefined</td>
<td>If the source type is set to Predefined, the positions generated by the position source are statically defined and no camera is used.</td>
</tr>
<tr>
<td>External</td>
<td>If the source type is set to External, an external sensor in the solution together with external position generators are used to define item positions.</td>
</tr>
<tr>
<td>Tip</td>
<td>If an indexed work area is used, external sensor function will be disabled.</td>
</tr>
</tbody>
</table>

Continues on next page
4 Working with PickMaster PowerPac

4.3.2.9 Position generator

Continued

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger Setting</td>
</tr>
<tr>
<td>Note</td>
</tr>
<tr>
<td>CAUTION</td>
</tr>
<tr>
<td>Tip</td>
</tr>
<tr>
<td>Base Frame Adjustment</td>
</tr>
</tbody>
</table>

Procedure

Note

Any modification on the source type or trigger setting requires a new calibration.

Use this procedure to set the Position Generator:

2. Click to choose one conveyor.
3. Set the source type and the trigger setting.
4. If needed, set the virtual base frame data accordingly.
5. Click to select the other conveyor and set for it.
6. Click OK.

Adjusting the base frame

When the default virtual base frame is inconsistent with the real base frame in the real station, adjust the base frame to ensure the accuracy of the pick and place in production.

The following table provides details about the Base Frame Adjustment setting dialog box.

<table>
<thead>
<tr>
<th>Base Frame Adjustment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>Select the desired conveyor or indexed work area to adjust its base frame.</td>
</tr>
</tbody>
</table>
### Base Frame Adjustment

<table>
<thead>
<tr>
<th>Base Frame Adjustment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Base Frame</td>
<td>Show current virtual base frame data and allows the user to edit the virtual base frame data.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip</strong></td>
</tr>
<tr>
<td></td>
<td>The virtual base frame data are automatically updated when the virtual station is calibrated. They can also be copied from the real station or manually edited.</td>
</tr>
<tr>
<td>Real Base Frame</td>
<td>Show current real base frame data acquired from the real controller.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip</strong></td>
</tr>
<tr>
<td></td>
<td>The real base frame data CANNOT be changed from PickMaster PowerPac.</td>
</tr>
<tr>
<td>Display Base Frame</td>
<td>Synchronize the real base frame data to virtual base frame data.</td>
</tr>
<tr>
<td>Apply</td>
<td>Select to show the base frame on the station view.</td>
</tr>
<tr>
<td>Acquire</td>
<td>Save and apply the edited virtual base frame data to the virtual controller.</td>
</tr>
<tr>
<td></td>
<td>Acquire the real base frame data from the real controller.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip</strong></td>
</tr>
<tr>
<td></td>
<td>The real base frame data only can be acquired when the real Runtime is connected. For more information about connecting to real Runtime, see <em>Switching to real Runtime on page 168</em>.</td>
</tr>
</tbody>
</table>

### Procedure

Use this procedure to adjust the virtual base frame:

1. Switch to real Runtime.
   
   For more information, see *Switching to real Runtime on page 168*.


3. Click to choose the desired conveyor or indexed work area.

4. Click in the Controller drop-down list to choose the desired controller.

5. Click Acquire to acquire the real base frame data from the real controller.

6. Click the Sync button to synchronize the data from real base frame to virtual base frame.

7. Click Yes.
8 Click **Apply**.
9 Click **Yes** in the popped-up message box to save the virtual base frame.

**Note**

If the user click **No** in this step, the virtual base frame data will not be saved.

10 If needed, click **Yes** in the coming popped-up message box to adjust the station components’ position in the station view.

After click **Yes**, the virtual base frame and real base frame will be coincident in the station view.
If the user click **No** in this step, the station components will not be moved accordingly.

11 Click to select the other conveyor and set for it.
12 Click **OK**.
4 Working with PickMaster PowerPac

4.3.2.10 Calibration

Overview

This section describes how to calibrate the created solution.

The calibration in PickMaster PowerPac is a prerequisite for running the simulation. The calibration is different with the calibration of the actual hardware (camera, conveyor, IO sensor, etc.). Running this calibration does not mean that the actual hardware calibration has been completed.

The calibration in PickMaster PowerPac is used to establish the relative relationship between the conveyor base frame and the robot base coordinate system in the virtual controller.

If a camera is used for a linear conveyor in the solution, the base frame of the conveyor is directly below the camera after calibration (x is the forward direction).

If an IO sensor is used for a linear conveyor, the base frame of the conveyor is located at the IO sensor. If predefined is used with the default setting for a linear conveyor, the base frame of the conveyor overlaps the hotspot0. For more information on frames, see Frame relationship on page 87.

Indexed work area calibration is consistent with linear conveyor’s calibration.

The calibrated base coordinate system of the circular conveyor belt is located at the center of the conveyor belt, and the x-direction points directly below the camera or along the IO sensor. If the circular conveyor uses a predefined point, the x direction points to a predefined coordinate point (hotspot).

Calibration

On the PickMaster PowerPac ribbon-tab, click Layout.

Use this procedure to calibrate:

1. Click Calibration on the ribbon-tab. Then it will start to calibrate the created solution automatically.

   The calibration runs automatically.

   Note

   If the layout in the solution changes, such as changing the camera position or robot position, redo the calibration.

For more details on frames, see Frame relationship on page 87.
4.3.3 Process

4.3.3.1 Item

Overview
An item is the object that is picked and placed by the robot. It is most common to use only one item for both pick and place but any number of items can be created. The grip location of an item defines the pick/place position relative to the item position.

This section describes how to add an item.

Adding an item
Click Item on the ribbon to add an item in the solution.

The following table provides details about the Item setting dialog box.

Item Properties tab

<table>
<thead>
<tr>
<th>Item Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Change the name.</td>
</tr>
<tr>
<td>Type</td>
<td>Change the shape of the item.</td>
</tr>
<tr>
<td></td>
<td>• Cylinder</td>
</tr>
<tr>
<td></td>
<td>• Box</td>
</tr>
<tr>
<td></td>
<td>• Customized: import predefined models.</td>
</tr>
<tr>
<td>Size(x,y,z)[mm]</td>
<td>Configure the size of the item.</td>
</tr>
</tbody>
</table>

Rapid properties

| Accepted Type   | Define the values for accepted item types. The values for the accepted item type are sent to the RAPID program and are supplied with the item targets. For more details see, GetItmTgt - Get the next item target on page 340. |
| Rejected Type   | Define the values for rejected item types. The values for the rejected item type are sent to the RAPID program and are supplied with the item targets. For more details see, GetItmTgt - Get the next item target on page 340. |

Note
If the Accepted Type or Rejected Type of different items in one solution set as the same value, the Picking Status will be influenced.
4 Working with PickMaster PowerPac

4.3.3.1 Item

Continued

Appearance Properties

| Description | Default Settings tab: choose one of the preset templates.  
|            | Default Name text box: enter the name for a new template.  
|            | Save icon: save your new template.  
|            | Delete icon: delete your templates.  
| Tip        | If you enter a new template name in the template text box, a new template will be created instead of being renamed.  
| Note       | If you directly modify the appearance of the default template instead of creating a new template, this will modify the default value of the default template. And all items created with default template will be modified too.  
| Color      | Change the color of the new item.  
| Use Texture | Use a texture image file for the item.  
| Label Location | Set the location of the label on the item.  
| Label Picture | Select an image file for the label picture.  
| Show Contour | Choose to show the contour or not.  
| Show Orientation Marker | Choose to show the orientation maker or not.  
| Browse      | Select and import a Customized model.  
| Offset [mm] | Set the offset value for the imported Customized models.  
| Orientation [deg] | Set orientation for the imported Customized models.  

Item Source tab

| Description | If the source type is set to Vision, a camera and vision models are used to find the object positions. The vision models are described in section Adding vision model on page 286.  
|            | For more information regarding Vision Models see the following section.  
| Vision     | If the source type is set to Predefined, the positions generated by the position source are statically defined and no camera is used.  
| Predefined | If the source type is set to External, an external sensor in the solution together with external position generators are used to define item positions.  
| External   | For more information, see Configuring external sensor on page 181.  

Continues on next page
### Vision

<table>
<thead>
<tr>
<th>Description</th>
<th>New Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a new vision model.</td>
<td><strong>Geometric</strong>: Add a geometric vision model. A geometric sub inspection model is configured in the same way as a PatMax model. See Configuring a geometric model with PatMax on page 289. In addition, the relative positions of the found items and the corresponding alignment hit must be trained.</td>
</tr>
<tr>
<td></td>
<td><strong>Blob</strong>: Add a blob vision model. A blob sub inspection model is configured in the same way as a blob model. See Configuring blob models on page 297. In addition, the number of required hits must be configured.</td>
</tr>
<tr>
<td></td>
<td><strong>Inspection</strong>: Add an inspection vision model.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Import Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import an existed vision model.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>External</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add an external sensor. An external sensor is a software component that gives external partners full control of how item positions are generated. An external sensor can use any type of item detection such as barcode readers, cameras, or a combination of photo sensors to generate item positions.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit the selected vision model.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Export</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export the selected vision model.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete the selected vision model.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Rename</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename the selected vision model.</td>
<td></td>
</tr>
</tbody>
</table>

### Predefined

<table>
<thead>
<tr>
<th>Description</th>
<th>Position(X,Y,Z)[mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the position for the predefined model.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Angle Z[deg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the angle on Z axis of the predefined model.</td>
<td></td>
</tr>
</tbody>
</table>

### External

**Tip**

The External configuration for items/containers can only be implemented when real Runtime is connected.

<table>
<thead>
<tr>
<th>Description</th>
<th>New position generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add an external position generator. When users have not created the position generator for this sensor before, they have to click the new position generator button first. Then the python interface of def configurePosGen(self, posGenId) will be automatically called, which is the same as the next operation “Configure”. The prerequisite of this operation is that the corresponding external sensor has already been configured according to section 3.2, otherwise there will be a message box showing “The current sensor is not configured. Please configure the sensor before creating the position generator.” For more information on configuring an external position generator, see Configuring external sensor on page 181.</td>
<td></td>
</tr>
</tbody>
</table>

Continues on next page
### 4 Working with PickMaster PowerPac

#### 4.3.3.1 Item

**Description**

| Description | Sync Time[MS] | The time of RT received strobe signal is calculated by the current system time (StrobeTime) minus the time of data process (ITimeSinceStrobe). But the time of from controller trigger strobe signal to RT received strobe signal cannot be calculated. So the value of Synchronization time is used to compensate for this value. This value will be set by users to compensate the time spent for signal transmission on hardware and invoking function. For different external sensor, this value may be set differently.

| Configure | Once the position generator is created and configured, users could click the button of Configure to do configuration again. This operation refers to the Python interface of `def configurePosGen(self, posGenId)`. Users should self-define the position generator configuration behavior in this interface in their own Python class. Although users could only create one position generator in PMPP UI, users could implement more position generation methods in this interface, so that positions could be generated based on one or more methods. The same as sensor configuration, the position generator configuration information should be serialized into a string, so that PMPP solution could get and save this string. This button could be clicked as long as its button state is enabled. If the current row is in disabled state, the corresponding position generator could not be configured until it enters configuration – enabled state.

| Delete | Delete the selected position generator.

| Save | In the save – enabled state, users could click “Save” button to get the configuration string from the Python program and update in PMPP. This button refers to the Python interface “def savePosGen(self, posGenId)” which is provided by PMTW developer in ExternalSensorInterface.py file and users should not modify the interface content. The content only contains returning the configuration string, so users should make sure that all configured information are included in this string in the “configurePosGen” interface. After “Save” button is clicked, all rows will enter configuration - enabled state.

| OK | The “OK” button is for the item/container view. When this button is clicked, all data will be saved, and the item/container view will be closed. If one external sensor position generator is in save – enabled state, the “savePosGen” Python interface will firstly be called before the view is closed.

| Cancel | The “Cancel” button is for the item/container view. When this button is clicked, all modified data will be abandoned, and the item/container view will be closed.

### Procedure

On the PickMaster PowerPac ribbon-tab, click **Process**.

Use this procedure to add an item:

1. On the ribbon-tab, click **Item**.

   The Item window opens.

2. In the RH Size part, define the item’s size.

   The height of the item defines the pick height and is always added to items found by a vision model or a position defined by a predefined position source.

3. If needed, define levels for accepted or rejected item types.

Continues on next page
4 Working with PickMaster PowerPac

4.3.3.1 Item
Continued

When inspection is used, a found item will be marked as either accepted or rejected. The values for accepted and rejected item type in the Item Configuration dialog are sent to the RAPID program and are processed there. See Configuring inspection models on page 304.

4 Click OK.

Related information

Configuring inspection models on page 304.
4.3.3.2 Container

Overview

A container defines which patterns to use and what items to use for each position in the patterns. This way, different containers can use the same patterns but with different items.

This section describes how to add a container.

Prerequisites

At least one item must be defined in the solution before configuring the container.

Adding a container

Click Container on the ribbon to add a container in the solution.

The following table provides details about the Container setting dialog box.

Container Properties tab

<table>
<thead>
<tr>
<th>Container Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container Name</td>
<td>Change the name.</td>
</tr>
<tr>
<td>LWH Size (mm)</td>
<td>Configure the size of the container.</td>
</tr>
<tr>
<td>Type</td>
<td>Define the type of the container.</td>
</tr>
<tr>
<td></td>
<td>• Box</td>
</tr>
<tr>
<td></td>
<td>• Customized: import predefined models.</td>
</tr>
</tbody>
</table>

Appearance Properties

<table>
<thead>
<tr>
<th>Appearance Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template</td>
<td>Default Settings tab: choose one of the preset templates. Default Name text box: enter the name for a new template. Save icon: save your new template. Delete icon: delete your templates.</td>
</tr>
<tr>
<td></td>
<td>Tip</td>
</tr>
<tr>
<td></td>
<td>If you enter a new template name in the template text box, a new template will be created instead of being renamed.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>If you directly modify the appearance of the default template instead of creating a new template, this will modify the default value of the default template. And all containers created with default template will be modified too.</td>
</tr>
<tr>
<td>Color</td>
<td>Change the color of the container.</td>
</tr>
<tr>
<td>Use Texture</td>
<td>Use a texture image file for the container.</td>
</tr>
<tr>
<td>Label Location</td>
<td>Set the location of the label on the container.</td>
</tr>
<tr>
<td>Label Picture</td>
<td>Select an image file for the label picture.</td>
</tr>
<tr>
<td>Show Contour</td>
<td>Choose to show the contour or not.</td>
</tr>
</tbody>
</table>
Choose to show the orientation maker or not.

Show Orientation Marker

Select and import a Customized model.

Browse

Set the offset value for the imported Customized models.

Offset [mm]

Set orientation for the imported Customized models.

Orientation [deg]

Container Pattern tab

A pattern defines a collection of positions. For example, a box with predefined locations for certain objects. You can change the order, delete, or rearrange the selected layers using the available options. You can adjust the vertical position of each layer by modifying the Offset (mm). You can also manage the sorting method. The Sorting Method section defines the order in which the items in the container pattern shall be handled by the robots.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Layer</td>
</tr>
<tr>
<td>Edit Layer</td>
</tr>
<tr>
<td>Copy</td>
</tr>
<tr>
<td>Delete Layer</td>
</tr>
<tr>
<td>Up</td>
</tr>
<tr>
<td>Down</td>
</tr>
<tr>
<td>Delete All</td>
</tr>
<tr>
<td>Total Weight</td>
</tr>
<tr>
<td>Total Height</td>
</tr>
<tr>
<td>Total Count</td>
</tr>
</tbody>
</table>

Add Layer

Select one available item that has been created.

Add icon: add the selected item onto the layer.
Delete icon: delete the selected items.
Select All icon: select all the items in the layer.

Align Style

Define the align style when you have more than one item in the layer.

Left Align icon: align all the items in this layer from the left.
Center Align icon: align all the items in this layer from the center.
Right Align icon: align all the items in this layer from the right.
Top Align icon: align all the items in this layer at from top.
Middle Align icon: align all the items in this layer from the middle.
Bottom Align icon: align all the items in this layer from the bottom.

Distribute Style

Define the distribution style when you have more than one item in the layer.

Horizontally icon: distribute all the items in the horizontal direction.
Vertically icon: distribute all the items in the vertical direction.
### Container Source tab

<table>
<thead>
<tr>
<th>Description</th>
<th>Vision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vision</td>
<td></td>
</tr>
<tr>
<td>Predefined</td>
<td></td>
</tr>
<tr>
<td>External</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

- **Vision:** If the source type is set to Vision, a camera and vision models are used to find the object positions. The vision models are described in section [Adding vision model on page 286](#). For more information regarding Vision Models see the following section.

- **Predefined:** If the source type is set to Predefined, the positions generated by the position source are statically defined and no camera is used.

- **External:** If the source type is set to External, an external sensor in the solution together with external position generators are used to define container positions. For more information, see [Configuring external sensor on page 181](#).

### Vision

<table>
<thead>
<tr>
<th>Description</th>
<th>New Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Model</td>
<td>Add a new vision model.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Geometric:</strong> Add a geometric vision model.</td>
</tr>
<tr>
<td></td>
<td>A geometric sub inspection model is configured in the same way as a PatMax model. See <a href="#">Configuring a geometric model with PatMax on page 289</a>. In addition, the relative positions of the found items and the corresponding alignment hit must be trained.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Blob:</strong> Add a blob vision model.</td>
</tr>
<tr>
<td></td>
<td>A blob sub inspection model is configured in the same way as a blob model. See <a href="#">Configuring blob models on page 297</a>. In addition, the number of required hits must be configured.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Inspection:</strong> Add an inspection vision model.</td>
</tr>
</tbody>
</table>

**Import Model**

Import an existed vision model.

---

Continues on next page
Add an external sensor.

An external sensor is a software component that gives external partners full control of how item positions are generated. An external sensor can use any type of item detection such as barcode readers, cameras, or a combination of photo sensors to generate item positions.

Edit

Edit the selected vision model.

Export

Export the selected vision model.

Delete

Delete the selected vision model.

Rename

Rename the selected vision model.

Predefined

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Position</strong>(X,Y,Z)[mm]</td>
</tr>
<tr>
<td><strong>Angle Z</strong>[deg]</td>
</tr>
</tbody>
</table>

External

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New position generator</strong></td>
</tr>
<tr>
<td><strong>SYNC TIME</strong>[MS]</td>
</tr>
</tbody>
</table>

Tip

The External configuration for items/containers can only be implemented when real Runtime is connected.
Once the position generator is created and configured, users could click the button of Configure to do configuration again. This operation refers to the Python interface of `def configurePosGen(self, posGenId)`. Users should self-define the position generator configuration behavior in this interface in their own Python class. Although users could only create one position generator in PMPP UI, users could implement more position generation methods in this interface, so that positions could be generated based on one or more methods.

The same as sensor configuration, the position generator configuration information should be serialized into a string, so that PMPP solution could get and save this string.

This button could be clicked as long as its button state is enabled. If the current row is in disabled state, the corresponding position generator could not be configured until it enters configuration-enabled state.

Delete the selected position generator.

Save

In the save-enabled state, users could click “Save” button to get the configuration string from the Python program and update in PMPP. This button refers to the Python interface “def savePosGen(self, posGenId)” which is provided by PMTW developer in ExternalSensorInterface.py file and users should not modify the interface content. The content only contains returning the configuration string, so users should make sure that all configured information are included in this string in the “configurePosGen” interface. After “Save” button is clicked, all rows will enter configuration-enabled state.

OK

The “OK” button is for the item/container view. When this button is clicked, all data will be saved, and the item/container view will be closed. If one external sensor position generator is in save-enabled state, the “savePosGen” Python interface will firstly be called before the view is closed.

Cancel

The “Cancel” button is for the item/container view. When this button is clicked, all modified data will be abandoned, and the item/container view will be closed.

### Procedure

On the PickMaster PowerPac ribbon-tab, click **Process**.

Use this procedure to add a container:

1. On the ribbon-tab, click **Container**.
   
   The **Container** window opens.

2. Define the container with your requirements in the **Container Properties** tab.

3. Define the container pattern with your requirements in the **Container Pattern** tab.

4. In the **Container Pattern** tab, click **New Layer** to define a layer in the container.

5. If need, adjust the layout of the items on the layer.
   
   A. Select all items on the layer.
   
   B. Click ‘Ctrl’ and select the base item at the same time.
   
   C. Click **Left** to align all items on the left edge according to the base item.

   Click **Right** to align all items on the right edge according to the base item.

Continues on next page
Click **Center** to align all items on the centre line vertically according to the base item.

Click **Middle** to align all items on the centre line horizontally according to the base item.

Click **Top** to align all items on the top edge according to the base item.

Click **Bottom** to align all items on the bottom edge according to the base item.

D Click **Horizontally** to set all items tangent in horizontal direction.

Click **Vertically** to set all items tangent in vertical direction.

6 Click **Save**.

The layer layout is saved.

7 Click **OK**.

The container is saved and the window is closed.
4 Working with PickMaster PowerPac

4.3.3.3 Flow

Overview

A flow is used to define how the items and containers are to be generated in the simulation. A flow can be used to simulate the random and irregular incoming material flow in reality. A flow is attached to a hotspot on a conveyor or an indexed work area. When attaching the flow, the hotspot becomes a source from where items and containers appear in the simulation according to the flow configuration. Following are the two types of flows:

- Layout: A Layout flow is a predefined layout that is periodically regenerated at the hotspot. The layout may have some random variation regarding the locations of items or containers and the availability of them. The layout may consists of different items or container patterns.

- Recorded: A recorded flow is a recording of a sensor from a simulation or production. The recording is exported from PickMaster PowerPac as an xml file having information of all the detected items and containers during a time interval. When the file is imported, the items detected are mapped to the configured items and container patterns.

Note

Only when the source type of the work area is set as Vision, the flow can used.

Adding a flow

Click Flow on the ribbon to add a flow in the solution.

The following table provides details about the Flow setting dialog box.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout</td>
</tr>
<tr>
<td>Recorded</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Flow Type</td>
</tr>
<tr>
<td>LW Size [mm]</td>
</tr>
<tr>
<td>Stability</td>
</tr>
<tr>
<td>Position Stability</td>
</tr>
</tbody>
</table>

Continues on next page
4 Working with PickMaster PowerPac

4.3.3.3 Flow

Continued

| Description | X pos dev min/max [mm] | Defines the minimum and maximum deviation of the X position from the correct value.

| Y pos dev min/max [mm] | Defines the minimum and maximum deviation of the Y position from the correct value.

| Orientation Stability | If set to 100%, the generated items always have correct orientation. A lower value defines the probability that an item gets correct orientation.

| Z pos dev min/max [deg] | Defines the minimum and maximum deviation of the Z position from the correct value.

| Rejection Ratio | Defines the probability that an item becomes rejected by a camera. If set to 0%, the item setting “Rejected” in the Layout will decide if the item is rejected.

Edit Layout

| Description | Available Obj | Select one item or container you have created for this system.

Add icon: add the selected item or container onto the layer.

Delete icon: delete the selected items.

Select All icon: select all the items in the layer.

Align Style | Define the align style when you have more than one item in the layer.

Left Align icon: align all the items in this layer from the left.

Center Align icon: align all the items in this layer from the center.

Right Align icon: align all the items in this layer from the right.

Top Align icon: align all the items in this layer at from top.

Middle Align icon: align all the items in this layer from the middle.

Bottom Align icon: align all the items in this layer from the bottom.

Distribute Style | Define the distribution style when you have more than one item in the layer.

Horizontally icon: distribute all the items in the horizontal direction.

Vertically icon: distribute all the items in the vertical direction.

Else Functions | Rotate icon: rotate the selected items.

Order | Define the order of the layer.

Position [mm] | Define the position of the item in the layer.

Angle [deg] | Define the angle of the item in the layer.

Show Item Name | Shows the name of the items.

Show Item Order | Shows the added order of the items.

Procedure

On the PickMaster PowerPac ribbon-tab, click Process.

Use this procedure to add a rectangle flow:

1. On the ribbon-tab, click Flow.

The Flow window is opened.

2. Select a type for the flow in Flow Type.

3. Click the Edit Layout icon to open the dialog.
4 Working with PickMaster PowerPac

4.3.3.3 Flow

Continued

4 Click the **Add** icon in the **Edit Layout** dialog to add an item.
   A Select all items on the layer.
   B Click 'Ctrl' and select the base item at the same time.
   C Click **Left** to align all items on the left edge according to the base item.
      Click **Right** to align all items on the right edge according to the base item.
      Click **Center** to align all items on the centre line vertically according to the base item.
      Click **Middle** to align all items on the centre line horizontally according to the base item.
      Click **Top** to align all items on the top edge according to the base item.
      Click **Bottom** to align all items on the bottom edge according to the base item.
   D Click **Horizontally** to set all items tangent in horizontal direction.
      Click **Vertically** to set all items tangent in vertical direction.

5 Click **OK** to apply the configuration.

6 Click **OK** to close the **Flow** dialog.
4.3.3.4 Recipe

Overview

This section describes how to add a recipe. In one solution, several recipes can be created. All elements (Robots, sensor and so on) in this solution can be added to any recipes with no limits.

Adding a recipe

Click Recipe on the ribbon to add a recipe in the solution. The following table provides details about the Recipe setting dialog box.

Properties

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Devices</td>
<td>Define the available devices, including robots and conveyors. All robots and conveyors in the same solution will be listed in every recipe, but they can have different attribute settings in different recipes. For example, the speed of the same robot can be different in different recipes. For more information regarding Available Device see the following section.</td>
</tr>
<tr>
<td>Available Workareas</td>
<td>Define the available work areas. All work areas in the same solution will be listed in every recipe, but they can have different attribute settings in different recipes. For more information regarding Available Work Areas see the following section.</td>
</tr>
</tbody>
</table>

Available Devices

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Robot Setting     | **Note**
|                   | If there are more than one robot in this system, all the robot will be listed here with their defined name. **Rapid Editor**: specify the editor to open Rapid. **Speed**: change the speed of the robot. **Rapid**: import/export/edit the Rapid program of the robot. |
|                   | **Note**
|                   | The default RAPID module is created for IRB 360. Alternative RAPID template modules for different robot type categories and for double picking can be imported from the installation folder: C:\Program Files (x86)\ABB\PickMaster Twin 2\PickMaster Twin Client 2\PickMaster PowerPac\RAPID. |
| Conveyor Setting  | **Speed**: change the speed of the conveyor. **Acceleration**: change the acceleration of the conveyor. **Deceleration**: change the deceleration of the conveyor. |

Continues on next page
### Available Work Areas

<table>
<thead>
<tr>
<th>Pick Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pick/place elevation</strong></td>
<td>is the distance, in negative z-direction relative to the tool, from where the robot approaches the item target.</td>
</tr>
<tr>
<td><strong>Pick/place time</strong></td>
<td>is the time the robot is in the pick/place position. If the conveyor is moving during the pick/place time, the robot will track along the conveyor to keep the relative position on the moving conveyor.</td>
</tr>
<tr>
<td><strong>Vacuum activation</strong></td>
<td>is the time in seconds before the middle of the corner path of the approaching position, when the vacuum I/O should be set. If a negative value is entered, the vacuum I/O will be set the time after the middle of the corner path. This value is only valid for work areas of type Pick.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Vacuum activation does not affect the picking of items in simulation. Items are attached to the picking tool using SimAttach events, for example, in the Pick Routine.</td>
</tr>
<tr>
<td><strong>Vacuum reversion</strong></td>
<td>is the time in seconds before the half place time in the place position, when the blow I/O should be set. If a negative value is entered, the blow I/O will be set the time after the half place time in the place position. This value is only valid for work areas of type Place.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Vacuum reversion does not affect the placing of items in simulation. Items are detached from the picking tool using SimDetach events, for example, in the Place Routine.</td>
</tr>
<tr>
<td><strong>Vacuum off</strong></td>
<td>is the time in seconds after the half place time in the place position, when the blow I/O should be reset. If a negative value is entered, the blow I/O will be reset the time before the half place time in the place position. This value is only valid for work areas of type Place.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Vacuum Off does not affect the placing of items in simulation. Items are detached from the picking tool using SimDetach events, for example, in the Place Routine.</td>
</tr>
</tbody>
</table>
After you define a start entry in a work area which may called Start X, you can define a same start entry which may called Start Y at the vertical direction of the Start X.

**Enter** is the limit from where the robot starts to execute item targets on the work area (Start X). The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving direction of the conveyor. Make sure that the enter limit can be reached by the robot.

**Start** is when the next item to execute on the conveyor is above this limit, the conveyor is started. The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving direction of the conveyor.

**Stop** is when an item on the conveyor reaches this limit, the conveyor is stopped. The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving direction of the conveyor.

**Exit** is the limit from where the robot considers an item target as lost on the work area (Start X). The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving direction of the conveyor. When the tracked item passes beyond this limit it will be dropped. This limit must be chosen well within the maximum reach of the robot. The robot must be able to reach this position from an arbitrary position in the robot’s working area before the position is out of reach.

Select the **Use Start/Stop** checkbox if the work area should supervise the start and stop limits.

**Note**

Start and Stop values should be within boundaries of Enter and Exit limits. The value of Enter MUST be smaller than the value of Start. The value of Stop MUST be smaller than the value of Exit.

Otherwise there will be some errors during simulation.

**Note**

When **Use Start/Stop** checkbox is selected, the distance between Stop and Exit should be larger than the size (x direction) of the container.

This is handled by the Conveyor start/stop signal, see *Work area on page 116*.

Select the **Start with production** checkbox if the work area should work with the conveyor when the production is started, and stopped when the production is stopped.

**ConveyorYMax** is the limit from where robot considers an item target as lost on the work area in End Y. The distance is calculated in millimeter from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving vertical direction of the conveyor.

Make sure that the ConveyorYMax can be reached by the robot. If the y coordinate value of the item’s position is greater than the ConveyorYMax, the robot will not grab the item. So when the tracked item passes beyond this limit it will be dropped. This limit must be chosen well within the maximum reach of the robot.
### Description

ConveyorYMin is the limit from where robot starts to execute item targets on the work area in Start Y. The distance is calculated in millimeter from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving vertical direction of the conveyor.

Use Y Max/Y Min checkbox if the work area should supervise the upper and lower limits.

---

### Diagram

![Diagram](image.png)

### Table

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Camera and Baseframe origin</td>
</tr>
<tr>
<td>B</td>
<td>Camera</td>
</tr>
<tr>
<td>C</td>
<td>Enter</td>
</tr>
<tr>
<td>D</td>
<td>Start</td>
</tr>
<tr>
<td>E</td>
<td>Stop</td>
</tr>
<tr>
<td>F</td>
<td>Exit</td>
</tr>
<tr>
<td>G</td>
<td>Robot</td>
</tr>
<tr>
<td>H</td>
<td>Image frame</td>
</tr>
<tr>
<td>I</td>
<td>Center of Robot</td>
</tr>
<tr>
<td>J</td>
<td>Y Max</td>
</tr>
<tr>
<td>K</td>
<td>Y Min</td>
</tr>
</tbody>
</table>

### Note

The reference origin for Enter, Exit, Start, and Stop is I (Center of Robot). The reference base for Y Max and Y Min is the conveyor base frame.

---

### Record Setting

Record the position of the items and containers in simulation and production.

#### Note

When Record scenes is selected and saved for any work area, the following message will pop up.

```plaintext
Scenes recording is activated for: {0}
```

After this, the recording will be activated automatically when the simulation or production is started.
Operation

The operation contains pick operation and place operation.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main Setting</strong></td>
</tr>
<tr>
<td><strong>Filter Setting</strong></td>
</tr>
<tr>
<td><strong>User Script</strong></td>
</tr>
<tr>
<td><strong>Distribution Setting</strong></td>
</tr>
</tbody>
</table>

Main Setting

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation Name</strong></td>
</tr>
<tr>
<td><strong>Operation Type</strong></td>
</tr>
<tr>
<td><strong>Associated Conveyor or Indexed WA</strong></td>
</tr>
<tr>
<td><strong>Select Flow</strong></td>
</tr>
<tr>
<td><strong>Select Hotspot</strong></td>
</tr>
<tr>
<td><strong>Select Object</strong></td>
</tr>
<tr>
<td><strong>Object Generation Distance[mm]/[degree]</strong></td>
</tr>
<tr>
<td><strong>Tip</strong></td>
</tr>
<tr>
<td><strong>Trigger Distance[mm]/[degree]</strong></td>
</tr>
<tr>
<td><strong>Note</strong></td>
</tr>
</tbody>
</table>
Different conditions for using **Object Generation** and **Trigger Distance**

As the **Object Generation Distance [mm] [degree]** and **Trigger Distance [mm] [degree]** are valid in different conditions, we list all conditions with their different options as below:

<table>
<thead>
<tr>
<th>Source Type</th>
<th>Trigger Setting</th>
<th>Object Generation Distance [mm] [degree]</th>
<th>Trigger Distance [mm] [degree]</th>
<th>Main Setting view</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conveyor</strong> Vision/External Sensor</td>
<td>Distance</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td><strong>Conveyor</strong> Vision/External Sensor</td>
<td>I/O</td>
<td>Available</td>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td><strong>Conveyor</strong> Predefine</td>
<td>I/O</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Indexed work area</strong> Vision/External Sensor</td>
<td>Distance</td>
<td>Unavailable</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Indexed work area</strong> Vision/External Sensor</td>
<td>Distance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Indexed work area</strong> Predefine</td>
<td>I/O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Indexed work area</strong> Predefine</td>
<td>I/O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Continues on next page
### Filter Setting

<table>
<thead>
<tr>
<th>Filter Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Position Filter Distance</strong></td>
<td>The position filter defines the minimum allowed distance between the different item positions found by a camera or an external sensor. For example, if two or more models are used to identify the same object, there might be one hit for each model at almost the same location. If two positions for the same item are closer in either x- or y-direction than the defined minimum item distance, only the position with the highest sort value will be sent to the robot controller. The sort value can be set for each vision model, see <a href="#">Adding vision model on page 286</a>. If <strong>Same level only</strong> is selected, the filtering will only be done between item positions with the same inspection level.</td>
</tr>
<tr>
<td><strong>Overlap Filter Distance</strong></td>
<td>For linear conveyor, items can be identified in two consecutive frames due to the overlap. The models can have a small variation in the pick/place position between these frames. Items that are found in two consecutive frames and whose pick/place position between these two frames does not vary by more than the overlap filter distance will be regarded as one item. The first identified hit is sent to the robot, and any subsequent hit is filtered out.</td>
</tr>
<tr>
<td><strong>Overlap Filter Angle</strong></td>
<td>For circular conveyor, items can be identified in two consecutive frames due to the overlap. The models can have a small variation in the pick/place position between these frames. Items that are found in two consecutive frames and whose pick/place position between these two frames does not vary by more than the overlap filter angle will be regarded as one item. The first identified hit is sent to the robot, and any subsequent hit is filtered out. For circular conveyor, <strong>Overlap Filter Distance</strong> and <strong>Overlap Filter Angle</strong> are both valid. Which one works depends on which filtering condition is more stringent.</td>
</tr>
</tbody>
</table>

### User Script

**User Script** is a software component that can be designed to customize item positions during runtime.

With this function, user can customize the item position generation, adjustment, filter, or distribution according to their own requirements to achieve user-defined picking and placing of items.

The **User Script** can be queried for positions instead of using predefined positions. It is also possible for **User Script** objects to adjust item positions generated by vision models in PickMaster PowerPac. Item positions carry some free usage
parameters that can be set by the user script. These parameters can later on be accessed in RAPID by the robot that handles the position.

**Note**

Only the qualified personnel can write or modify the script files. It is the responsibility of the writer to make sure that the cell is safety when running with the script files.

**Note**

Only native Python is supported in PickMaster® Twin products. Any third-party libraries CANNOT be directly referenced in the script.

**Tip**

Syntax errors will cause the script files fail to run. With the following way to avoid the syntax errors:

1. Keep to use the same editor for the same script file.
2. It is recommended to use PyCharm or Notepad++ to edit the script files, as they have syntax checking capabilities for Python files.

**CAUTION**

It is the responsibility of the integrator to implement that local presence is set up in a correct way.

It is the responsibility of the integrator to implement that single point of control is set up in a correct way.

**DANGER**

Protect the script carefully if it is used in the production. Anyone who has access to the script can modify the script directly. This may cause serious danger.

**Note**

The user script and external sensor cannot be used at the same time in one recipe.

**Note**

Python script files will not be included in the Pack&Go file. Copy the Python script files to the desired destination.

Continues on next page
When the User Script checkbox is selected, the User Script setting content will show up.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Script Name</strong></td>
</tr>
<tr>
<td><strong>Tip</strong></td>
</tr>
<tr>
<td><strong>Configure Interface</strong></td>
</tr>
<tr>
<td><strong>Object List</strong></td>
</tr>
</tbody>
</table>

**Supported User Script interface types**

**PickMaster Twin supports four types of User Script.**

<table>
<thead>
<tr>
<th>User script interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialize Interface</strong></td>
<td>This interface is used to provide the user to initialize the User Script program, such as: initialize the parameters, etc.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>Initialize Interface will be executed only once when the the Start is clicked. The other three interfaces will be executed when DSQC 2000 or DSQC 377 signals are triggered.</td>
</tr>
<tr>
<td><strong>Adjuster Interface</strong></td>
<td>This interface is used to provide the user to realize the customized item position generation and adjustment. Each time the model generates positions, an array with the positions is sent to the User Script object. The User Script object can then control the positions in any desired way. Positions can be changed, removed, or added.</td>
</tr>
<tr>
<td><strong>Vision Interface</strong></td>
<td>This interface is used to provide the user to realize the customized item position filter and adjustment by vision result. This interface will be invoked when the Runtime execute to the item recognition section in production.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>The Vision Interface can only be used in Production. The other three interfaces can be used in Production and Simulation.</td>
</tr>
<tr>
<td><strong>Distribution Interface</strong></td>
<td>This interface is used to provide the user to realize the customized distribution function. This interface will be invoked when the item distribution executes.</td>
</tr>
</tbody>
</table>

**Files provided in the installation**

There are two ways to customize PickMaster PowerPac with a User Script. Either create a new User Script, or implement an existing user script file.

*Continues on next page*
## Creating a new User Script

### Initialize Interface

**PyInitialize: Initialize data**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Explain</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Runtime type</td>
<td></td>
</tr>
</tbody>
</table>
| itemInfo | Item information, which contains 
{Key} Name:{}, Id:{}. For example: 
itemInfo = |
|          |             | 0: VRT  |
|          |             | 1: RRT  |
|          |             | Name: name of the item |
|          |             | Id: ID of the item |

**Example:**

```python
# PyInitialize Initialize

def PyInitialize(type, itemInfo):
    global BType
    global Item_1
    global Item_2
    BType = type
    f = open(logpath, 'w')
    f.write("PyInitialize()")
    BType = "BType:\n" + str(BType) + "\n"
    f.write(BType)"x
    itemInfo = itemInfo.keys()
    for key in keys:
        if itemInfo[key]["Name"] == "Item_1":
            Item_1 = itemInfo[key]["Id"]
        elif itemInfo[key]["Name"] == "Item_2":
            Item_2 = itemInfo[key]["Id"]
        f.write((strLine))
    f.close
```

xx2200001780
### Adjuster Interface PyAdjuster: Modify position

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Explain</th>
</tr>
</thead>
</table>
| items    | Item information, which contains Time: X: Y: Z: RX: RY: RZ: Tag: Val1: Val2: Val3: Val4: Val5: Level: Id: | • Time: time stamp(s), get the number of milliseconds since 1 Jan 1970  
  • X: the location value of the item in X direction  
  • Y: the location value of the item in Y direction  
  • Z: the location value of the item in Z direction  
  • RX: the rotation angle value of the item in X direction  
  • RY: the rotation angle value of the item in Y direction  
  • RZ: the rotation angle value of the item in Z direction  
  • Tag: used in rapid  
  • Val1, Val2, Val3, Val4, Val5: optional value, used in rapid  
  • Level: inspection level  
  - 0: Discarded  
  - 1: Rejected  
  - 2: Accepted  
  • Id: ID of the item |

For example:
```python
items = {
    'Time': 1666849507.969,
    '0': {'X': 0.0,
          'Y': 150.0,
          'Z': 0.0,
          'RX': 0.0,
          'RY': 0.0,
          'RZ': 0.0,
          'Tag': 0,
          'Val1': 0.0,
          'Val2': 0.0,
          'Val3': 0.0,
          'Val4': 0.0,
          'Val5': 0.0,
          'Level': 2,
          'Id': '35139a6c-56a8-437d-b180-7f40e49bf6ff'
}
```

Example:
```python
items = {
    'Time': 1666849507.969,
    '0': {'X': 0.0,
          'Y': 150.0,
          'Z': 0.0,
          'RX': 0.0,
          'RY': 0.0,
          'RZ': 0.0,
          'Tag': 0,
          'Val1': 0.0,
          'Val2': 0.0,
          'Val3': 0.0,
          'Val4': 0.0,
          'Val5': 0.0,
          'Level': 2,
          'Id': '35139a6c-56a8-437d-b180-7f40e49bf6ff'
}
```

Continues on next page
Vision Interface PyVision: Recognize items by reanalyzing image

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Explain</th>
</tr>
</thead>
<tbody>
<tr>
<td>imageData</td>
<td>Image data, which contains Width: {} Height: {} IsColor: {}</td>
<td>- Width: image width in pixel</td>
</tr>
<tr>
<td></td>
<td>• Grey: {}</td>
<td>• Height: image height in pixel</td>
</tr>
<tr>
<td></td>
<td>• Blue: {}\nGreen: {}\nRed: {}\n</td>
<td>• IsColor:</td>
</tr>
<tr>
<td></td>
<td>For example: Grey image</td>
<td>- 0: Grey image</td>
</tr>
<tr>
<td></td>
<td>imageData = {</td>
<td>- 1: Colorful image</td>
</tr>
<tr>
<td></td>
<td>'Width': 481,</td>
<td>• Grey: grey data, valid from 0 to 255</td>
</tr>
<tr>
<td></td>
<td>'Height': 409,</td>
<td>• Blue: blue data, valid from 0 to 255</td>
</tr>
<tr>
<td></td>
<td>'IsColor': 0,</td>
<td>• Green: green data, valid from 0 to 255</td>
</tr>
<tr>
<td></td>
<td>'Grey': [56,...,67]</td>
<td>• Red: red data, valid from 0 to 255</td>
</tr>
<tr>
<td></td>
<td>}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Colorful image</td>
<td></td>
</tr>
<tr>
<td></td>
<td>imageData = {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'Width': 481,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'Height': 409,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'IsColor': 1,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'Blue': [56,...,67],</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'Green': [56,...,67],</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'Red': [56,...,67]</td>
<td></td>
</tr>
<tr>
<td>calibData</td>
<td>Calibration data, which contains UpperLeftX: {} UpperLeftY: {}</td>
<td>- UpperLeftX: the upper left point on the X direction in the coordinate system in pixel</td>
</tr>
<tr>
<td></td>
<td>LowerRightX: {}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LowerRightY: {}</td>
<td>- UpperLeftY: the upper left point on the Y direction in the coordinate system in pixel</td>
</tr>
<tr>
<td></td>
<td>XScale: {}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YScale: {}</td>
<td>- LowerRightX: the lower right point on the X direction in the coordinate system in pixel</td>
</tr>
<tr>
<td></td>
<td>For example: calibData = {</td>
<td>- LowerRightY: the lower right point on the Y direction in the coordinate system in pixel</td>
</tr>
<tr>
<td></td>
<td>'UpperLeftX': -313,</td>
<td>- XScale: X axial scale of real item and image in pixel</td>
</tr>
<tr>
<td></td>
<td>'UpperLeftY': -265,</td>
<td>- YScale: Y axial scale of real item and image in pixel</td>
</tr>
<tr>
<td></td>
<td>'LowerRightX': 168,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'LowerRightY': 144,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'XScale': 0.415,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'YScale': 0.415</td>
<td></td>
</tr>
</tbody>
</table>
### Argument | Description | Explain
--- | --- | ---
items | Item information, which contains: Time() and | • Time: time stamp(s), get the number of milliseconds since 1 Jan 1970<br>• X: the location value of the item in X direction<br>• Y: the location value of the item in Y direction<br>• Z: the location value of the item in Z direction<br>• RZ: the rotation angle value of the item in Z direction<br>• SortValue: sort value<br>• ZValid: - 1: valid<br> 0: invalid<br>• XImgPos: item position in image on X direction<br>• YImgPos: item position in image on Y direction<br>• Val1, Val2, Val3, Val4, Val5: optional value, used in rapid<br>• Level: inspection level<br> 0: Discarded<br> 1: Rejected<br> 2: Accepted<br>• Id: ID of the item<br>• ModelType: - 1: Geometric<br> 2: Blob<br> 3: Inspection<br>• Score: how closely the found item matches the trained model.<br>• XScale: X axial scale of real item and image in pixel<br>• YScale: Y axial scale of real item and image in pixel.<br>• Contrast: the image contrast of each item that is found in the image.<br>• FitError: a measure of the variance between the shape of the trained pattern and the shape of the pattern found in the search image.<br>• Coverage: a measure of the extent to which all parts of the trained pattern are also present in the search image.<br>• Clutter: a measure of the extent to which the found pattern contains features that are not present in the trained pattern.<br>For more information, see Configuring a geometric model with PatMax on page 289.

For example:

**Geometric**

```javascript
resResult = {
  'Time': 1666849507.969,
  'O': {'X': -80.1,
        'Y': -77.2,
        'Z': 0.0,
        'RZ': -7.22,
        'SortValue': 0.976,
        'ZValid': 0,
        'XImgPos': -80.1,
        'YImgPos': -77.2,
        'Val1': 0.0,
        'Val2': 0.0,
        'Val3': 0.0,
        'Val4': 0.0,
        'Val5': 0.0,
        'Level': 2,
        'Id': '3513a6c-5e8-4374-b180-769de599f3ff',
        'ModelType': 1,
        'Score': 0.747174859046936,
        'XScale': 0.9955959997177114,
        'YScale': 0.9955959997177114,
  }
```

Continues on next page
### blobResult

```javascript
Blob
resResult = {
'Time': 1666849507.969,
'O': {'X': -80.1,
'Y': -77.2,
'Z': 0.0,
'RX': -7.22,
'SortValue': 0.976,
'ZValid': 0,
'XImgPos': -80.1,
'YImgPos': -77.2,
'Val1': 0.0,
'Val2': 0.0,
'Val3': 0.0,
'Val4': 0.0,
'Val5': 0.0,
'Level': 2,
'Id': '55105b0c-56a8-473d-bb180-760e49b6f6ff',
'ModelType': 'blobResult',
'Area': 0,
'Perimeter': 0,
'Elongation': 0,
'Circularity': 0
}
}
```

### inspectionResult

```javascript
Inspection
resResult = {
'Time': 1666849507.969,
'O': {'X': -80.1,
'Y': -77.2,
'Z': 0.0,
'RX': -7.22,
'SortValue': 0.976,
'ZValid': 0,
'XImgPos': -80.1,
'YImgPos': -77.2,
'Val1': 0.0,
'Val2': 0.0,
'Val3': 0.0,
'Val4': 0.0,
'Val5': 0.0,
'Level': 2,
'}
```

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Explain</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Contrast':12.289325714111328,</td>
<td></td>
<td>• Area: expressed in mm²</td>
</tr>
<tr>
<td>'FitError':0.36996814608573914,</td>
<td></td>
<td>• Perimeter: expressed in mm</td>
</tr>
<tr>
<td>'Coverage':0.747174859046936,</td>
<td></td>
<td>• Elongation: the ratio of the feature's second moment of inertia about its second principal axis to the feature's second moment of inertia about its first principal axis.</td>
</tr>
<tr>
<td>'Clutter':0.10466811060905457</td>
<td></td>
<td>• Circularity: defines the circularity. A value of 1 means perfectly circular and completely filled (no holes).</td>
</tr>
</tbody>
</table>

For more information, see Configuring blob models on page 297.
### Distribution Interface

**PyDistribution**: Adjust the target items information after distribution and before push them to robot

#### Argument | Description | Explain
---|---|---
**Wald** | Workarea ID, which contains Wald(). For example: Wald = (98838CC-2965-4054-92A5-3CC38D648657) | • Wald: Workarea Id

---

Continues on next page
### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Explain</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>items</strong></td>
<td>Item information, which contains Time: X: Y: Z: q1: q2: q3: q4: Tag: Val1: Val2: Val3: Val4: Val5: Type: Index: State: Container: Layer: Group: Id:</td>
<td>- Time: time stamp(s), get the number of milliseconds since 1 Jan 1970 - X: the location value of the item in X direction - Y: the location value of the item in Y direction - Z: the location value of the item in Z direction - q1, q2, q3, q4: the quaternion values of the item - Tag: used in rapid - Val1, Val2, Val3, Val4, Val5: optional value, used in rapid - Type: item type, used in rapid - Index: Index number - State: item state, - 0: Use - 1: Bypass - 2: Used - Container: container number, 0 means that it is an item - Layer: layer number, 0 means that it is an item - Group: sorting method - 0: None or X direction - 1: Strict - Id: ID of the item</td>
</tr>
</tbody>
</table>

For example:

```python
items = {
    'Time': 1666849507.969,
    'X': 0.0,
    'Y': 150.0,
    'Z': 0.0,
    'q1': 0.0,
    'q2': 1.0,
    'q3': 0.0,
    'q4': 0.0,
    'Val1': 0.0,
    'Val2': 0.0,
    'Val3': 0.0,
    'Val4': 0.0,
    'Val5': 0.0,
    'Type': 2,
    'Tag': 0,
    'Index': 0,
    'State': 0,
    'Container': 2,
    'Layer': 1,
    'Group': 0,
    'Id': '35139a5c-56a8-437d-b180-7f40e49b6f6f'
}
```
Example:

```python
# Example of a user script

# Import necessary modules
import sys
import os

# Define global variables
x = 1
y = 2

# Define a function
def my_function():
    global x
    global y
    x = x + 1
    y = y + 1
    print(f"x={x}, y={y}")

# Call the function
my_function()
```

All the user script example files are provided in the folder `C:\Program Files (x86)\ABB\PickMaster Twin 2\PickMaster Twin Client 2\PickMaster PowerPac\Template` when PickMaster Client is installed.

Configuring the User Script function

Follow this procedure to configure the user script function:

1. Put the predefined script files into the destination folder.

   **Tip**
   
   The predefined script file(s) should be put into `C:\Users\xxxx\Documents\PickMaster\PMscripts` folder before use any script function.

   ```
   /mnts > PickMaster > PM_scripts
<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor2Functions.py</td>
<td>4/27/2023 1:30 PM</td>
<td>JetBrains PyCharm ...</td>
</tr>
<tr>
<td>Sensor1Functions.py</td>
<td>4/27/2023 11:11 AM</td>
<td>JetBrains PyCharm ...</td>
</tr>
<tr>
<td>ExternalSensors.py</td>
<td>4/27/2023 10:54 AM</td>
<td>JetBrains PyCharm ...</td>
</tr>
<tr>
<td>ExternalISensors.py</td>
<td>4/27/2023 10:41 AM</td>
<td>JetBrains PyCharm ...</td>
</tr>
<tr>
<td>PyDistributionDemo.py</td>
<td>3/27/2023 8:54 PM</td>
<td>JetBrains PyCharm ...</td>
</tr>
<tr>
<td>PyAdjusterDemo.py</td>
<td>12/20/2022 1:11 PM</td>
<td>JetBrains PyCharm ...</td>
</tr>
<tr>
<td>PyVisionDemo.py</td>
<td>11/15/2022 10:38 AM</td>
<td>JetBrains PyCharm ...</td>
</tr>
</tbody>
</table>
   ```

2. Select the User Script checkbox to open the configuration page.
3. Input the predefined script file name into the Script Name text box.
4. Click Configure Interface to open the interface type page.

Continues on next page
5 On the popped-up page, select the desired interface type.

Tip

The four types can be used at the same time.

6 Click Done to finish the user script function setting.

Distribution Setting

By default all positions are sent to the same work area. It is possible to distribute item positions to more than one work area to balance the load between several robots or to guarantee that all positions are accessed.

All positions for a specific item type are distributed to the robots by a single item distributor. There are four types of item distributors.

- Work area: The item positions are handled by a single conveyor or indexed work area.
- ByPass: The item positions are discarded, that is not handled by any work area. If no distributor is selected for an item type it will be considered as ByPass.
- LB group: The item positions are handled by the work areas included in a load balance group. A load balance group is a collection of Work area, ByPass, and ATC group distributors. Item positions will be distributed among the work areas in an optimal way to avoid sending two adjacent positions to the same work area.
- ATC group: Positions are handled by the work areas included in an Adaptive Task Completion (ATC) group. An ATC group is a collection of ordered work areas that will get the same item positions. The first robot accesses as many positions as possible. The other robots in the ATC group will access any missed positions. If the last work area in the group is a conveyor work area with start and stop it is guaranteed that all positions will be accessed.

To use either load balancing or ATC the work areas must be arranged in the order that they occur after the position source (for example: the camera or sensor).

The work area that triggers the position source is set automatically. When starting a production, the work area for the robot that is first up and running is set to be the trigger work area. If the robot for a trigger work area is stopped, a work area for another robot that is running will be the one that triggers the position source.
The item distribution tree control shows the items for which positions are to be generated. Accepted and rejected items can be distributed differently.

### Distribution

<table>
<thead>
<tr>
<th>Item distribution</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set the distribution strategy as Accept or Reject for all available items for this operation.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Make sure that at least there is one group valid distribution setting under Item distribution Accept or Reject for all available items. Otherwise an error will pop up when this recipe is selected to do the simulation or production.</td>
</tr>
<tr>
<td></td>
<td>{0} lacked valid distribution. Please check settings in Recipe -&gt; Operation.</td>
</tr>
</tbody>
</table>

| Available Distributor | Shows the available distributor for this operation. |

### Load balance

Item positions that are distributed by a load balance group are divided among the distributors in the group. A load balance group can contain any number of item distributors and a single distributor can appear several times. The ratio between the number of times a single distributor is added and the total number of distributors defines the ratio of the item positions that are sent by that particular distributor. Item positions are arranged to the distributors in the group in an optimal way to avoid adjacent positions to be sent to the same work area.

If **Adaptive Task Completion** is selected, any defined ATC groups will be listed among the available distributors. Additionally, ATC groups can be added to load balance groups. However, to achieve task completion, the load balance group should only contain ATC groups.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Balance Group</td>
</tr>
<tr>
<td>Available Distributor</td>
</tr>
<tr>
<td>New LBGroup</td>
</tr>
<tr>
<td>Delete Group</td>
</tr>
</tbody>
</table>

### ATC

**Adaptive Task Completion** guarantees the item positions to be accessed by any robot in an ATC group. An ATC group contains ordered work areas and a single work area is allowed to exist once in a group. All item positions distributed to an ATC group are sent to every work area in the group and the positions not accessed by the first work area will be accessed by any of the other work areas. If the last work area is on a conveyor with start and stop it is guaranteed that all item positions will be accessed by one of the robots in the ATC group.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Task Completion Group</td>
</tr>
</tbody>
</table>
4 Working with PickMaster PowerPac

4.3.3.4 Recipe

Continued

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Distributor</td>
</tr>
<tr>
<td>Shows the available distributor for this operation.</td>
</tr>
<tr>
<td>New ATCGroup</td>
</tr>
<tr>
<td>Create a adaptive task completion group.</td>
</tr>
<tr>
<td>Delete Group</td>
</tr>
<tr>
<td>Delete a adaptive task completion group.</td>
</tr>
</tbody>
</table>

Procedure

On the PickMaster PowerPac ribbon-tab, click Process.
Use this procedure to add a recipe:

1. On the ribbon-tab, click Recipe.
   The Recipe window opens.
2. Click on the Add Operation to add a new operation.
3. Click on the Operation 1 to open the setting window for the operation.
4. Select the operation type as Pick or Place.
5. If need, click to select the applicable flow in Select Flow.
6. Click to select the item in Available Objects.
7. Click to select the work area in Available Work Areas.
8. In the Trigger/Filter Setting tab, define the trigger or filter setting according to your requirements.
9. If need, click to select and configure the User Script according to your requirements.
10. In the Distribution tab, drag distributors from the Available distributors list to the Distribution list.
    There can be only one distributor for each item type. If an item type is missing a distributor, it will be regarded as ByPass.
11. If using load balancing, in the Load balance tab, drag a distributor from the Available distributors list to a group in the list Load balance groups.
    To create a new load balance group, double-click <New LbGroup> in the Available distributors list.
    Select rebalancing strategy.
12. If using Adaptive Task Completion, in the ATC tab, drag a work area from the Available work areas list to the Adaptive Task Completion groups list.
13. Click OK.
    The window is closed.

Redistributing items from one robot to downstream robots

It is possible to modify the distribution of already distributed item positions when they enter a conveyor work area of a robot. The Rapid program, that controls the robot, based on current flow conditions decides to skip an item position and change the type of it. As a result, PickMaster PowerPac will redistribute the item position to downstream robots according to the configured distribution strategy for the selected item type.
4.3.3.5 Simulation

Overview
This section describes how to do the simulation with the created solution.

Control
All operations in the simulation production are reflected in the station view, and all data comes from the solution.

Select one recipe from the tree view and click Control on the ribbon to open the control dialog box in the solution.

The following table provides details about the Control dialog box.

<table>
<thead>
<tr>
<th>Description</th>
<th>Control the status of the current recipe and have an overview of the production data. For more information regarding Recipe see the following section.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipes</td>
<td></td>
</tr>
<tr>
<td>Tuning</td>
<td>Adjust the parameters of the item, work area and robot. For more information regarding Tuning see the following section.</td>
</tr>
<tr>
<td>Flow Control</td>
<td>Adjust the speed of the conveyor. For more information regarding Flow Control see the following section.</td>
</tr>
</tbody>
</table>

Recipe

<table>
<thead>
<tr>
<th>Description</th>
<th>Control the status of the production.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipe Status</td>
<td></td>
</tr>
<tr>
<td>Picking Status</td>
<td>Shows the overview of the picking status in summary or detail.</td>
</tr>
</tbody>
</table>

Tuning

Sometimes, the exact pick and place positions are not exactly where expected. This might be caused by a small error in the calibration of either the camera or the work area. It is possible to adjust the positions while running a project. This is called tuning.

Tip

For item tuning, the tuning value only affects the new generated item targets. The tuning value will not be effective on the recognized item targets in the queue. For the work area and robot tuning, the tuning value will be effective immediately.

Tuning the Item

<table>
<thead>
<tr>
<th>Description</th>
<th>Set the location of the gripper when doing the picking and placing operation in X direction.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GripX</td>
<td></td>
</tr>
<tr>
<td>GripY</td>
<td>Set the location of the gripper when doing the picking and placing operation in Y direction.</td>
</tr>
<tr>
<td>GripZ</td>
<td>Set the location of the gripper when doing the picking and placing operation in Z direction.</td>
</tr>
</tbody>
</table>

Continues on next page
### Description

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
</table>
| **GripAngleX** | Set the angle of the gripper when doing the picking and placing operation in X direction.  
- **Note**  
  The angle cannot be out of the physical limits. Otherwise the robot will not work normally.  
  For example, trying to rotate the gripper of an IRB 360 robot in X or Y direction will cause an error. Redo the simulation after the error occurred. |
| **GripAngleY** | Set the angle of the gripper when doing the picking and placing operation in Y direction.  
- **Note**  
  The angle cannot be out of the physical limits. Otherwise the robot will not work normally.  
  For example, trying to rotate the gripper of an IRB 360 robot in X or Y direction will cause an error. Redo the simulation after the error occurred. |
| **GripAngleZ** | Set the angle of the gripper when doing the picking and placing operation in Z direction.  
- **Note**  
  The angle cannot be out of the physical limits. Otherwise the robot will not work normally. |

### Configuring the grip location

Use this procedure to configure the item’s grip location.

1. Select the **Type** as **Item** and select the required item.
2 Define the positions in millimeters for the grip position of the item specified in X', Y', and Z' coordinates. The positions are relative to the origin of the taught model (Vision model grip point). See the following graphic.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Camera</td>
</tr>
<tr>
<td>B</td>
<td>Adjusted grip point</td>
</tr>
<tr>
<td>C</td>
<td>Vision model grip point</td>
</tr>
<tr>
<td>D</td>
<td>Item height</td>
</tr>
<tr>
<td>E</td>
<td>Angle X</td>
</tr>
<tr>
<td>F</td>
<td>Angle Y</td>
</tr>
<tr>
<td>G</td>
<td>Conveyor direction</td>
</tr>
</tbody>
</table>

3 Define the Euler orientation in degrees for the grip orientation on the item. A four axes robot can only rotate around the z-axis and therefore only GripAngleZ can be used.

Six axes robots can pick/place 3D items by defining Euler orientation GripAngleX, GripAngleY and the item height. The grip orientation has an orientation in relation to the origin of the taught model (Vision model grip point). The item height must be specified in the Item configuration dialog, as a distance from the base frame to the item origin (vision model grip point).

It is important to define a correct calibration tool when calibrating the base frame of the conveyor, so the orientation in relation to the items grip point (place/pick) will be correct. It is also important to do the camera calibration at the same height as the item’s grip point, that is vision model grip point.

Tuning the work area

**Note**

The parameters of in tuning work area are synchronized with the parameters in the recipe. Any modification in one place will modify the parameters in the other place.
### Description

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TuneX</strong></td>
<td>Tune the position of the work area along the X direction when running simulation or production. Tuning the position of the work area along the X direction is equivalent to offsetting the conveyor base frame along the X direction.</td>
</tr>
<tr>
<td><strong>TuneY</strong></td>
<td>Tune the position of the work area along the Y direction when running simulation or production. Tuning the position of the work area along the Y direction is equivalent to offsetting the conveyor base frame along the Y direction.</td>
</tr>
<tr>
<td><strong>TuneZ</strong></td>
<td>Tune the position of the work area along the Z direction when running simulation or production. Tuning the position of the work area along the Z direction is equivalent to offsetting the conveyor base frame along the Z direction.</td>
</tr>
<tr>
<td><strong>Enter</strong></td>
<td>Enter is the limit from where the robot starts to execute item targets on the work area. The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving direction of the conveyor. Make sure that the enter limit can be reached by the robot. For more details, see <em>Available Work Areas on page 142</em>.</td>
</tr>
<tr>
<td><strong>Exit</strong></td>
<td>Exit is the limit from where the robot considers an item target as lost on the work area. The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving direction of the conveyor. When the tracked item passes beyond this limit it will be dropped. This limit must be chosen well within the maximum reach of the robot. The robot must be able to reach this position from an arbitrary position in the robot’s working area before the position is out of reach. For more details, see <em>Available Work Areas on page 142</em>.</td>
</tr>
<tr>
<td><strong>Pick/placeElevation</strong></td>
<td>Pick/place elevation is the distance, in negative z-direction relative to the tool, from where the robot approaches the item target.</td>
</tr>
<tr>
<td><strong>Pick/placeTime</strong></td>
<td>Pick/place time is the time the robot is in the pick/place position. If the conveyor is moving during the pick/place time, the robot will track along the conveyor to keep the relative position on the moving conveyor.</td>
</tr>
<tr>
<td><strong>VacuumActivation</strong></td>
<td>Vacuum activation is the time in seconds before the middle of the corner path of the approaching position, when the vacuum I/O should be set. If a negative value is entered, the vacuum I/O will be set the time after the middle of the corner path. This value is only valid for work areas of type Pick. <strong>Note</strong> Vacuum activation does not affect the picking of items in simulation. Items are attached to the picking tool using SimAttach events, for example, in the Pick Routine.</td>
</tr>
<tr>
<td><strong>VacuumReversion</strong></td>
<td>Vacuum reversion is the time in seconds before the half place time in the place position, when the blow I/O should be set. If a negative value is entered, the blow I/O will be set the time after the half place time in the place position. This value is only valid for work areas of type Place. <strong>Note</strong> Vacuum reversion does not affect the placing of items in simulation. Items are detached from the picking tool using SimDetach events, for example, in the Place Routine.</td>
</tr>
<tr>
<td><strong>VacuumOff</strong></td>
<td>Vacuum off is the time in seconds after the half place time in the place position, when the blow I/O should be reset. If a negative value is entered, the blow I/O will be reset the time before the half place time in the place position. This value is only valid for work areas of type Place. <strong>Note</strong> Vacuum Off does not affect the placing of items in simulation. Items are detached from the picking tool using SimDetach events, for example, in the Place Routine.</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>ConveyorYMax</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Note**

To enable this function, you need to select the *Use Start/Stop* checkbox for this function in the recipe configuration page.

**Note**

The *ConveyorYMax* function in the **Tuning** window has a slight delay. If there is any update for this value, you need to wait a while to see the results.

*ConveyorYMax* is the limit from where robot considers an item target as lost on the work area in End Y. The distance is calculated in millimeter from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving vertical direction of the conveyor.

Make sure that the *ConveyorYMax* can be reached by the robot. If the y coordinate value of the item's position is greater than the *ConveyorYMax*, the robot will not grab the item. So when the tracked item passes beyond this limit it will be dropped. This limit must be chosen well within the maximum reach of the robot.
4 Working with PickMaster PowerPac

4.3.3.5 Simulation

Continued

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ConveyorYMin</strong></td>
</tr>
<tr>
<td><strong>Note</strong></td>
</tr>
<tr>
<td>To enable this function, you need to select the Use Start/Stop checkbox for this function in the recipe configuration page.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
</tr>
<tr>
<td>The <strong>ConveyorYMin</strong> function in the Tuning window has a slight delay. If there is any update for this value, you need to wait a while to see the results. <strong>ConveyorYMin</strong> is the limit from where robot starts to execute item targets on the work area in Start Y. The distance is calculated in millimeter from the center of the robot. The range is positive if the limit is beyond the center of the robot, relative to the moving vertical direction of the conveyor.</td>
</tr>
</tbody>
</table>

![Diagram](image)

A | Camera and Baseframe origin |
---|-----------------------------|
B | Camera |
C | Enter |
D | Start |
E | Stop |
F | Exit |
G | Robot |
H | Image frame |
I | Center of Robot |

**Tuning the robot**

The robot settings can be tuned when a production is running, using the **Tuning the robot** window.

**Limitations**

All tunings, including robot tuning, item tuning, and work area tuning, are only valid while the simulation or production is running.

Continues on next page
Flow Control

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conveyor</td>
</tr>
<tr>
<td>Indexed Work Area</td>
</tr>
</tbody>
</table>

Simulation

**Note**

It is recommended to calibrate the solution when its virtual controller is used in other solution before simulation.

If different solutions use the same virtual controller, any modification to the controller of one solution will affect other solutions. This will cause unexpected and misleading behavior of other solutions.

Use this procedure to do the simulation:

1. On the PickMaster PowerPac ribbon-tab, click Operation.
2. Click to choose one recipe from the tree view browser.
3. Click Start on the ribbon-tab. Then it will start the simulation of created solution.
   - The simulation runs automatically.
4. Click Stop on the ribbon-tab. Then it will stop the simulation.
4.4 Configuration in real Runtime (RRT)

4.4.1 Switching to real Runtime

Switch Runtime

---

**Note**

After install PickMaster Twin Client and PickMaster Twin Host on different PC as recommended, there will be two real Runtime available but only the one connected to controller or camera should be used.

The real Runtime on Host PC and Client PC are identical but the one on Host is for production. Robot controllers and cameras should also be connected to this one.

**Tip**

The PickMaster® Runtime (VRT and RRT) is defined to use 50000 port. If 50000 port is occupied by other program, you will have this warning and not be able to connect to Runtime,

Release the 50000 port and restart the PickMaster® Runtime.

Use this procedure to release the 50000 port:

1. Enter the command `netstat -aon|findstr "50000"` in the CMD window.
2. The process that occupies port 50000 will be listed in the window. Obtain the PID code of the process.
3. Find the process corresponding to this PID in the task manager and close the it (Make sure that this process is allowed to be closed on this computer).
4. Restart PickMaster® Runtime and connect.

Right-click on Runtime to set the connection to the virtual Runtime (VRT) in simulation mode or the real Runtime (RRT) for operating the real robots on the Host computer in emulation mode.

**Tip**

Before connecting to RRT, start the PickMaster Runtime on the Host computer.

When selecting Connect to RRT, the Sign in window is displayed.

---

*Continues on next page*
The following table provides details about the Connect to RRT dialog box.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IP Address</strong></td>
</tr>
<tr>
<td>Locate the IP address of the Runtime computer.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
</tr>
<tr>
<td>Check the IPv4 address of the computer which the PickMaster Runtime is installed on.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
</tr>
<tr>
<td>Loopback address is NOT allowed to use as the real PickMaster Runtime IP address, for example 127.0.0.1.</td>
</tr>
<tr>
<td>Loopback address will cause errors in vision function.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Credential</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UserName</strong></td>
</tr>
<tr>
<td>The default user name is admin. And it CANNOT be changed.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
</tr>
<tr>
<td>Enter the password of your account in the Runtime.</td>
</tr>
</tbody>
</table>

A default user and password have been created for each role.

Administrator Username: admin with Password: password

**Note**

If the solution will be used in the PickMaster Operator, it must have been connected to a real controller with the same configuration on PickMaster PowerPac.

**Procedure**

To connect to Runtime.

1. Right-click the Runtime in the tree view Layout and select Start Local RRT.
2. Right-click the Runtime in the tree view Layout and select Connect to RRT.
   The ConnectToRRT window is opened.
3. In the Sign in dialog, enter the correct information.
4. Click OK.

**Tip**

If switch failed, the message box will show up.

Message: Failed to connect to RRT, please make sure the RRT starts or login information is correct.
4 Working with PickMaster PowerPac

4.4.1 Switching to real Runtime

Continued

---

**Note**

If the user meets any problem when building connection between PickMaster PowerPac and real Runtime, please check from below possible reasons:

1. Using a host account that is not administrator;
2. Firewall blocking;
3. VPN interference;
4. Host IP address incorrect, or not in the same IP segment as the client port.
5. The PickMaster PowerPac is not allowed to communicate in all networks.

---

Select a real controller

**Note**

Make sure that at least one real controller has been selected for the controller which need to run the production.

Otherwise an error will pop up when this recipe is selected to do the production.

{0} lacked real controller setting. Please connect to a real controller first.

Use this procedure to select a real controller:

1. Right-click the Controller in the tree view Layout and select Edit Controller.
   The Edit Controller dialog is opened.
2 Click on the Select Real Controller icon to open the Select Real Controller dialog.

**Note**

User must modify the firewall settings before selecting a real OmniCore controller in PickMaster PowerPac.

For WAN port, under Configuration/Communication/Firewall Manager, the following functions must be enabled.

"syslog" -EnableOnPublicNet
"Bonjour" -EnableOnPublicNet
"RobiCl" -EnableOnPublicNet
"EtherNetIP" -EnableOnPublicNet
"RapidSockets" -EnableOnPublicNet
"RobotWebServices" -EnableOnPublicNet
"IEEE1588" -EnableOnPublicNet
"Netscan" -EnableOnPublicNet
"RobAPI" -EnableOnPublicNet

**Note**

Make sure that the setting 'Allow connection to controller from RobotStudio on public network' is enabled.

---

The Select Real Controller dialog is opened.

3 In the dialog box, choose the real controller to be connected.

Continues on next page
4 Click OK to apply the configuration.
5 Click Close to close the Edit Controller dialog.

Modifying I/O signals in work area

Note

Make sure that it is NOT set as Default signal type for the work areas which need to run the production.
Otherwise an warning will pop up when this recipe is selected to do the production.
{0} used default signal type and lacked customized signal type setting. Please check the signal configuration in work area.

Use this procedure to modify the I/O signals in work area which is in used:
1 Right-click on Conveyor WorkArea 1 in the tree view Layout and select Setting.
   The Conveyor work area setting window is opened.
2 Select the Customized Settings in the Signal Type tab.
3 Enter the required data into the I/O signal setting table. For more information, see Configuring the I/O on page 172.
   For example:

   ![](signal_type_table.png)

4 Click OK to close the Recipe setting window.
5 Repeat step 1 - 4 to the other Conveyor WA.

Configuring the I/O

I/O signals

I/O signals are configured using RobotStudio or the FlexPendant.
The predefined signals can be used without modifications. Edit the predefined signals or add additional signals if needed.

**Note**
The maximum name length for a signal is 15 characters.

The following I/O signals are used in PickMaster PowerPac. Some of them are used or referenced to when configuring the solution. The encoder signals are described in *Application manual - Conveyor tracking*.

<table>
<thead>
<tr>
<th>I/O signal name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>diX_1</td>
<td>Digital input signals for custom use, such as generating I/O triggered position or checking a gripper pressure switch.</td>
</tr>
<tr>
<td>doStartCnvX</td>
<td>Digital output for starting/stopping conveyors.</td>
</tr>
<tr>
<td>doTrigVisX/cXTrigVis</td>
<td>Digital output for triggering an image acquisition. This signal is used by Runtime to order the camera to acquire an image. For DSQC 377, this output should be connected to the doTrigVisX on the corresponding encoder board. For DSQC 2000, this output should be connected to the cXTrigVis. For more detail information, see the circuit diagram.</td>
</tr>
<tr>
<td>doManSyncX</td>
<td>Digital output used for triggering predefined positions in a conveyor work area. For DSQC 377, this output should be connected to the StartSig (input 9) on the corresponding encoder board. For DSQC 2000, this output should be connected to the cXTrigVis. For more detail information, see the circuit diagram.</td>
</tr>
<tr>
<td>doVacuumX</td>
<td>Digital output for activating vacuum. For example, for gripping a product. The output signal is set when an item shall be attached to the tool. <strong>Note</strong> The signal is controlled from the RAPID program. In simulation, the RAPID trigdata SimAttachX controls when the signal is set. On a real robot, the RAPID trigdata VacuumActX controls when the signal is set.</td>
</tr>
<tr>
<td>doBlowX</td>
<td>Digital output for activating air blow. For example, for releasing a product gripped by the robot. The output signal is set when an item shall be detached from the tool. <strong>Note</strong> The Release signal is controlled from the RAPID program. In simulation, the RAPID trigdata SimDetachX controls when the signal is set. On a real robot, the RAPID trigdata VacuumRevX and VacuumOffX controls when the signal is set/pulsed.</td>
</tr>
</tbody>
</table>

Continues on next page
### Conveyor work area default I/O signals

The default I/O signals are used for simulation.

<table>
<thead>
<tr>
<th>Item</th>
<th>DSQC 377</th>
<th>DSQC 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conveyor start/stop</td>
<td>cnvX_doStartCnv</td>
<td>cnvX_doStartCnv</td>
</tr>
<tr>
<td>Queue idle</td>
<td>cnvX_doQIdle</td>
<td>cnvX_doQIdle</td>
</tr>
<tr>
<td>Position available</td>
<td>cnvX_doPAvail</td>
<td>cnvX_doPAvail</td>
</tr>
<tr>
<td>Position generator</td>
<td>cnvX_diPosGen</td>
<td>cnvX_diPosGen</td>
</tr>
<tr>
<td>Trig</td>
<td>doTrigVisX</td>
<td>cXTrigVis</td>
</tr>
<tr>
<td>Strobe</td>
<td>cXNewObjStrobe</td>
<td>cXNewObjStrobe</td>
</tr>
</tbody>
</table>

Note:
For DSQC 2000, there is no predefined port for this signal. Define the real connected port on the board as the signal name.

xx2100001673
Conveyor work area customized I/O signals

The customized I/O signals are used for production.

<table>
<thead>
<tr>
<th>Item</th>
<th>DSQC 377</th>
<th>DSQC 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conveyor start/stop</td>
<td>doStartCnvX</td>
<td>doStartCnvX</td>
</tr>
</tbody>
</table>

Note: This signal can be left as empty if the conveyor is running.

Indexed work area default I/O signals

The default I/O signals are used for simulation.

<table>
<thead>
<tr>
<th>Item</th>
<th>DSQC 377</th>
<th>DSQC 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conveyor start/stop</td>
<td>indX_doQIdle</td>
<td>indX_doQIdle</td>
</tr>
<tr>
<td>Queue idle</td>
<td>indX_doPAvail</td>
<td>indX_doPAvail</td>
</tr>
<tr>
<td>Position available</td>
<td>indX_diPosGen</td>
<td>indX_diPosGen</td>
</tr>
<tr>
<td>Trig</td>
<td>indX_doTrigVis</td>
<td>indX_doTrigVis</td>
</tr>
<tr>
<td>Strobe</td>
<td>indX_diStrobe</td>
<td>indX_diStrobe</td>
</tr>
</tbody>
</table>

Indexed work area customized I/O signals

The customized I/O signals are used for production.

<table>
<thead>
<tr>
<th>Item</th>
<th>DSQC 377</th>
<th>DSQC 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conveyor start/stop</td>
<td>doTrigVisX</td>
<td>cXTrigVis</td>
</tr>
<tr>
<td>Queue idle</td>
<td>doTrigVisX</td>
<td>cXTrigVis</td>
</tr>
<tr>
<td>Position available</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position generator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trig</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strobe</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The Queue idle signal and Strobe signal should be the same one.

Any available do signals can be used.
Note

Make sure that the activator signal setting of gripper is exactly same with the connected controller.
Otherwise the gripper will not pick or place the items in PickMaster PowerPac.
4.4.2 Configuring camera

Introduction

Note

If any firewall or antivirus software is installed, add `pickmasteru.exe` and `visionclient.exe` to the white list. Otherwise the PickMaster PowerPac cannot connect Runtime and the vision function cannot work normally.

Cameras together with vision models are used to locate objects in a specific area. When a camera is created in the tree view, it is not connected to any physical camera. This must be done manually in the camera configuration dialog box. The camera in the tree view is configured to use one specific physical camera. The camera should also be configured to give an optimal image.

To configure a camera.

1. Right-click the camera in the tree view Layout and select Configuration. The Camera Configuration dialog and the Image dialog are opened.
2. In the Imaging device list, select the Gigabit Ethernet camera to which the camera is connected.
3 In the Video format list, select the type of the connected camera. The image in Image dialog shows up.

4 If the camera should strobe when it is not in production mode, select the Enable strobe offline checkbox. This is necessary if, for example, the camera is used together with a strobe light. This setting applies only to Gigabit Ethernet cameras.

5 If the selected camera is a color camera and will be used together with the color video format, it is necessary to calibrate the white balance of the camera using this procedure:
   a Put a white sheet of paper under the camera. The sheet must cover the entire field of view.
   b Adjust the light settings so that the image looks medium gray. Use either the camera aperture or the exposure time.
   c In the White balance part, click Calculate. This will calculate the white balance calibration parameters.
   d Click Apply. This will modify the camera’s internal settings.
   e Click Save on camera. This will store the settings in the camera.

For more information about color vision, see Using color vision on page 317.

6 If needed, adjust Exposure, Brightness, and Contrast and click Apply in the Camera settings part.

Adjust the exposure to achieve the best image possible. The exposure together with the camera aperture defines the focus depth and possible motion blur. These two parameters must be suitably adjusted depending on the type of objects to look for and the speed of the conveyor.
Brightness and contrast can be changed to give an optimal image. Some objects might be easier to find by adjusting the ambient lighting together with the brightness and contrast parameters.

The effect of changing these parameter values is not seen until clicking Apply.

7 Click OK.

**Note**

If the user meets any problem when opening VisionClient, please check from below possible reasons:

1 Firewall blocking;
2 The VisionClient is not allowed to communicate in all networks.

**Configuring a simulated camera**

The vision functions in PickMaster can be used without having a physical camera connected. The purpose is to allow vision modeling and evaluation offline on any laptop or PC and only a vision dongle connected. Instead of acquiring images from a physical camera, images are instead loaded from files. The function is for offline purposes only, and is not supported in production mode.

There are two types of dongles, the standard camera dongle and the simulation dongle. With the standard dongle connected, PickMaster automatically enters simulated mode if no camera is present when the program is started. With the simulation dongle, image acquisitions from cameras are not enabled, so all images must be loaded from files.

Use the following procedure to configure a simulated camera.

1 Right-click the camera in the tree view Cell and select **Configuration**.

The Camera Configuration dialog is opened.

Continues on next page
2 In the Imaging device list, select the Simulated framegrabber.

Note
The 8100d framegrabber is not compatible from PickMaster 3.41 onwards.

3 Configure a port.

4 Set the Video format to show color or monochrome images.

5 Load the images. There are two ways to load images in the various vision dialogs.
   - Load images from any folder using the "Import" button in the various vision dialogs.
   - Read image files from a registered image folder. Each camera has a default image for modeling, and a set of images for calibration which can be toggled by pressing "Acquire" in the calibration dialog. This requires some additional configuration to install a registered image as described below.

Set the file paths that PickMaster will use to locate the images. This is done by running the file "DongleSettings.reg" found on the PickMaster CD under "\PickMaster\DongleData\". The search paths are stored in the Windows registry, and may be edited. The default location for the image folder is "C:\DongleImages", so create this directory and copy the images included on the PickMaster CD under "\PickMaster\DongleData\DongleImages\". The configured port of the simulated camera determines which image is loaded for that camera.

6 Click OK.

Related information
Using color vision on page 317.
Calibrating camera on page 276.
4.4.3 Configuring external sensor

**External sensor**

An external sensor is a software component that gives the user full control of how item positions are generated. An external sensor can use any type of item detection such as barcode readers, cameras, or a combination of photo sensors to generate item positions. If cameras are used, any vision hardware or image searching algorithms can be used. PMTW supports to use Python to implement external sensors programs.

---

**Note**

Only the qualified personnel can write or modify the script files. It is the responsibility of the writer to make sure that the cell is safety when running with the script files.

---

**Note**

Only native Python is supported in PickMaster® Twin products. Any third-party libraries CANNOT be directly referenced in the script.

---

**Tip**

Syntax errors will cause the script files fail to run. With the following way to avoid the syntax errors:

1. Keep to use the same editor for the same script file.
2. It is recommended to use PyCharm or Notepad++ to edit the script files, as they have syntax checking capabilities for Python files.

---

**CAUTION**

It is the responsibility of the integrator to implement that local presence is set up in a correct way.

It is the responsibility of the integrator to implement that single point of control is set up in a correct way.

---

**DANGER**

Protect the script carefully if it is used in the production. Anyone who has access to the script can modify the script directly. This may cause serious danger.

---

**Tip**

If an external sensor is used on the conveyor, Flow function will be disabled.

Continues on next page
### 4.4.3 Configuring external sensor

**Note**

The user script and external sensor cannot be used at the same time in one recipe.

**Note**

Python script files will not be included in the Pack&Go file. Copy the Python script files to the desired destination.

**Tip**

If an indexed work area is used, external sensor function will be disabled.

Right-click on an **External Sensors** in the tree view and select **Configuration** to configure the external sensor.

The following table provides details about the **External Sensor Configuration** dialog box.

<table>
<thead>
<tr>
<th><strong>Description</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Script Name</strong></td>
<td>Type the predefined script file name with <code>.py</code>. It's recommended to use the name <code>ExternalSensors.py</code> as template.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>The predefined script file(s) should be put into <code>C:\Users\xxxx\Documents\PickMaster\PMScripts</code> folder before use any script function.</td>
</tr>
<tr>
<td><strong>Configure in user program</strong></td>
<td>When click on this button, it will refer to the Python interface <code>def configureSensor(self, sensorId)</code> . Users should self-define the configuration behavior in this interface in their own Python class. When users click this button, the interface will be called.</td>
</tr>
<tr>
<td><strong>Get sensor information</strong></td>
<td>When click on this button, it will refer to the Python interface <code>def getSensorInfo(self)</code> which is provided by PMTW in the file <code>ExternalSensorInterface.py</code>. In this interface, the information of user program, such as name, author, version and description will be sent back to and shown in the <strong>External Sensor Configuration</strong> dialog. The information of user program can be modified directly in Python code or in <code>configureSensor</code> interface in the file <code>ExternalSensors.py</code>.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The file <code>ExternalSensorInterface.py</code> is not allowed to be modified.</td>
</tr>
</tbody>
</table>

If an external vision system is used it may be necessary to turn off PickMaster's internal vision system to avoid that both systems try to connect to the same camera.

*Continues on next page*
Turn off the internal vision system in real Runtime.

Note
All the contents in file ExternalSensor.py should inherit the predefined basic classes from the file ExternalSensorInterface.py.

Predefined External Sensor interface classes

Except these four classes, a tool class StoppableThread is provided which is inherited from the threading.Thread class of native Python and has the function of stop thread.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SensorInfo</td>
<td>This interface is used to provide the user to initialize the Python program, such as: initialize all basic information and settings.</td>
</tr>
<tr>
<td>SensorConfig</td>
<td>This interface is used to provide the user to realize the operations about the sensor.</td>
</tr>
<tr>
<td></td>
<td>- def configureSensor(self, sensorId): the content should be implemented by users in their own class.</td>
</tr>
<tr>
<td></td>
<td>- def getSensorInfo(self): provided by PMTW to get the information of name, author, version and description of user program. This interface is called when the Get sensor information button is clicked.</td>
</tr>
<tr>
<td></td>
<td>- def loadSensor(self, sensorId, configurationInfo): provided by PMTW to load the previously saved configuration information string to the Python dictionary data sensorConfigurationDict[sensorId] in the user program to update their last settings. This interface is called when the Python environment is in initialization and the sensor has already been configured previously.</td>
</tr>
<tr>
<td></td>
<td>- def saveSensor(self, sensorId): provided by PMTW to get the latest configuration information string including possible modification in configureSensor. This interface is called when the OK button is clicked in the sensor configuration dialog or Save button is clicked in PositionGenerator dialog.</td>
</tr>
</tbody>
</table>

Continues on next page
4 Working with PickMaster PowerPac

4.4.3 Configuring external sensor

Continued

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PositionGenerator</td>
<td>This interface is used to provide the user to realize the customized sensor position generation.</td>
</tr>
<tr>
<td></td>
<td>• def initializePosGenRelatedMap(self, sensorId, posGenId, objectName): provided by PMTW to initialize the relationship between sensor, position generator and object (item/container). Users could then use objectName (item/container name) in their Python logic to help distinguish different position generators. This interface is called when the Python environment is in initialization and one position generator is turned from configuration-enabled state into save-enabled state.</td>
</tr>
<tr>
<td></td>
<td>• def configurePosGen(self, posGenId): the content should be implemented by users in their own class.</td>
</tr>
<tr>
<td></td>
<td>• def loadPosGen(self, posGenId, positionGeneratorInfo): provided by PMTW to load the previously saved position generator configuration information string to the Python dictionary data posGenConfigurationDict[posGenId] in the user program to update their last settings. This interface is called when one existing position generator is turned from configuration-enabled state into save-enabled state.</td>
</tr>
<tr>
<td></td>
<td>• def savePosGen(self, posGenId): provided by PMTW to get the latest position generator configuration information string including possible modification in configurePosGen. This interface is called when the “Save” button is clicked, or when the “OK” button is clicked and one position generator is in save-enabled state in the item/container source type view.</td>
</tr>
<tr>
<td>SensorRuntime</td>
<td>This interface is used to provide the user to realize the customized sensor operations during the production. A flag mechanism is provided to handle the finishing logic of startSensor interface.</td>
</tr>
<tr>
<td></td>
<td>• def startSensor(self, callBackFunc): the content should be implemented by users in their own class and based on the flag mechanism to avoid unnecessary crash.</td>
</tr>
<tr>
<td></td>
<td>• def stopSensor(self): the content should be implemented by users in their own class.</td>
</tr>
<tr>
<td></td>
<td>• def monitorRecipeStatus(self, callBackFunc), def checkRecipeStatus(self, callBackFunc), def waitForRecipeStop(self): flag mechanism provided by PMTW to monitor the recipe status in the Runtime. Users could use it with referring to the template in the template folder C:\Program Files (x86)\ABB\PickMaster Twin 2\PickMaster Twin Client 2\PickMaster PowerPac\Template.</td>
</tr>
</tbody>
</table>

The script template ExternalSensor.py is an example which users should follow the class format to develop their own class. Please note that the class must inherit from the four base classes defined in ExternalSensorInterface.py and the class name must be the same as the main file name, otherwise errors exist when PMTW tries to load the Python file.

**def configureSensor(self, sensorId) interface**

In this interface, the serialization of configuration data into a string and saving the string in the Python dictionary sensorConfigurationDict[sensorId] must be included.
This interface will be called when the Configure in user program button is clicked in the external sensor configuration view.

For more information on dictionary data, see *Data structure used in Python program on page 191.*

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>self</td>
<td>Python syntax</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Refer to the class</td>
<td></td>
</tr>
<tr>
<td>sensorId</td>
<td>Sensor id automatically generated by PMPP</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```python
def configureSensor(self, sensorId):
    if self.sensorIdNameMapDict[sensorId] == 'ExternalSensor_1':
        # Configuration code
    else:
        # Configuration code

def configurePosGen(self, posGenId):
    if self.sensorIdNameMapDict[posGenId] == 'ExternalSensor_2':
        # Configuration code
    else:
        # Configuration code
```

Continues on next page
Example:

```python
def configureSensor(sil, posSensorId):
    if self.sensorManagerMapDict[sil].posSensorMapDict[posSensorId] == 'ExternalSensor_1':
        inputTitle = sil.sensorMapDict[posSensorId].sensorProps['inputTitle']
        if posSensorId in self.sensorMapDict[posSensorId].sensorProps:
            positionGeneratorInfo = sil.sensorMapDict[posSensorId].sensorProps
        else:
            positionGeneratorInfo = sil.sensorMapDict[posSensorId].sensorProps

    log = ('logLevel', 0, 'Log': positionGeneratorInfo)
    self.toggleCallback.ShowPythonLogging(log)
    elif self.sensorMapDict[sil].posSensorMapDict[posSensorId] == 'ExternalSensor_2':
        inputTitle = sil.sensorMapDict[posSensorId].sensorProps['inputTitle']
        if posSensorId in self.sensorMapDict[posSensorId].sensorProps:
            positionGeneratorInfo = sil.sensorMapDict[posSensorId].sensorProps
        else:
            positionGeneratorInfo = sil.sensorMapDict[posSensorId].sensorProps

    log = ('logLevel', 0, 'Log': positionGeneratorInfo)
    self.toggleCallback.ShowPythonLogging(log)
```

**def startSensor(self, callBackFunc) interface**

In this interface, users should handle the logic about letting the sensor run and generate positions. This interface will be called when the running recipe is starting. The PMTW `callBackFunc` is given as interface input argument, and includes two concrete callback functions, namely `GetStrobeTime()` and `NewPosition(pos)`.

`GetStrobeTime()` should be called when the sensor gets the trigger signal and will get the strobe time. When the position is generated, `NewPosition(pos)` should be called to send newly generated position to PMTW together with the strobe time.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>self</td>
<td>Python syntax</td>
<td>Refer to the class</td>
</tr>
</tbody>
</table>

Continues on next page
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>callBackFunc</td>
<td>Which contains GetStrobeTime() and NewPosition(pos)</td>
<td>The position format should follow this structure:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>newPos = {'SensorId': sensorId, # sensor id</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Time': strobeTime, # time stamp, get from PMTW by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>calling GetStrobeTime() callback function, unit is ms.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'key': {'X': 0.0, # key refers to the position index, start from 0. X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>refers to the location value of the item in X direction, unit is mm.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Y': 100.0, # Y refers to the location value of the item in Y direction, unit is mm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Z': 5.0, # Z refers to the location value of the item in Z direction, unit is mm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'RX': 0.0, # RX refers to the rotation angle value of the item in X direction, unit is degree</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'RY': 0.0, # RY refers to the rotation angle value of the item in Y direction, unit is degree</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'RZ': 0.0, # RZ refers to the rotation angle value of the item in Z direction, unit is degree</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Tag': 0, # used in rapid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Score': 1.0, # refers to the score of different position generator methods, could be used to sort the results</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Val1': 0.0, # optional value, used in rapid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Val2': 0.0, # optional value, used in rapid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Val3': 0.0, # optional value, used in rapid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Val4': 0.0, # optional value, used in rapid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Val5': 0.0, # optional value, used in rapid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'Level': 2, # level, 0: Discarded, 1: Rejected, 2: Accepted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'PosGenId': posGenId), # external sensor position generator id</td>
</tr>
</tbody>
</table>
The following example shows the format which contains two positions:

```plaintext
newPos = {
'SensorId': '11548258-b028-470a-b399-b780084acc59',
'Time': 376910718,
'O': {'X': 0.0,
'Y': 100.0,
'Z': 5.0,
'RX': 0.0,
'RY': 0.0,
'RZ': 0.0,
'Tag': 0,
'Score': 1.0,
'Val1': 0.0,
'Val2': 0.0,
'Val3': 0.0,
'Val4': 0.0,
'Val5': 0.0,
'Level': 2,
'PosGenId': '17dec9b3-8624-45fa-b8cb-366b457e6024'},
'1': {'X': 0.0,
'Y': 100.0,
'Z': 5.0,
'RX': 0.0,
'RY': 0.0,
'RZ': 0.0,
'Tag': 0,
'Score': 1.0,
'Val1': 0.0,
'Val2': 0.0,
'Val3': 0.0,
'Val4': 0.0,
'Val5': 0.0,
'Level': 2,
'PosGenId': '5413832d-dbbe-44b9-aab3-6bb64d8d30a1'}
}
```

Continued
Example:

```python
def startSensor(self, callBackFunc):
    try:
        ExternalSensors.monitorRecipeStatus(self, callBackFunc)
        for posenId in self.posen2SensorMapDict:
            windowTitle = self.sensorIDNameMapDict[self.sensorID] + str(posenId)
            windoW = ExternalSensors.getBySensorID(posenId)
            if windoW is None:
                print('Sensor ID {} not found in sensor list'.format(posenId))
            else:
                windoW.show()
                windoW.start()
                windoW.wait()
                windoW.destroy()
                windoW = None
        for td in self.allThreads:
            td.start()
        for td in self.allThreads:
            td.stop()
    except:
        log = ['LogLevel': 2, 'Log': 'Python Error: Failed to start sensor.'].
        self.flogCallback.ShowPythonLog(log)
```

**Using `startSensor` with flag mechanism**

The users should implement the `startSensor` content based on the flag mechanism provided by PMTW. The following codes show an example.

- **At the beginning of the interface**, the `monitorRecipeStatus` method provided by PMTW should be called first to start a thread to monitor whether the recipe is running.

- **Then users could implement their own logic about position generation logic**. In this example, a thread is created to start a position generation simulator. Since it should be guaranteed that interface content can be finished, the thread should be able to be stopped. Therefore, the `stoppableThread` provided by PMTW developer is used in this example.

- It is possible that there are more than one position generator. If one position generator runs in one thread, there will be more than one thread. Thus, the next step is to start all threads and the content in all threads will be executed.

- **The next step is to wait for the flag signal that the recipe is stopped**. If the recipe is running, the flag will be 1. If the recipe is stopped, the flag will be 0. The method of `waitForRecipeStop` is to get the flag value, so it should be called after all threads are called.

- If the flag signal shows that the recipe is stopped, all threads in this interface should be stopped. Please note that before directly stopping the thread, stop the content in the thread first if needed, e.g. close the connection port inside the thread.
def stopSensor(self, callbackFunc):
    try:
        # Step 1: call classname.recipeStatusAware(self, callbackFunc) to monitor the recipe status running in PNEK
        classname.recipeStatusAware(self, callbackFunc)
        # Step 2: start logic defined by users. For each sensor, a stopableThread must be created to generate positions and appended to self.allThreads
        for posGenId in self.posGenSensorMapDict:
            sensorId = self.sensorIDHashMapDict[posGenId]
            windowTitle = self.sensorIDHashMapDict[posGenId] + ' ' + self.posGenObjMapDict[posGenId]
            if self.sensorIDHashMapDict[posGenId] == 'ExternalSensor_1':
                thread_1 = StopableThread(target=classname.ExternalSensorDemo, args=(self, callbackFunc, sensorId), posGenId, self.posGenObjMapDict[posGenId], windowTitle)
                thread_1.start()
                self.allThreads.appendThread(thread_1)
                # Step 3: start all threads in self.allThreads
                for td in self.allThreads:
                    td.start()
            # Step 4: call classname.waitForRecipeStop(self) to wait for the stop signal from PNEK
            classname.waitRecipeStop(self)
            # Step 5: stop all threads in self.allThreads. Note that the stop behavior of startSensor interface should be handled by users here.
            for td in self.allThreads:
                td.stop()
        log = {'LogLevel': 0, 'Log': 'StartSensor: stopped all threads.'}
        self.flogCallback.ShowPythonLogLog(log)
    except:
        log = {'LogLevel': 0, 'Log': 'Python Error: Failed to stop sensor.'}
        self.flogCallback.ShowPythonLogLog(log)
Data structure used in Python program

In the `ExternalSensorInterface.py` file, several data structures are defined to save the relationship between sensor, position generator and object (item/container) and users could use them in their own class to realize the Python logic.

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>sensorIdNameMapDict</strong></td>
<td>Store the relationship between sensor id and name. <strong>SensorId</strong> is generated automatically in PMPP, which users could not modify. <strong>SensorName</strong> refers to the name of external sensor node, e.g., <code>ExternalSensor_1</code>. Users could use the default name provided by PMPP or modify the name according to their requirements. For most Python interfaces, <strong>SensorId</strong> is given as input argument, so users could directly use the <strong>SensorId</strong> to search in the dictionary to find the corresponding sensor name. If users use the sensor name in their logic and modify the name in PMPP, the string in Python must be updated to keep the same as in PMPP. In most cases, users don’t need to define or assign key and values in this dictionary, they only use <code>self.sensorIdNameMapDict[sensorId]</code> to get the sensor name based on the <strong>SensorId</strong>. <strong>Example:</strong> <code>sensorIdNameMapDict = {'11548258-b028-470a-b399-b780084acc59': 'ExternalSensor_1'}</code></td>
</tr>
<tr>
<td><strong>sensorConfigurationDict</strong></td>
<td>Store the relationship between sensor id and configuration information string. According to previous sections, the configuration information string reflects how the sensor is configured and the data is serialized in a string and saved in PMPP solution. When <code>loadSensor</code> is called, the saved string will be updated in this dictionary, which is implemented in <code>ExternalSensorInterface.py</code>. When <code>configureSensor</code> is called, if the sensor has already been configured, users may need to analyze or deserialize the configuration string first. They could use <strong>SensorId</strong> (interface input argument) to find the string first: <code>self.sensorConfigurationDict[sensorId]</code>. At the end of <code>configureSensor</code>, users may change the configuration settings in this interface, so a new string should be generated to reflect the latest setting and saved in the dictionary: <code>self.sensorConfigurationDict[sensorId] = configInfoString</code>. When <code>saveSensor</code> is called, the configuration string could be found in the dictionary with the latest setting and returned to PMPP, which is implemented in <code>ExternalSensorInterface.py</code>. <strong>Example:</strong> <code>sensorConfigurationDict= {'11548258-b028-470a-b399-b780084acc59': 'SensorType:Camera;IP:192.169.10.10;Brightness:10'}</code></td>
</tr>
</tbody>
</table>
### posGenConfigurationDict

Store the relationship between position generator id and position generator configuration information string. Similar to sensor configuration, the position generator configuration information string reflects how the position generator is configured and the data is serialized in a string and saved in PMPP solution.

When `loadPosGen` is called, the saved string will be updated in this dictionary, which is implemented in `ExternalSensorInterface.py`.

When `configurePosGen` is called, if the position generator has already been configured, users may need to analyze or deserialize the configuration string first. They could use `posGenId` (interface input argument) to find the string first:

```python
self.posGenConfigurationDict[posGenId]
```

At the end of `configurePosGen` users may change the configuration settings in this interface, so a new string should be generated to reflect the latest setting and saved in the dictionary:

```python
self.posGenConfigurationDict[posGenId] = posGenConfigInfoString
```

When `savePosGen` is called, the configuration string could be found in the dictionary with the latest setting and returned to PMPP, which is implemented in `ExternalSensorInterface.py`.

**Example:**

```python
posGenConfigurationDict =
    {'17dec9b3-8624-45fa-b8cb-366b457e6024':
     'PositionGeneratorType:Blob;Type:White;ThresholdValue:100'}
```

### posGenSensorMapDict

Store the relationship between position generator id and sensor id. According to previous introduction, the external sensor position generator can be created for each sensor under different item/containers. If there are more than one external sensors in the PMPP solution, under the same item/container, `externalsensor_1` can own its position generator, and `externalsensor_2` could also have its own position generator. These two position generators belong to different external sensors, so their relationship must be clarified to avoid data confusion.

Their relationship will be initialized in the interface `initializePosGenRelatedMap` which is implemented in the `ExternalSensorInterface.py` and users could use them when needed:

```python
self.posGenSensorMapDict[posGenId]
```

**Example:**

```python
posGenSensorMapDict =
    {'17dec9b3-8624-45fa-b8cb-366b457e6024':
     '11548258-b028-470a-b399-b780084acc59'}
```
posGenObjectMapDict

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
</table>
| Store the relationship between position generator id and object name. Here object refers to item or container. According to previous introduction, the external sensor position generator can be created for each sensor under different item/containers. If there are more than one items in the PMPP solution, for the same external sensor, item_1 can own its position generator, and item_2 could also have its own position generator. These two position generators belong to different items, so their relationship must be clarified to avoid data confusion.

As the object name may be needed in the user program logic, in this map dictionary, the object name is stored instead of item id.

Their relationship will be initialized in the interface initializePosGenRelatedMap which is implemented in the ExternalSensorInterface.py and users could use them when needed: self.posGenObjectMapDict [posGenId]

Example: posGenObjectMapDict = {'17dec9b3-8624-45fa-b8cb-366b457e6024': 'Item_1'}

| All the script example files are provided in the folder C:\Program Files (x86)\ABB\PickMaster Twin 2\PickMaster Twin Client 2\PickMaster PowerPac\Template when PickMaster Client is installed.
| Users could also overwrite the information properties of name, description, author and version. Except these interfaces and properties, users could implement more functions in this class or in other Python files for their own use. It’s just that PMTW software will only be aware of and invoke these four interfaces and ignore others.
| All used Python files should locate in the same path under C:\Users\..\Documents\PickMaster\PMScripts on the Host PC.

Configuring the external sensor

When an external sensor is created in the tree view, it is not connected to any physical sensor. This must be done manually in the external sensor configuration dialog box. The external sensor in the tree view is configured to use one specific physical external sensor. The external sensor should also be configured to give an optimal image.

To configure an external sensor.

1. Put the predefined script files into the destination folder.

Tip

The predefined script file(s) should be put into C:\Users\xxxx\Documents\PickMaster\PMScripts folder before use any script function.
2 Right-click the external sensor in the tree view **Layout** and select **Configuration**.

The **External Sensor Configuration** dialog is opened.

3 Input the name of the predefined main file in **Script Name**.

4 Click **Configure in user program** to configure the external sensor.

5 Click **OK** to save the configuration in PMPP.

6 Right-click the desired item/container in the tree view **Process** and select **Setting**.

7 Switch to **Item Source/Container Source** page.

8 Click **New Position Generator** to create the position generator for desired external sensor under the **External** tab.

**Note**

Only the external sensor that has been configured in the **External Sensor Configuration** can create the position generator.

**Tip**

All created external sensors in this solution will be listed in **External** tab.

9 Configure the position generator according to the user defined in external sensor script file.

10 Click **Save** to save the configuration data to PMPP for the position generator.

11 Click **OK**.
4.4.4 Calibrating robot

Instruction

Detailed information about how to calibrating the robot are described in the robot product manual.
4.4.5 Calibrating linear conveyor

Overview

Note

The following calibration process is required when running production and emulation. Calibration under the simulation tab in PickMaster PowerPac will not complete the following calibration process.

The calibrations needed for the conveyors are camera and work area calibrations. The work area calibration is a base frame calibration for conveyor work areas and a work object definition for indexed work areas. The key concept is to define a coordinate system origin that is the same for a camera and a robot base frame or work object.

Each camera must be calibrated separately. The base frame calibration is needed whenever conveyor systems are used.

The camera calibration is stored in the solution so all recipes in that solution could share the same calibration. If you need to re-calibrate a camera, all recipes in the solution will be updated with the new calibration.

The camera calibration and the work area calibration can be performed independently of each other, but it is very hard to make an accurate new camera calibration after the work area is calibrated.

The work area calibration is stored in the robot controller.

To calibrate the linear conveyor:

1. Define the parameter Counts Per Meter (for conveyors only), see Defining the parameter Counts Per Meter on page 220, Defining the parameter Counts Per Meter on page 197.

2. Calibrate the camera, see Defining the base frame on page 222, Defining the base frame on page 199.
4.4.5.1 Calibrating linear conveyor with DSQC 2000

4.4.5.1.1 Defining the parameter Counts Per Meter

Introduction

The Counts Per Meter system parameter is used to calibrate the conveyor encoder. The Counts Per Meter system parameter belongs to the type Conveyor Ici, in the topic Process.

Calculation for Counts Per Meter

The value for the Counts Per Meter system parameter is calculated as follows:

\[
\frac{\text{counts value}}{\text{measured_meters (m)}}
\]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>counts value</td>
<td>The conveyor position after moving. For DSQC 2000: Read from predefined I/O signal on the FlexPendant or RobotStudio. For example, CNV1, the signal name is c1counts.</td>
</tr>
<tr>
<td>measured_meters (m)</td>
<td>The manually measured distance in meters that the conveyor has been moved.</td>
</tr>
</tbody>
</table>

Defining Counts Per Meter

Use the following procedure to define Counts Per Meter for the conveyor encoder.

1. Put a mark on the conveyor belt, for example draw a line or attach a piece of tape, and a mark on the side of the conveyor at the same location.
2. In the FlexPendant Program Editor, load and run the program ppacal.prg. This sets the current position of the conveyor to zero. The value is shown as CNV value in the Position part of the FlexPendant Jogging window.
3. Run the conveyor belt approximately 1 meter.
4. In the FlexPendant Jogging window, read the position of the conveyor. This is position1.

Continues on next page
5 Measure the physical distance between the two marks. This is the value measured\_meters.

6 Calculate Counts Per Meter using the read and measured values.
   For example: $20200/1.005 = 20099$

7 In RobotStudio, click Configuration and select topic Process and type Conveyor Ici.

8 Edit the unit IC\textsubscript{x} (where x is the number of the conveyor) and update the value for parameter Counts Per Meter.

9 Tap OK.

10 Restart the controller.

Related information

* Application manual - Conveyor tracking.
4.4.5.1.2 Defining the base frame

Introduction

For each conveyor work area on a conveyor, a conveyor base frame calibration must be performed. The base frame calibration gives a reference point for the robot when a picking or placing sensor detects objects at the work area.

Preparations

- Define the Counts Per Meter system parameter for each conveyor work area. For more details, see Defining the parameter Counts Per Meter on page 220, Defining the parameter Counts Per Meter on page 197.

- Prepare a calibration tool that can be mounted temporarily on the robots. The calibration tool shall have a pointed TCP. Measure the TCP offset accurately.

- Create a tooldata for the calibration tool in the rapid program for each robot. Update the TCP offset with the measured values. In the FlexPendant Jogging Window, select the tooldata for the robot.

- If a camera is used, calibrate the camera, see Calibrating camera on page 276. After calibrating the camera, keep the camera calibration pattern attached to the conveyor.

Procedure for OmniCore

Use the following procedure to calibrate all the base frames for a conveyor in the line with OmniCore controller:

1. Make sure the reference point for calibration is marked accurately on the conveyor belt.
   - If a camera is used, the reference point is the local origin of the camera view. If the camera has been just calibrated, the reference point is already marked by the origin of the camera calibration pattern that is attached to the conveyor.
   - If an I/O sensor is used to generate predefined positions, the reference point should be marked on the conveyor at the point where the objects are detected by the sensor. This point becomes the local origin of the detected items or containers.

2. Reset the conveyor (encoder board) positions.

   Note

   Do not move the conveyor until this step is completely finished.

Continues on next page
Do the following for all the robots having work areas that needs to be calibrated along the conveyor:

- In the FlexPendant, click **Calibrate**.

- **Click Option Tab** on the up left corner.
4 Working with PickMaster PowerPac

4.4.5.1.2 Defining the base frame

Continued

- Click **Service Routines**.

- Click **PrepareCalibration**.

Continues on next page
4 Working with PickMaster PowerPac

4.4.5.1.2 Defining the base frame

*Continued*

- Click Yes in the popped up dialog.
4 Working with PickMaster PowerPac

4.4.5.1.2 Defining the base frame

Continued

- Set the controller to Manual mode.

- Enable the Thumb button to motors on the controller.

Continues on next page
4 Working with PickMaster PowerPac

4.4.5.1.2 Defining the base frame

Continued

• Click Play.

• Select the work area type Linear Conveyor.
• Select conveyor: for example, CNV2. Then click OK

• Wait for the message ...is prepared for calibration. The conveyor position in the jogging window for CNV2 should now be displayed as “0” mm.

3 Move the conveyor belt forward until the reference point is just inside the working range of the next robot to calibrate. The conveyor positions for all the conveyor work areas, in the jogging window should indicate the same total travel distance for the reference point. The nearest robot to the camera or sensor is calibrated first, followed by the next nearest robot and so on until all the robots along the conveyor have been calibrated.

4 Mount the calibration tool on the robot.
5 Open the Calibration window in Calibrate on the FlexPendant.

6 Select the conveyor, for example, CNV2.
7 Tap Define Base Frame.

8 Tap 4 Point and click Next.

9 Select the robot, for example, T_ROB1.
   This step is required for MultiMove robots.
10 Select the first point Point 1.
11 Jog or move the robot by hand. Point out the reference point on the conveyor accurately with the calibration tool TCP.

Continues on next page
12 Modify the selected point (Point 1) by tapping the Modify Position function key.

13 Move the conveyor belt forward a distance where the reference point still can be reached by the robot.

Long and equally spaced distances between the four calibration points (Point 1-4) are preferred since this increases the accuracy of the calibration.

14 Repeat the steps 10-13 for the points Point 2, Point 3, and Point 4.

15 Tap Next to calculate the base frame.
16 Check if the displayed mean error and max error of the base frame calculation is acceptable. If the estimated error is acceptable, tap Finish to confirm and store the new base frame.

**Note**

A mean error of less than 1 mm is acceptable in most cases.

If the estimated error is not ok, this base frame must be re-calibrated:

- Move the conveyor belt backward until the reference point is just inside the working range of the robot. Repeat the steps 10-13 for all the points Point 1, Point 2, Point 3, and Point 4.
- If the conveyor belt cannot be moved backward, start over from step 1.

17 If there are more robots to calibrate along the conveyor, continue from step 3.

18 Restart the controllers to activate the new base frames.

**Procedure for IRC5**

Use the following procedure to calibrate all the base frames for a circular conveyor with IRC5 controller:

1 Make sure the reference point for calibration is marked accurately on the conveyor belt.
   - If a camera is used, the reference point is the local origin of the camera view. If the camera has been just calibrated, the reference point is already marked by the origin of the camera calibration pattern that is attached to the conveyor.
   - If an I/O sensor is used to generate predefined positions, the reference point should be marked on the conveyor at the point where the objects
are detected by the sensor. This point becomes the local origin of the detected items or containers.

2 Reset the conveyor (encoder board) positions.

**Note**

Do not move the conveyor until this step is completely finished.

Do the following for all the robots having work areas that needs to be calibrated along the conveyor:

- In the FlexPendant, click **Menu** to open the drop-down list.

- **Click Program Editor** in the drop-down list.
4 Working with PickMaster PowerPac

4.4.5.1.2 Defining the base frame

Continued

- Click **Tasks and Programs**.

- Click **File** and **Load Program**.
• Click **Don't Save** in the popped up dialog.

• Click **Program Files (Old style)(.prg)** on the right upper corner drop-down list.
• Select PrepareCalib.prg and click OK.

• Set the controller to Manual mode.
4 Working with PickMaster PowerPac

4.4.5.1.2 Defining the base frame

Continued

- Enable the Thumb button to motors on the controller.

- Click Play.
4  Working with PickMaster PowerPac

4.4.5.1.2  Defining the base frame

Continued

- Select the work area type Lin CNV.

- Select conveyor: for example, CNV1.
4 Working with PickMaster PowerPac

4.4.5.1.2 Defining the base frame

Continued

- Wait for the message ...is prepared for calibration. The conveyor position in the jogging window for CNV1 should now be displayed as “0” mm.

3 Move the conveyor belt forward until the reference point is just inside the working range of the next robot to calibrate.

The conveyor positions for all the conveyor work areas, in the jogging window should indicate the same total travel distance for the reference point. The nearest robot to the camera or sensor is calibrated first, followed by the next nearest robot and so on until all the robots along the conveyor have been calibrated.

4 Mount the calibration tool on the robot.

5 Open the Calibration window on the FlexPendant.

Continues on next page
6 Select the conveyor, for example, CNV1.

7 Tap Base Frame and select 4 Point.

8 Select the robot, for example, T_ROB1.
   This step is required for MultiMove robots.
9 Select the first point Point 1.

10 Jog or move the robot by hand. Point out the reference point on the conveyor accurately with the calibration tool TCP.

11 Modify the selected point (Point 1) by tapping the Modify Position function key.

12 Move the conveyor belt forward a distance where the reference point still can be reached by the robot.

Long and equally spaced distances between the four calibration points (Point 1-4) are preferred since this increases the accuracy of the calibration.

13 Repeat the steps 10-13 for the points Point 2, Point 3, and Point 4.
14 Tap OK to calculate the base frame.

15 Check if the displayed mean error and max error of the base frame calculation is acceptable. If the estimated error is acceptable, tap OK to confirm and store the new base frame.

**Note**

A mean error of less than 1 mm is acceptable in most cases.

If the estimated error is not ok, this base frame must be re-calibrated:
- Move the conveyor belt backward until the reference point is just inside the working range of the robot. Repeat the steps 10-13 for all the points Point1, Point 2, Point 3, and Point 4.
- If the conveyor belt cannot be moved backward, start over from step 1.

16 If there are more robots to calibrate along the conveyor, continue from step 3.

17 Restart the controllers to activate the new base frames.
4 Working with PickMaster PowerPac

4.4.5.2 Calibrating linear conveyor with DSQC 377

4.4.5.2.1 Defining the parameter Counts Per Meter

**Introduction**

The *Counts Per Meter* system parameter is used to calibrate the conveyor encoder. The *Counts Per Meter* system parameter belongs to the type *DeviceNet Command*, in the topic *I/O System*.

**Calculation for Counts Per Meter**

The value for the *Counts Per Meter* system parameter is calculated as follows:

\[
\frac{(\text{position1} \times \text{old\_counts\_per\_meter})}{\text{measured\_meters}}
\]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>position1</td>
<td>The conveyor position after moving. Read from FlexPendant Jogging window.</td>
</tr>
<tr>
<td>old_counts_per_meter</td>
<td>The encoder's old value.</td>
</tr>
<tr>
<td>measured_meters</td>
<td>The manually measured distance in meters that the conveyor has been moved.</td>
</tr>
</tbody>
</table>

**Note**

The encoders delivered from factory have a preset value. For an IRC5 system this value is 20,000. This value can be used to start the calibration with.

**Defining Counts Per Meter**

Use the following procedure to define *Counts Per Meter* for the conveyor encoder.

1. Put a mark on the conveyor belt, for example draw a line or attach a piece of tape, and a mark on the side of the conveyor at the same location.

2. In the FlexPendant *Program Editor*, load and run the program ppacal.prg. This sets the current position of the conveyor to zero. The value is shown as CNV value in the Position part of the FlexPendant Jogging window.

3. Run the conveyor belt approximately 1 meter.

4. In the FlexPendant Jogging window, read the position of the conveyor. This is position1.

5. Measure the physical distance between the two marks. This is the value measured\_meters.

6. Calculate *Counts Per Meter* using the read and measured values.

Continues on next page
For example: \((1010 \times 20000) / 1005 = 20099\)

7 In RobotStudio, click Configuration and select topic I/O System and type DeviceNet Command.

8 Select the unit \(Qtrackx\) (where \(x\) is the number of the conveyor) and update the value for parameter Counts Per Meter.

9 Tap OK.

10 Restart the controller.

Related information

Application manual - Conveyor tracking.

4.4.5.2.2 Defining the base frame

Introduction

For each conveyor work area on a conveyor, a conveyor base frame calibration must be performed. The base frame calibration gives a reference point for the robot when a picking or placing sensor detects objects at the work area.

Preparations

- Define the Counts Per Meter system parameter for each conveyor work area. For more details, see Defining the parameter Counts Per Meter on page 220, Defining the parameter Counts Per Meter on page 197.
- Prepare a calibration tool that can be mounted temporarily on the robots. The calibration tool shall have a pointed TCP. Measure the TCP offset accurately.
- Create a tooldata for the calibration tool in the rapid program for each robot. Update the TCP offset with the measured values. In the FlexPendant Jogging Window, select the tooldata for the robot.
- If a camera is used, calibrate the camera, see Calibrating camera on page 276. After calibrating the camera, keep the camera calibration pattern attached to the conveyor.

Procedure

Use the following procedure to calibrate all the base frames for a conveyor in the line with IRC5 controller:

1. Make sure the reference point for calibration is marked accurately on the conveyor belt.
   - If a camera is used, the reference point is the local origin of the camera view. If the camera has been just calibrated, the reference point is already marked by the origin of the camera calibration pattern that is attached to the conveyor.
   - If an I/O sensor is used to generate predefined positions, the reference point should be marked on the conveyor at the point where the objects are detected by the sensor. This point becomes the local origin of the detected items or containers.
2. Reset the conveyor (encoder board) positions.

   Note

Do not move the conveyor until this step is completely finished.

Do the following for all the robots having work areas that needs to be calibrated along the conveyor:

- In the FlexPendant Program Editor, load the program ppacal.prg. If the robot is a MultiMove robot, load ppacal.prg for this robot task (for example, T_ROB1), and select only this task for execution.
• Start the loaded rapid program
  - Select calibration type: Conveyor.
  - Select conveyor: for example, CNV1.
  - Wait for the message READY FOR CALIB. The conveyor position in the jogging window for CNV1 should now be displayed as “0” mm.

3 Move the conveyor belt forward until the reference point is just inside the working range of the next robot to calibrate.

The conveyor positions for all the conveyor work areas, in the jogging window should indicate the same total travel distance for the reference point. The nearest robot to the camera or sensor is calibrated first, followed by the next nearest robot and so on until all the robots along the conveyor have been calibrated.

4 Mount the calibration tool on the robot.

5 Open the Calibration window on the FlexPendant.
6 Select the conveyor, for example, CNV1.

7 Tap Base Frame and select 4 Point.

8 Select the robot, for example, T_ROB1.
   This step is required for MultiMove robots.
9 Select the first point Point 1.

10 Jog or move the robot by hand. Point out the reference point on the conveyor accurately with the calibration tool TCP.

11 Modify the selected point (Point 1) by tapping the Modify Position function key.

12 Move the conveyor belt forward a distance where the reference point still can be reached by the robot.

   Long and equally spaced distances between the four calibration points (Point 1-4) are preferred since this increases the accuracy of the calibration.

13 Repeat the steps 10-13 for the points Point 2, Point 3, and Point 4.

Continues on next page
14 Tap OK to calculate the base frame.

15 Check if the displayed mean error and max error of the base frame calculation is acceptable. If the estimated error is acceptable, tap OK to confirm and store the new base frame.

**Note**

A mean error of less than 1 mm is acceptable in most cases.

If the estimated error is not ok, this base frame must be re-calibrated:

- Move the conveyor belt backward until the reference point is just inside the working range of the robot. Repeat the steps 10-13 for all the points Point 1, Point 2, Point 3, and Point 4.
- If the conveyor belt cannot be moved backward, start over from step 1.

16 If there are more robots to calibrate along the conveyor, continue from step 3.

17 Restart the controllers to activate the new base frames.
4.4.6 Calibrating circular conveyor

Overview

Note

The following calibration process is required when running production and emulation. Calibration under the simulation tab in PickMaster PowerPac will not complete the following calibration process.

The calibrations needed for the circular conveyors are camera and work area calibrations. The work area calibration is a base frame calibration for conveyor work areas and a work object definition for indexed work areas. The key concept is to define a coordinate system origin that is the same for a camera and a robot base frame or work object.

Each camera must be calibrated separately. The base frame calibration is needed whenever conveyor systems are used.

The camera calibration is stored in the solution so all recipe in that solution could share the same calibration. If you need to re-calibrate a camera, all recipes in the solution will be updated with the new calibration.

The camera calibration and the work area calibration can be performed independently of each other, but it is very hard to make an accurate new camera calibration after the work area is calibrated.

The work area calibration is stored in the robot controller.

To calibrate the circular conveyor:

1 Define the parameter Counts Per Meter (for conveyors only), see Defining the parameter Counts Per Meter on page 252, Defining the parameter Counts Per Meter on page 228.

Note

In the circular conveyor, the parameter Counts Per Meter indicates counts per radian.

2 Define the base frame, see Defining the base frame on page 254, Defining the base frame on page 230.
4.4.6.1 Calibrating circular conveyor with DSQC 2000

4.4.6.1.1 Defining the parameter Counts Per Meter

**Introduction**

The Counts Per Meter system parameter is used to calibrate the conveyor encoder. The Counts Per Meter system parameter belongs to the type Conveyor Ici, in the topic Process.

**Calculation for Counts Per Meter**

The value for the Counts Per Meter system parameter is calculated as follows:

\[
\text{counts value} / \text{measured radians}
\]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>position1/counts value</td>
<td>Read from predefined I/O signal on the FlexPendant or RobotStudio. For example, CNV1, the signal name is c1counts.</td>
</tr>
<tr>
<td>old_counts_per_meter</td>
<td>The encoder’s old value.</td>
</tr>
<tr>
<td>measured_radians(rad)</td>
<td>The manually measured radians that the conveyor has been moved.</td>
</tr>
</tbody>
</table>

**Defining Counts Per Meter**

Use the following procedure to define Counts Per Meter for the conveyor encoder.

1. Put a mark on the conveyor belt, for example draw a line or attach a piece of tape, and a mark on the side of the conveyor at the same location.

When this variable is applied to a circular conveyor, the actual meaning is counts per radian.
2 Hot start it to set the current position of the conveyor to zero. This sets the current position of the conveyor to zero. The value is shown as CNV value in the Position part of the FlexPendant Jogging window.

3 Rotate the conveyor belt approximately 180 degrees.

4 In the FlexPendant Jogging window, read the position of the conveyor. This is position1.

5 Measure the physical radians between the two marks. This is the value measured_radians.

6 Calculate Counts Per Meter using the read and measured values. For example: $30000 / 0.5 = 60000$

7 In RobotStudio, click Configuration and select topic Process and type Conveyor Ici.

8 Edit the unit ICix (where x is the number of the conveyor) and update the value for parameter Counts Per Meter.

9 Tap OK.

10 Restart the controller.

Related information

Application manual - Conveyor tracking.

4.4.6.1.2 Defining the base frame

Introduction

For each conveyor work area on a circular conveyor, a conveyor base frame calibration must be performed. The base frame calibration gives a reference point for the robot when a picking or placing sensor detects objects at the work area.

Preparations

- Define the Counts Per Meter system parameter for each conveyor work area. For more details, see Defining the parameter Counts Per Meter on page 252, Defining the parameter Counts Per Meter on page 228.
- Prepare a calibration tool that can be mounted temporarily on the robots. The calibration tool shall have a pointed TCP. Measure the TCP offset accurately.
- Create a tooldata for the calibration tool in the rapid program for each robot. Update the TCP offset with the measured values. In the FlexPendant Jogging Window, select the tooldata for the robot.
- If a camera is used, calibrate the camera, see Calibrating camera on page 276. After calibrating the camera, keep the camera calibration pattern attached to the conveyor.

Recommendation

This section describes how to use TCP measurements and RAPID programs to calculate the conveyor base frame position and quaternion for a circular conveyor. This method uses three measured points on the circular conveyor to calculate the center of rotation. The three points should be spaced as far apart as possible around the periphery.

Procedure for OmniCore

Use the following procedure to calibrate all the base frames for a circular conveyor with OmniCore controller:

1. Make sure the reference point for calibration is marked accurately on the conveyor belt.
   - If a camera is used, the reference point is the local origin of the camera view. If the camera has been just calibrated, the reference point is

Continues on next page
already marked by the origin of the camera calibration pattern that is attached to the conveyor.

- If an I/O sensor is used to generate predefined positions, the reference point should be marked on the conveyor at the point where the objects are detected by the sensor. This point becomes the local origin of the detected items or containers.

2 Mount the calibration tool on the robot.

3 Place the calibration grid X-aligned with the center line(a).

4 Rotate the belt to make the center line be parallel with the X-axis(b) of the calibrating robot.

Center line is a line connecting the centre point(c) of the circular conveyor and the X-axis on the calibration grid paper.

5 Reset the conveyor (encoder board) positions.

Note

Do not move the conveyor until this step is completely finished.
Do the following for the robot having work areas that needs to be calibrated along the conveyor:

A In the FlexPendant, click Calibrate.

B Click Option Tab on the up left corner.
4 Working with PickMaster PowerPac

4.4.6.1.2 Defining the base frame

Continued

C Click Service Routines.

D Click PrepareCalibration.
E Click Yes in the popped up dialog.

Continued
F  Set the controller to Manual mode.

G  Enable the Thumb button to motors on the controller.
4  Working with PickMaster PowerPac

4.4.6.1.2  Defining the base frame

Continued

H  Click Play.

I  Select the work area type Circular Conveyor.

Continues on next page
J Select conveyor: for example, CNV1. Then click OK

K Wait for the message ...is prepared for calibration. The conveyor position in the jogging window for CNV1 should now be displayed as “0” mm.
4 Working with PickMaster PowerPac

4.4.6.1.2 Defining the base frame

Continued

L  Wait for the message **Select rotation** and click the direction of the conveyor.

![Select rotation message](image)

xx2100000692

M  The program will continue automatically.

![Calibrating robots](image)

xx2100000693

6  If there are more robots need to be calibrating, repeat from step 4 to step 5 for each robot.

Continues on next page
7 Rotate the belt to make the calibration grid under the camera (d) (zero position).

8 Click Play on the FlexPendant of the robot(s) which have been reset.

9 Move the conveyor belt forward a distance where the reference point still can be reached by the robot.

Long and equally spaced distances between the three calibration points (Point 1-3) are preferred since this increases the accuracy of the calibration.

10 Jog or move the robot by hand. Point out the reference point on the conveyor accurately with the calibration tool TCP.
11 Modify the point (Pos 1) by tapping Play.
12 Repeat the steps for the points Pos 2 and Pos 3.
13 Check if the displayed mean error and max error of the base frame calculation is acceptable. If the estimated error is acceptable, restart the system to confirm and store the new base frame.

**Note**

A mean error of less than 1 mm is acceptable in most cases.

If the estimated error is not ok, this base frame must be re-calibrated.

**Tip**

Read the value of CAMERA OFFSET FROM CENTER. This value will be used as the input of Sensor offset in Type configuration for circular conveyor on page 258.

14 If there are more robots to calibrate along the conveyor, continue from step 2.

**Procedure for IRC5**

Use the following procedure to calibrate all the base frames for a circular conveyor with IRC5 controller:

1 Make sure the reference point for calibration is marked accurately on the conveyor belt.
   • If a camera is used, the reference point is the local origin of the camera view. If the camera has been just calibrated, the reference point is already marked by the origin of the camera calibration pattern that is attached to the conveyor.
• If an I/O sensor is used to generate predefined positions, the reference point should be marked on the conveyor at the point where the objects are detected by the sensor. This point becomes the local origin of the detected items or containers.

2 Mount the calibration tool on the robot.

3 Place the calibration grid X-aligned with the center line(a).

4 Rotate the belt to make the center line be parallel with the X-axis(b) of the calibrating robot.

   Center line is a line connecting the centre point(c) of the circular conveyor and the X-axis on the calibration grid paper.

5 Reset the conveyor (encoder board) positions.

   **Note**

   Do not move the conveyor until this step is completely finished.
Do the following for the robot having work areas that needs to be calibrated along the conveyor:

A  In the FlexPendant, click Menu to open the drop-down list.

B  Click Program Editor in the drop-down list.
4 Working with PickMaster PowerPac

4.4.6.1.2 Defining the base frame

Continued

C Click Tasks and Programs.

D Click File and Load Program.

Continues on next page
E Click **Don't Save** in the popped up dialog.

F Click **Program Files (Old style)(.prg)** on the right upper corner drop-down list.
G Select PrepareCalib.prg and click OK.

H Set the controller to Manual mode.
I Enable the Thumb button to motors on the controller.

J Click Play.
K. Select the work area type Circ CNV.

L. Select conveyor: for example, CNV1.
M Wait for the message ...is prepared for calibration. The conveyor position in the jogging window for CNV1 should now be displayed as “0” mm.

N Wait for the message Select rotation and click the direction of the conveyor.
The program will continue automatically.

6 If there are more robots need to be calibrating, repeat from step 4 to step 5 for each robot.

7 Rotate the belt to make the calibration grid under the camera(d) (zero position).
8 Click **Play** on the FlexPendant of the robot(s) which have been reset.

9 Move the conveyor belt forward a distance where the reference point still can be reached by the robot.

   Long and equally spaced distances between the three calibration points (Point 1-3) are preferred since this increases the accuracy of the calibration.

10 Jog or move the robot by hand. Point out the reference point on the conveyor accurately with the calibration tool TCP.

11 Modify the point (Pos 1) by tapping **Play**.

12 Repeat the steps for the points Pos 2 and Pos 3.

13 Check if the displayed mean error and max error of the base frame calculation is acceptable. If the estimated error is acceptable, restart the system to confirm and store the new base frame.

**Note**

A mean error of less than 1 mm is acceptable in most cases.
If the estimated error is not ok, this base frame must be re-calibrated.

**Tip**

Read the value of **CAMERA OFFSET FROM CENTER**. This value will be used as the input of **Sensor offset** in **Type configuration for circular conveyor on page 258**.
4.4.6.2 Calibrating circular conveyor with DSQC 377

4.4.6.2.1 Defining the parameter Counts Per Meter

Introduction

The Counts Per Meter system parameter is used to calibrate the conveyor encoder. The Counts Per Meter system parameter belongs to the type DeviceNet Command, in the topic I/O System.

Calculation for Counts Per Meter

The value for the Counts Per Meter system parameter is calculated as follows:

\[(\text{position1} \times \text{old_counts_per_meter})/\text{measured_radians}\]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>position1</td>
<td>Read from FlexPendant Jogging window.</td>
</tr>
<tr>
<td>old_counts_per_meter</td>
<td>The encoder's old value.</td>
</tr>
<tr>
<td>measured_radians</td>
<td>The manually measured radians that the conveyor has been moved.</td>
</tr>
</tbody>
</table>

Note

The encoders delivered from factory have a preset value. For an IRC5 system this value is 20,000. This value can be used to start the calibration with.

Defining Counts Per Meter

Use the following procedure to define Counts Per Meter for the conveyor encoder.

1. Put a mark on the conveyor belt, for example draw a line or attach a piece of tape, and a mark on the side of the conveyor at the same location.
2. In the FlexPendant Program Editor, load and run the program ppacal.prg. This sets the current position of the conveyor to zero. The value is shown as CNV value in the Position part of the FlexPendant Jogging window.
3. Rotate the conveyor belt approximately 180 degrees.
4. In the FlexPendant Jogging window, read the position of the conveyor. This is position1.
5. Measure the physical radians between the two marks. This is the value measured_radians.
6. Calculate Counts Per Meter using the read and measured values.

When this variable is applied to a circular conveyor, the actual meaning is counts per radian.
For example: \( \frac{1.5 \times 20000}{0.5} = 60000 \)

7 In RobotStudio, click Configuration and select topic I/O System and type DeviceNet Command.

8 Select the unit \( Qtrackx \) (where \( x \) is the number of the conveyor) and update the value for parameter Counts Per Meter.

9 Tap OK.

10 Restart the controller.

Related information

*Application manual - Conveyor tracking.*

*Technical reference manual - System parameters.*
4.4.6.2.2 Defining the base frame

Introduction

For each conveyor work area on a circular conveyor, a conveyor base frame calibration must be performed. The base frame calibration gives a reference point for the robot when a picking or placing sensor detects objects at the work area.

Preparations

- Define the Counts Per Meter system parameter for each conveyor work area. For more details, see Defining the parameter Counts Per Meter on page 252, Defining the parameter Counts Per Meter on page 228.
- Prepare a calibration tool that can be mounted temporarily on the robots. The calibration tool shall have a pointed TCP. Measure the TCP offset accurately.
- Create a tooldata for the calibration tool in the rapid program for each robot. Update the TCP offset with the measured values. In the FlexPendant Jogging Window, select the tooldata for the robot.
- If a camera is used, calibrate the camera, see Calibrating camera on page 276. After calibrating the camera, keep the camera calibration pattern attached to the conveyor.

Recommendation

This section describes how to use TCP measurements and RAPID programs to calculate the conveyor base frame position and quaternion for a circular conveyor. This method uses three measured points on the circular conveyor to calculate the center of rotation. The three points should be spaced as far apart as possible around the periphery.

Continues on next page
Defining the base frame orientation and start window start calibration

The base frame quaternion defines where the 0.0 rad point is for the robot motion. The following figure shows an example of the angles that are used when defining the base frame orientation for the circular conveyor.

Calculating the x and y positions for the base frame

Use this procedure to calculate the x and y positions for the base frame.

1. Use Wobj0 on the FlexPendant. Pick out a reference point on the circular conveyor, jog the TCP to this point and record \( p_0 \).
2. Run the conveyor to another position. Jog the TCP to the reference point and record \( p_1 \).
3. Run the conveyor to a third position, jog the TCP to the reference point and record \( p_2 \).
4. Use the function `CNVUTL_cirCntr` with the points \( p_0, p_1, \) and \( p_2 \), to calculate the center of the circle, \( p_{centre} \).

The system module `cnv_utl.sys` can be found in Robotware.
5 Take the x and y values from \textit{p\_centre} and enter them into the base frame values for the conveyor, converting to meters, see \\textit{Application manual - Conveyor tracking}. The z value will be entered later, once the work object zero position has been chosen.

### Calculating the quaternion

Use this procedure to calculate the quaternion for the base frame orientation.

1. With the recorded angle in step 5 when calculating the x and y positions for the base frame. This is angle $T P_\theta$, see example measurement points in \\textit{Defining the base frame orientation and start window start calibration on page 255}.

2. Calculate $P \theta$ from the $XP1$ and $YP1$ coordinates of $P0$ and the \texttt{atan} function.
   - If the point is at first quartile or fourth quartile: $P \theta = \text{atan}(YP1/XP1)$
   - If the point is at second quartile or third quartile: $P \theta = \pi + \text{atan}(YP1/XP1)$

\textbf{Tip}

If the calculation tool provide the \texttt{atan2} function, there is no need to judge the quartile and use $P \theta = \text{atan2}(XP1, YP1)$ directly.
3 Calculate the value of Base.

\[ Base_\theta = P_\theta - TP_\theta \]

4 Calculate the quaternion for the base frame taking into account the direction of rotation:

**Counter clockwise rotation:**

\[ q_1 = \cos(Base_\theta / 2) \]
\[ q_2 = 0.0 \]
\[ q_3 = 0.0 \]
\[ q_4 = \sin(Base_\theta / 2) \]

**Clockwise rotation:**

\[ q_1 = 0.0 \]
\[ q_2 = \cos(Base_\theta / 2) \]
\[ q_3 = -\sin(Base_\theta / 2) \]
\[ q_4 = 0.0 \]

5 Enter the value for z (in meters) from \( p_0 \), and the values for the quaternions, \( q_1, q_2, q_3, \) and \( q_4 \), into the base frame for the conveyor.
4.4.6.3 Type configuration for circular conveyor

Introduction

For each conveyor work area on a circular conveyor, the type parameters, Sensor offset, Mechanics and Rotating Move, must be set.

Sensor offset defines the distance between the sensor and the conveyor base frame original point. For example, when using a camera, this parameter represents the distance of the projection point of the camera on the conveyor belt from the center of the circle.

Note

For DSQC 377, the distance for Sensor offset is measured manually.
For DSQC 2000, the distance for Sensor offset is read from the program result. See the value for IRC5 on page 251 and for OmniCore on page 240.

Mechanics defines the moving trajectory of the conveyor. The default value is EXT_LIN (linear conveyor). So when the circular conveyor is used, this parameter must be set as EXT_ROT.
Rotating Move defines the conveyor’s rotating status. The default value is No (linear conveyor). So when the circular conveyor is used, this parameter must be set as Yes.
4.4.7 Calibrating indexed work area

Introduction

For indexed work areas a work object calibration must be performed. The work object calibration gives a reference point for the robot when picking or placing sensor detected objects at the work area.

Preparations for calibrating the indexed work area

- Prepare a calibration tool that can be mounted temporarily on the robot. The calibration tool shall have a pointed TCP. Measure the TCP offset accurately.
- Create a tooldata for the calibration tool in the rapid program for the robot. Update the TCP offset with the measured values. In the FlexPendant Jogging Window, select the tooldata for the robot.
- Calibrate the camera, see Calibrating camera on page 276. After calibrating the camera, keep the camera calibration pattern attached to the conveyor.
- Make sure the reference x- and y-axes for work object calibration is marked accurately on the indexed work area. Three reference points are needed for the calibration: two points on the x-axis and one point on the y-axis.
  - If a camera is used, the reference x- and y-axes should be marked with respect to the local origin of the camera view. If the camera just has been calibrated, the local origin is marked by the camera calibration pattern attached to the indexed work area.
  - If a position generator I/O signal is used to generate predefined positions, the reference x- and y-axes should be marked at the desired location for the local origin where items or containers are to be generated.
Procedure(OmniCore)

1 Make sure the reference point for calibration is marked accurately on the conveyor belt.
   - If a camera is used, the reference point is the local origin of the camera view. If the camera has been just calibrated, the reference point is already marked by the origin of the camera calibration pattern that is attached to the conveyor.
   - If an I/O sensor is used to generate predefined positions, the reference point should be marked on the conveyor at the point where the objects are detected by the sensor. This point becomes the local origin of the detected items or containers.

2 Reset the conveyor (encoder board) positions.

   ![Image of FlexPendant with Calibrate button highlighted]

   Note

   Do not move the conveyor until this step is completely finished.

   Do the following for all the robots having work areas that needs to be calibrated along the conveyor:
   - In the FlexPendant, click Calibrate.
4 Working with PickMaster PowerPac

4.4.7 Calibrating indexed work area

Continued

• **Click Option Tab** on the up left corner.

![Option Tab](image1)

• **Click Service Routines.**

![Service Routines](image2)

Continues on next page
• Set the controller to **Manual** mode.

• Enable the **Thumb** button to motors on the controller.

*Continues on next page*
4 Working with PickMaster PowerPac

4.4.7 Calibrating indexed work area

Continued

- Click **PrepareCalibration**.

- Click **Yes** in the popped up dialog.
4 Working with PickMaster PowerPac

4.4.7 Calibrating indexed work area

Continued

Click Play.

Continues on next page
4 Working with PickMaster PowerPac

4.4.7 Calibrating indexed work area

Continued

• Select the work area type Stationary.

![Image of software interface with selected work area type]

• Select conveyor: for example, ldxwobj1. Then click OK

![Image of software interface with conveyor selection]

Continues on next page
• Wait for the message ...is prepared for calibration. The conveyor position in the jogging window for CNV1 should now be displayed as “0” mm.

3 Return to the home page and select Program Data.
4 Select Workobject in Data Type.

5 In the Workobject, tap on the ... to select Define.
6 In the Define User frame window, set the User Method as User defined with 3 points.

![Image of Define User frame window]

7 Select X1. Point out a location on the x-axis close to the origin with the robot's TCP.

![Image of Define User frame window with X1 selected]
8 Press Modify.

9 Select X2. Move the TCP a distance in the direction the x-axis. Point out a location on the x-axis with the robot's TCP.

10 Press Modify.

11 Select Y1. Point out a location on the positive y-axis with the robot's TCP.

12 Press Modify.

13 Tap Next.
14 In the Define Object frame window, tap Next.

![Image of Define Object frame window]

15 Check if the displayed mean error and max error of the user frame calculation is acceptable. If the estimated error is acceptable, tap Finish to confirm and store the new user frame.

![Image of Workobject Definition window showing user frame calculation results]

16 Enable the Thumb button to motors on the controller.

17 Click Play.
18 Click Yes on the question: Do you want to save this work object definition.

![Image](xx2100000393)

19 The definition is saved in the rapid data array NonCnvWOData located in the ppaUser system module.

Procedure (IRC5)

1 Select the work object to be calibrated.

- In the FlexPendant Program Editor, load the program ppcal.prg (DSQC 377)/ PrepareCalib.prg (DSQC 2000). If the robot is a MultiMove robot, load ppcal.prg (DSQC 377)/ PrepareCalib.prg (DSQC 2000) for this robot task (for example, T_ROB1), and select only this task for execution.
• Start the loaded rapid program
  - Select calibration type: Fixed/Indexed.
  - Select work object: For example,IdxWobj1.
  - Wait for the message DEFINE CURRENT WORKOBJECT.

**Note**
Do not move the program pointer until the calibration has been completed. Otherwise, the calibration is not properly saved.

2 In the FlexPendant Jogging window, tap and select Workobject. Then tap Edit and select Define.

3 Select Object method: No Change. Select User method: 3 points.

4 Select User Point X 1. Point out a point on the x-axis close to the origin with the robot's TCP. Press Modify Position.

5 Select User Point X 2. Move the TCP a distance in the direction the x-axis. Point out a point on the x-axis with the robot's TCP. Press Modify Position.

6 Select User Point Y 1. Point out a point on the positive y-axis with the robot's TCP. Press Modify Position.

7 Tap OK.

8 Restart the RAPID program (without moving the PP) to save the selected work object definition.

   The definition is saved in the rapid data array NonCnvWOData located in the ppaUser system module.
4.4.8 Verifying conveyor calibrations

Introduction
The calibration is verified by using a calibration verification paper. The paper has a model that is taught and used as a bull’s eye for the robot to find. The same tool is used here as for the base frame calibration.

The file with the calibration verification paper is found in the PickMaster package.
To achieve a very good calibration, the camera calibration tune and the base frame calibration tune steps can be performed more than once. Each time the result should be closer to the optimal calibration.

Note
The calibration tuning should only be used for small errors. If the error is large then the line should be recalibrated.

Tuning the camera and base frame calibrations
Use this procedure to tune the camera and base frame calibrations.

1. Place the calibration verification paper on the conveyor under the camera. The center column of object should be placed close to the center of the camera view. Align the paper with the conveyor as accurately as possible.

2. Use one of the objects on the calibration verification paper as model. See Calibrating camera on page 276.

3. Place the grip position in the center of the model.

4. Examine how the robot is placing the holes to adjust possible errors in the camera calibration or the base frame calibration.

Continues on next page
If the holes are rotated too much compared to the center of the objects, which affects the accuracy of the grasp, then recalibrate the cameras.

If the holes are off center of the objects too much, which affects the accuracy of the grasp, then recalibrate the base frame of the conveyor.

4 The angle between the hole to the center and the X axis.
4.4.9 Calibrating camera

Introduction

Overview

The camera calibration defines the origin for the coordinate system shared by the camera and the robot base frame or work object. If the camera is used with a conveyor work area the camera calibration must be performed before the base frame calibration because the camera calibration origin works as a common reference point for the two calibrations. When a camera calibration is done, the origin is saved and the user can graphically display this origin when the base frame calibration is performed.

Note

If any firewall or antivirus software is installed, add pickmasteru.exe, sshd.exe, and visionclient.exe to the white list.
Otherwise the PickMaster PowerPac cannot connect Runtime and the vision function cannot work normally.

Checkerboard calibration

The camera calibration method is called checkerboard calibration. The calibration is performed in two steps. First the whole image is analyzed and warped into a correct image and then the region of the resulting image is defined.

The algorithm uses the scale in the center of the image, which means that it makes all the tiles the same size as the tile at the center of the original image.

Multi-view calibration

The camera can be calibrated using one or several images. The difference when using more than one image is that the camera’s position in space is calculated. This space information is used both for 2.5D applications when the product height needs to be determined, and for compensating parallax errors in pure 2D applications.

See Working with products of varying height (2.5D vision) on page 326.

The accuracy of the multi-view calibration increases with the number of input images.

Use atleast 10-15 images with the following specifications:

- A set of images with different heights where the calibration pattern is flat under the camera (3 to 5 images).
- A set of images where the calibration pattern has different tilt and heights. (Minimum 3 images but more images give better results.)
- Place the calibration pattern down on the conveyor surface. This should be the origin image.
Note

Using multiple images of calibration plates in parallel planes does not increase accuracy.

Prerequisites

Camera calibration is done using calibration papers that you must print out. The calibration papers are found in the `C:\Program Files (x86)\ABB\PickMaster Twin 2\PickMaster Twin Client 2\PickMaster Runtime\Documentation\CalibrationPapers` or `C:\Program Files (x86)\ABB\PickMaster Twin 2\PickMaster Twin Host 2\PickMaster Runtime\Documentation\Calibration Papers`.

The printed image must have a high contrast and the surface must not be reflective (high gloss). Make sure that the calibration paper covers the full field of view of the camera. Verify with a ruler that the squares are proportional. Enter the correct width and length of the squares. The precision must be at a minimum of a tenth of mm. To obtain an accurate value, measure the full length of the calibration sheet in each direction and divide by the number of squares.

The calibration paper must be adequately illuminated and free from shadows.

If a conveyor is used, the x-axis of the calibration paper must be aligned with the positive motion direction of the conveyor.

Calibrating the camera

The Camera Calibration dialog can be used to handle camera calibrations for the specified camera. Calibrations can be created, edited, imported, and exported.

Use this procedure to calibrate the camera.

1. Right-click the camera in the tree view Cameras and select Calibration.

The Camera Calibration window is opened.
4 Working with PickMaster PowerPac

4.4.9 Calibrating camera

Continued

2 Place the verification paper on the conveyor under the camera. Align the paper with the conveyor as accurately as possible.

Tip

All images used for calibration should be taken with the central part of the calibration board always in the image.

3 Select the default calibration from the list and click Edit.

The Camera Calibration Feature and Calibration dialog are opened.
4 In the Image part, click Live to get and show new images continuously, or click Acquire to get one new image. To use an image from file or save the current image, click Import or Export.

5 For single-view calibration: When the calibration plate is in position, acquire an image and click Set Origin in the Calibration images part. This stores the image and marks it as the origin image (the origin of this image will be the physical origin of the camera’s coordinate system).

6 For multi-view calibration: When calibrating a camera with multiple images it is important that the origin image is still in place after finishing the camera calibration. This is because the origin image is used to define the coordinate system of the robot.

There are two ways of achieving this. One way is to acquire additional views first (click Acquire and Add) and acquire the origin image last (click Acquire and Set origin), leaving the calibration plate in the correct place for calibration of the work object/base frame.

The other way is to use two calibration plates with the exact same grid pitch. Put one calibration plate in the position to represent the origin of the camera. Acquire an image and click Set origin. Leave this plate in place while acquiring images of the second calibration plate at different angles and altitudes and click Add to save them to the list.

Tip

With multi-view calibration, the space information is calculated automatically.
4 Working with PickMaster PowerPac

4.4.9 Calibrating camera

Continued

7 In the Calibration part, click Calibrate to start calibration.
   The image is analyzed and calibration is performed with the specified parameters. A corrected image is shown together with an adjustable rectangle used to define the final image area. The calibration is not complete until the region is defined.

8 Adjust the rectangle to the desired region and click Set Region to define the resulting image size.
   The calibration is now completed and the result is displayed in the Calibration result part. See Calibration result on page 281.

9 If needed, click:
   • Calib Image to show the original image used to calibrate the camera.
   • Warp Live to show continuously acquired and corrected images.
   • Warp Image to correct the current image.

10 If needed, click:
   • Show features to show the checkerboard vertices used during the calibration. The features are only shown in the calibration images.
   • Show origin to show the origin of the resulting coordinate system. The origin is only shown in corrected images.

11 Click OK.

Tip
For conveyors, leave the calibration paper as it is until the base frame has been calibrated.

Continues on next page
Note

You can export or import camera calibrations. The exported file is stored in .pmcalib format. It is also possible to export images from the camera calibration window for storing the images used for a certain calibration.

Calibration result

Single-view calibration:

<table>
<thead>
<tr>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max residual</td>
<td>The maximum residual error for the calibration.</td>
</tr>
<tr>
<td>Average residual</td>
<td>The average residual error for the calibration.</td>
</tr>
<tr>
<td>Warp time</td>
<td>The time required correcting an image. This time has to be considered when calculating the total time for the image analysis.</td>
</tr>
<tr>
<td>Image size</td>
<td>The resulting size in pixels of the corrected image</td>
</tr>
<tr>
<td>Camera view</td>
<td>The resulting size of the camera view calculated with the new calibration.</td>
</tr>
<tr>
<td>Camera location</td>
<td>The position of the camera in relation to the origin of the origin calibration plate.</td>
</tr>
</tbody>
</table>

Multi-view calibration:

<table>
<thead>
<tr>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max residual</td>
<td></td>
</tr>
<tr>
<td>Average residual</td>
<td></td>
</tr>
<tr>
<td>Warp time</td>
<td></td>
</tr>
<tr>
<td>Image size</td>
<td></td>
</tr>
<tr>
<td>Camera view</td>
<td></td>
</tr>
<tr>
<td>Camera location</td>
<td></td>
</tr>
</tbody>
</table>
4.4.9.1 Showing live images

It is possible to view images from each camera when a production is running.

**Note**

Showing Runtime images requires much processing power and should not be used for a long period of time if complex vision models are used.

To show images, click Control. The camera images are shown in the Vision tab. The found objects are shown as green or blue crosses, depending on if they are marked as accepted or rejected by the vision model. See *Vision modeling on page 286*.
4.4.9.2 Detailed vision information

Detailed vision information

More detailed information than given by the live images is shown in the Detailed Vision Information dialog. This dialog box keeps a buffer of images and information about the corresponding vision model hits.

Sequences of images can be recorded to the buffer and then analyzed individually. While recording, images are saved in the buffer in a first in, first out basis and the latest image is shown in the dialog.

When switching off the Image Buffer function, images are no longer added and the images in the buffer can be analyzed. Save the images in the current buffer to file for later analysis with the Vision Analyzer program, see Vision Analyzer on page 284.

You can switch to different cameras from the drop-down list.

The maximum size of the buffer depends on the RAM memory on the computer.

Illustration

Click Vision tab under Control to open the dialog. By default, the recording state is activated and the buffer max volume is set to 10 images.

| Image buffer | Used to switch between recording or pause and set the image buffer size. Click Save As to save all images in the buffer to a .pmv file. Step through the image buffer when recording is paused. LEFT or RIGHT ARROW button can also be used to step. Click Export to save the current image to file (.bmp format). |
| Display options | Select which vision models to display, all together or individually, and other settings for what to show in the images. The settings are valid both for recording and pause. |
| Search Result | The list view at the bottom shows information about all the hits. When an individual model is selected, the columns change depending on its type. |
| Image Dialog | The pan and zoom buttons can be used to analyze the image more closely. |

Continues on next page
Vision Analyzer

Image buffers recorded in the Detailed Vision Information dialog can be saved as .pmv files. These files can be viewed with a separate program called PickMaster Vision Analyzer.

Start Vision Analyzer from the PickMaster Twin Client installation folder or from Windows Start menu.

Click **Load** to open a .pmv file.

Click **Camera** to see detailed information about the camera that took the images.

Other settings in Vision are identical to settings in Detailed Vision Information.
4.4.9.3 The image windows

The image windows

When configuring a camera or a vision model the camera image is shown in a separate window. The image window is resizable and provides tools to quickly zoom and pan the shown image. Some tools change the appearance of the mouse pointer.

To zoom using the keyboard and mouse, place the pointer over the image, press CTRL and scroll the mouse wheel.

The current zoom level and the world coordinate of the mouse pointer is shown in the status bar. When live images are shown, the current frame rate is also shown in the status bar.
4.4.10 Adding vision model

4.4.10.1 Vision modeling

Introduction to vision modeling

There are three different tools available for generating models in a solution. The three tools are:

- Geometric PatMax which is a pattern recognition tool. See Configuring a geometric model with PatMax on page 289.
- Blob which is a detection of two-dimensional shapes within images. See Configuring blob models on page 297.
- Inspection tool (Inspection II) which makes it possible to combine the PatMax, Blob, Histogram and Caliper to generate a model. See Configuring inspection models on page 304.

Note

Vision modeling can only be created or edited when the software is connected to real Runtime.

Note

You can import vision models from PickMaster 3 solutions and other PickMaster PowerPac solutions.

Importing an existing vision model

Use this procedure to import an existing vision model.

1. Right-click on one Item in the tree view Process and select Setting. The Item Setting window is opened.
2. Click to select the Item Source tab.
3. In the Item Source dialog, click Import Model under the required camera. The Import Vision Model window is opened.
4 Select the valid vision model (.pmmodel or .pmmodel.zip) and click Open

5 Click OK.

Classification of Items

Items identified by vision models can be classified as either accepted or rejected. These two types can be distributed to different work areas and be given different item type values accessible from the RAPID program. Item classification can be done by PatMax, Blob, and the Inspection tool.

Vision model parameters in item targets

Item targets identified by a vision model can store a selection of up to 5 vision model parameters in the components Val1, Val2, Val3, Val4, and Val5. These parameters can be accessed in the RAPID program.

Item targets identified by an inspection model can store a selection of parameters from the alignment model and from the included subinspection models.

For each kind of vision model, a target storage can be selected for some vision parameters.

External vision models

This function is reserved for next version.
Related information

- Configuring a geometric model with PatMax on page 289.
- Configuring blob models on page 297.
- Configuring inspection models on page 304.
4.4.10.2 Configuring a geometric model with PatMax

Introduction to the geometric model PatMax

*PatMax* is a pattern location search technology. This tool measures:

- Position of the pattern.
- Size relative to the originally trained pattern.
- Angle relative to the originally trained pattern.

*PatMax* differs from other pattern location technologies as it is not based on pixel grid representations that cannot be efficiently and accurately rotated or scaled. Instead, *PatMax* uses a feature based representation that can be transformed quickly and accurately for pattern matching.

When creating a pattern the following things should be considered.

- Select a representative pattern with consistent features. Reduce needless features and image noise. Train only important features. If necessary, export the image and use an external program to erase noise.
- Larger patterns will provide greater accuracy because they contain more boundary points to resolve at run-time.
- High frequency features are more significant at the outer edges of the pattern.

Models can be classified with the function *Inspection I*. A model can either be defined as accepted or rejected, see *Item Properties tab on page 127*.

To increase the contrast in images where parts have similar grayscale tone, it is possible to use color filtering. See *Using color vision on page 317*.

There are several parameters that can be adjusted to make an efficient model. The configuration is done in the Geometric Model tab view and the result is displayed in the Search Result window and the Image Dialog.

Illustration geometric model Configuration
Configuring a geometric model with PatMax

Use this procedure to configure a geometric model with PatMax.

1 Right-click on one Item in the tree view Process and select Setting.
   The Item Setting window is opened.
2 Click to select the Item Source tab.
3 In the Item Source dialog, click New model under the required camera and select Geometric.
   The Image Dialog and Geometric dialog are opened.
4 In the Model Definition, click Live, Acquire, or Import to get an image. Select the calibration that has set in the Camera Calibration from the Calibration list. Select the Calibration grid checkbox to display help lines for the coordinate system.
The help lines can be moved with the mouse to make it easier to train a pattern.

5 If color filtering should be used select the Color filter checkbox to enable the filter. Configure the filter parameter in the Color Filter tab. See Using color vision on page 317.

6 In the Model definition part, define a model for the pattern using an image in front of the camera or using an imported image. The selected calibration will be used.

**Note**

When importing a vision model it is required to enter model configuration and re-select which calibration to use from the calibration drop-down menu. This is required even if there is only one calibration defined. If this is not performed then further actions may produce the error No valid calibration for the PatMax model.

- If the height of the item is to be defined, choose an appropriate calculation method before training the item. Model Height is used as the basic height for the trained item. Pick Offset is used to make up the deviation of the picking point with this calculation method. For more information, see Working with products of varying height (2.5D vision) on page 326.

- Click Define to define a model. Drag the rectangle so it covers the pattern and move the cross to the desired pick/place position. To maintain the greatest accuracy, the pick/place position should be placed close to the center of the trained pattern.

- Click Train to train the pattern.
d Select Show Model to show the features of the trained models in the search image.

e If needed, click Advanced to access more model settings.

f Click Adjust Granularity to define the levels in the Fine and Coarse boxes. Granularity is a radius of influence, in pixels, which determines the detection of a feature in a pattern. PatMax locates patterns in the search image by first searching only for large features. After locating one or more pattern instances, it uses smaller features to determine the precise transformation between the trained pattern and the pattern in the search image. PatMax uses the same range of granularity that is computed when training the pattern to detect features in the search image. The granularity parameters fine and coarse are auto-selected when training the pattern and often these values are the best. These can also be set manually. The lower limit is 1 and upper limit is 25.5.

g Select Ignore polarity to ignore if the features are dark on bright or bright on dark.

**Note**

PatMax will not care if a product is light on a dark background or dark on a light background. This is useful when the background is, for example, a grid.

h Increase the value of Elasticity to allow for any expected non-linear shape distortion, for example, for organic products and so on. The value represents the maximum distance between a trained feature and a matched feature in pixels. The lower limit is 0 and upper limit is 25. This setting is useful for products of varying shape.

7 In the Search parameters part, set parameters to limit the search procedure and the analysis time.

Score Limit indicates how closely the found item matches the trained model. A score of 1 indicates a perfect match while a score of 0 indicates that the pattern does not match at all. The higher a score threshold is defined the faster PatMax will be able to perform a search.

Target Storage indicates the variables in Rapid. For more details, see GetItmTgt - Get the next item target on page 340

Items to Find is the number of items that is expected to be present in the image. If there are more items present in the image these will not be reported by PatMax.

Contrast Limit defines the minimum image contrast of each item that is found in the image. The contrast is the average difference in gray-level values for all of the boundary points that PatMax matched between the trained model and the found item in the search image. PatMax considers only items with a contrast value that exceeds the contrast limit.

Area Overlap defines how much multiple patterns in the image are allowed to largely overlap each other. PatMax assumes that these patterns actually

Continues on next page
represent the same item in the image. When two patterns overlap by a percentage greater than the area overlap threshold they are treated as a single pattern.

Enable Angle variation defines the acceptable rotation for the items. If an item has a rotation outside the valid range it will be discarded by the vision system. Default +/- 180 degrees.

Enable Uniform Scale is a threshold that accepts hits that differ in size relative to the taught vision model. A scale value of 1 indicates that there are no differences between the found item and the taught vision model. A value <1 indicates a smaller model.

Score Using Clutter defines a measure of the extent to which the found item contains features that are not present in the trained vision model. By default the PatMax analysis ignores clutter when scoring which means that the patterns receive the same score regardless of the presence of extra features. If this checkbox is selected, clutter is included in the calculation of the score. If the application is an alignment application in which the background does not change, Score Using Clutter should be selected.

Limit Search Region limits the search area for the PatMax analysis. Only objects within this area will be found. A smaller search area will decrease the search time.

Note

When combining Fine/Coarse Granularity and Uniform Scale a slight difference in the score can appear between design time and running time. Therefore, the model should be tested in running time to verify that items are identified as expected.

8 In the Post search filters part, define the score values for each pattern in the search image.

Fit Error Limit is a measure of the variance between the shape of the trained pattern and the shape of the pattern found in the search image. If the found pattern in the search image is a perfect fit for the trained pattern, the fit error is 0.

Coverage is a measure of the extent to which all parts of the trained pattern are also present in the search image. If the entire trained pattern is also present in the search image, the coverage score is 1. Lower coverage scores indicate that less of the pattern is present. This parameter can be used to detect missing features.

Clutter is a measure of the extent to which the found pattern contains features that are not present in the trained pattern. A clutter of 0 indicates that the found pattern contains no extra features. A clutter score of 1 indicates that for every feature in the trained pattern there is an additional extra feature in the found pattern. The clutter can exceed 1.0.
If more settings are required, click Advanced to open the Advanced Search Settings dialog where the following settings are found:

- **Use Inspection Levels**
  - **Score Limit:** Determines the score threshold.
  - **Min Angle:** Minimum angle for classification.
  - **Max Angle:** Maximum angle for classification.
  - **Contrast Limit:** Contrast threshold.
  - **Min UScale:** Minimum uniform scale.
  - **Max UScale:** Maximum uniform scale.
  - **Fit Error:** Error threshold.
  - **Coverage:** Coverage threshold.
  - **Clutter:** Clutter threshold.

- **Limit Position Region**
  - Define the position region with the orange box in Image Dialog.

**Use Inspection Levels - Inspection I**, this inspection is also called *Inspection I* in PickMaster PowerPac. With this function it is possible to classify the found models into two categories. A model can either be classified as accepted or rejected. An accepted model has better search results than the rejected model. The item type number is defined for the accepted and rejected model in the *Item* dialog, see *Item Properties tab on page 127*. An item type can be read in the RAPID code, see *RAPID programs on page 367*.

In the Inspection parameter section, all models that fulfill the conditions specified for the search parameters and the post filters will be classified. Select **Use Inspection Levels** to define the parameter that will divide the found items into the two categories. If **Use inspection levels** is not selected all found models are classified as an accepted model.

- **Score**, **Contrast**, and **Coverage**, items with a value larger than the defined value in **Inspection Parameter** will be defined as accepted.
- **Angle** and **Uniform Scale**, items with a value between the defined values in **Inspection Parameter** will be defined as accepted.
- **Fit Error** and **Clutter** a value less than the defined one will be classified as accepted.

**Limit Position Region** defines if the *PatMax* analysis is done on the whole image. Objects found within this area will be handled as normal. Object found outside this area will be discarded.
To define an item region, select **Use Item Region** checkbox and click **Define Region**. Adjust the polygon showed around the found object using vertices. Then click **Train**. The polygon can have 2 to 16 vertices.

9 In the **Display options** part, select the type of information to display in graphics.

**Match Info** displays the quality of the matched boundary points in the search image. Boundary points drawn in:
- Red are poor matches.
- Yellow are fair matches.
- Green are good matches.

**Item Score** displays the score for the selected item in the image window. **Item Region** displays the regions in the image window. Red regions indicate an overlap and the corresponding hits will be considered as discarded.

**Item Angle** displays the angle of the item that will be sent to the robot. This angle is relative to the trained model.

**Sort value** is used if there is more than one hit for the same item. Only the hit with the highest sort value will be sent to the robot controller. The sort value can be set individually for all models or the **PatMax** score can be used by selecting **Score as sort value**.

10 Click **Search** to analyze the image. If needed, define sort value.

The result is displayed as an image with numbered hits in the **Image** dialog, and a corresponding result list.

![Image](image.png)

Model hits are normally classified as accepted. If inspection is used, hits can be classified as either accepted or rejected. See **Item Properties tab on page 127**. Hits that do not fulfill all the requirements or hits with overlapping regions will not be accessed by any robot and are classified as discarded. The hits shown in the result list are marked with an icon identifying its classification. For hits that are not accepted, the parameter that failed is marked with either red or blue in the result list.

**Search Time** displays the time it takes to analyze the image in ms.

Continues on next page
11 Click OK.

**Note**

Items located after a search operation in the PatMax configuration window is presented as discarded due to item region overlap even if they are actually rejected due to another parameter (fit error, clutter, and so on). This happens only if the item region is activated and the item regions overlap with each other in running time. However, the discarded items are removed before applying the item region.

**PatMax parameters in item targets**

The PatMax parameters Score, fit error, coverage, and clutter can be selected for the target storage.

**Related information**

- *Item Properties tab on page 127.*
- *Using color vision on page 317.*
- *RAPID programs on page 367.*
4.4.10.3 Configuring blob models

Introduction to blob models

The simplest kinds of images that can be used for machine vision are two-dimensional shapes or blobs. Blob analysis is the detection of two-dimensional shapes within images. It finds objects by identifying groups of pixels that fall into a predefined grayscale range.

This kind of analysis is well suited for applications where:

- Objects vary much in size, shape, and/or orientation.
- Objects are of a distinct shade of gray not found in the background.

Blob analysis works best with images that can be easily segmented into foreground and background pixels. Typically, strong lighting of scenes with opaque objects of interest produces images suitable for an analysis like this.

To increase the contrast in images where parts have similar grayscale tone, it is possible to use color filtering. See *Using color vision on page 317.*

Illustration Blob Configuration

![Blob Configuration Illustration](image-url)
4 Working with PickMaster PowerPac

4.4.10.3 Configuring blob models

Continued
Configuring a blob vision model

Use this procedure to configure a blob vision model.

1. Right-click on one Item in the tree view Process and select Setting.
   The Item Setting window is opened.

2. Click to select the Item Source tab.

3. In the Item Source dialog, click New model and select Blob.

4. In the Image part, click Live, Acquire, or Import to get an image. Select the Calibration origin checkbox to display help lines for the coordinate system. Click Histogram to display a graph of the pixel distribution in the acquired image.

   If color filtering should be used, select the Color filter checkbox to enable the filter and configure the filter parameter in the Color Filter tab. See Using color vision on page 317.

5. Click to select White in the Segmentation under Model Definition. In the Segmentation part, select segmentation method and blob type.

   Segmentation is the division of the pixels in an image into object pixels and background pixels. Typically objects are assigned a value of 1 while background pixels are assigned a value of 0.

   Static method uses gray values to divide blob pixels and background pixels. All pixels with a grayscale value below the threshold are assigned as object pixels, while all pixels with values above the threshold are assigned as background pixels.

   Relative method uses a relative threshold expressed as the percentages of the total pixels between the left and right tail to divide blob pixels and background pixels. Tails represent noise-level pixels that lie at the extremes of the histogram (the lowest and the highest values).

   Static is faster than relative segmentation because the gray levels corresponding to the percentages do not have to be computed. Static

Continues on next page
segmentation can test for absence of a feature in a scene, whereas relative segmentation will always find a blob in the scene.

6 Adjust the parameters in the Search Parameter according to your requirements.

In the Search Parameters part, define the values for the feature. 
Area is expressed in mm\(^2\).

Perimeter is expressed in mm.
Circularity defines the circularity. A value of 1 means perfectly circular and completely filled (no holes).
Elongation is the ratio of the feature’s second moment of inertia about its second principal axis to the feature’s second moment of inertia about its first principal axis.

Angle defines how the found item is sent to the controller.

- No Orientation means that the found item is sent to the controller with angle 0 (zero).
- First Principal Axis means that the found item is sent down with the angle around the first principal axis. The angle is relative to the x-axis and can be ±90 degrees.

Use boundary box center defines if the position of a blob will be at the center of its boundary box instead of at its center of mass.

No Blob On Edge defines if a blob connected to the edge of the search area should be reported.

Use Inspection Levels defines if the found models should be classified. See Item Properties tab on page 127. The item type can be read in the RAPID code, see RAPID programs on page 367. Select Use Inspection Levels to open the Inspection Parameters part.
If Use Inspection Levels is not selected all found models are classified as accepted. All models that fulfill the conditions specified for the Search Parameters will be classified.

Limit Search Region limits the search area for the blob analysis. Only objects within this area will be found.

Note

Tune the blob tool by pressing Search and the blob algorithm lists all the blobs. Adjust the size threshold limit to filter out blobs that are too large or too small. Tune other parameters if necessary.
7 If needed, in the MorphOp part, select the Morphological and/or Clean Up checkboxes and define the settings.

Morphological settings:

- **Erode** reduces or eliminates object features, increases the thickness of holes within an object. This operation replaces each pixel in the image with the maximum value of the pixels and each of its eight vertical and horizontal neighbors.
- **Dilation** reduces or eliminates holes within an object, increases the thickness of an object’s features. This operation replaces each pixel in the image with the minimum value of the pixel and each of its eight vertical and horizontal neighbors.
- **Closing** eliminates holes. Preserves small features. An erosion operation is applied to the image, followed by a dilation operation.
- **Opening** preserves holes. Eliminates small object features. A dilation operation is applied to the image, followed by an erosion operation.

Clean up settings:

- **Prune** is used to ignore, but not remove features, that are below a specified size (connectivity size). When an image is pruned of all features below a certain size, the blob measures returned for the blob that enclosed the pruned features are computed as though the pruned features still existed, but the pruned features themselves are not counted.
- **Fill** is used to fill in pruned features with gray values from neighboring pixels on the left. The pixels value that is used to fill the feature is the value of the pixel to the immediate left of the feature being filled.

Continues on next page
each row of pixels in the feature is filled, the pixel value to the immediate left of that row of pixels is used as the fill value for that row.

- **Connectivity** defines the minimum size (in pixels) that a blob can have to be considered. Is used with either prune or fill.

8 In the **Item region** part, select the **Use Item Region** checkbox and click **Define Region**. Adjust the polygon showed around the found object using vertices. Then click **Train**.

The polygon can have 2 to 16 vertices.

9 Click **Search** in the **Display Options**.

**Tip**

If the search result matches with the **Image Dialog**, the configuration succeeds.

In the **Display Options** part, select **Segmentation image** to display the processed image. Select how the result will be displayed.

- **Item Area** displays the area of the blob in the image window.
- **Boundary Box** displays the minimum horizontal rectangle that contains the whole blob.
- **Item region** displays the regions in the image window. Red regions indicate an overlap and the corresponding hits will be considered as discarded.
- **Blob angle** displays the angle of the item that will be sent to the robot.
- **Score Value** displays the score for the selected item in the image window.

10 Click **OK**.

**Blob parameters in item targets**

The blob parameters **Area**, **perimeter**, **circularity**, and **elongation** can be selected for the target storage.
4 Working with PickMaster PowerPac

4.4.10.3 Configuring blob models

Continued

Related information

Using color vision on page 317.

Item Properties tab on page 127.

RAPID programs on page 367.
4.4.10.4 Configuring inspection models

Introduction to inspection models

Inspection models make it possible to combine several models of PatMax, blob, histogram and Caliper. This is sometimes referred to as Inspection II.

An inspection model always consists of an alignment model. The alignment model can either be a PatMax or blob works as the reference for the inspection model. It is this model’s position and rotation that is the pick/place position and rotation for the item.

Inspection areas are defined relative to the alignment model and either blob, histogram, Caliper or PatMax can be done within each of these areas. Conditions such as number of found items and location relative to the alignment model can be set.

For a found item to be classified as accepted, all inspection areas and the alignment model must be classified as accepted. If one of the inspection areas does not fulfill the given conditions the corresponding item is classified as rejected.

Illustration Inspection Configuration

Configuring inspection models

Use this procedure to configure inspection models.

1. Right-click on one Item in the tree view Process and select Setting. The Item Setting window is opened.
2. Click to select the Item Source tab.
3. In the Item Source dialog, click New model and select Inspection.
4. In the Image part, click Live, Acquire, or Import to get an image.
5. In the Inspection model part, define the relationships between the alignment model and its corresponding inspection areas.

The created models are shown in a tree view.

Continues on next page
Alignment Model defines the position and orientation of any found items. For more information on the alignment model configuration dialog, see Vision modeling on page 286.

Sub Inspection Model adds inspection areas to an alignment model. See Sub inspection models on page 306.

Edit opens the configuration dialog for the selected model. When an existing alignment model is modified the relations to the inspection areas must be retrained.

Delete is used to delete the selected model and corresponding inspection area.

Edit Area shows the current model’s area. The area can be rearranged for the selected sub inspection model.

6 Click Create Alignment Model to open the Select Model Type drop-down list.
7 Select Geometric or Blob in the drop-down list to create the alignment model. For detail procedures on how to create a geometric model or a blob model, see Configuring a geometric model with PatMax on page 289 or Configuring blob models on page 297.
8 Click + Sub Inspection Model to open the Select Model Type drop-down list.
9 Select Geometric, Blob, Histogram or Caliper in the drop-down list to create the sub model.
10 Click OK on the popped-up dialog to edit area.
11 Drag the rectangle so it covers the pattern.
12 Click **Edit** button to open the corresponding model creating window. For detail procedures on how to create a Geometric, Blob, Histogram or Caliper model, see *Configuring a geometric model with PatMax on page 289 Configuring blob models on page 297, Histogram on page 307 and Caliper on page 310.*

**Note**

For Geometric sub model, after **Define** and **Train** the models, another **Train** need to be done.

13 Click **Search**.

The result is displayed as an image with numbered hits in the Image Dialog and a corresponding detailed list in the Search Result window.

**Tip**

If the search result matches with the Image Dialog, the configuration succeeds.

14 Click **OK**.

**Sub inspection models**

**Introduction**

Sub inspection models are used to add inspection areas to an alignment model. Each area uses a specified sub inspection model. The inspection area defines where the sub model is to perform its analysis relative to the alignment model. The areas are shown in the image and should be moved and resized to cover the area to analyze.
Sub inspection models are configured in their own dialogs. When testing a sub inspection model the alignment hit is shown in the image window together with the corresponding inspection area. Sub inspection models only analyze the part of the image defined by its inspection area.

Geometric

A geometric sub inspection model is configured in the same way as a PatMax model. See Configuring a geometric model with PatMax on page 289. In addition, the relative positions of the found items and the corresponding alignment hit must be trained.

**Required hits** defines the number of hits with the sub inspection model within the inspection area that are required for the result to be considered as accepted.

**Deviation limits** defines the allowed deviations from the trained positions.

After a search and the items are found within the inspection area their positions must be trained. The relative positions are listed as $x_{\text{Diff}}$, $y_{\text{Diff}}$, and $\text{Angle}_{\text{Diff}}$. Click **Train** to save the positions of the found items relative to the alignment hit.

Geometric subinspection parameters in item targets

The parameter **Number of hits** can be selected for the target storage.

Blob

A blob sub inspection model is configured in the same way as a blob model. See Configuring blob models on page 297. In addition, the number of required hits must be configured.

**Required hits** defines the number of hits with the sub inspection model within the inspection area that are required for the result to be considered as accepted.

Blob subinspection parameters in item targets

The parameter **Number of hits** can be selected for the target storage.

Histogram

The histogram tool measures the color or the gray level within any given area. While using a monochrome camera the histogram tool measures the gray level within a given area. Similarly, if a color camera is used each of the three color channels (Red, Green, and Blue) is measured separately. The histogram tool is useful when the objects to be identified and classified have similar shapes but different colors.

The inspection area for a histogram sub inspection model is graphically represented as a circle. But the area used in the histogram analysis is actually a square aligned with the image but enclosed by the inspection area.

1. Click **+ Sub Inspection Model** to open the **Select Model Type** drop-down list.
2. Select **Histogram** in the drop-down list to create the sub model.
3. Click **OK** on the popped-up dialog to edit area.
4 Drag the circle so it covers the pattern.

5 Click Edit icon under Action to open the histogram model editing window.

6 Press Auto Settings to automatically get an appropriate range limits (Min. and Max. values) for the histogram. Alternatively, the Min. and Max. values can be set manually by sliding the red and green bars across the histogram or by simply entering values into the text boxes. For a product to be accepted, both the standard deviation and the mean value have to be within the specified limits. When using color vision the histograms for all channels must fall within the limits.
7 If change to Tab RGB or HSI, the window for the colors will show up.

8 Click OK.

To classify the inspection area as accepted or rejected the histogram tool evaluates two different magnitudes within the specified region:

- **Mean** defines the min and max value for the inspection model. If the inspection area has a mean value less than min or higher than max the inspection area will be classified as rejected.

- **Std dev** is a statistical measure that illustrates how closely all the various pixel values are clustered around the mean value. An even color tone gives a narrow histogram with low standard deviation while a speckled pattern gives a wide histogram and a high value for **Std dev**.

Histogram subinspection parameters in item targets

- The **Mean and standard deviation** parameters can be selected for the target storage.

Continues on next page
The **Caliper** tool identifies edges and measures the distance between them. The analysis is only done within the corresponding inspection area. To increase the contrast in images where parts have similar grayscale tone, it is possible to use color filtering. For more information, see *Using color vision on page 317*

1. Click **+ Sub Inspection Model** to open the **Select Model Type** drop-down list.
2. Select **Caliper** in the drop-down list to create the sub model.
3. Click **OK** on the popped-up dialog to edit area.
4. Drag the rectangle so it covers the pattern.
5. Click **Edit** icon under **Action** to open the Caliper model editing window.
6. Move the line so the end points are located on the edges of the area under the **Image settings**.

7. Adjust the parameters in the **Search parameter** according to the **Defined distance** in the **Analyze area**.
8 Click Search.

The result is displayed in the Search Result tab.

9 Click OK.

To make a Caliper analysis a rectangle is defined around the search line. Defined distance is the distance between the end points of the green line located in the Image Dialog. Move the line so the end points are located on the edges of the area.

Analyze area length is the length of the rectangle within which the Caliper analysis will be performed. To increase the Analyze area length either increase the Delta length value or resize the Defined distance line.

Analyze area width is the width of the rectangle within which the Caliper analysis will be performed. To increase the Analyze area width increase the Delta width value.

Delta length define the extra mm to add to the Defined distance to get an Analyze area length.

\[
\text{Analyze area length} = 2 \times \text{Delta length} + \text{Defined distance}
\]
Delta width defines the width of the analyze area.

\[ \text{Analyze area width} = 2 \times \text{Delta width} \]

From the analyze area a production image is created. The operation sums all the information in the analyze area, accentuating the strength of edges that lie parallel to the Analyze area width and reducing the effects of noise.

Edge property defines the polarity of the edge. The polarity is defined as the measure from Edge1 to Edge2.

The Search parameter defines filters using a Gaussian curve. The filter controls how the Caliper tool removes noises, how it accentuates the peaks of interest in the image, contrast, and distance.

The Search is used to search for two edges with the specified distance (Defined distance) and the defined polarity.

The checkboxes in the Search parameter define which results should be displayed in the Image Dialog.

Caliper subinspection parameters in term targets

The Distance parameter can be selected for the target storage.

External model

This function is reserved for next version.
4.4.11 Starting production

Production

After switching to the real controller and real Runtime, all operations in the production are reflected in the real cell, and all data comes from the real system.

Select one recipe from the tree view and click Control on the ribbon to open the control dialog box in the solution.

The following table provides details about the Control dialog box.

<table>
<thead>
<tr>
<th>Description</th>
<th>Control the status of the production and have an overview of the production data. For more information regarding Statics see Recipe on page 161.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipes</td>
<td>Adjust the parameters of the item, work area and robot. For more information regarding Tuning see Tuning on page 161.</td>
</tr>
<tr>
<td>Tuning</td>
<td>Adjust the speed of the conveyor. For more information regarding Flow Control see Flow Control on page 167.</td>
</tr>
<tr>
<td>Flow Control</td>
<td>See the live video of the camera. For more information regarding Vision see Vision on page 313.</td>
</tr>
<tr>
<td>Vision</td>
<td></td>
</tr>
</tbody>
</table>

Vision

For more information, see Detailed vision information on page 283.

Emulation

When running the production, the movement of 3D models in PickMaster PowerPac is called as emulation.

Use this procedure to do the emulation:

1. On the PickMaster PowerPac ribbon-tab, click Operation.
2. On the Operation ribbon-tab, click Control.
   The Control dialog is opened.

Continues on next page
3 Click **Start** to run the production.
   The emulation starts running.

4 Click **Stop** to stop the emulation.

---

**Note**

When running the production, the movement of 3D models in PickMaster PowerPac follows the actual system. However, since the 3D models dimension in PickMaster PowerPac cannot be completely consistent with the real cell. The layout of conveyor, camera, I/O sensor and robot in the emulation may need to be adjusted according to the actual dimension to make the emulation as close to the actual system as possible.

If the item is missing during the emulation, it may be caused by that the size of the PickMaster PowerPac station is not exactly the same with the real station. The item is hidden in the conveyor model. Adjust the height of the conveyor model to show the item normally.
4.4.12 Managing the robot in production

Starting production
Start and stop the production from the Control menu. During production, the robots are accessed from the Control tab in the Workspace area. For more details, see Production on page 313.

Prerequisites
The solution must be configured to start production. The recipe must be open and active.

Pick rate
The pick rate is shown as icons in the Production tab when a robot is running. The following values are shown:

- Number of pick during the last minute.
- Total number of picks since the production was started.

Robot states
The robot can be in different states.

<table>
<thead>
<tr>
<th>State</th>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>Green</td>
<td>The robot can pick and place items.</td>
</tr>
<tr>
<td>Paused</td>
<td>Red</td>
<td>The robot is paused in motors off state, or the RAPID program has stopped.</td>
</tr>
<tr>
<td>Emergency State</td>
<td>Red</td>
<td>The robot is in emergency stop state.</td>
</tr>
<tr>
<td>Stopped</td>
<td>Red</td>
<td>The robot is stopped, that is no items are handled by the robot or distributed to the robot.</td>
</tr>
</tbody>
</table>

Stopping and resuming the robot
It is possible to stop a robot during runtime. Click a robot icon in the Production tab and select action from the popup menu. If more than one robot is connected to a controller (MultiMove):

- Restart from stopped state must be performed at the same time for all robots. To do this, right-click the controller icon in the production tab and select Restart Robots.
- Stopping one robot will also stop the other robots on the same robot controller.

Emergency stop
In case of emergency:

1. Press the emergency stop button on the robot controller or the FlexPendant to stop the robot immediately. This sets the controller in emergency state and a warning is displayed on the FlexPendant and in PickMaster PowerPac and Runtime.
2. Fix the problem.

Continues on next page
3 Release the emergency button.
4 Then acknowledge and reset the emergency state on the FlexPendant or using the popup menu before you restart the robot.

⚠️ CAUTION

Emergency stop should not be used for normal program stops as this causes extra, unnecessary wear on the robot.
4.5 Using color vision

Introduction to color vision

PickMaster PowerPac can either be used with monochrome or color cameras. The difference between the two is that an image acquired with a color camera represents each pixel with three 8-bit values (decimal 0-255) instead of only one 8-bit value for monochrome (grayscale) images. In a monochrome image the 8-bit value represents the gray level from white to black, whereas in a color image the three values represent the content of three separate color channels. These three channels represent red, green, and blue (color space RGB) or hue, saturation, and intensity (color space HSI). Which color space to work with, depends on the content of the image.

Color spaces

When working with RGB the color of each pixel is represented by its content of red, green, and blue. The numerical representation is straightforward for the three base colors - red (255, 0, 0) green (0, 255, 0), and blue (0, 0, 255). However, it can be difficult to understand the composition of other mixed colors.

HSI is a color space that is more easily translated to the human perception of colors.

- Hue: The location of the color on the electromagnetic spectrum. See graphic below.
- Saturation: The purity of the color.
- Intensity: The brightness of the color.

Because the hue spectrum wraps around (both 0 and 255 represent red), it is suitable to display it as a circle.

When using color filtering it is easier to distinguish between colors if they are dissimilar. The level of similarity may be interpreted as the distance between the colors in color space. The difference may be more pronounced in one or the other of the two color spaces and for this reason it is wise to try out filters in both color spaces.

Continues on next page
4 Working with PickMaster PowerPac

4.5 Using color vision

Continued

Lighting

Because a color system provides more information about the color contents of an image it is also more sensitive to lighting conditions. It is very important to provide uniform light, that is consistent over time.

Computer performance

Color vision is very resource consuming: acquisition, warping, and filtering all take more time. It is important to keep the number of cameras and frame rate moderate. The performance limit can vary greatly as it is a combination of the vision task and the computer resources.

Color vision in PickMaster PowerPac

PickMaster PowerPac provides color vision in the form of a filter. This filter is accessible from the PatMax, Blob and Caliper configuration dialogs, both as standalone, alignment and sub-inspection models. The filter is a pre-processing step which takes place before the object recognition or measurement. Every model can have its own individual filter setting.

The camera acquires a color image, that is converted into a grayscale image by passing it through a color filter, as shown in the following figure.

<table>
<thead>
<tr>
<th>A</th>
<th>Vision model</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Color image</td>
</tr>
<tr>
<td>C</td>
<td>Color filter</td>
</tr>
<tr>
<td>D</td>
<td>grayscale image</td>
</tr>
<tr>
<td>E</td>
<td>Object recognition</td>
</tr>
</tbody>
</table>
The result of the color filter is a grayscale image in which certain colors have been accentuated or attenuated according to the filter settings. The object recognition tools (Blob/PatMax) operate on this grayscale image.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>An image acquired with a color camera.</td>
</tr>
<tr>
<td>B</td>
<td>The same scene acquired with a monochrome camera.</td>
</tr>
<tr>
<td>C</td>
<td>The color image after having passed through a filter which is set to extract green. This is the image that will be used by PatMax/Blob.</td>
</tr>
</tbody>
</table>

**Prerequisites**

- The camera must be a color camera.
- The color video format must be configured for the camera.
- The Cognex vision license must contain the color tool option.

**Calibrating the camera’s white balance**

A camera is delivered with default settings. These include three parameters which represent the white balance of the camera. Depending on the light source, the image can get an undesired color tone. Different light sources emit light of different temperatures (color content) and the camera needs to be color calibrated in order to compensate for this light.

The basic concept is to present the camera with a gray scene, that is a scene that has equal contents of red, green, and blue. The most accurate method is to take a sheet of white paper and adjust the light settings of the camera in order to make the scene appear gray.

Use this procedure to calibrate the white balance for the camera.

1. In the tree view, right-click on the camera and select Configuration. The Camera Configuration dialog is opened.
2. Place a white sheet of paper under the camera. The sheet must cover the whole field of view.
3. Adjust the light settings (aperture or exposure time) to make the scene appear mid-gray. The number of saturated pixels (completely black or white) should be kept to a minimum.
4. Press Calculate. This will calculate the white balance calibration parameters.
5 Click **Apply**.

The camera’s internal settings are now modified. If the calibration is successful the color image and the grayscale image of the white paper sheet should now look the same (gray).

6 Click **OK**.

The settings are stored in the camera. If the parameters are not saved, the camera will lose the calibration when PickMaster PowerPac is restarted.

---

**Illustration Color Filter Settings**

The *PatMax* and *Blob* configuration dialogs contain a checkbox to enable color filtering (Color filter), and a tab page to display the filter settings.

---

**Continues on next page**
Use this procedure to configure color vision.

1. In the **PatMax** or **Blob** configuration dialog, select **Color Filter**. This will enable the filter.

   ![Color Filter Configuration Dialog](image)

   The **Color Filter Settings** tab is opened together with a second video window showing the color image.

2. In the **Color Filter** tab, select **RGB** or **HSI**.

3. In the **Define color** tab, color samples can be collected from the display to indicate which colors should be enhanced.
   
   a. Click **Define**. An adjustable rectangle will appear in the color dialog.
   
   b. Move/resize the rectangle to indicate what color should pass through the filter. The indicated color range will be converted to white in the
output grayscale image. Colors that are dissimilar to the specified color will be converted to black.

c Click Get color to store this color range.
4 In the **Manual color filter** tab, adjust each color channel to improve the result if needed.

- **Low** specifies the lower limit of the color range that will translate into white pixels in the output image. Minimum is 0 and maximum is 255, except for Hue which has no boundary.
- **High** specifies the upper limit of the color range that will translate into white pixels in the output image. Minimum is 0 and maximum is 255 except for Hue which has no boundary.
- **Fuzzy** specifies how colors outside the minimum and maximum thresholds should be filtered to the output grayscale image. A value of 0 indicates that colors outside the range specified by Low and High will be completely removed by the filter - the result is a black and white image. A non-zero value means that colors outside the Low/High range will be weighted in the output image. A higher value produces a smoother grayscale image. Minimum is 0, maximum is 255.

5 If needed, add a new color range to the list in the **Colors** section.

Each pixel of the output image is computed as the corresponding maximum output pixel of all individual color range filters.

6 If needed, adjust the smoothing factor to reduce noise in the resulting grayscale image.

7 Proceed to define the object recognition model.
4 Working with PickMaster PowerPac

4.5 Using color vision

Continued

Tip

Filter ranges should be narrow to provide an output image with high contrast. From an image quality perspective, it is often better to select small homogeneously colored samples and add several ranges to the list of colors.

Tip

Try to filter with both RGB and HSI. Sometimes one may work significantly better than the other.

Example 1

This example describes how to locate a part with PatMax and inspect the color with Blob.

1 Create an inspection model, see Configuring inspection models on page 304.
2 Create a PatMax alignment model. Use color filtering if contrast needs to be increased, or use the unfiltered monochrome image if there is sufficient contrast.
3 Add a Blob sub inspection model.
   a Select Color filter checkbox. This opens the Color Filter Settings tab.
   b Extract the color to be inspected by clicking Define color. This filters the desired color into white in the Blob image window.
   c Switch to other tab to do further configuration.
   d Adjust the Blob settings so as to find the white blob.
   e If necessary, adjust the settings of the color filter and the Blob analysis.
4 Test the result in the Inspection Configuration dialog.

Example 2

Color image before filter

Continues on next page
4 Working with PickMaster PowerPac

4.5 Using color vision

Continued
4 Working with PickMaster PowerPac

4.6 Working with products of varying height (2.5D vision)

Introduction to height settings

The vision tools in PickMaster PowerPac typically return a result in a 2D coordinate system: X and Y angle, based on a calculation made at a certain height. The trained model is assumed to be located in the plane of the camera calibration.

Working with objects located above or below the calibrated plane will result in parallax position problems.

Assuming a calculation on a defined height, any object of a different height will be shifted by the resulting parallax.

The camera is taught to calculate based on the blue block top surface. Without the parallel, the camera will employ this surface for the lower green block or higher orange block. As a result, x- and y-coordinates for the green block and the orange block would move based on the misused calculation plane. That would mislead the robot to target a wrong position.

With 2.5D calibration, a full 3-dimensional space is calibrated, which allows the system to compensate for the parallax error, given that the system knows the correct height of the object. This raises the following questions:

1. At what height is the object located (z-coordinate)?
2. What are the true x- and y-coordinates? The object recognition tools assumes that the object is still located in the calibration plane, and thus will provide coordinates projected on this plane.

To calculate the true x- and y-coordinates the camera’s height above the calibration plane, and the product’s distance (above/below) to the calibration plane must be
known, based on the camera location, provided by performing a multi-view calibration. See *Calibrating camera on page 276.*

Determining the height at which an object is located can be done in three ways with PickMaster PowerPac.

1. Manual input
2. Automatic calculation based on the scale change in relation to the trained object.
3. External input

All three methods will return the parallax compensated x- and y-coordinates, and method 2 and 3 will also return an estimated z-coordinate.

Effectively, the tools described in this section can be used to compensate for parallax error (find the true x- and y-coordinates) and for determining the height of a product.

**Prerequisites**

The camera must be calibrated with multiple images (Multi-view). The height settings can be used together with a geometric standalone model, or main geometric-based inspection model.

**Configuring height settings**

The height settings belong to a specific model and can only be configured together with Geometric.

Use this procedure to configure the height settings.

1. Create a Geometric model.
2. In the Image part, select calibration from the Calibration list. This must be a multi-view calibration.
3. In the Model definition part, click Advanced. This opens the Geometric advanced model settings dialog.

Continues on next page
4. Choose an appropriate calculation method before training the item.

5. If the calculation method is set as Item Height,
   
   Item height: Manually enter the value for the picking/placing height.

   One parameter should be fulfilled for this calculation method. Model Height is literally used to describe the height from the calibration panel to the
calculation plane. Z-coordinate is defined as the true picking/placing height for the object and would be sent to the robot controller.

- Measure the height (H0) of the calibration board.
- Measure the height (H1) from the conveyor to the maximum contour panel for the picking/placing object.
- Measure the height (H2) from the conveyor to the picking/placing panel for the picking/placing object.
- Enter the value (H1-H0) to the Model Height.
- Enter the value (H2-H0) to z-coordinate of the item setting $\text{Size}(x,y,z)\text{[mm]}$/RH Size[mm]. See Item on page 127.

6 If the calculation method is set as Vision Height, Vision height: The value from the calibration panel to the calculation plane is calculated from the scale change (relative to the trained pattern) of the found object.

Two parameters should be fulfilled for this calculation method. Model Height follows the same meaning defined in the Item Height. Pick Offset is the deviation from the calculation plane to the picking/placing panel. The
calculation plane is defined as the maximum contour panel of the identified object.

**Note**

Enable uniform scale must be enabled. The maximum and minimum values must allow for sufficient scale variation.

PickMaster PowerPac would consider the calculate and compensation the z-coordinate for the robot controller to target the final picking/placing height (H3).

a Measure the height (H0) of the calibration board.

b Measure the height (H1) from the conveyor to the maximum contour panel for the picking/placing object.

c Measure the height (H3) from the maximum contour panel to the picking/placing panel of the picking/placing object.

d Enter the value (H1-H0) to the *Model Height*.

**Tip**

If this value is a positive number, that means the calculation panel is higher than the calibration panel on the z-direction.

If this value is a negative number, that means the calibration panel is higher than the calculation panel on the z-direction.

e Enter the value (H3) to the *Pick Offset*.

f Enable the *Enable Uniform Scale* and enter a proper range for the scaling.

7 If the calculation method is set as *External height*,

Continues on next page
External height: The product’s distance (above/below) to the calculation plane is calculated by the external source. This may be a height sensor or information from a cell PLC or any other external device. The z-coordinate is sent through a UDP port to PickMaster Runtime.

One parameter should be fulfilled for this calculation method.

a Enter the UDP port in UDP Listening Port. Then the calculated z-coordinate will be sent to PickMaster Runtime with the UDP message through this port.

With the height setting configured during the model training, the search results will contain the space information for all searched objects.

Note

The Vision height method may be inaccurate. The accuracy depends on many factors such as camera the camera calibration, camera resolution, model size relative to image etc, thus the obtainable accuracy must be tested for a specific application.

Note

Defining a value for Model height, and selecting Item height as height method results in parallax compensation but no z-coordinate is calculated by the vision system.
4 Working with PickMaster PowerPac

4.6 Working with products of varying height (2.5D vision)

Continued

**Note**

If there is only one object type, and it is always located at the same height, it is most accurate to calibrate the camera at this height instead of using **Model height** to compensate.

**Tip**

To filter out erroneous height information when using the **Vision height** method, set appropriate scale limits under the **Post search filters** part in the **Geometric model dialog**.

**Related information**

*Calibrating camera on page 276.*

*Item on page 127.*
4.7 Production with flow (Ghost Picking)

Overview

Ghost picking flow is used by the application engineer to run the dry cycle function before the production. The user sees the robot picking up empty objects on the real workstation. This feature differs from the production in that its incoming material is virtual and is provided by the flow generated by the previous record.

Creating a ghost picking flow

Use this procedure to create a ghost picking flow:

1. Open the solution need to do the ghost picking.
2. Right click on the recipe you need in the tree view Process and select Setting. The Recipe setting window is opened.
3. Click the Conveyor WA in the Available Workareas which need to be recorded to open the work area setting window.
4. Select Record Scenes checkbox in the Record Setting.

Note

When Record scenes is selected and saved for any work area, the following message will pop up.
Scenes recording is activated for: {0}
After this, the recording will be activated automatically when the simulation or production is started.

5. Set the record time according to your requirements.

Continues on next page
Adding a ghost picking flow to a solution

Use this procedure to add a ghost picking flow:

2. On the ribbon-tab, click Flow.
   The Flow window is opened.
3. Click Recorded tab.
4. Select the type for the flow in Flow Type.

   **Note**
   The type selected here MUST be same with the imported flow type. Otherwise the flow cannot work normally

5. Click the Import Flow icon to import the predefined work area .xml file.
6. Click Open to apply the configuration.
7. Click Item/ContainerPattern drop-down list to select the desired item or container.
8. Click OK to save and close the Flow dialog.
Modifying position generator
Use this procedure to modify the position generator for the ghost picking flow:
1 Right-click on one **Position Generator** in the tree view **Layout** and select **Setting**.
The **Position Generator** window is opened.
2 Click on the conveyor which has added the recorded flow.
3 Select the **Vision** in the **Source Type**.
4 Select the used camera for this conveyor in the drop-down list.
5 Select **Distance** in the **Trigger Setting**.
6 Click **OK** to close the **Position Generator** window.

Selecting ghost picking flow (Modify recipe)
Use this procedure to select a ghost picking flow:
1 Right-click on one **Recipe** in the tree view **Process** and select **Setting**.
The **Recipe setting** window is opened.
2 Click on the **Operation_1** to open the setting window for the operation.
3 In the **Select Flow**, select **Flow1** in the drop-down list.

**Tip**
If the source type of conveyor is set as Predefined or no camera is added to the conveyor, the flow cannot be selected for this conveyor.

4 Add the **Conveyor WA 1** to the **Accept** by dragging under the **Distribution Setting** tab.
5 Click **OK** to close the **Recipe setting** window.

Ghost picking flow
Use this procedure to run a ghost picking flow:
1 On the PickMaster PowerPac ribbon-tab, click **Operation**.
2 Select the recipe which will be running in the tree view and click **Control** on the ribbon tab **Operation**.
The **Control** dialog is opened.
3 Click **Start** to run the production.

**Note**
The ghost picking is default set as looped. It will repeat sending the recorded position data to the real controller until the **Stop** icon is clicked.

**Note**
The detailed vision is not applicable when running ghost picking flow.
The emulation starts running.
4.7 Production with flow (Ghost Picking)

Continued

4 Click Stop to stop the ghost picking in the production.
5 RAPID reference

5.1 Instructions

5.1.1 AckItmTgt - Acknowledge an item target

Usage

AckItmTgt is used to acknowledge that an itmtgt received with GetItmTgt from an item source has been used (for example, handled by the robot, skipped or put back in the queue for later usage). Normally, acknowledge is setup as a TriggL event on the path (using the Ack or Nack trigndata from sourcedata) to make sure acknowledge does not occur before any movements related to the target has been finished. However, if the received itmtgt shall be skipped or put back in the queue for later usage, movements related to the target may not be needed. Then it is convenient to use this instruction instead. Only after the acknowledge has been made, a new itmtgt can be fetched from the item source.

Basic example

VAR itmtgt PlaceTarget;
GetItmTgt ItmSrcData{Index}.ItemSource, PlaceItem;
AckItmTgt ItmSrcData{Index}.ItemSource, PlaceItem, FALSE
\Skip:=TRUE;

Arguments

AckItmTgt ItemSource ItemTarget Acknowledge [\Skip] [\Type]

ItemSource

Data type: itmsrc
The item source from where the item target has been received with GetItmTgt.

ItemTarget

Data type: itmtgt
The item target to acknowledge.

Acknowledge

Data type: bool
The status of acknowledge. TRUE if the itmtgt has been handled (picked or placed) by the robot and FALSE otherwise, in which case the itmtgt is put back into the queue.

Skip

Data type: bool
Indicates if the itmtgt shall be skipped. If set to TRUE it will not be possible to receive the itmtgt again with GetItmTgt. If combined with Acknowledge = FALSE the itmtgt will be passed on for possible handling by downstream robots. If combined with Acknowledge = TRUE, skip will have no effect. If Skip is set to FALSE the itmtgt will either be considered as handled by the robot (when

Continues on next page
5 RAPID reference

5.1.1 AckItmTgt - Acknowledge an item target

Continued

combined with Acknowledge = TRUE), or put back in the queue for later usage (when combined with Acknowledge = FALSE).

Type

Data type: num

Modifies the type of the itmtgt. If combined with Acknowledge = FALSE and Skip = TRUE, the item will be passed on to downstream robots according to the configured distribution of the new item type.

If combined with Acknowledge = FALSE and Skip = FALSE, the item will be put back in the queue with the new item type and can still be received with GetItmTgt. The item type will only be changed locally; the item type and the distribution of the item will not change for downstream robots.

If combined with Acknowledge = TRUE, type change will have no effect.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable ERRNO will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined.</td>
</tr>
</tbody>
</table>

Limitations

The itmtgt must be received with the instruction GetItmTgt.

Syntax

AckItmTgt

[ItemSource ':=' ] <variable (VAR) of itmsrc>,
[ItemTarget ':=' ] <var or pers (INOUT) of itmtgt>,
[Acknowledge':=' ] <expression (IN) of bool>,
[\Skip ':=' ] <expression (IN) of bool>,
[\Type ':=' ] <expression (IN) of num>;

Related information

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data type itmtgt</td>
<td>itmtgt - Item target data on page 360.</td>
</tr>
</tbody>
</table>
5.1.2 FlushItmSrc - Flush an item source

Usage

FlushItmSrc is used to flush an item source. The instruction clears the item source buffers, sets the scene number to one and flushes the encoder board.

Basic example

FlushItmSrc PlaceSource;

Flushes the earlier created item source object PlaceSource.

Arguments

FlushItmSrc ItemSource

ItemSource

Data type: itmsrc

The created item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable ERRNO will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined</td>
</tr>
</tbody>
</table>

Limitations

To avoid potential problems, this instruction should be executed only when the last item target definitely has been acknowledged.

Syntax

FlushItmSrc

[ItemSource ':='] <variable (VAR) of itmsrc>;
5 RAPID reference

5.1.3 GetItmTgt - Get the next item target

Usage

GetItmTgt is used to get the next available itmtgt in the item source queue between the enter and the exit limit of the work area. The RAPID program waits in this instruction until the next item is possible to reach or the timeout occurs.

Basic examples

Basic examples of the instruction GetItmTgt are illustrated below.

Example 1

GetItmTgt PlaceSource, PlaceItem;

Receives a place item from the PlaceSource when there is one that can be used.

Example 2

... 

VAR selectiondata neg_y_sort;

neg_y_sort.ShapeType:=BOX;
neg_y_sort.ConsiderType:=BitOr(ITEMS_TO_USE,ITEMS_BYPASS);
neg_y_sort.GeometricData.x:=60;
neg_y_sort.GeometricData.y:=500;
neg_y_sort.GeometricData.z:=10;
neg_y_sort.GeometricData.radius:=0;
neg_y_sort.Offset.OffsetRelation:=FRAME_COORD_DIR;
neg_y_sort.Offset.OffsetPose.trans.x:=0;
neg_y_sort.Offset.OffsetPose.trans.z:=0;
neg_y_sort.Offset.OffsetPose.rot.q1:=1;
neg_y_sort.Offset.OffsetPose.rot.q2:=0;
neg_y_sort.Offset.OffsetPose.rot.q3:=0;
neg_y_sort.Offset.OffsetPose.rot.q4:=0;
IF pick_type = 2 THEN pick_type := 1; ELSE
  pick_type := 2
ENDIF

GetItmTgt PickSource, PickItem \ItemType:=pick_type \Limit:=100 \Selection:=neg_y_sort;

Retrieves a pick item from the PickSource with negative y-sorting and type request. The type is alternating between two types. The Limit argument tells from where to start the search.

In the example graphic below, the sorting is in positive x-direction, negative y-direction, and operating on two different object types. The two object types should
be chosen in an alternating pattern starting with the circular. This will give the order as numbered 1-10 in the graphic.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Enter</td>
</tr>
<tr>
<td>B</td>
<td>Check limit</td>
</tr>
<tr>
<td>C</td>
<td>Exit</td>
</tr>
<tr>
<td>D</td>
<td>Product flow direction</td>
</tr>
<tr>
<td>E</td>
<td>Sort direction</td>
</tr>
<tr>
<td>1-10</td>
<td>Sort order</td>
</tr>
</tbody>
</table>

**Arguments**


**ItemSource**

*Data type: itmsrc*

The item source from which the item target should be received.

**ItemTarget**

*Data type: itmtgt*

The received item target.

\[\MaxTime\]

*Data type: num*

The maximum waiting time permitted, expressed in seconds. If this time runs out before the item target is retrieved and no TimeOut flag is given, the error handler

Continues on next page
will be called with the error code \texttt{ERR_PPA_TIMEOUT}. If there is no error handler, the execution will be stopped.

[\textbf{\texttt{\textbackslash TimeFlag}}]

\textbf{Data type:} \texttt{bool}

The output parameter that contains the value \texttt{TRUE} if the maximum permitted waiting time runs out before an item target is received. If this parameter is included in the instruction, it is not considered to be an error if the max time runs out. This argument is ignored if the \texttt{MaxTime} argument is not included in the instruction.

[\textbf{\texttt{\textbackslash ItemType}}]

\textbf{Data type:} \texttt{num}

Specifies which item type number is requested. The instruction waits until an item target with the requested type number is available to be executed.

[\textbf{\texttt{\textbackslash Limit}}]

\textbf{Data type:} \texttt{num}

Modifies the distance from where the item target is received. The instruction will return the next item target above this limit. If this argument is excluded, the instruction will return the next item target above the exit limit.

The distance is specified in millimeters from the center of the robot. The value is positive if the limit is beyond the center of the robot, in the moving direction of the feeder. This argument is only valid when a conveyor is used.

[\textbf{\texttt{\textbackslash SortData}}]

\textbf{Data type:} \texttt{sortdata}

This data structure defines how the items shall be sorted.

[\textbf{\texttt{\textbackslash Selection}}]

\textbf{Data type:} \texttt{selectiondata}

This data structure defines how the items are selected.

[\textbf{\texttt{\textbackslash Val1Min}}]

\textbf{Data type:} \texttt{num}

Specifies minimum value for \texttt{itmtgt} parameter Val1. The instruction waits until an item target fulfilling this condition is available for execution.

[\textbf{\texttt{\textbackslash Val1Max}}]

\textbf{Data type:} \texttt{num}

Specifies maximum value for \texttt{itmtgt} parameter Val1. The instruction waits until an item target fulfilling this condition is available for execution.

[\textbf{\texttt{\textbackslash Val2Min}}]

\textbf{Data type:} \texttt{num}

Specifies minimum value for \texttt{itmtgt} parameter Val2. The instruction waits until an item target fulfilling this condition is available for execution.

[\textbf{\texttt{\textbackslash Val2Max}}]

\textbf{Data type:} \texttt{num}

Continues on next page
Specifies maximum value for `itmtgt` parameter `Val2`. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val3Min]`

**Data type:** `num`

Specifies minimum value for `itmtgt` parameter `Val3`. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val3Max]`

**Data type:** `num`

Specifies maximum value for `itmtgt` parameter `Val3`. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val4Min]`

**Data type:** `num`

Specifies minimum value for `itmtgt` parameter `Val4`. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val4Max]`

**Data type:** `num`

Specifies maximum value for `itmtgt` parameter `Val4`. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val5Min]`

**Data type:** `num`

Specifies minimum value for `itmtgt` parameter `Val5`. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val5Max]`

**Data type:** `num`

Specifies maximum value for `itmtgt` parameter `Val5`. The instruction waits until an item target fulfilling this condition is available for execution.

**Program execution**

If there is no item target in buffer or any item targets available in the working area, the program execution waits in this instruction until an item is considered as inside the working area.

If the `MaxTime` argument is specified then the wait time is supervised. If the waiting time exceeds the value of `MaxTime` and the `TimeFlag` argument is used, then the program will continue. If `TimeFlag` is not used, then an error is raised. If `TimeFlag` is specified, it will be set to `TRUE` if the time is exceeded, otherwise it will be set to `FALSE`.

The `Limit` argument modifies the limit from where the item target shall be received. If the `SortData` argument is specified the instruction will return the item target that is the closest to the exit limit in x-direction and depending of the absence of other objects in direction of the sorting, the first object in the sort direction will be selected. The `CheckBoundary` distance defines the required clearance distance.

Continued on next page
around an object. The sorting will check both upwards and downwards the
production flow for presence of other item targets. If this argument is combined
with the Limit argument the sorting algorithm will also take all objects between
the limit and the exit limit into consideration when checking the safety distance for
the nearest objects. If more than one robot is used in a shared position source
system, that is load balancing or ATC, we strongly recommend using the Selection
argument instead with a proper selection data, as SortData does not take items
that are bypassing in consideration when sorting.

If the Selection argument is specified, the instruction will return the item target
that is the closest to the exit limit in x-direction, which has no other item targets
inside the specified shape. If this argument is combined with the Limit argument
the selection algorithm will also take all objects between the limit and the exit limit
into consideration when checking the distance for the nearest objects. This is
highly recommended to avoid collisions.

If values are specified for the optional arguments ValXmin or ValXmax, the
instruction will return an item target that fulfills the required maximum and minimum
values for ValX.

Error handling

The following recoverable errors can be generated. The errors can be handled in
an error handler. The system variable ERRNO will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined.</td>
</tr>
<tr>
<td>ERR_PPA_TIMEOUT</td>
<td>Timeout without any error flag.</td>
</tr>
</tbody>
</table>

Syntax

GetItmTgt

[ItemSource ':=' ] <variable (VAR) of itmsrc>,
[ItemTarget ':=' ] <var or pers (INOUT) of itmtgt>
[\MaxTime ':=' ] <expression (IN) of num>
[\TimeFlag ':=' ] <var or pers (INOUT) of bool>
[\ItemType ':=' ] <expression (IN) of num>
[\Limit ':=' ] <expression (IN) of num>
[\SortData ':=' ] <expression (IN) of sortdata>
[\Selection ':=' ] <expression (IN) of selectiondata>
[\Val1Min ':=' ] <expression (IN) of num>
[\Val1Max ':=' ] <expression (IN) of num>
[\Val2Min ':=' ] <expression (IN) of num>
[\Val2Max ':=' ] <expression (IN) of num>
[\Val3Min ':=' ] <expression (IN) of num>
[\Val3Max ':=' ] <expression (IN) of num>
[\Val4Min ':=' ] <expression (IN) of num>
[\Val4Max ':=' ] <expression (IN) of num>
[\Val5Min ':=' ] <expression (IN) of num>
[\Val5Max ':=' ] <expression (IN) of num>
### Related information

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data type <code>itmtgt</code></td>
<td><code>itmtgt - Item target data on page 360.</code></td>
</tr>
<tr>
<td>The data type <code>selectiondata</code></td>
<td><code>selectiondata - Selection data on page 363.</code></td>
</tr>
<tr>
<td>The data type <code>sortdata</code></td>
<td><code>sortdata - Sort data on page 366.</code></td>
</tr>
</tbody>
</table>
5.1.4 NextItmTgtType - Get the type of the next item target

**Usage**

NextItmTgtType is used to get the type of the next item target (itmtgt) in the item source buffer. If the Limit distance parameter is given, the instruction will return the type of the next item target above the limit. The RAPID program waits in this instruction until there is an item in this queue.

**Basic examples**

NextItmTgtType PlaceSource, PlaceType

Retrieves the type of the next itmtgt in the PlaceSource.

**Arguments**

NextItmTgtType ItemSource ItemType [\Limit] [\MaxTime] [\TimeFlag]

*ItemSource*

Data type: itmsrc

The item source that the item target type should be retrieved from.

*ItemType*

Data type: num

The retrieved item target type.

[\Limit]

Data type: num

This is the limit from where the type is retrieved. The instruction will return the type of the next item target above this limit. If this argument is excluded, the instruction will return the type of the next item target above the exit limit.

The distance is calculated in millimeters from the center of the robot. The value is positive if the limit is beyond the center of the robot, in the moving direction of the conveyor.

This argument is only valid when a conveyor is used.

[\MaxTime]

Data type: num

The maximum waiting time permitted, expressed in seconds. If this time runs out before the item target is retrieved and no TimeOut flag is given, the error handler will be called with the error code ERR_PPA_TIMEOUT. If there is no error handler, the execution is stopped.

[\TimeFlag]

Data type: bool

The output parameter that contains the value TRUE if the maximum permitted waiting time runs out before an item target is retrieved. If this parameter is included in the instruction it is not considered to be an error if the max time runs out.

This argument is only used if the MaxTime argument is used.

Continues on next page
### Program execution

If there is no item target in buffer or any item targets above the Limit, the program execution waits in this instruction until there is an item in the buffer.

If the MaxTime argument is specified then the wait time is supervised. If the waiting time exceeds the value of MaxTime and the TimeFlag argument is used, then the program will continue. If TimeFlag is not used, then an error is raised. If TimeFlag is specified, this will be set to TRUE if the time is exceeded, otherwise it will be set to FALSE.

### Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable ERRNO will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined.</td>
</tr>
<tr>
<td>ERR_PPA_TIMEOUT</td>
<td>Timeout without any error flag</td>
</tr>
</tbody>
</table>

### Syntax

NextItmTgtType

[ItemSource ':=' ] <variable (VAR) of itmsrc>, [ItemType ':=' ] <var or pers (INOUT) of num> [\Limit ':=' ] <expression (IN) of num> [\MaxTime ':=' ] <expression (IN) of num> [\TimeFlag ':=' ] <var or pers (INOUT) of bool>;

### Related information

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data type itmtgt</td>
<td>itmtgt - Item target data on page 360.</td>
</tr>
</tbody>
</table>
5 RAPID reference

5.1.5 QStartItmSrc - Start queue in item source

Usage

QStartItmSrc is used to start the queue in an item source. This instruction must be used when starting a new program or after flushing.

Basic example

QStartItmSrc PlaceSource;
The queue of objects in the item source PlaceSource is started.

Arguments

QStartItmSrc ItemSource

ItemSource

Data type: itmsrc

The started item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable ERRNO will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined</td>
</tr>
</tbody>
</table>

Syntax

QStartItmSrc

[ItemSource ':=' ] <variable (VAR) of itmsrc>;

Related information

For information about | See
---|---
The instruction QStopItmSrc | QStopItmSrc - Stop queue in item source on page 349.
5.1.6 QStopItmSrc - Stop queue in item source

Usage

QStopItmSrc is used to stop the queue in an item source.

Basic example

QStopItmSrc PlaceSource;

The queue of objects in the item source PlaceSource is stopped.

Arguments

QStopItmSrc ItemSource

ItemSource

Data type: itmsrc

The stopped item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable ERRNO will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined</td>
</tr>
</tbody>
</table>

Syntax

QStopItmSrc
[ItemSource ':=' ] <variable (VAR) of itmsrc>;

Related information

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The instruction QStartItmSrc</td>
<td>QStartItmSrc - Start queue in item source on page 348.</td>
</tr>
</tbody>
</table>
5.1.7 ResetFlowCount - Reset flow counter

Usage

ResetFlowCount is used to reset the flow counter. The flow counter indicates the number of objects that has passed the exit limit of a conveyor work area since last reset. The value of the flow counter can be retrieved with the function GetFlowCount.

Basic example

ResetFlowCount PlaceSource;
Resets the flow counter for an item source.

Arguments

ResetFlowCount ItemSource

ItemSource

Data type: itmsrc
The item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an errorhandler. The system variable ERRNO will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined</td>
</tr>
</tbody>
</table>

Syntax

ResetFlowCount[ItemSource ':='] <variable (VAR) of itmsrc>;

Related information

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The function GetFlowCount</td>
<td>GetFlowCount - Get number of passed items on page 359.</td>
</tr>
</tbody>
</table>
5.1.8 ResetMaxUsageTime - Reset max measured usage time

Description

ResetMaxUsageTime is used to reset the maximum measured usage time of the previously handled objects. This is the time between receiving a target with GetItmTgt, until the object is handled by the robot (acknowledge time).

ResetMaxUsageTime is only available with the PickMaster Ready.

Example

ResetMaxUsageTime ItmSrcData{PickWorkArea{1}}.ItemSource;

Resets the maximum usage time for an item source.

Arguments

ResetMaxUsageTime ItemSource

Item Source

ItemSource

Data type: itmsrc

The item source.

Error handling

The following recoverable errors are generated. They are handled in an error handler. The system variable ERRNO will be set to:

| ERR_ITMSRC_UNDEF | The itmsrc is undefined. |

Syntax

ResetMaxUsageTime[ItemSource ':='] <variable (VAR) of itmsrc>;
5.1.9 UseReachableTargets - Use reachable targets

Description

*UseReachableTargets* is used to activate a functional mode, where the robot only receives reachable targets for object handling.

When activated, non-reachable targets are filtered out for target requests with *GetItmTgt*.

*UseReachableTargets* sets an optimal target release zone with a variable size. The size of the release zone depends on the robot’s reach and the real-time speed of the conveyor. When the conveyor speed increases, the size of the release zone decreases, thereby decreasing the amount of targets available for use. If the conveyor speed is too high, the release zone disappears completely and no targets will be received until the speed is reduced.

*UseReachableTargets* is available only with the *PickMaster Ready* option.

**WARNING**

The target release zone depends on the selection of the enter/exit limits, see Application manual - PickMaster 3. The resulting target release zone will be the intersection of the optimal target release zone and the enter/exit region.

The recommended exit/enter values to avoid any impact on the optimal target zone are as follows:

- **Enter** = -10000 mm (this signifies, a distance well outside the robot reach in an upstream direction)
- **Exit** = 10000 mm (this signifies, a distance well outside the robot reach in a downstream direction)

The following figure shows the target release zone for an IRB 360 (as seen from above) at 4 different conveyor speeds. The light blue area is the working range of the robot and the green area is the target release zone.

![Diagram of target release zone](image)

**Example**

```rapid
UseReachableTargets ItmSrcData{PlaceWorkArea{1}}.ItemSource, TRUE, 0.7 \ReleaseTime:=0.1;
GetItmTgt PlaceSource, PlaceItem;
```

Activate *UseReachableTargets* in the place work area of a linear conveyor. The targets in use are expected to be placed within a maximum time of 0.7 seconds after being received with *GetItmTgt*. Targets become available for use 0.1 second...
before they enter robot reach. Then, the targets remain available for use until they leave the release zone.

## Arguments

UseReachableTargets ItemSource, Enable, UsageTime [\ReleaseTime]

### Item Source

**ItemSource**

Data type: itmsrc

The item source where *UseReachableTargets* is activated.

### Enable

**Enable**

Data type: bool

This activates/deactivates *UseReachableTargets*.

### Usage Time

**UsageTime**

Data type: num

The expected usage time of the targets. This is the time between receiving the target with *GetItmTgt*, until the object is handled (for example picked) by the robot (acknowledge time). The actual usage time is continuously measured and the maximum measured usage time can be received with *GetMaxUsageTime*. To avoid reach errors, the *UsageTime* value should be defined as a sum of the maximum measured usage time and a margin. For example, set UsageTime = Maximum measured usage time + 0.1 second. The drawback of having a large safety margin is an unnecessary reduction of the target release zone, which may decrease the pick rate.

[\ReleaseTime]

### Release Time

**Release Time**

Data type: num

The *Release Time* defines the time when the targets enter the release zone, before entering robot reach. If the value is negative, targets enter the release zone after they enter robot reach. A value of 0.1 or less is recommended to avoid reach errors. A higher value can be useful to handle high speed conveyors. The drawback of a higher value is an increasing risk of having upstream reach errors at low speeds.

### Note

It is possible to change *UsageTime* or *ReleaseTime* at any time. For example, a temporary reduction in the robot speed requires a longer usage time to avoid reach errors.

### Syntax

UseReachableTargets

```rapid
[ItemSource ':=' ] <variable (VAR) of itmsrc>,
[Enable ':=' ] <var or pers (IN) of bool>
```
Limitations

If the robot work area is limited in motion configuration, there is a possibility that targets upto 20 mm outside of the working area perpendicular to the conveyor moving direction, may be retrieved by the GetItmTgt instruction.

A work around to avoid the outside reach errors is to put an extra check on the Y-value of the itemtarget before moving towards it.
5.2 Functions

5.2.1 GetMaxUsageTime - Get max measured usage time

Description

GetMaxUsageTime is used to get the maximum measured usage time of the previously handled objects. It is the time between receiving a target with GetItmTgt, until the object is handled by the robot (acknowledge time). The actual usage time is continuously measured for each handled object. GetMaxUsageTime is only available with the PickMaster Ready.

Example

VAR num usetime;
usetime := GetMaxUsageTime(ItmSrcData{PickWorkArea{1}}.ItemSource);

usetime is the the maximum measured usage time since starting production or since executing ResetMaxUsageTime.

Return value

Data type: num
The maximum measured usage time since starting production or since executing ResetMaxUsageTime.

Arguments

GetMaxUsageTime (ItemSource)

Item Source

ItemSource
Data type: itmsrc
The item source.

Error handling

The following recoverable errors can be generated. They can be handled in an error handler. The system variable ERRNO will be set to:

| ERR_ITMSRC_UNDEF | The itmsrc is undefined. |

Syntax

GetMaxUsageTime ('([ItemSource ':=' ] <variable (VAR) of itmsrc> ')');

This function returns the value of the data in num type.
5.2.2 GetQueueLevel - Get queue level

Usage

GetQueueLevel is used to get current number of item targets in an item source fulfilling certain conditions.

Basic example

reg1 := GetQueueLevel(PlaceSource);

reg1 is assigned the current number of item targets in the item source PlaceSource.

Return value

Data type: num
The current number of item targets in the item source.

Arguments

GetQueueLevel (ItemSource [\ItmType] [\MinLimit] [\MaxLimit])

ItemSource

Data type: itmsrc
The item source that the current number of item targets should be retrieved from.

\ItmType

Data type: num
Only items of the specified type number will be counted.

\MinLimit

Data type: num
Defines the minimum distance to the robot center from where an item will be counted. A negative value indicates that the limit is upstreams from the robot center. A positive value indicates that the limit is downstreams. The parameters does not affect indexed work areas.

\MaxLimit

Data type: num
Defines the maximum distance to the robot center from where an item will be counted. A negative value indicates that the limit is upstreams from the robot center. A positive value indicates that the limit is downstreams. The parameter does not affect indexed work areas.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable ERRNO will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined</td>
</tr>
</tbody>
</table>
5 RAPID reference

5.2.2 GetQueueLevel - Get queue level

Continued

Syntax

GetQueueLevel '(
    [ItemSource ':=' ] <variable (VAR) of itmsrc> ')
[\ItmType ':=' ] <expression (IN) of num>
[\MinLimit ':=' ] <expression (IN) of num>
[\MaxLimit ':=' ] <expression (IN) of num>;

A function with a return value of the data type num.
5.2.3 GetQueueTopLevel - Get queue top level

**Usage**

GetQueueTopLevel is used to get the maximum number of item targets that simultaneously have been in the buffer of an item source.

**Basic examples**

```plaintext
reg1 := GetQueueTopLevel(PlaceSource);
```

`reg1` is assigned the maximum number of item targets that simultaneously have been in the item source `PlaceSource`.

**Return value**

- **Data type**: num
- The maximum number of item targets that simultaneously have been in the item source.

**Arguments**

- **GetQueueTopLevel (ItemSource)**

  - **ItemSource**
    - **Data type**: itmsrc
    - The item source that the current number of item targets should be retrieved from.

**Error handling**

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable `ERRNO` will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined</td>
</tr>
</tbody>
</table>

**Syntax**

```plaintext
GetQueueTopLevel '('
[ItemSource ':='] <variable (VAR) of itmsrc> ')';
```

A function with a return value of the data type `num`. 
5.2.4 GetFlowCount - Get number of passed items

Usage

GetFlowCount is used to get the total number of items that has passed the exit limit of a conveyor work area since ResetFlowCount was executed. Items that the robot handles will not be counted (even if they pass the exit limit before picking/placing occurs).

Basic example

```rapid
VAR num counter;
ResetFlowcount PlaceSource;
WaitTime 10;
counter := GetFlowCount(PlaceSource);
```

`counter` is assigned the number of items originating from `PlaceSource` that has passed the exit limit.

Return value

Data type: num
The number of items that has passed the exit limit since `ResetFlowCount` was executed.

Arguments

GetFlowCount (ItemSource)

ItemSource

Data type: itmsrc
The item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an errorhandler. The system variable `ERRNO` will be set to:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_ITMSRC_UNDEF</td>
<td>itmsrc undefined</td>
</tr>
</tbody>
</table>

Syntax

```rapid
GetFlowCount '('[ItemSource ':= ' ] <variable (VAR) of itmsrc> ')';
```

A function returns value of the data type num.

Related information

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The instruction</td>
<td><code>ResetFlowCount - Reset flow counter on page 350.</code></td>
</tr>
</tbody>
</table>
5 RAPID reference

5.3 Data types

5.3.1 itmtgt - Item target data

Usage

\texttt{itmtgt} is used to describe one pick or place item.

Description

\texttt{Itmtgt} identifies an item to pick or place. It contains the position and some additional data.

Components

\texttt{tag}

Data type: \texttt{num}

Sequential number identifying the item. Can be modified by a user hook for free usage. Is restricted to integer values.

\texttt{type}

Data type: \texttt{num}

Type of item.

\texttt{scene}

Data type: \texttt{num}

Sequential number identifying the scene, corresponding for example to a picture taken by the vision system.

\texttt{robtgt}

Data type: \texttt{robtgt}

The pick or place position.

\texttt{val1}

Data type: \texttt{num}

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

\texttt{val2}

Data type: \texttt{num}

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

\texttt{val3}

Data type: \texttt{num}

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

\texttt{val4}

Data type: \texttt{num}
Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

val5

Data type: num

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

Examples

Example 1

```
CONST itmtgt pickpos :=
    [1,2,1,0,0,0,0,0,
     [20,40,8],[1,0,0,0],[0,0,0,0],
     [9E+9,9E+9,9E+9,9E+9,0,0]];
```

A pick position is defined. The external axis related to the used conveyors must be set to zero, that is not marked as unused (by stating 9E+9). Example: if you have two conveyors, set the two last external axis positions to zero.

Structure

```
<dataobject of itmtgt>
  <tag of num>
  <type of num>
  <scene of num>
  <val1 of num>
  <val2 of num>
  <val3 of num>
  <val4 of num>
  <val5 of num>
  <dataobject of robtarget>
    <trans of pos>
      <x of num>
      <y of num>
      <z of num>
    <rot of orient>
      <q1 of num>
      <q2 of num>
      <q3 of num>
      <q4 of num>
    <robconf of confdata>
      <cf1 of num>
      <cf4 of num>
      <cf6 of num>
      <cfx of num>
    <extax of extjoint>
      <eax_a of num>
      <eax_b of num>
      <eax_c of num>
      <eax_d of num>
      <eax_e of num>
      <eax_f of num>
```

Continues on next page
## 5 RAPID reference

### 5.3.1 itmtgt - Item target data

*Continued*

#### Related information

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positioning instructions</td>
<td>Technical reference manual - RAPID Overview</td>
</tr>
<tr>
<td>Coordinate systems</td>
<td>Technical reference manual - RAPID Overview</td>
</tr>
<tr>
<td>Handling configuration data</td>
<td>Technical reference manual - RAPID Overview</td>
</tr>
<tr>
<td>Configuration of external axes</td>
<td>Technical reference manual - System parameters</td>
</tr>
<tr>
<td>What is a quaternion?</td>
<td>Technical reference manual - RAPID Overview</td>
</tr>
</tbody>
</table>
5.3.2 selectiondata - Selection data

Usage

`selectiondata` is used to describe the selection criteria. It is also used to describe item sorting.

Description

`selectiondata` is used to set the criteria for sorting and clearance area when retrieving item targets from an item source.

Components

**ShapeType**

Data type: `shapetype`

Specifies the shape of the clearance area that should be used.

- `SHAPE_UNDEFINED` specifies that no selection is used.
- `BOX` specifies that there must be a clear box shape around the item target position where no other item targets are present.
- `CYLINDER` specifies there must be a clear cylinder shape around the item target position where no other item targets are present.
- `SPHERE` specifies that there must be a clear sphere shape around the item target position where no other item targets are present.

**ConsiderType**

Data type: `aconsidertype`

Specifies which items in the queue that should be taken in consideration when selecting.

- `ITEMS_TO_USE` specifies that only items marked for use by this queue are considered in the selection.
- `ITEMS_BYPASS` specifies that only items marked to pass by this queue are considered in the selection.
- `ITEMS_PICKED` specifies that only items marked as already picked, by this queue or by a former queue in the line, are considered in the selection.
- `ITEMS_PLACED` specifies that only items marked as already placed, by this queue or by a former queue in the line, are considered in the selection.

If items with different marks should be taken into consideration when selecting an item, then use a bit-or operation with the consideration types. (RAPID function `BitOr(<byte>,<byte>)`.)

**GeometricData**

Data type: `geodata`

The data that defines the geometric shape dimensions (x, y, z and radius).

- A `BOX` shape is defined by the x, y, and z-values.
- A `CYLINDER` shape is defined by the radius value and the height is defined by the z-value.
A SPHERE shape is defined by the radius value. The orientation of the shape’s coordinate system is defined by the offset data component. By default it is the coordinate system of the shape aligned to the workobject or conveyor frame. Note that all shapes origin are placed in the center of the shape and the values are the distance to every plane in both positive and negative direction. That is, if a box is defined as x: 10, y: 15 and z: 20 the box will have a size of 20 mm in x-direction, 30 mm in y-direction and 40 in z-direction. If no offset is used the check for other items in range will be done 10 mm before, 10 mm after, 15 mm left of, 15 mm right of, 20 mm above, and 20 mm underneath every item.

Offset

Data type: offsetdata

The offset consists of OffsetRelation (offsetreltype) and OffsetPose (pose).

The OffsetRelation can be of two different types.

- FRAME_COORD_DIR indicates that the rotation in the OffsetPose is relative to the workobject or conveyor frame coordinate system.
- ITEM_COORD_DIR indicates that the rotation in the OffsetPose is relative to the item coordinate system of the item to check.

The OffsetPose is used to move the center of the shape away from the item position, for example, if the grip position of the item is not at the center of real object to pick.

Examples

```
VAR selectiondata clear_rect:= [BOX, ITEMS_TO_USE, [22, 15, 5, 0], [FRAMECOORD_DIR, [[0, 7, 0], [1, 0, 0, 0]]]);
```

Limitations

The orientation must be normalized; that is the sum of the squares must equal 1.

\[ q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \]

Structure

```
<dataobject of selectiondata>
  <ShapeType of shapetype>
  <ConsiderType of considertype>
  <GeometricData of geodata>
    <x of num>
    <y of num>
    <z of num>
    <radius of num>
  <Offset of offsetdata>
    <OffsetRelation of offsetreltype>
    <OffsetPose of pose>
      <trans of pos>
        <x of num>
        <y of num>
        <z of num>

Continues on next page
Related information

<table>
<thead>
<tr>
<th>For information about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data type pose</td>
<td>Technical reference manual - RAPID Instructions, Functions and Data types.</td>
</tr>
<tr>
<td>The function BitOr</td>
<td>Technical reference manual - RAPID Instructions, Functions and Data types.</td>
</tr>
<tr>
<td>Example using selectiondata</td>
<td>Example: Selecting item depending on clearance zone on page 387.</td>
</tr>
</tbody>
</table>
5.3.3 sortdata - Sort data

**Usage**

sortdata is used to describe the sorting criteria.

**Description**

sortdata is used to set the criteria for sorting item targets from an item source.

**Components**

**SortType**

- **Data type:** sorttype

  Type of sorting that is going to be used.
  - UNSORT_TYPE tells that no sorting is used.
  - POS_Y_SORT_TYPE tells that the sorting shall be done from the positive y-direction of the work area.
  - NEG_Y_SORT_TYPE tells that the sorting shall be done from the negative y-direction of the work area.

**CheckBoundary**

- **Data type:** num

  The clearance distance for sorting, in millimeters. The distance is defined as the minimum distance to the next item in the sorting direction.

**SortDirOffset**

- **Data type:** num

  An offset distance beyond the item target in the sort direction. Is used to define the inner limit for the corridor in which no other item targets are allowed.

**Examples**

VAR sortdata y_sort:=[NEG_Y_SORT_TYPE, 78, 52];

**Structure**

<dataobject of sortdata>
<SortType of sorttype>
<CheckBoundary of num>
<SortDirOffset of num>
5.4 RAPID programs

5.4.1 RAPID programs

Introduction

Overview

Each robot has a default RAPID program that can be edited using a normal text editor from the robot settings of the job dialog. When a job is started, the program is downloaded by PickMaster in the picking controller. The program contains the Main routine where the program execution starts.

Note

Due to the download procedure, this program cannot be modified directly on the robot system.

The installation contains the following program template files:

<table>
<thead>
<tr>
<th>Template</th>
<th>Customized for</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMppa360.mod</td>
<td>Four axes FlexPicker IRB 360.</td>
</tr>
<tr>
<td>PMppaDelta.mod</td>
<td>Five axes FlexPicker IRB 365 and IRB 390.</td>
</tr>
<tr>
<td>PMppa6Axes.mod</td>
<td>SCARA robots, for example, IRB 910. Four axes robots of articulated arm type IRB 460 and IRB 660.</td>
</tr>
<tr>
<td></td>
<td>Six axes robots of articulated arm type, for example, IRB 120.</td>
</tr>
<tr>
<td></td>
<td>Seven axes robots of articulated arm type, for example, IRB 14050.</td>
</tr>
</tbody>
</table>

Program execution - General

The RAPID program is loaded and started from the Main routine by PickMaster when a new job is started.

For every cycle, the default RAPID program performs:

- a pick on a pick work area.
- a place on a place work area.

If there are more than one pick work area with a robot, it uses the one having the lowest configured work area index. If there are more than one place work area with a robot, it uses the one having the lowest configured work area index. The RAPID program can be modified to implement another sequence, for example, to double pick with single place.

Program execution – Work areas

In RAPID, a work area is always associated with an item source object. The item source is sometimes referred to as a queue. The item source holds all target positions related to this work area. Target positions are continuously received in the item source, while being detected with the associated flow handler sensor.

Continues on next page
Program execution – Target positions

For each pick, a pick target is fetched from the pick item source. The target position gives the location of the next item to be picked.

For each place, a place target is fetched from the place item source. The target position gives the location of the next empty place location for the item to be placed.

Movements

The RAPID program is built with six different movements.

For a six axis robot, the following two intermediate points must be used:

- Between position 3 and position 4.
- Between position 6 and the next loop’s position 1.

The following six movements are included.

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>Approach position above the pick target. The distance above the pick target is the pick elevation value, in negative z-direction of the tool, given in the Work Area Properties dialog in the job dialog. The target is of corner path type and the vacuum activation occasion is calculated as the time before the middle of the corner path. The time is entered in the Work Area Properties dialog.</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>This is the pick target. The robot TCP is coordinated relative to the conveyor during the pick time entered in the Work Area Properties dialog. The TCP follows the pick target during the pick time.</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>Last position in the pick sequence. The distance above the pick target is calculated in the same way as the approach position. The position is coordinated to the conveyor until the middle of the corner path. Therefore the used item target must be acknowledged, so the item source can start tracking the next item target in the pick work area buffer. The target cannot be a fine point.</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>Approach position above the place target. The distance above the place target is the pick or place elevation value, in negative z-direction of the tool, given in the Work Area Properties dialog.</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>This is the place target. The robot TCP is coordinated relative to the conveyor during the place. The moment for the vacuum reversion event is calculated as the time before the half place time. The vacuum off moment is calculated as a time after the half place time.</td>
</tr>
</tbody>
</table>
Last position of the sequence. The position is coordinated to the conveyor until the TCP passes the middle of the corner path or goes into the fine point. Therefore the used item target must be acknowledged, so the item source can start tracking the next item target in the pick work area buffer. The target cannot be a fine point.

**Note**

When running a pick and place cycle over moving conveyors, the RAPID program pointer runs in advance and picks out a target long before it is going to be used. By the time the robot uses the target it may already have moved past the exit limit. RAPID moves the program pointer in advance about 100ms. In a coordinated fine point the "running in advance" is triggered at the beginning of the fine point movement as the robot locks above the conveyor. If the PickTime is long (for example, 50ms) the next target will be taken out of the queue long before (50ms) the robot is physically going to go there. If the conveyor speed is high 50ms may mean that the target to pick is already beyond the exit limit. Still the robot will try to pick it.

**Program modules**

The default RAPID program contains three program modules.

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMTWMMAIN</td>
<td>Handles the main program initiations and execution sequence. Do not edit this module for customization purpose.</td>
</tr>
</tbody>
</table>

**System modules**

An ABB robot controller with the RobotWare option *PickMaster Ready* will always contain the loaded system modules *ppaBase* (crypted) and *ppaUser* (open).

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppaBase</td>
<td>Contains variables for communication with PickMaster, event routines and routines for creating, initiating, and deleting item sources.</td>
</tr>
<tr>
<td>ppaUser</td>
<td>Contains declarations of public data types and holds the work object data for indexed work areas. It also contains the declaration of default tool data, for example, <em>PickAct1</em> and <em>PickAct2</em>.</td>
</tr>
</tbody>
</table>

**Public data types**

**Overview**

The system module *ppaUser* contains two record definitions, *sourcedata* and *noncnvwoIIData*.
The sourcedata is used in the variable array ItmSrcData. This array holds data about every item source.

The record can be extended for other purposes, but do not change or delete any component in the structure.

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>bool</td>
<td>Flag to indicate that the array index is used.</td>
</tr>
<tr>
<td>ItemSource</td>
<td>itmsrc</td>
<td>Descriptor to the item source.</td>
</tr>
<tr>
<td>SourceType</td>
<td>itmsrctype</td>
<td>Type of source, PICK_TYPE, PLACE_TYPE or UNDEFINED_TYPE.</td>
</tr>
<tr>
<td>Ack</td>
<td>triggdata</td>
<td>Triggdata for acknowledging the item targets.</td>
</tr>
<tr>
<td>Nack</td>
<td>triggdata</td>
<td>Triggdata for negative acknowledging the item targets.</td>
</tr>
<tr>
<td>SimAttach1</td>
<td>triggdata</td>
<td>Triggdata for attaching a nearby item to activator 1 in simulation.</td>
</tr>
<tr>
<td>SimAttach2</td>
<td>triggdata</td>
<td>Triggdata for attaching a nearby item to activator 2 in simulation.</td>
</tr>
<tr>
<td>SimDetach1</td>
<td>triggdata</td>
<td>Triggdata for detaching an item held by activator 1 in simulation.</td>
</tr>
<tr>
<td>SimDetach2</td>
<td>triggdata</td>
<td>Triggdata for detaching an item held by activator 2 in simulation.</td>
</tr>
<tr>
<td>VacuumAct1</td>
<td>triggdata</td>
<td>Triggdata for vacuum activation on real robot.</td>
</tr>
<tr>
<td>VacuumAct2</td>
<td>triggdata</td>
<td>Triggdata for vacuum activation on real robot.</td>
</tr>
<tr>
<td>VacuumRev1</td>
<td>triggdata</td>
<td>Triggdata for vacuum blow on real robot.</td>
</tr>
<tr>
<td>VacuumRev2</td>
<td>triggdata</td>
<td>Triggdata for vacuum blow on real robot.</td>
</tr>
<tr>
<td>VacuumOff1</td>
<td>triggdata</td>
<td>Triggdata for vacuum off on real robot.</td>
</tr>
<tr>
<td>VacuumOff2</td>
<td>triggdata</td>
<td>Triggdata for vacuum off on real robot.</td>
</tr>
<tr>
<td>Wobj</td>
<td>wobjdata</td>
<td>Work object data for the source.</td>
</tr>
<tr>
<td>VacActDelay</td>
<td>num</td>
<td>Vacuum activation delay</td>
</tr>
<tr>
<td>VacRevDelay</td>
<td>num</td>
<td>Vacuum reversion delay</td>
</tr>
<tr>
<td>VacOffDelay</td>
<td>num</td>
<td>Vacuum off delay</td>
</tr>
<tr>
<td>TunePos</td>
<td>pos</td>
<td>Position tuning for the work area.</td>
</tr>
<tr>
<td>TrackPoint</td>
<td>stoppointdata</td>
<td>Follow time data.</td>
</tr>
<tr>
<td>OffsZ</td>
<td>num</td>
<td>Height for the offset point above the pick or place position.</td>
</tr>
</tbody>
</table>
noncnwobjdata

The **noncnwobjdata** is used in the persistent variable array **NonCnvWOData**. This is only used for indexed work areas. The work object data is stored in this array. This data is then used when the item sources are created. The record can be extended for other purposes, but do not change or delete any component in the structure.

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>bool</td>
<td>Flag to indicate that the array index is used.</td>
</tr>
<tr>
<td>NonCnvWobjName</td>
<td>string</td>
<td>Name of the work area.</td>
</tr>
<tr>
<td>Wobj</td>
<td>wobjdata</td>
<td>The stored work object data.</td>
</tr>
</tbody>
</table>

**AlwaysClearPath**

Clear path

The robot path is cleared before the restart when a stop occurs during a motion that is coordinated to a moving work object. Otherwise the coordinated motion continues the stored path, but the position of the object in the conveyor may have changed to a position that is out of reach by the robot.

Unconditional path clearing

The **AlwaysClearPath (bool always)** routine unconditionally clears the path before the restart, if the input parameter value is set to TRUE.
5.4.2 Variables

Introduction to variables
The PickMaster robot controller contains many RAPID variables. The variables are declared in both ppaBase and ppaUser. Many are not used in customized programs.

Public variables in ppaUser
Overview
The following variables in ppaUser can be used.

VAR sourcedata ItmSrcData{MaxNoSources}
This array variable keeps information about all work areas. The index given in the work area configuration is the index of the ItmSrcData array.

PERS noncnvwobjdata NonCnvWOData{MaxNoSources}:=[…]
This array variable stores the work object frames for the indexed work areas. The key to find a certain work object calibration is the name, that must be same as the name in the work area configuration.

TASK PERS tooldata PickAct1:=[…]
This tooldata is used for pick and place operations.

Note
The direction of tool must fit the direction of items that are retrieved from the queue. The target positions of the items, which are retrieved from the queue, are rotated 180 degrees around their x-axis from the defined direction.

In an installation with a hanging IRB 360 and items lying on a horizontal conveyor, the tool’s z-direction will point out from the nose and down into the conveyor, like tool0.

Public variables in ppaBase
The following variables in ppaBase can be used.

TASK PERS num Vtcp:=1000
Used for speed adjustment from PickMaster.

TASK PERS speeddata MaxSpeed:=[…]
Highest speed used for movements.

TASK PERS speeddata LowSpeed:=[…]
Low speed used for movements.

TASK PERS speeddata VeryLowSpeed:=[…]
Lowest speed used for movements.

Continues on next page
Public variables in PickMaster template programs

The following public variables are used in the PickMaster template program.

VAR num PickWorkArea{X}:=0

The PickWorkArea array is used to specify from which work area the robot will pick
an item. The pick work areas are ordered with respect to selection index.

PickWorkArea{1} has the lowest work area selection index.
PickWorkArea{2} has the second lowest selection index.

VAR num PlaceWorkArea{X}:=0

The PlaceWorkArea array is used to specify on which work area the robot will place
an item. The place work areas are ordered with respect to selection index.

PlaceWorkArea{1} has the lowest work area selection index.
PlaceWorkArea{2} has the second lowest selection index.

VAR num OtherWorkArea{X}:=0

The OtherWorkArea array is used to specify to which work area the robot will go
for a user defined purpose. The other work areas are ordered with respect to
selection index.

OtherWorkArea{1} has the lowest work area selection index.
OtherWorkArea{2} has the second lowest selection index.

VAR itmtgt PickTarget:=[...]

Used to retrieve a pick target from a pick item source.

VAR itmtgt PlaceTarget:=[...]

Used to retrieve a place target from a place item source.

TASK PERS wobjdata WObjPick:=[...]

Holds the wobjdata for the work area. The information is moved from ItmSrcData
to WObjPick in the Pick routine because the motion instructions need to have the
wobjdata as PERS type.

TASK PERS wobjdata WObjPlace:=[...]

Holds the wobjdata for the work area. The information is moved from ItmSrcData
to WObjPlace in the Place routine because the motion instructions need to have the
wobjdata as PERS type.

TASK PERS robtarget SafePos:=[...]

Defined start position for the robot. Edit this robtarget to fit the application.

TASK PERS robtarget IntPosPickX:=[...]

Defined intermediate position for every pick work area robot. Edit this robtarget to
fit each work area.

TASK PERS robtarget IntPosPlaceX:=[...]

Defined intermediate position for every place work area robot. Edit this robtarget
to fit each work area.
Load data (loaddata) used for pick and place operations. Edit this loaddata to fit the picked item. If different item types are used, declare one loaddata for each type. It is important that correct loaddata is used to get the best performance of the robot.

The default loaddata is the same as tooldataload0.
5.4.3 Routines

Introduction to routines
The PickMaster RAPID modules contain many routines, some are very useful for the end user, others are only to be used internally by the PickMaster program.

Public routines in PickMaster template programs
The following public routines are available in the PickMaster template programs.

PROC main()
Start routine for the RAPID program. The program will always start from this routine.

PROC InitSafeStop()
Initiates the SafeStop trap. It must be executed at the beginning of the program execution to get a correct robot stop when the PickMaster project is paused or stopped.

PROC InitTriggs()
Sets trigger events for the vacuum activation, reversion and turning off, at the project start for every used work area index. See more at SetTriggs.

PROC InitPickTune()
Initiates the PickTune trap. Must be executed at the beginning of the project start so the work areas can be tuned.

PROC SetTriggs(num Index)
Sets trigger events for the vacuum activation, reversion and turning off. The default program only sets up events for one vacuum ejector on the I/O group goVacBlow1. If more than one vacuum ejector is used, the new vacuum ejector I/O group must be setup for the correct work area and the default routine must be edited to get the right vacuum ejector to each work area.

PROC SetSimulatedTriggs(num Index)
The offline version of PROC SetTriggs(num Index).
No need to change if the tool has 1 - 2 activators.

PROC SetSimulatedDummyTriggs(num Index)
Sets up all trigger events used in the RAPID code that not is relevant for simulated mode.
No need to change if the tool has 1 - 4 activators.

PROC SetDummyTriggs(num Index)
Sets up all trigger events used in the RAPID code that not is relevant for online mode.
No need to change if the tool has 1 - 4 activators.

PROC InitSpeed()
Sets the robot speed used in the program. The instruction VelSet is executed in this routine, which sets the maximum allowed speed for the robot. If a six axes robot is used, this limit can be tuned to avoid motion errors.

Continues on next page
PROC PickPlace()

Starts the item queues and initiates the final settings. The pick and place sequence is called from this routine. Do not make changes in this routine. This routine is called when the pick and place execution is started.

PROC SafeStop()

When the project is stopped or paused this routine will be called either from the SafeStopTrap routine or the PickPlace routine. The slow motion to the safe position is called from this routine.

PROC GotoRestartPos()

Runs the slow motion to the safe position and sends a negative acknowledge to all item sources. This must be done to tell the sources that the execution was interrupted.

PROC Home()

Service routine that moves the robot to the safe position.

PROC WashDown()

Wash down service routine.

PROC TestCycle()

Test service routine.

PROC Homepos()

Service routine that moves the robot to the synchronization position.

PROC EnumerateWorkAreas()

Sets up the arrays of work areas for Pick, Place, and Other.

PROC PickPlaceSeq()

Specifies the sequence of the application, that is the logic of how the robot will pick and place from different queues.

This routine is called once every loop, which is counted as one pick in the pick rate statistics shown in the PickMaster production tab.

PROC Pick(num Index)

Executes one pick. The index defines which work area the item will be picked from.

PROC Place(num Index)

Executes one place. The given index defines which work area the item will be placed on.

TRAP SafeStopTrap

Trap routine to catch the stop I/O signal. This is executed if the stop I/O signal is set before SafeStop is called from the PickPlace routine.

TRAP PickTuneTrap

Trap routine to attach the tuned values from the PickMaster to the corresponding variables.
Hidden routines in ppaBase module

Overview

Following are the hidden routines in the ppaBase module.

PROC ResetEvent()

Resets some variables. This routine is only executed in the RESET system event shelf.

PROC PowerOnEvent()

Resets some variables. This routine is executed only in the POWER_ON system event shelf.

PROC StopEvent()

Clears the robot path if the robot is in a coordinated motion when the stop occurs. This routine is only executed in the STOP system event shelf.

PROC RestartEvent()

This routine is only executed in the RESTART system event shelf. If the robot is currently in a coordinated motion, this routine will force the program to restart the program from the level that has an error handler for the raised error PPA_RESTART.

PROC NewSource()

Creates a new item source and initiates the ItmSrcData variable. PickMaster calls this routine for each work area when the project starts.

PROC ClearAll()

Resets all important variables and deletes all item sources. This routine is called when the project is stopped.

PROC PickRateInit()

Initiates the pick rate calculation.

PROC PickRateReset()

Resets the pick rate calculation.

PROC CheckAx4Rev ()

Checks if it is necessary to reset the fourth axis on the IRB340.

PROC ResetAx4 (VAR mecunit MechUnit)

Resets the fourth axis.

PROC NotifyClearAll ()

Tells PickMaster that ClearAll is executed.

PROC NotifySafeStop ()

Tells PickMaster that SafeStop is executed.

PROC NotifyRunning ()

Tells PickMaster that the process is running.

PROC NotifyWaitForExe ()

Tells PickMaster that the RAPID program is waiting for new order.

Continues on next page
PROC WaitForExeOrder ()

Instruction where the RAPID program waits for PickMaster to give the next execution order. If no order is given, the RAPID execution will wait and idle on this instruction.

PROC IncrPicks ()

Increments the pick calculation.

PROC ppaDropWobj(PERS wobjdata Wobj)

Encapsulates the DropWobj instruction. See Application manual - Conveyor tracking for more information

PROC WalkTheData()

Traces the content of the array variables ItmSrcData and NonCnvWOData, which can be useful when trying to find an error. It prints the file TheData.log on the system directory on the controller.

TRAP PickRateTrap

Trap routine to calculate the correct pick rate for the robot.

PROC AlwaysClearPath(bool always)

For more details, see AlwaysClearPath on page 371
5.5 Program examples

5.5.1 Example: Mixing one pick work area and two place work areas

Description of example
In this example we use one pick work area with two types of items. The items are put on two out work areas depending on type of item.

1. Pick item from pick work area
2. Define type of item
3. Place on out work area

Example code

PROC PickPlaceSeq()
  Pick PickWorkArea{1};
  IF PickTarget.Type = 1 THEN
    Place PlaceWorkArea{1};
  ELSEIF PickTarget.Type = 2 THEN
    Place PlaceWorkArea{2};
  ENDIF
ENDPROC
5.5.2 Example: Mixing two pick work areas and one place work area

Description of example

In this example, we use the place work area as master to decide which item is needed to fill a pattern, which in turn defines pick work area to pick from.

1. Check next item target type
2. Decide which work area to pick from
3. Pick item from pick work area
4. Place on out work area

Example code

```rapid
PROC PickPlaceSeq()
    VAR num PlaceType:=0;
    NextItmTgtType
        ItmSrcData{PlaceWorkArea{1}}.ItemSource,
        PlaceType;
    IF PlaceType = 1 THEN
        Pick PickWorkArea{1};
    ELSEIF PlaceType = 2 THEN
        Pick PickWorkArea{2};
    ENDIF
    Place PlaceWorkArea{1};
ENDPROC
```
5.5.3 Example: Mixing with one pick and one place work area

Description of example

In this example we use the place work area as master to decide which item is needed to fill a pattern, which in turn defines which item to pick.

1. Check next item target type
2. Pick item from pick work area
3. Place on out work area

Note

It's recommended to use the Use Start/Stop in the Available Work Areas setting.

Example code

PROC Pick(num Index)
VAR num PickType:=0;
VAR num PlaceType:=0;

WObjPick:=ItmSrcData{Index}.Wobj;
NextItmTgtType
   ItmSrcData{PlaceWorkArea{1}}.ItemSource,PlaceType;
TEST PlaceType
CASE 4:
   PickType:=1;
CASE 5:
   PickType:=2;
CASE 6:
   PickType:=3;
ENDTEST
GetItmTgt ItmSrcData{Index}.ItemSource, PickTarget
 \ItemType:=PickType;
TriggL \Conc, RelTool(PickTarget.RobTgt, 0, 0,
   -ItmSrcData{Index}.OffsZ), MaxSpeed,
   ItmSrcData{Index}.VacuumAct1, z20, PickAct1 \WObj:=WObjPick;
MoveL \Conc, PickTarget.RobTgt, LowSpeed, z5 \Inpos:=
   ItmSrcData{Index}.TrackPoint, PickAct1 \WObj:=WObjPick;
GripLoad ItemLoad;
TriggL RelTool(PickTarget.RobTgt, 0, 0, -ItmSrcData{Index}.OffsZ),
   LowSpeed, ItmSrcData{Index}.Ack, z20, PickAct1
   \WObj:=WObjPick;
ENDPROC
5.5.4 Example: Double pick single place

Description of example
The robot shall pick up two items, one-by-one, on the infeeder conveyor, and then place both items on the outfeed conveyor. This operation requires a picking tool with two vacuum ejectors.

Implementation
As a starting point, create a simple working setup with one robot.

The RAPID program needs to be modified. To edit the RAPID program, go to the Recipe Setting, select a robot and display the drop down menu, select the Rapid program and select Edit....

The PickPlaceSeq routine shall perform two Pick routine calls to handle the first and the second pick. It will then perform one Place routine call to handle the simultaneous placing of the picked up items. See the following example code.

```plaintext
!***********************************************************
PROC PickPlaceSeq()
    Pick PickWorkArea{1}, 1;
    Pick PickWorkArea{1}, 2;
    Place PlaceWorkArea{1};
ENDPROC
```

For the Pick routine, see the following example code. Note the usage of PickAct2 and VacuumAct2 for the second pick.

```plaintext
!***********************************************************
PROC Pick(num Index, num pickNo)
    IF Index > 0 THEN
        WObjPick:=ItmSrcData{Index}.Wobj;
        GetItmTgt ItmSrcData{Index}.ItemSource,PickTarget;
        IF pickNo = 1 THEN
            TriggL\Conc,RelTool(PickTarget.RobTgt,0,0,
                                 -ItmSrcData{Index}.OffsZ),
        ENDIF;
    ENDIF;
ENDPROC
```

Continues on next page
MaxSpeed, ItmSrcData{Index}.VacuumAct1, z20,
PickAct1\WObj:=WObjPick;
TriggL\Conc, PickTarget.RobTgt, LowSpeed, ItmSrcData{Index}.SimAttach1,
z5\Inpos:=ItmSrcData{Index}.TrackPoint,
PickAct1\WObj:=WObjPick;
GripLoad ItemLoad;
RelTool(PickTarget.RobTgt, 0, 0, -ItmSrcData{Index}.OffsZ),
LowSpeed, ItmSrcData{Index}.Ack, z20, PickAct1\WObj:=WObjPick;
ELSEIF pickNo = 2 THEN
TriggL\Conc, RelTool(PickTarget.RobTgt, 0, 0, -ItmSrcData{Index}.OffsZ),
MaxSpeed, ItmSrcData{Index}.VacuumAct2,
z20, PickAct2\WObj:=WObjPick;
TriggL\Conc, PickTarget.RobTgt, LowSpeed, ItmSrcData{Index}.SimAttach2,
z5\Inpos:=ItmSrcData{Index}.TrackPoint,
PickAct2\WObj:=WObjPick;
GripLoad ItemLoad;
RelTool(PickTarget.RobTgt, 0, 0, -ItmSrcData{Index}.OffsZ),
LowSpeed, ItmSrcData{Index}.Ack, z20,
PickAct2\WObj:=WObjPick;
ENDIF
ELSE
ErrWrite "Missing item distribution", "Cannot pick because no
item distribution contains current work area."
\RL2:="Please check configuration";
SafeStop;
ENDIF
ENDPROC

The tooldata PickAct1 is used at the first pick. The tooldata PickAct2 is used
at the second pick. Update PickAct1 and PickAct2 (defined in module
ppaUser.sys): Define the tool center point in the center of the controlled vacuum
ejector. Update also the weight and the center of mass. Save the updates of the
RAPID program, close the editor, and apply the updates.

For the Place routine see the following example. Note the usage of VacuumOff1
and VacuumOff2 for the simultaneous placing of both held items.

!***********************************************************
!
! Procedure Place
!
! Executes a place movement.
! Edit this routine to modify how the robot shall
! execute the place movements.
! Needs to be changed if more than one activator is used.
!
!***********************************************************
PROC Place(num Index)
IF Index > 0 THEN
WObjPlace:=ItmSrcData{Index}.Wobj;
GetItmTgt ItmSrcData{Index}.ItemSource, PlaceTarget;
Continues on next page
MoveL\Conc,RelTool(PlaceTarget.RobTgt,0,0,-ItmSrcData{Index}.OffsZ),
MaxSpeed,z20,PlaceAll\WObj:=WObjPlace;
TriggL\Conc,PlaceTarget.RobTgt,LowSpeed,ItmSrcData{Index}.VacuumRev1\T2:=
ItmSrcData{Index}.VacuumOff1\T3:=ItmSrcData{Index}.VacuumOff2\T4:=
ItmSrcData{Index}.VacuumRev2\T5:=ItmSrcData{Index}.SimDetach1\T6:=
ItmSrcData{Index}.VacuumOff2\T7:=ItmSrcData{Index}.TrackPoint,PlaceAll\WObj:=WObjPlace;
GripLoad load0;
TriggL RelTool(PlaceTarget.RobTgt,0,0,-ItmSrcData{Index}.OffsZ),
LowSpeed,ItmSrcData{Index}.Ack,z20,PlaceAll\WObj:=WObjPlace;
ELSE
ErrWrite "Missing item distribution", "Cannot place because no
item distribution contains current work area."
\RL2:="Please check configuration";
SafeStop;
ENDIF
ENDPROC

The tooldata PlaceAll (defined in module ppaUser.sys) is used at place. Update
PlaceAll: Define the tool center point in the center of the controlled vacuum
ejectors. Update also the weight and the center of mass. Save the updates of the
RAPID program, close the editor, and apply the updates.

**Note**

Use the same method to setup a tool with more than two activators. However, a
few additional setup steps are required. For example, using a tool with 3-4
activators requires the following additional steps:

1. **Select two I/O boards as controller option.** Alternatively, create additional
   signals goVacBlow3, goVacBlow4, doVacuum3, doVacuum4, doBlow3,
   and doBlow4. The first bit of goVacBlowX shall overlap the signal
doVacuumX. The second bit of goVacBlowX shall overlap the signal
doBlowX.

2. **Update the SetTriggs routine.** Enable the TriggEquip events
   VacuumAct3, VacuumOff3, VacuumAct4, and VacuumOff4 by removing
   the comments on these lines.
Description of example

In this example we place a predefined pattern on an indexed work area. The position generator signal is triggered from RAPID.

Four new signals must be defined.

1. Position generator signal set from RAPID, `doSIMPosGen`.
2. Position generator signal that generates an event from the controller to the computer, `diSIMPosGen`.
3. Trigger signal that tells the system on the computer to send a predefined position, `doSIMTrig`.
4. Strobe signal that tells the system a position is sent, `diSIMStrobe`.

The signals can be defined on the PPASIM board. For example:

```
{Name "doSIMPosGen" -SignalType "DO" -Unit "PPASIM" -UnitMap "6" -Access "ALL"
{Name "doSIMTrig" -SignalType "DO" -Unit "PPASIM" -UnitMap "7" -Access "ALL"
{Name "diSIMPosGen" -SignalType "DI" -Unit "PPASIM" -UnitMap "6" -Access "ALL"
{Name "diSIMStrobe" -SignalType "DI" -Unit "PPASIM" -UnitMap "7" -Access "ALL"
```

Cross connect the trigger and strobe signal and the position generator signals.

For example:

```
EIO_CROSS
  -Res "diSIMPosGen" -Act1 "doSIMPosGen"
  -Res "diSIMStrobe" -Act1 "doSIMTrig"
```

In the RAPID code, create a control of the place queue. If the queue is empty (all positions in the pattern are used) set the signal `doSIMPosGen` high (in the RAPID code). This signal is cross connected with the `diSIMPosGen` and an event will be sent to the computer from the controller that a new pattern has to be sent to the controller. The trigger strobe signals are also cross connected and the `diSIMStrobe` will be used to strobe the system.

Example code

```
PROC Place(num Index)
  VAR bool flagplace:=TRUE;
  WObjPlace:=ItmSrcData{Index}.Wobj;
  flagplace:=TRUE;
  WHILE flagplace=TRUE DO
    GetItmTgt ItmSrcData{Index}.ItemSource,
    PlaceTarget\MaxTime:=1\TimeFlag:=flagplace;
    IF flagplace=TRUE THEN
      PulseDO\PLength:=0.2,doSIMPosGen;
    ENDIF
  ENDWHILE
```

Continues on next page
5 RAPID reference

5.5.5 Example: Placing a predefined pattern on indexed work area

Continued

MoveL\Conc, RelTool(PlaceTarget.RobTgt, 0, 0, ItmSrcData(Index).OffsZ), MaxSpeed, z20, PickAct1\WObj:=WObjPlace;
TriggL\Conc, PlaceTarget.RobTgt, LowSpeed, ItmSrcData(Index).VacuumRev1 \T2:=ItmSrcData(Index).VacuumOff1, z5 \Inpos:=ItmSrcData(Index).TrackPoint, PickAct1\WObj:=WObjPlace;
GripLoad load0;
Trigg RelTool(PlaceTarget.RobTgt, 0, 0, ItmSrcData(Index).OffsZ), LowSpeed, ItmSrcData(Index).Ack, z20, PickAct1\WObj:=WObjPlace;
ENDPROC
5.5.6 Example: Selecting item depending on clearance zone

Description of example

In this example, we select items on a conveyor belt depending on the clearance zone around the item, that is if there is any other item target within a specified area. This is useful when it is important that the gripper does not touch surrounding objects.

The selection algorithm selects the object that is closest to the exit limit in x-direction and has no locking objects in the selection shape.

Use the check limit in x-direction as a parameter to the GetItmTgt instruction. This makes it possible to define the starting point from where the first object will be picked. The instruction will try to retrieve the first object between the check and enter limits. This will cause the selection algorithm to take all objects between the check limit and the exit limit into consideration when checking for the nearest objects. Therefore the distance between the check limit and the exit limit will be at least the diameter of the largest item.

The illustration below shows how the items are selected depending on the position and the orientation. The robot will first pick item 4 and then item 3. The other two will never be picked.

- Item 1 cannot be picked because it has passed the check limit, and item 2 is inside its selection shape.
- Item 2 cannot be picked because the positions of items 1 and 3 are inside its selection shape.
- Item 3 cannot be picked because item 4 is inside its selection area.
- Item 4 can be picked because no other item is its selection shape.
- Item 3 will be picked after item 4 is no longer present.
5 RAPID reference

5.5.6 Example: Selecting item depending on clearance zone

Continued

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Grippers</td>
</tr>
<tr>
<td>B</td>
<td>Selection shape</td>
</tr>
<tr>
<td>C</td>
<td>Item</td>
</tr>
<tr>
<td>D</td>
<td>Item target position</td>
</tr>
<tr>
<td>E</td>
<td>Product flow direction</td>
</tr>
<tr>
<td>F</td>
<td>Exit</td>
</tr>
<tr>
<td>G</td>
<td>Check limit</td>
</tr>
<tr>
<td>H</td>
<td>Enter</td>
</tr>
</tbody>
</table>

See *selectiondata - Selection data on page 363.*

Example code

```rapid
PROC Pick(num Index)
VAR selectiondata sel_data;
VAR robtarget draw_target;
VAR num check_limit;

sel_data.ShapeType:=BOX;
sel_data.ConsiderType:=BitOr(ITEMS_TO_USE,ITEMS_BYPASS);
sel_data.GeometricData.x:=60;
sel_data.GeometricData.y:=70;
sel_data.GeometricData.z:=10; sel_data.GeometricData.radius:=0;
sel_data.Offset.OffsetRelation:=ITEMCOORD_DIR;
sel_data.Offset.OffsetPose.trans.x:=0;
sel_data.Offset.OffsetPose.trans.y:=0;
sel_data.Offset.OffsetPose.trans.z:=0;
sel_data.Offset.OffsetPose.rot.q1:=1;
sel_data.Offset.OffsetPose.rot.q2:=0;
```

Continues on next page
Continued
In this example, we shuffle items off a conveyor belt without touching surrounding objects. The shuffle movement is done perpendicular on the horizontal plane to the right side of the conveyor and the manipulator motion is coordinated with the conveyor motion.

The sorting algorithm selects the item closest to the exit limit in x-direction and has no locking objects in its selection shape.

The selection shape is defined as a long box. The shape’s x-value is used to define the corridor width, the y-value must be more than half the width of the conveyor belt and the z-value must be greater than the largest difference in height among all items.

Set the y-value in the OffsetData to the negative y-value of the shape, the selection box will be moved out to the right.

As a result there must be a clear corridor to the right of every item before it is shuffled.

The algorithm will check both upwards and downwards the production flow for other items.

Use the check limit in the x-direction as a parameter to the GetItmTgt instruction, to define the starting point from where the first item will be shuffled. The instruction will try to shuffle the first item between the check and enter limits. This will also cause the selection algorithm to take all items between the check limit and the exit limit into consideration when checking for the nearest items. Therefore the distance between the check limit and the exit limit will be at least the diameter of the largest item.

In the illustration below, all items will be shuffled off to the right side of the conveyor belt. Because each item needs a clear zone, that is the shape of theShapeType,
the items will be shuffled off in the order 1 to 10 as numbered in the illustration below.

Example code

```rapid
PROC Pick(num Index)
VAR selectiondata y_sort;
VAR robtarget draw_target;
VAR num check_limit;

y_sort.ShapeType:=BOX;
y_sort.ConsiderType:=BitOr(ITEMS_TO_USE,ITEMS_BYPASS);
y_sort.GeometricData.x:=41;
y_sort.GeometricData.y:=160;
y_sort.GeometricData.z:=5;
y_sort.GeometricData.radius:=0;
y_sort.Offset.OffsetRelation:=FRAME_COORD_DIR;
```

Continued on next page
y_sort.Offset.OffsetPose.trans.x:=0;
y_sort.Offset.OffsetPose.trans.z:=0;
y_sort.Offset.OffsetPose.rot.q1:=1;
y_sort.Offset.OffsetPose.rot.q2:=0;
y_sort.Offset.OffsetPose.rot.q3:=0;
y_sort.Offset.OffsetPose.rot.q4:=0;
check_limit:=150;

WObjPick:=ItmSrcData{Index}.Wobj;
GetItmTgt ItmSrcData{Index}.ItemSource,PickTarget
    \Limit:=check_limit\Selection:= y_sort;
Triggl\Conc, RelTool(PickTarget.RobTgt, 0, 0,
    -ItmSrcData{Index}.OffsZ), MaxSpeed,
    ItmSrcData{Index}.VacuumAct1, z20, Gripper\WObj:=WObjPick;
MoveL\Conc, PickTarget.RobTgt, LowSpeed, z5
    \Inpos:=ItmSrcData{Index}.TrackPoint, Gripper
    \WObj:=WObjPick;
GripLoad ItemLoad;
draw_target:=PickTarget.RobTgt;
draw_target.trans.y:=-200;
draw_target.rot:=[0,1,0,0];
Triggl draw_target, LowSpeed, ItmSrcData{Index}.Ack, z20,
    Gripper\WObj:=WObjPick;
ENDPROC
5.5.8 Example: Indexed work area with predefined position

Description of example

In this example we use an indexed work area with predefined positions. When using predefined positions with the indexed work area, we must modify the configuration, that is the EIO.cfg file. We will cross connect the trigger and strobe signals because with predefined positions there is no system generating the strobe signal. Without the predefined positions, the trigger signal is sent to the vision system to acquire an image. The strobe is then sent back from the vision system to acknowledge that the image has been acquired.

This is an example setup for a line that is triggered externally by an I/O signal and the position source is a predefined positions type. We recommend defining unique signal names for all new signals when setting up a system that is much different from the standard system.

Two new signals are used in this line:

- The trigger signal, doTrigSignal.
- The strobe signal, diStrobeSignal.

Modify the signal configurations by adding the two signals.

```
EIO_SIGNAL:
-Name "doTrigSignal" -SignalType "DO" -Unit "PPASIM" -UnitMap "6" -Access "ALL"
-Name "diStrobeSignal" -SignalType "DI" -Unit "PPASIM" -UnitMap "6" -Access "ALL"
```

The trigger and strobe signals are cross connected since there is no vision system that can send back a strobe signal.

```
EIO_CROSS
-Res "diStrobeSignal" -Act1 "doTrigSignal"
```

The Position generator signal in this case is di1_1, which is connection 1 on the DSQC 328A:X3 board, see Circuit diagram - PickMaster Twin, 3HAC024480-020.

When the di1_1 goes high (by an external I/O signal) the trigger signal is pulsed. Since the trigger and strobe signals are cross connected, the strobe will be received immediately. An event will then be sent from the controller to the computer, which it is ready for new item positions and the predefined positions will then be sent to the controller. If a pattern is used, several positions are sent for every signal. In this example the robot execution signal is not used and was therefore removed.
5 RAPID reference

5.5.9 Example: Automatically generating new positions to indexed work area

Description of example

In this example we configure an indexed work area and the queue will automatically be refilled with new positions when it is empty.

The trigger and strobe signals are set up as in Example: Indexed work area with predefined position on page 393.

Instead of using an external input I/O signal, we will use a new simulated input I/O signal as position generator signal. This signal is set by a cross connected simulated output signal.

Two new signals are used in this line:

- The output position generator signal, doPosGenSignal.
- The input position generator signal, diPosGenSignal.

Modify the signal configurations by adding the two signals.

```
EIO_SIGNAL:
- Name "doPosGenSignal" - SignalType "DO" - Unit "PPASIM" - UnitMap "7" - Access "ALL"
- Name "diPosGenSignal" - SignalType "DI" - Unit "PPASIM" - UnitMap "7" - Access "ALL"
```

The position generator signals are cross connected.

```
EIO_CROSS
- Res "diPosGenSignal" - Act1 "doPosGenSignal"
```

diPosGenSignal is defined in the line as the position generator signal and doPosGenSignal is defined as queue idle signal.

When the queue goes empty the queue idle signal doPosGenSignal will go high. This cross connection will make diPosGenSignal go high and new positions will be pushed to the queue according to the earlier described principles.
5.5.10 Example: Item buffer

Description of example

In this example we use item buffer. The items are put on the predefined buffer position.

1. Pick item from pick work area
2. Place on buffer position

Note

The buffer position must be out of the range in X axis and Y axis within the conveyor. Otherwise the robot will place the item on the conveyor directly rather than on the buffer position.

Example code

Define the buffer position in the RAPID program. See the following example code.

```rapid
!***********************************************************
! Global BUFFER Variables
! Robtarget BufferPos must be defined in wobj0
!***********************************************************
TASK PERS robtarget BufferPos{3}:=
    [[-200,-10,-1084],[0,-1,0,0],[0,0,0,0],[0,0,0,0,0]],
    [[0,-10,-1084],[0,-1,0,0],[0,0,0,0],[0,0,0,0,0]],
    [[200,-10,-1084],[0,-1,0,0],[0,0,0,0],[0,0,0,0,0]];
VAR num BufferMax{3}:=3,3,3; !Number of items in a buffer
VAR num BufferPitch{3}:=50,50,50; !distance between buffer positions
VAR num DropAction:=0; !What to do if an item can not be used.
    0=Ack 1=Nack 2=Skip
VAR num BufferZ{3}:=[0,0,0]; !Buffer Z-adjustments
VAR num InFlowEnter:=-250; !Set to same as pickarea Enter limit
VAR num InFlowExit:=250; !Set to little before pickarea Exit limit
VAR num OutFlowEnter:=-250; !Set to same as placearea Enter limit
VAR num OutFlowExit:=250; !Set to little before placearea Exit limit
VAR num BufferX{3};
VAR num BufferY{3};
VAR num BufferIndex{3}:=[0,0,0];
VAR num IType:=0;
VAR num Picked:=0;

Modify the PickPlaceSeq routine to perform Pick routine and Place routine on the buffer position. See the following example code.

!***********************************************************
!
!Procedure PickPlaceSeq
!
!The Pick and Place sequence.
```

Continues on next page
! Edit this routine to modify from which work areas to pick and place.
! Needs to be changed if more than one pick work area is used.
! Needs to be changed if more than one place work area is used.
!
!***********************************************************
PROC PickPlaceSeq()
VAR num GQL:=0;
Picked:=0;
WHILE Picked=0 DO
IType:=0;
!=== PickPosAvailable ? ===
GQL:=GetQueueLevel (ItmSrcData{PickWorkArea{1}}.ItemSource
\MinLimit:=InFlowEnter \MaxLimit:=InFlowExit);
IF GQL>0 THEN
NextItmTgtType ItmSrcData{PickWorkArea{1}}.ItemSource, IType;
!=== Matching Item on outfeeder ? ===
GQL:=GetQueueLevel (ItmSrcData{PlaceWorkArea{1}}.ItemSource
\ItmType:=IType \MinLimit:=OutFlowEnter
\MaxLimit:=OutFlowExit);
IF GQL>0 THEN
Picked:=1;
ELSE
!=== Empty pos in buffer? ===
IF BufferIndex{IType}<BufferMax{IType} THEN
Picked:=3;
ELSE
!=== What to do with the item on the infeeder? ===
GetItmTgt
  ItmSrcData{PickWorkArea{1}}.ItemSource,PickTarget\ItemType:=IType;
TEST DropAction
CASE 0:
  AckItmTgt ItmSrcData{PickWorkArea{1}}.ItemSource,PickTarget,TRUE;
CASE 1:
  AckItmTgt ItmSrcData{PickWorkArea{1}}.ItemSource,PickTarget,FALSE;
CASE 2:
  AckItmTgt
    ItmSrcData{PickWorkArea{1}}.ItemSource,PickTarget,FALSE\Skip:=TRUE;
ENDTEST
Picked:=0;
ENDIF
ENDIF
ELSE
GQL:=GetQueueLevel (ItmSrcData{PlaceWorkArea{1}}.ItemSource
\MinLimit:=OutFlowEnter \MaxLimit:=OutFlowExit);
IF GQL>0 THEN
NextItmTgtType ItmSrcData{PlaceWorkArea{1}}.ItemSource, IType;
!=== Matching Item on infeeder ? ===
GQL:=GetQueueLevel (ItmSrcData{PlaceWorkArea{1}}.ItemSource
\ItmType:=IType \MinLimit:=InFlowEnter \MaxLimit:=InFlowExit);
IF GQL>0 THEN
Picked:=1;
ELSE
!=== Matching pos in buffer? ===
IF BufferIndex{IType}>0 THEN
Picked:=2;
ELSE
Picked:=0;
ENDIF
ENDIF
ELSE
Picked:=0;
ENDIF
ENDIF

TEST Picked
CASE 0:
!=== No pick, no place ===
WaitTime 0.1;
CASE 1:
!=== pick infeed, place outfeed ===
Pick PickWorkArea{1},IType;
Place PlaceWorkArea{1},IType;
CASE 2:
!=== pick buffer, place outfeed ===
PickBuffer PickWorkArea{1},IType;
Place PlaceWorkArea{1},IType;
CASE 3:
!=== pick infeed, place buffer ===
Pick PickWorkArea{1},IType;
PlaceBuffer PlaceWorkArea{1},IType;
ENDTEST
ENDWHILE
ENDPROC

Edit this routine to modify how the robot shall execute the pick and place movements.

!***********************************************************
! Procedure PickBuffer
! Executes a pick movement
!***********************************************************
PROC PickBuffer(num Index,num TypeNr)
VAR num zboffs;
BufferX{TypeNr}:=(BufferIndex{TypeNr}-1)*BufferPitch{TypeNr};
WObjPick:=Wobj0;
zboffs:=BufferZ{TypeNr};
TriggL\Conc,Offs(RelTool(BufferPos{TypeNr},0,0,-ItmSrcData{Index}.OffsZ),
BufferX{TypeNr},0,0),MaxSpeed,ItmSrcData{Index}.VacuumAct1,z20,
PickAct1\WObj:=WobjPick;
TriggL\Conc,Offs(BufferPos{TypeNr},BufferX{TypeNr},0,zboffs),
LowSpeed,ItmSrcData{Index}.SimAttach1,z5\Inpos:=ItmSrcData{Index}.TrackPoint,
PickAct1\WObj:=WObjPick;
GripLoad ItemLoad;
MoveL Offs(RelTool(BufferPos{TypeNr},0,0,-ItmSrcData{Index}.OffsZ),
BufferX{TypeNr}, 0,0), LowSpeed,z20,PickAct1\WObj:=WObjPick;
Decr BufferIndex{TypeNr};
ENDPROC

***********************************************************
!
! Procedure PlaceBuffer
!
! Executes a place movement
!
******************************************************************************
PROC PlaceBuffer(num Index,num TypeNr)
VAR num zboffs;
Incr BufferIndex{TypeNr};
BufferX{TypeNr}:=(BufferIndex{TypeNr}-1)*BufferPitch{TypeNr};
WObjPlace:=Wobj0;
zboffs:=BufferZ{TypeNr};
MoveL\Conc,Offs(RelTool(BufferPos{TypeNr},0,0,-ItmSrcData{Index}.OffsZ),
BufferX{TypeNr}, 0,0), MaxSpeed,z20,PickAct1\WObj:=WObjPlace;
TriggL\Conc,Offs(BufferPos{TypeNr},BufferX{TypeNr},0,zboffs),LowSpeed,
ItmSrcData{Index}.VacuumRev1\T2:=ItmSrcData{Index}.VacuumOff1\T3:=
ItmSrcData{Index}.SimDetach1,z5\Inpos:=ItmSrcData{Index}.TrackPoint,
PickAct1\WObj:=WObjPlace;
GripLoad load0;
MoveL Offs(RelTool(BufferPos{TypeNr},0,0,-ItmSrcData{Index}.OffsZ),
BufferX{TypeNr}, 0,0),LowSpeed,z20,PickAct1\WObj:=WObjPlace;
ENDPROC
6 Troubleshooting

6.1 Introduction to troubleshooting

Troubleshooting

This chapter describes some of the most common troubles known when installing, configuring, or running PickMaster PowerPac.

A fault in the robot system first appears as a symptom, which can be:

- An event log message that can be viewed using PickMaster PowerPac, FlexPendant, RobotStudio, or Windows Event Viewer.
- The system is performing poorly or displaying mechanical disturbances.
- The system can not be started or displays irrational behavior during start.
- Indications on the hardware, such as LEDs.
- Other types of symptoms. The robot system is complex and has a large number of functions and function combinations.

Related information

Generic troubleshooting and all error messages in the robot system are listed in Operating manual - Troubleshooting IRC5.

Administering the log on page 401.
6 Troubleshooting

6.2 Safety during troubleshooting

General

All normal service work; installation, maintenance and repair work, is usually performed with all electrical, pneumatic and hydraulic power switched off. All manipulator movements are usually prevented by mechanical stops etc.

Troubleshooting work differs from this. While troubleshooting, all or any power may be switched on, the manipulator movement may be controlled manually from the FlexPendant, by a locally running robot program or by a PLC to which the system may be connected.

Dangers during troubleshooting

This implies that special considerations unconditionally must be taken when troubleshooting:

- All electrical parts must be considered as live.
- The manipulator must at all times be expected to perform any movement.
- Since safety circuits may be disconnected or strapped to enable normally prohibited functions, the system must be expected to perform accordingly.

⚠️ DANGER

Troubleshooting on the controller while powered on must be performed by personnel trained by ABB or by ABB field engineers.
6.3 Administering the log

The log

The log messages that are displayed in the log area of PickMaster PowerPac.

Administering the log

Use this procedure to administer the event log.

1. Click the LOG ribbon
2. If you need to view the event log in the PickMaster PowerPac, select Viewer. The event log will show up.
3. If you need to view the event log without the PickMaster PowerPac, select Save. The event log will be saved as .xlsx file.
6 Troubleshooting

6.4 Fault symptoms or errors

6.4.1 Warnings 4326 - 4329

Verification actions

The following are the general verification actions for the warning 4326, 4327, 4328, and 4329. For more detailed explanation, see Warning 4326 on page 403, Warning 4327 on page 403, Warning 4328 and 4329 received together on page 404, Warning 4328 received without 4329 on page 405, and Warning 4329 received without 4328 on page 405.

Action 1
Check the selection of signals for trigger and strobe in the work area configuration of the PickMaster line. Check that the I/O configurations of these signals correspond to the wiring.

Action 2
Check all the trig/strobe wiring. Check if the trig and strobe cables are mixed up. Make sure that the cables are shielded, properly attached and grounded the right way. There should be no current in the shield. Make sure that sources for 24 volt are not mixed. The controller system parameter SyncSeparation (Topic: I/O, Type: Fieldbus Command, Name: CNVX) can be modified to filter strobe input events from a camera or sensor.

Action 3
Check all the LAN cables on the robot network. Make sure that the cables are shielded and properly attached. Check that the right IP address, default gateway, and subnet mask is defined (on both PC and robot controller). Note that all three values must be defined even if there is only one computer and one robot controller on the network. For more information, see Configuring networks on page 54.

Action 4
See Configuring networks on page 54.

Action 5
Check that the IP address (goto File and click Options in RRT) in the field "Controller Network Adapter" is the address of the network interface card in the PC that communicates with the robot controller. Check if time sync service has trouble to connect to controller. Stop the service for 30 seconds and then restart it again. Check that there are no firewalls active that are affecting the time synchronization services.

Action 6
Reduce the trigger frequency Sometimes the trigger distance is very short causing the system to trigger much more often than it can handle. How often a trigger can be handled depends on how complicated the models are that are used on the system. Sometimes the frequent triggering can be caused by faulty trigger/strobe wiring or electrical noise.

Continues on next page
Action 7

Some switches are buffering data that needs to be present. This buffering time might be too long. Try to switch to a simple hub or to decrease this buffer time. Make sure that you have the newest software running on the hub/switch. Make sure that there are no infinite loops in the RAPID code because it will affect the robot network communication.

Action 8

Debug the implementation of the external sensor.

Action 9

For external sensors there might be a small constant delay between the strobe pulses and the recording of time stamps (For example, if the trigger signal is cross connected with the strobe). Modify the Position Source parameter Synchronization tune to modify all time stamps sent to PickMaster with a constant time value.

Warning 4326

For verification actions, see the preceding section.

Error description:

4326 Item positions lost on %s due to missing strobe. See Application manual.

Probable causes:

The following table provides the probable causes of the warning 4326:

<table>
<thead>
<tr>
<th>Probable cause</th>
<th>Verification actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>If work area is conveyor:</td>
<td></td>
</tr>
<tr>
<td>The conveyor board does not receive any strobe pulses on the start input.</td>
<td>Action 1 on page 402,</td>
</tr>
<tr>
<td></td>
<td>Action 2 on page 402</td>
</tr>
<tr>
<td>The strobe signal is not configured as cXNewObjStrobe.</td>
<td>Action 1 on page 402</td>
</tr>
<tr>
<td>PickMaster has no connection with the robot controller.</td>
<td>Action 3 on page 402</td>
</tr>
<tr>
<td>If work area is indexed:</td>
<td></td>
</tr>
<tr>
<td>The configured strobe signal does not receive a strobe pulses.</td>
<td>Action 1 on page 402,</td>
</tr>
<tr>
<td></td>
<td>Action 2 on page 402</td>
</tr>
<tr>
<td>PickMaster has no connection with the robot controller.</td>
<td>Action 3 on page 402</td>
</tr>
</tbody>
</table>

Warning 4327

Error description:

4327 Expected item positions missing from %s. See Application manual.

Probable causes:

The following table provides the probable causes of the warning 4327:

<table>
<thead>
<tr>
<th>Probable cause</th>
<th>Verification actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>If source type is camera:</td>
<td></td>
</tr>
<tr>
<td>The camera does not receive trigger pulses.</td>
<td>Action 1 on page 402,</td>
</tr>
<tr>
<td></td>
<td>Action 2 on page 402</td>
</tr>
<tr>
<td>PickMaster has no connection with the camera.</td>
<td>Action 4 on page 402</td>
</tr>
</tbody>
</table>

Continues on next page
### Probable cause and Verification actions

<table>
<thead>
<tr>
<th>Probable cause</th>
<th>Verification actions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>If source type is external sensor:</strong></td>
<td></td>
</tr>
<tr>
<td>The external sensor does not receive any trigger pulses.</td>
<td>Action 1 on page 402,</td>
</tr>
<tr>
<td></td>
<td>Action 2 on page 402</td>
</tr>
<tr>
<td>The external sensor does not send any positions to PickMaster.</td>
<td>Action 8 on page 403</td>
</tr>
<tr>
<td><strong>If source type is external sensor:</strong></td>
<td></td>
</tr>
<tr>
<td>The external sensor does not receive any trigger pulses.</td>
<td>Action 1 on page 402,</td>
</tr>
<tr>
<td></td>
<td>Action 2 on page 402</td>
</tr>
<tr>
<td>The external sensor does not send any positions to PickMaster.</td>
<td>Action 8 on page 403</td>
</tr>
<tr>
<td><strong>If source type is predefined and work area is conveyor:</strong></td>
<td></td>
</tr>
<tr>
<td>The conveyor board does not receive any strobe pulses on the start input.</td>
<td>Action 1 on page 402,</td>
</tr>
<tr>
<td></td>
<td>Action 2 on page 402</td>
</tr>
<tr>
<td>The strobe signal is not configured as cXNewObjStrobe.</td>
<td>Action 8 on page 403</td>
</tr>
<tr>
<td>PickMaster has no connection with the robot controller</td>
<td>Action 3 on page 402</td>
</tr>
<tr>
<td><strong>If source type is predefined and work area is indexed:</strong></td>
<td></td>
</tr>
<tr>
<td>The configured strobe signal does not receive any strobe pulses.</td>
<td>Action 1 on page 402,</td>
</tr>
<tr>
<td></td>
<td>Action 2 on page 402</td>
</tr>
<tr>
<td>PickMaster has no connection with the robot controller</td>
<td>Action 3 on page 402</td>
</tr>
</tbody>
</table>

### Warning 4328 and 4329 received together

**Error description:**

Typically, a pair of 4328 and 4329 is received for one, several or every trigger/strobe related to a work area.

4328 Trigger/strobe time mismatch (%.1f s). Item positions from %s to %s lost. See Application manual.

4329 Trigger/strobe time mismatch (%.1f s). Strobe from %s was ignored. See Application manual.

**Probable causes:**

The following table provides the probable causes of the warning 4328 and 4329:

<table>
<thead>
<tr>
<th>Probable cause</th>
<th>Verification actions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In order of probability:</strong></td>
<td></td>
</tr>
<tr>
<td>The time synchronisation between controllers and PickMaster is not working.</td>
<td>Action 6 on page 402</td>
</tr>
<tr>
<td>The trigger frequency is set too high.</td>
<td>Action 5 on page 402</td>
</tr>
<tr>
<td>Low robot network performance</td>
<td>Action 7 on page 403</td>
</tr>
<tr>
<td>Low camera network performance</td>
<td>Action 4 on page 402</td>
</tr>
<tr>
<td><strong>Additional causes for external sensors:</strong></td>
<td></td>
</tr>
<tr>
<td>Time stamps are not enough synchronized with strobes.</td>
<td>Action 9 on page 403</td>
</tr>
<tr>
<td>The external sensor does not send positions with a correct time stamp..</td>
<td>Action 8 on page 403</td>
</tr>
</tbody>
</table>
Warning 4328 received without 4329

Error description:

4328 Trigger/strobe time mismatch (%.1f s). Item positions from %s to %s lost. See Application manual.

Probable causes:

The following table provides the probable causes of the warning 4328 and 4329:

<table>
<thead>
<tr>
<th>Probable cause</th>
<th>Verification actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The trigger signal is not stable.</td>
<td>Action 2 on page 402</td>
</tr>
</tbody>
</table>

Warning 4329 received without 4328

Error description:

4329 Trigger/strobe time mismatch (%.1f s). Strobe from %s was ignored. See Application manual.

Probable causes:

The following table provides the probable causes of the warning 4328 and 4329:

<table>
<thead>
<tr>
<th>Probable cause</th>
<th>Verification actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The strobe signal is not stable.</td>
<td>Action 2 on page 402</td>
</tr>
</tbody>
</table>
6 Troubleshooting

6.4.2 The camera does not take pictures

6.4.2 The camera does not take pictures

Error description

The camera does not take pictures.

Probable causes

There can be several causes why the camera does not take pictures. To check all the possible causes the following must be verified.

- Check that the trig cable is properly connected.
- Check that the camera cable is connected to the correct port.

If the camera is distance trigged, the encoder might not be recording any conveyor movement due to

- bad encoder connection or
- wrong conveyor selected in the work area.

If the camera is I/O trigged, the photo eye might not be sensing any part, due to:

- Wrong connection.
- Bad reflection.
6.4.3 Robot does not move

Error description
The camera is identifying objects, but the robot does not move.

Probable causes
There can be several causes why the robot does not move although the camera takes pictures properly. To check all the possible causes the following must be verified.

- To check that the strobe cable is connected, check the StartSig LED on the encoder board.
- Check the distribution in the Position Source.
- Check the AI c*Speed in the I/O list if any speed is detected. If not, check encoder signals.
- Check the AI c*Position in the I/O list if any position is tracked. If not, check the distribution in the Position Source.
- Check the direction of travel on the DI c*DirOfTravel.
- Monitor the signal Queue Idle, to see if the queue gets any positions.
- Monitor the Position Available signal, to see if the parts are detected.
6 Troubleshooting

6.4.4 Bad or varying position accuracy

## 6.4.4 Bad or varying position accuracy

**Error description**

The position accuracy is bad or varying.

**Probable causes**

There can be several causes why the position accuracy is bad or varying. To check all the possible causes the following must be verified.

- Verify that the *Counts Per Meter* calibration is accurate. Verify several times. Include verification in scheduled maintenance.
- Avoid drive shaft encoders, since belt slippage between roller and belt can vary.
- Check the camera calibration. Poor quality of calibration grid will give inaccurate calibration result.
- Check if there are differences between calibration paper height and product height.
- Check if there are parallax errors when identifying high products.
- Make sure that the camera is not mounted on robot frame because this can cause camera vibrations.
6.4.5 Positions are used twice

Error description
The robot uses every position twice.

Probable causes
There can be several causes why the robot uses every position twice. To check all the possible causes the following must be verified.

- If I/O trigged predefined positions or containers are used, set the SyncSeparation filter distance to avoid double and ghost triggers.
- If vision is used, increase the overlap and position filter.
- Clear the checkbox Same level only in the Position Source.

If a robot downstream in an ATC group tries to use an already used item, then the Work Area order in the Position Source is incorrect.
6 Troubleshooting

6.4.6 Problem with camera resolution in PickMaster

Error description

Camera image size decreases to lower resolution as compared to calibration image resolution.

Probable causes

There can be several causes why camera resolution is decreased. To check all the possible causes the following must be verified:

- Is the factory default configuration is active.
- There could be custom configuration activated. Verify if the custom configuration is having reduced ROI (region of interest).
6.4.7 The Image Dialog cannot show

Error description

When users try to use camera related functions (camera configuration, camera calibration, geometric model, blob model, inspection model, live video, detail vision), the specific image dialog shown below cannot be displayed. Sometimes a "pure virtual function call" error pops up.

![Image Dialog](image.png)
Probable causes

Windows firewall blocks VisionClient.exe, a camera function related software engine, in some networks. Users should check whether VisionClient is available within the network of the computer in the Allowed apps window. If not all network settings of VisionClient are selected, this issue can occur as shown in the following case.

Recommended actions

The following procedure is recommended to change the firewall settings manually:

1. Open Windows Defender Firewall.
2 Click Allow an app of feature through Windows Defender Firewall to open the Allowed apps window.

3 Click Change settings.
6 Troubleshooting

6.4.7 The Image Dialog cannot show

Continued

4 Find VisionClient in the list and check that all network checkboxes for all VisionClient or visionclient.exe apps are selected.

5 Click OK.
6.4.8 Robot fails to grip item when using camera on a circular conveyor

Error description
In a station with using camera(s) on a circular conveyor, the robot fails to grip item and error message of Failed to grip item by tool 'PickPlaceTool_1'. Ignoring... shows up in the log when run the production.

Probable causes
If Enable vision width is not enabled, the vision scope will include these items in the red circle showed in the following image. Then wrong position information will be sent to the robot, and the gripping error will occur.

When using camera(s) on a circular conveyor, Enable vision width is preferred to be enabled to limit the vision scope. Otherwise, this vision scope will cover items on the other side of circular conveyor.
Recommended actions

Select to enable **Enable vision width** in the camera setting view if the camera is used on a circular conveyor.

![Camera Setting View](image)
6.4.9 Robot fails to start when clicking on Start button after Arm check point limit error for an indexed work area

Error description
In a station with using indexed work areas, the robot fails to start directly when clicking on Unhold button in PackML or clicking on Start button after the error message of Arm check point limit... shows up in the log.

Probable causes
Index work area doesn’t support this scenario.

Recommended actions
Stop the robot first and then click the Start button of the robot or the Unhold button in PackML.
### 6.5 Error codes

#### Common error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| 4097       | Error | Undefined error  
Reason: The occurred error has not been given a correct error ID but the error message should explain the reason. |
| 4098       | Status| Information only |
| 4099       | Error | Command line options  
Reason: PickMaster was given an unknown command line option, e.g. /p, at startup. |
| 4100       | Error | Description: Unexpected error  
Reason: An unexpected error occurred in PickMaster. See the log message for more information. |
| 4101       | Error | XML parsing error  
Reason: There was a problem reading either a pmline or pmproj file. See the log message for further information about where in the file the error occurred. |
| 4197       | Error | The project has been upgraded to a later version and the file is marked as modified. The file needs to be saved to make changes permanent. |
| 4198       | Error | The line has been upgraded to a later version. If the line itself was opened it is marked as modified and needs to be saved. If a project was opened, the line should be opened and saved before continuing. |
| 4199       | Error | The project file has an invalid format. It was either created with a beta version of PickMaster or the file is corrupt. |
| 4200       | Error | The PickMaster program failed to access the Windows registry when writing or reading its configuration |
| 4202       | Warning | The project is not designed on the current line. When trying to open a project, there is already a project open that is built upon a different line.  
Reason: Only one line can be used at the same time.  
Solution: Close any open projects and try to open the project again. |
| 4203       | Error | Failed to load the corresponding line when opening a project. The line file may be corrupt |
| 4204       | Error | Failed to load a line. The file may be corrupt. |
| 4205       | Error | The imported line may need to be recalibrated  
Reason: If the imported line was designed with other cameras or lenses, the cameras as well as the robot’s base frame must be recalibrated. |
| 4206       | Error | The selected RIS plug-in could not be loaded at program startup. The file may be corrupt. |
| 4207       | Error | The selected RIS plug-in could not be found at program startup. |
| 4208       | Error | One of the previously available lines has been overwritten by another line. The old line will not show up as an available line and projects designed on that line cannot be used. |
### Error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4209</td>
<td>Error</td>
<td>The line file is invalid and cannot be opened.</td>
</tr>
<tr>
<td>4210</td>
<td>Error</td>
<td>Failed to load resources for the selected language. The default language (English) will be used instead.</td>
</tr>
<tr>
<td>4211</td>
<td>Status</td>
<td>A notification about the total number of picks done by a robot until the project was stopped.</td>
</tr>
<tr>
<td>4212</td>
<td>Error</td>
<td>Failed to remove the line file. The file must be removed manually.</td>
</tr>
<tr>
<td>4213</td>
<td>Warning</td>
<td>Failed to find the html help file for the selected language. Make sure the &quot;Application manual xxx.chm&quot; file is in the Documentation folder in the PickMaster folder.</td>
</tr>
<tr>
<td>4216</td>
<td>Error</td>
<td>An attempt to open a file not recognized by PickMaster.</td>
</tr>
<tr>
<td>4217</td>
<td>Error</td>
<td>No time synchronization service available. Reason: The PickMaster Time Synchronization Service might not be properly installed or not started.</td>
</tr>
<tr>
<td>4218</td>
<td>Warning</td>
<td>Two or more network adapters are configured on the same subnet: x.x.x Refer to the user guide and review the recommended network settings.</td>
</tr>
<tr>
<td>4297</td>
<td>Status</td>
<td>Attempt to start a project that is already running.</td>
</tr>
<tr>
<td>4298</td>
<td>Status</td>
<td>Attempt to stop a project that was not started.</td>
</tr>
<tr>
<td>4300</td>
<td>Error</td>
<td>A camera is currently in use by another project. Reason: When starting a project, one of the position sources is configured with a camera that is currently in use by another project. Solution: A camera can only be used in production in one project at the same time. Reconfigure one project or run them one at a time.</td>
</tr>
<tr>
<td>4301</td>
<td>Error</td>
<td>Failed to start project execution Reason: Internal error probably caused by out of memory. Solution: Try restarting the PickMaster program.</td>
</tr>
<tr>
<td>4302</td>
<td>Error</td>
<td>When starting a project, a vision defined position source has no camera defined. Solution: Either remove the position source or configure it with the camera to use.</td>
</tr>
<tr>
<td>4303</td>
<td>Error</td>
<td>When starting a project, a position source has no work area defined Solution: Either remove the position source or configure it with the work area to use</td>
</tr>
<tr>
<td>4304</td>
<td>Warning</td>
<td>When starting a project, a vision defined position source has no configured vision models. Solution: Either remove the position source or define which vision models to use.</td>
</tr>
<tr>
<td>4305</td>
<td>Error</td>
<td>When starting a project, a predefined position source has no object defined. Solution: Edit the position source and define the predefined object to use.</td>
</tr>
</tbody>
</table>
### 6.5 Error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4306</td>
<td>Status</td>
<td>A model was edited on a different camera than it was created on. Solution: Check that the correct camera is selected in the position source and retrain the model.</td>
</tr>
<tr>
<td>4307</td>
<td>Warning</td>
<td>A vision model was created on a camera that has not been calibrated. Solution: Open the corresponding line and calibrate the camera. Then retrain the model.</td>
</tr>
<tr>
<td>4308</td>
<td>Error</td>
<td>When running a project, a vision model found an object but could not find the item or container to refer to. Solution: Stop the project, remove the vision model in question and create a new one for the correct item.</td>
</tr>
<tr>
<td>4309</td>
<td>Warning</td>
<td>A container is incorrectly configured. Solution: Check the error message for more information.</td>
</tr>
<tr>
<td>4310</td>
<td>Status</td>
<td>Production was successfully started.</td>
</tr>
<tr>
<td>4311</td>
<td>Status</td>
<td>Production was successfully stopped.</td>
</tr>
<tr>
<td>4312</td>
<td>Warning</td>
<td>Indication that PickMaster is running on a demo license with limited production time. Reason: There is only a demo license installed. Solution: Request a fully qualified license to run projects for an unlimited time.</td>
</tr>
<tr>
<td>4313</td>
<td>Error</td>
<td>PickMaster is running on a demo license and the allowed production time is exceeded. Solution: Request a fully qualified license or restart the PickMaster program to be able to start a project again</td>
</tr>
<tr>
<td>4314</td>
<td>Error</td>
<td>Got scene information from an unknown work area.</td>
</tr>
<tr>
<td>4315</td>
<td>Status</td>
<td>The work area that triggers a Position Source has changed. This occurs at project startup or when the robot controller with the previous trigger work area has stopped.</td>
</tr>
<tr>
<td>4319</td>
<td>Warning</td>
<td>Received item acknowledgment from an unknown work area.</td>
</tr>
<tr>
<td>4320</td>
<td>Warning</td>
<td>A project that used load balancing has been upgraded and a work area order was generated. The work area order must be verified in the Position Source configuration dialog box.</td>
</tr>
<tr>
<td>4321</td>
<td>Warning</td>
<td>An item acknowledge was received from a work area but the corresponding item position could not be found. Following work areas will not be notified that an item position has already been accessed.</td>
</tr>
<tr>
<td>4326</td>
<td>Warning</td>
<td>Item positions lost on work area due to missing strobe. For more information, see Warnings 4326 - 4329 on page 402.</td>
</tr>
<tr>
<td>4327</td>
<td>Warning</td>
<td>Expected item positions missing from position source. For more information, see Warnings 4326 - 4329 on page 402.</td>
</tr>
<tr>
<td>4328</td>
<td>Warning</td>
<td>Trigger/strobe time mismatch. Item positions from position source to work area lost. For more information, see Warnings 4326 - 4329 on page 402.</td>
</tr>
<tr>
<td>4329</td>
<td>Warning</td>
<td>Trigger/strobe time mismatch. Strobe from work area was ignored. For more information, see Warnings 4326 - 4329 on page 402.</td>
</tr>
<tr>
<td>4396</td>
<td>Error</td>
<td>A COM error occurred in when using an External Sensor. The log message provides more information.</td>
</tr>
</tbody>
</table>

Continues on next page
### Error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4397</td>
<td>Error</td>
<td>An error occurred when calling a function on an External Sensor COM object. The log message provides more information.</td>
</tr>
<tr>
<td>4398</td>
<td>Error</td>
<td>When opening a project with an external position generator, its corresponding sensor could not be found in the used line.</td>
</tr>
<tr>
<td>4399</td>
<td>Error</td>
<td>An external sensor failed to start when the project was started. The position source will not be used during production.</td>
</tr>
<tr>
<td>4596</td>
<td>Error</td>
<td>General User Hook error. See description for more information.</td>
</tr>
<tr>
<td>4797</td>
<td>Error</td>
<td>General license error. See description for more information.</td>
</tr>
<tr>
<td>4798</td>
<td>Error</td>
<td>More cameras are used than allowed by the currently installed license.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Either remove cameras or request a new license.</td>
</tr>
<tr>
<td>4799</td>
<td>Error</td>
<td>More robot controllers are used than allowed by the currently installed license.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Either remove robot controllers or request a new license.</td>
</tr>
<tr>
<td>4800</td>
<td>Error</td>
<td>More cameras are using inspection vision models than allowed by the currently installed license.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Either remove inspection models or request a new license.</td>
</tr>
<tr>
<td>4804</td>
<td>Error</td>
<td>More robot controllers are using camera distribution than allowed by the currently installed license.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Either make sure not to use more camera distribution or request a new license.</td>
</tr>
<tr>
<td>4805</td>
<td>Error</td>
<td>Attempt to start a project with ATC without an appropriate license.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Request a new license including the ATC option or remove ATC from the project.</td>
</tr>
<tr>
<td>4806</td>
<td>Warning</td>
<td>The licence will expire in less than 14 days.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Request a new license.</td>
</tr>
<tr>
<td>4807</td>
<td>Error</td>
<td>More External Sensors are used than allowed by the currently installed license.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Either remove External Sensors or request a new license.</td>
</tr>
<tr>
<td>4808</td>
<td>Error</td>
<td>Attempt to start a project with conveyors without an appropriate license.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Request a new license including the ATC option or remove all conveyors from the project.</td>
</tr>
<tr>
<td>4809</td>
<td>Error</td>
<td>The network adapter (IP-address) not found.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Make sure that the specified network card is enabled and that the IP address of the card has not changed.</td>
</tr>
<tr>
<td>4810</td>
<td>Error</td>
<td>Access to Service denied.</td>
</tr>
<tr>
<td>4811</td>
<td>Error</td>
<td>Cannot access PickMaster Time Synchronization Service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: PickMaster Time Synchronization Service is not installed.</td>
</tr>
<tr>
<td>4812</td>
<td>Error</td>
<td>Cannot stop PickMaster Time Synchronization Service.</td>
</tr>
<tr>
<td>4813</td>
<td>Error</td>
<td>Cannot start PickMaster Time Synchronization Service.</td>
</tr>
</tbody>
</table>

Continued on next page
## 6 Troubleshooting

### 6.5 Error codes

Continued

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8193</td>
<td>Status</td>
<td>The robot is running.</td>
</tr>
<tr>
<td>8194</td>
<td>Status</td>
<td>The robot is stopped.</td>
</tr>
<tr>
<td>8195</td>
<td>Status</td>
<td>The robot is paused.</td>
</tr>
<tr>
<td>8196</td>
<td>Warning</td>
<td>Please set the robot in auto mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: The robot is started but the controller is not set to auto mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Switch the controller to auto mode.</td>
</tr>
<tr>
<td>8197</td>
<td>Warning</td>
<td>Please confirm auto mode (on the FlexPendant).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: The robot is started and is set to auto mode but the auto mode is not confirmed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Confirm the auto mode on the FlexPendant.</td>
</tr>
<tr>
<td>8198</td>
<td>Status</td>
<td>The robot is in auto mode.</td>
</tr>
<tr>
<td>8199</td>
<td>Error</td>
<td>Robot error X (where X is the robot error number).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: See the robot documentation for the specific error.</td>
</tr>
<tr>
<td>8200</td>
<td>Warning</td>
<td>Robot warning X (where X is the robot warning number).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: See the robot documentation for the specific warning.</td>
</tr>
<tr>
<td>8201</td>
<td>Warning</td>
<td>Robot program controller in unknown state.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: The robot was started but the program controller is in an unknown state.</td>
</tr>
<tr>
<td>8202</td>
<td>Warning</td>
<td>Guard stop</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: The robot has been stopped because a guard has been activated.</td>
</tr>
<tr>
<td>8203</td>
<td>Warning</td>
<td>Emergency stop</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: The robot has been stopped because of an activation of the emergency stop</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Remove the reason for the stop and reset the emergency stop. Restart the robot (can be done without stopping the project).</td>
</tr>
<tr>
<td>8204</td>
<td>Status</td>
<td>Rapid program stopped</td>
</tr>
<tr>
<td>8205</td>
<td>Status</td>
<td>Rapid program has been restarted</td>
</tr>
<tr>
<td>8209</td>
<td>Status</td>
<td>Robot controller is in system failure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: See event log on the controller for more information</td>
</tr>
<tr>
<td>8211</td>
<td>Error</td>
<td>Lost connection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: The computer lost the connection to the controller. The network connection can be down. The controller can be shut off or lost its power.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Solution: Make sure that the controller is on and has power supply. Also make sure that the network connection is working.</td>
</tr>
<tr>
<td>8212</td>
<td>Warning</td>
<td>A robot controller is used by another project</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: A robot controller may only be used by one project at a time</td>
</tr>
<tr>
<td>8213</td>
<td>Warning</td>
<td>Robot controller not in use and may not be accessed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reason: An attempt was made to access a robot controller that was not configured to be used in the project.</td>
</tr>
</tbody>
</table>
### Error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8293</td>
<td>Error</td>
<td>Failed to set motors on. Reason: PickMaster failed to set motors on. Some system state prevents PickMaster from setting the motors to on (e.g. emergency stop, guard stop etc.).</td>
</tr>
<tr>
<td>8294</td>
<td>Error</td>
<td>Failed to start the RAPID program.</td>
</tr>
<tr>
<td>8295</td>
<td>Error</td>
<td>Failed to prepare the RAPID program for start.</td>
</tr>
<tr>
<td>8297</td>
<td>Error</td>
<td>Failed to set the RAPID variable “RoutineName” to “ClearAll” Reason: The variable “RoutineName” is probably missing or is of the wrong type (should be a string type). Solution: Ensure that the variable exists and is of the string type.</td>
</tr>
<tr>
<td>8298</td>
<td>Error</td>
<td>Failed to get the robot controller states. Solution: Ensure that the controller is up and running OK. If not, reboot the controller.</td>
</tr>
<tr>
<td>8299</td>
<td>Error</td>
<td>Failed to get events from the robot controller. Solution: Ensure that the controller is up and running OK. If not, reboot the controller. Ensure that the correct network adapter is used for the specific controller in the line.</td>
</tr>
<tr>
<td>8300</td>
<td>Error</td>
<td>Failed to set the RAPID variable “StopProcess” to TRUE. Solution: Ensure that the RAPID variable “StopProcess” exists and is of type bool.</td>
</tr>
<tr>
<td>8302</td>
<td>Error</td>
<td>Failed to set the RAPID variable “RoutineName” to “Pick-Place”. Reason: The variable “RoutineName” is probably missing or is of the wrong type (should be a string type). Solution: Ensure that the variable exists and is of the string type.</td>
</tr>
<tr>
<td>8303</td>
<td>Internal Error</td>
<td>The system failed to apply a new work area tune because the work area ID does not exist.</td>
</tr>
<tr>
<td>8304</td>
<td>Internal Error</td>
<td>The system failed to apply new work area settings because the work area ID does not exist.</td>
</tr>
<tr>
<td>8305</td>
<td>Internal Error</td>
<td>The system failed to apply a new work area setting.</td>
</tr>
<tr>
<td>8306</td>
<td>Error</td>
<td>Failed to set DO signal “doSafeStop”. Solution: Verify that the signal exists and is correctly set-up.</td>
</tr>
<tr>
<td>8307</td>
<td>Error</td>
<td>Failed to connect to the controller. Solution: Verify that the network address (IP address) to the controller is correct. Verify that the network settings on the computer are correct. Verify that the correct network adapter is used (in the line) to connect to the robot controller.</td>
</tr>
<tr>
<td>8308</td>
<td>Error</td>
<td>Failed to write the IP address to the controller. Solution: Verify that the RAPID variable “RemoteIPNode” exists and is of the correct type (should be of the string type).</td>
</tr>
<tr>
<td>8309</td>
<td>Error</td>
<td>Failed to initiate events from the robot controller. Solution: Verify that the robot controller is up and running correctly. If not, reboot the controller.</td>
</tr>
<tr>
<td>8310</td>
<td>Error</td>
<td>Failed to get the robot controller states. Solution: Ensure that the controller is up and running OK. If not, reboot the controller.</td>
</tr>
</tbody>
</table>

Continues on next page
### Error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8313</td>
<td>Error</td>
<td>Failed to set the IO signal ppaExe. Solution: Ensure that the signal ppaExe exists and is set-up correctly.</td>
</tr>
<tr>
<td>8314</td>
<td>Error</td>
<td>Failed to set the RAPID variable “RoutineName” to “NewSource”. Reason: The variable “RoutineName” is probably missing or is of the wrong type (should be a string type). Solution: Ensure that the variable exists and is of the string type.</td>
</tr>
<tr>
<td>8315</td>
<td>Error</td>
<td>The system failed to apply the new robot speed.</td>
</tr>
<tr>
<td>8316</td>
<td>Error</td>
<td>Failed to set the IO signal doTune. Solution: Ensure that the signal doTune exists and is set-up correctly.</td>
</tr>
<tr>
<td>8317</td>
<td>Error</td>
<td>The system failed to apply a new work area tune. Solution: Verify that the following RAPID variables exist. Num SourceIndex Num TunePosX Num TunePosY Num TunePosZ</td>
</tr>
<tr>
<td>8318</td>
<td>Error</td>
<td>Failed to load the RAPID program. Solution: Verify that there are no errors in the RAPID program (otherwise it will fail to load).</td>
</tr>
<tr>
<td>8319</td>
<td>Error</td>
<td>Failed to download the RAPID program to the controller.</td>
</tr>
<tr>
<td>8320</td>
<td>Error</td>
<td>Failed to stop execution of the RAPID program.</td>
</tr>
<tr>
<td>8321</td>
<td>Error</td>
<td>Failed to delete the RAPID program.</td>
</tr>
<tr>
<td>8322</td>
<td>Error</td>
<td>Failed to reset emergency stop.</td>
</tr>
<tr>
<td>8323</td>
<td>Error</td>
<td>Failed to restart the RAPID program. Solution: Stop the project and restart it.</td>
</tr>
<tr>
<td>8324</td>
<td>Error</td>
<td>Failed to get local IP address. Reason: The network set-up is not correct (e.g. wrong IP settings, faulty network adapter configuration, etc.). Solution: Solve the local network problem on the computer.</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| 8325       | Error | Failed to init queues.  
Reason: PickMaster failed to initiate an item queue. The queue is initiated by setting several RAPID variables. Those variables must not be removed or changed. The variables are:  
String ItmSrcName  
String CnvName  
String NonCnvWobjName  
Num SourceType  
Num SourceIndex  
Num TunePosX  
Num TunePosY  
Num TunePosZ  
Num FollowTime  
Num Vtcp  
Num OffsZ  
Num VacActDelay  
Num VacRevDelay  
Solution: Ensure that all variables exist and are of the correct type (string or num etc.) in the RAPID program or in the PPA sys module (ppasys.sys). |
| 8326       | Error | Failed to synchronize the time on the robot controller with the PickMaster compute |
| 8327       | Error | There is no Rapid program defined for a robot controller when starting a project.  
Reason: Attempt to start a project without having configured which Rapid program to use for a robot controller.  
Solution: Select a Rapid program to use for the robot controller in question and restart the project. |
| 8337       | Error | Failed to flush item source queue (ItmSrcCnvxx). C0040403: No response from the controller.  
Reason: For large robots where working range is large, CPU takes more time for indexing it because of GetReachableTarget functionality.  
Solution: The accuracy of the release zone (indexed working range) associated with the function UseReachableTargets can be adjusted from 0% to 100% with a new process system parameter, Reach Zone Accuracy, in Type Conveyor. Default value is 100%. To make CPU load less make this value zero or very low. If the UseReachableTargets functionality is not used, it may be turned off by setting the Reach Zone Accuracy value to 0. |
| 8338       | Error | Not connected to controller.  
Reason: The communication with the controller could not be completed. |
| 8339       | Error | Unexpected error when using ABB Industrial Robot Communication Runtime to communicate with controller.  
Reason: See error log for more information. |
| 8340       | Error | Unexpected robot error.  
Reason: See error log for more information. |
| 8341       | Error | Failed to get write access to controller. |
## 6 Troubleshooting

### 6.5 Error codes

Continued

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8342</td>
<td>Error</td>
<td>Item source failed to send positions to the controller. No response from the controller.</td>
</tr>
<tr>
<td>8343</td>
<td>Error</td>
<td>The RobotWare version is later than the ABB Industrial Robot Communication Runtime on the PC. The Communication Runtime needs to be updated. Solution: If possible update PickMaster to the latest version. If this does not solve the problem or for some reason is not possible, update the ABB Industrial Robot Communication Runtime on the PC. The installation can be downloaded from the <a href="https://www.robostudio.com">RobotStudio Online Community</a>, where it is included in the <strong>Tools and Utilities</strong> package.</td>
</tr>
<tr>
<td>8345</td>
<td>Error</td>
<td>Failed to start program in Auto. Possible reason: The RW role setting 'Remote start/stop program in Auto' is not selected.</td>
</tr>
<tr>
<td>8393</td>
<td>Error</td>
<td>The motion server already exists as an instance (only one instance is allowed).</td>
</tr>
<tr>
<td>8394</td>
<td>Error</td>
<td>The robot ID already exists (IDs shall be unique).</td>
</tr>
<tr>
<td>8395</td>
<td>Error</td>
<td>No robot defined with that ID.</td>
</tr>
<tr>
<td>8396</td>
<td>Error</td>
<td>Work areas still exist. The conveyor cannot be removed before the work areas are removed. Solution: Remove all work areas for the conveyor.</td>
</tr>
<tr>
<td>8397</td>
<td>Error</td>
<td>A work area with that ID already exists. (All IDs shall be unique).</td>
</tr>
<tr>
<td>8398</td>
<td>Error</td>
<td>No work area with that ID exists. An operation was executed on a non-existing work area. The work area has probably been removed.</td>
</tr>
<tr>
<td>8399</td>
<td>Error</td>
<td>Settings on the work area failed due to a bad work area ID.</td>
</tr>
<tr>
<td>8400</td>
<td>Error</td>
<td>The system failed to apply new work area settings due to a bad work area ID.</td>
</tr>
<tr>
<td>8401</td>
<td>Error</td>
<td>The system failed to set a new work area because the work area ID does not exist.</td>
</tr>
<tr>
<td>8402</td>
<td>Error</td>
<td>The system failed to apply a new work area tune because the work area ID does not exist.</td>
</tr>
<tr>
<td>8403</td>
<td>Error</td>
<td>The system failed to apply new robot settings because the robot ID does not exist.</td>
</tr>
<tr>
<td>8404</td>
<td>Error</td>
<td>The system failed to set new robot settings because the robot ID does not exist.</td>
</tr>
<tr>
<td>8406</td>
<td>Error</td>
<td>The system failed to set a new robot speed because the robot ID does not exist.</td>
</tr>
<tr>
<td>8407</td>
<td>Error</td>
<td>Failed to update the work area due to wrong work area type (indexed work area / conveyor work area).</td>
</tr>
<tr>
<td>8408</td>
<td>Warning</td>
<td>There are no work areas defined for the robot. Solution: Define work areas and set up position sources for the work areas for the robot before project start</td>
</tr>
<tr>
<td>8418</td>
<td>Status</td>
<td>Downloading elog files from controller. Reason: If elog files are missing at production start they will be downloaded automatically.</td>
</tr>
</tbody>
</table>

Continues on next page
### Vision error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12298</td>
<td>Status</td>
<td>There is no frame grabber/Gigabit Ethernet camera installed</td>
</tr>
<tr>
<td>12299</td>
<td>Internal Error</td>
<td>Could not find the camera in question in the vision server.</td>
</tr>
<tr>
<td>12300</td>
<td>Internal Error</td>
<td>Could not find the vision model in question in the vision server.</td>
</tr>
<tr>
<td>12301</td>
<td>Internal Error</td>
<td>The camera is locked.</td>
</tr>
<tr>
<td>12302</td>
<td>Internal Error</td>
<td>Attempt to create or load a camera that already exists.</td>
</tr>
<tr>
<td>12305</td>
<td>Error</td>
<td>The current frame grabber does not support the selected video format.</td>
</tr>
<tr>
<td>12306</td>
<td>Internal Error</td>
<td>Failed to create camera.</td>
</tr>
<tr>
<td>12307</td>
<td>Internal Error</td>
<td>The vision server could not find the acquired camera during runtime.</td>
</tr>
<tr>
<td>12308</td>
<td>Warning</td>
<td>A camera is triggered too fast. Reason: A camera was triggered before it was done analyzing the last image. As long as there only are a few messages there will be no lost images. Solution: Adjust the vision models on the camera to yield a faster analyzing time. Adjust models on other cameras since it is the system performance in total that should be improved. Lowering the conveyor speed will also reduce the problem, if applicable.</td>
</tr>
<tr>
<td>12309</td>
<td>Error</td>
<td>Failed to get an image from a camera when running a project. Reason: This error probably occurred because the system is too heavily loaded or the frame grabber is triggered way too fast. Solution: Verify system load and make sure the robot control-ler does not send faulty vision triggers.</td>
</tr>
<tr>
<td>12310</td>
<td>Internal Error</td>
<td>Failed to create a geometric model. Reason: See error message for more information.</td>
</tr>
<tr>
<td>12312</td>
<td>Internal Error</td>
<td>Attempt to access a camera port on a frame grabber that does not exist.</td>
</tr>
<tr>
<td>12313</td>
<td>Internal Error</td>
<td>There is no camera port on the frame grabber specified for the camera. Solution: Open the corresponding line and configure the camera with a camera port.</td>
</tr>
<tr>
<td>12315</td>
<td>Error</td>
<td>Could not initiate the camera at project start. Reason: The system is probably out of resources.</td>
</tr>
<tr>
<td>12316</td>
<td>Error</td>
<td>External model failed to analyze image. Reason: See log message for more information</td>
</tr>
<tr>
<td>12317</td>
<td>Error</td>
<td>Failed to initiate external model at project start. Reason: See log message for more information</td>
</tr>
<tr>
<td>12318</td>
<td>Error</td>
<td>Failed to convert image to a format supported by external vision model.</td>
</tr>
</tbody>
</table>
6 Troubleshooting

6.5 Error codes

Continued

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12319</td>
<td>Error</td>
<td>External model failed to inspect image. Reason: See log message for more information</td>
</tr>
<tr>
<td>12321</td>
<td>Error</td>
<td>When the line was opened, more than one camera was defined to use the same port on the same frame grabber. Only one camera can be configured to use a single camera port and hence the other cameras were reset and must be configured again.</td>
</tr>
<tr>
<td>12322</td>
<td>Error</td>
<td>When the line was opened, a camera was defined on a frame grabber that was not available. The camera was reset and must be configured again.</td>
</tr>
<tr>
<td>12323</td>
<td>Error</td>
<td>Could not initiate the camera. More information is provided in the log message.</td>
</tr>
<tr>
<td>12324</td>
<td>Error</td>
<td>Failed to save camera configuration. More information is provided in the log message</td>
</tr>
<tr>
<td>12325</td>
<td>Error</td>
<td>Failed to load camera configuration. More information is provided in the log message.</td>
</tr>
<tr>
<td>12326</td>
<td>Error</td>
<td>Failed to load vision model configuration. More information is provided in the log message.</td>
</tr>
<tr>
<td>12329</td>
<td>Warning</td>
<td>Failed to communicate with Gigabit Ethernet camera. Reason: Bad Ethernet connection or excessive Ethernet communication.</td>
</tr>
<tr>
<td>12330</td>
<td>Warning</td>
<td>Images are triggered too frequently. Solution: Adjust vision models to be less time consuming, or decrease trigger frequency.</td>
</tr>
<tr>
<td>12331</td>
<td>Warning</td>
<td>Connection to camera is lost, attempting to reconnect. Reason: Ethernet cable or power cable has been disconnected.</td>
</tr>
<tr>
<td>12332</td>
<td>Warning</td>
<td>Image Buffer Full. More information is provided in the log message.</td>
</tr>
<tr>
<td>12333</td>
<td>Warning</td>
<td>A Gigabit Ethernet camera was found, but no such license was detected. Reason: No USB stick with vision license is inserted in the PC.</td>
</tr>
<tr>
<td>12334</td>
<td>Warning</td>
<td>A license for Gigabit Ethernet vision was detected, but no such camera was found. Reason: Camera is not connected, not turned on, or has an invalid IP-address.</td>
</tr>
<tr>
<td>12337</td>
<td>Warning</td>
<td>Failed to read parameter from camera. Reason: Check if the appropriate Cognex Drivers are installed. If the problem persists, check network connections.</td>
</tr>
<tr>
<td>12341</td>
<td>Status</td>
<td>Cognex USB License dongle is attached.</td>
</tr>
<tr>
<td>12342</td>
<td>Warning</td>
<td>Cognex USB License dongle is removed.</td>
</tr>
</tbody>
</table>

User script error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>41989</td>
<td>Message</td>
<td>The execution of {%s} in {%s} timed out.</td>
</tr>
<tr>
<td>41990</td>
<td>Message</td>
<td>The module of {%s} in {%s} load failed.</td>
</tr>
<tr>
<td>Error code</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>41991</td>
<td>Message</td>
<td>The interface of {%s} in {%s} load failed.</td>
</tr>
<tr>
<td>41992</td>
<td>Message</td>
<td>The return of {%s} in {%s} is not correct.</td>
</tr>
<tr>
<td>41993</td>
<td>Message</td>
<td>The returned data structure of {%s} in {%s} is incorrect.</td>
</tr>
<tr>
<td>41995</td>
<td>Message</td>
<td>The element of {%s} was not found in returned data of {%s} in {%s}.</td>
</tr>
<tr>
<td>41996</td>
<td>Message</td>
<td>Position at [{%.1f}, {%.1f}] discarded due to unknown object Id. It is from {%s} in {%s}.</td>
</tr>
<tr>
<td>41997</td>
<td>Message</td>
<td>The element type of {%s} is incorrect in returned data of {%s} in {%s}.</td>
</tr>
<tr>
<td>42003</td>
<td>Message</td>
<td>Failed to obtain the documents folder path when {%s} was executed in {%s}, please check.</td>
</tr>
</tbody>
</table>
This page is intentionally left blank
ABB spare parts are categorized into two levels, L1 and L2. Always check the part level before conducting a service work on a spare part.

- **L1 spare parts**
  The L1 parts can be replaced in the field. The maintenance and replacement instructions given in the related product manuals must be strictly followed. If there are any problems, contact your local ABB for support.

- **L2 spare parts**
  To replace the L2 parts require specialized training and might need special tools. Only ABB field service personnel or qualified personnel trained by ABB can replace L2 parts.
## 7.1 Licenses

<table>
<thead>
<tr>
<th>Spare part number</th>
<th>Description</th>
<th>Type</th>
<th>Spare part level</th>
</tr>
</thead>
<tbody>
<tr>
<td>3HAC072144-001</td>
<td>PickMaster Runtime license</td>
<td></td>
<td>L1</td>
</tr>
</tbody>
</table>
7.2 Camera parts

Spare part - PickMaster camera

<table>
<thead>
<tr>
<th>Spare part number</th>
<th>Description</th>
<th>Type</th>
<th>Spare part level</th>
</tr>
</thead>
<tbody>
<tr>
<td>3HAC072140-001</td>
<td>PickMaster camera</td>
<td>DSQC1066</td>
<td>L1</td>
</tr>
</tbody>
</table>

xx1900001574
The Basler acA1440-73gc GigE camera with the Sony IMX273 CMOS sensor delivers 73 frames per second at 1.6 MP resolution.

For more details on the camera’s installation, see the documentation on the Basler Ace website, Basler Ace.
### Spare part - PickMaster cam I/O cable

<table>
<thead>
<tr>
<th>Spare part number</th>
<th>Description</th>
<th>Type</th>
<th>Spare part level</th>
</tr>
</thead>
<tbody>
<tr>
<td>3HAC072141-001</td>
<td>PickMaster cam I/O cable</td>
<td></td>
<td>L1</td>
</tr>
</tbody>
</table>

**Power-I/O Cable HRS 6p/open, twisted, 10 m - IOs / Power Cables**

Cable for power supply and trigger of opto coupled I/Os of Basler ace GigE cameras at a length of 10 meters.

The cable has an HRS 6-pin connector on the camera side. The other end is open so that the cable can be shortened to match individual requirements.

**Wiring information:**

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Wire Color</th>
<th>Ace GigE (without GPIO)</th>
<th>Ace GigE (with GPIO)</th>
<th>Aviator CL runner</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Brown</td>
<td>Camera Power</td>
<td>Camera Power</td>
<td>Camera Power</td>
</tr>
<tr>
<td>2</td>
<td>Pink</td>
<td>Opto-isolated IN (Line1)</td>
<td>Opto-isolated IN (Line1)</td>
<td>Camera Power</td>
</tr>
<tr>
<td>3</td>
<td>Green</td>
<td>Not connected</td>
<td>GPIO (Line3)</td>
<td>Not connected</td>
</tr>
<tr>
<td>4</td>
<td>Yellow</td>
<td>Opto-isolated OUT (Out1)</td>
<td>Opto-isolated OUT (Out1)</td>
<td>Not connected</td>
</tr>
<tr>
<td>5</td>
<td>Gray</td>
<td>Opto-isolated I/O Ground</td>
<td>Opto-isolated I/O Ground</td>
<td>Camera Power Groud</td>
</tr>
<tr>
<td>6</td>
<td>White</td>
<td>Camera Power Ground</td>
<td>Camera Power and GPIO Ground</td>
<td>Camera Power Groud</td>
</tr>
</tbody>
</table>

Continues on next page
7 Spare parts

7.2 Camera parts

Continued

Spare part - PickMaster cam com cable

<table>
<thead>
<tr>
<th>Spare part number</th>
<th>Description</th>
<th>Type</th>
<th>Spare part level</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>PickMaster cam com cable</td>
<td>L1</td>
<td></td>
</tr>
</tbody>
</table>

Cable GigE Cat 6, S/STP, 1x screw lock horizontal, DrC, 20 m
GigE cable for data transmission with RJ-45 plug with horizontal locking screws on the camera side at a length of 20 meter.
The twisted, shielded cable has an RJ-45 click-lock plug on the host side and is suitable for drag chain applications.

Spare part - Camera mount adapter

<table>
<thead>
<tr>
<th>Spare part number</th>
<th>Description</th>
<th>Type</th>
<th>Spare part level</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Camera mount adapter</td>
<td>L1</td>
<td></td>
</tr>
</tbody>
</table>

Camera mount for Basler ace cameras.
For mounting the camera onto tripod threads.
## 7.3 USB dongle parts

<table>
<thead>
<tr>
<th>Spare part number</th>
<th>Description</th>
<th>Type</th>
<th>Spare part level</th>
</tr>
</thead>
<tbody>
<tr>
<td>3HAC072139-001</td>
<td>USB dongle (small)(^i)</td>
<td>Vision license for up to 2 cameras</td>
<td>L1</td>
</tr>
<tr>
<td>3HAC073341-001</td>
<td>USB dongle (large)(^i)</td>
<td>Vision license for up to 10 cameras</td>
<td>L1</td>
</tr>
<tr>
<td>3HAC039556-001</td>
<td>USB dongle (sim)(^i)</td>
<td>Vision simulation license for up to 10 simulated cameras</td>
<td>L1</td>
</tr>
</tbody>
</table>

\(^i\) The dongle can be connected to any USB interface on host computer.

\(^i\) The dongle can be connected to any USB interface on client computer.
7 Spare parts

7.4 GigE Network card parts

<table>
<thead>
<tr>
<th>Spare part number</th>
<th>Description</th>
<th>Type</th>
<th>Spare part level</th>
</tr>
</thead>
<tbody>
<tr>
<td>3HAC078753-001</td>
<td>GigE network card</td>
<td>DSQC1083</td>
<td>L1</td>
</tr>
</tbody>
</table>

* Standard height, half length, PCI express card.
8 Circuit diagram

8.1 Circuit diagrams

Overview

The circuit diagrams are not included in this manual, but are available for registered users on myABB Business Portal, www.abb.com/myABB.

See the article numbers in the tables below.

Controllers

<table>
<thead>
<tr>
<th>Product</th>
<th>Article numbers for circuit diagrams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit diagram - OmniCore C30</td>
<td>3HAC059896-009, 3HAC063898-009, 3HAC072448-009</td>
</tr>
<tr>
<td>Circuit diagram - OmniCore C30 for IRB 14050</td>
<td></td>
</tr>
<tr>
<td>Circuit diagram - OmniCore C30 for CRB 15000</td>
<td></td>
</tr>
<tr>
<td>Circuit diagram - IRC5</td>
<td>3HAC024480-011</td>
</tr>
</tbody>
</table>

RobotWare options

<table>
<thead>
<tr>
<th>Product</th>
<th>Article numbers for circuit diagrams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit diagram - PickMaster Twin</td>
<td>3HAC024480-020</td>
</tr>
</tbody>
</table>
Index

A
AckItmTgt, instruction, 337
Adjuster Interface, 149–152, 155

B
blob
   concept, 297
   configuring, 297
   sub inspection model, 307

calibrating
   calibration papers, 277
   checkerboard, 276
   circle conveyor, 227
   line conveyor, 196
   white balance, 319
   calibration
   papers, 277
   result, 281
caliper
   sub inspection model, 310
cameras
   CCD, 27
   checkerboard, 276
   connecting, 49
   IP address, 56
   maximum number, 24
   no pictures taken, 406
   requirements, 27
   sensor chip, 27
   CCD, 27
   checkerboard calibration, 276
   circle
   calibrating, 227
   color filtering
   description, 317
   color space
   configuring, 320
   color spaces
   description, 317
   color vision
   configuring, 320
   description, 317
   prerequisites, 319
   color wheel, 317
   configurations
   system parameters, 60
   configuring
   blob, 297
   blob sub inspection model, 307
   caliper sub inspection model, 310
   color space, 320
   color vision, 320
   controller network, 54, 58
   external sensor, 181
   external sub inspection model, 312
   geometric sub inspection model, 307
   grip locations, 162
   height settings, 327
   histogram sub inspection model, 307
   inspection models, 304
   item height, 327
   networks, 54
   product height, 327
   vision network, 55
   connecting
   cameras, 49
   I/O, 52
   container
   definition, 17
   controller
   I/O boards, 52
   IP address, 54
   RobotWare, 60
   settings, 60
   controller network adapter, 42
   conveyor start/stop signal, 117, 174–175
   counts_per_meter
   system parameter, 197, 220, 228, 252

D
data types
   itmtgt, 360
   selectiondata, 363
   sortdata, 366
diodes, 52
Distribution Interface, 149
dll
   PMHook.dll, 149
   User Hook, 157
DSQC, 52

E
emergency stop, 315
emulation
   definition, 17
error logs, 399
Ethernet card
   IP address, 54
   external sensor
   configuring, 181
   external sub inspection model, 312
field of view, 27
firewall settings
   UDP, 40
FlushItmSrc, instruction, 339
focal length
   calculating, 28
   description, 27
FOV, 27
functions
   GetFlowCount ¶, 359
   GetQueueLevel, 356
   GetQueueTopLevel, 358

G
GetItmTgt, instruction, 340
GetQueueLevel, function, 356
GetQueueTopLevel, function, 358
ghost picking
   definition, 17
Gigabit Ethernet
   description, 24
   requirements, 24
   grayscale, 317
   grip locations
   configuring, 162

Application manual - PickMaster® Twin - PowerPac
3HAC080435-001 Revision: F
© Copyright 2023 ABB. All rights reserved.
Index

H
height method, 327
height settings, 326
histogram
  sub inspection model, 307
HookTester.exe, 149
HSI
description, 317
hue
description, 317
I
I/O
  configuring, 172
  connecting, 52
  connections, 52
  installed boards, 52
  predefined, 174–175
image windows, 285
Initialize Interface, 149
inspection II, 304
inspection models
  concept, 304
  configuring, 304
instructions
  AckItmTgt, 337
  FlushItmSrc, 339
  GetItmTgt, 340
  NextItmTgtType, 346
  QStartItmSrc, 348
  QStopItmSrc, 349
  ResetFlowCount ¶, 350
intensity
description, 317
IP address
  controller, 54
item
definition, 17
height, 326
itemTgt, data type, 360
L
languages, 42
lenses
  calculating, 28
  example, 29
  recommendation, 27
line
calibrating, 196
logs
  RIS, 42
status messages, 42
M
max_dist, 61
maximum distance, 61
monochrome, 317
MultiMove
  restarting, 315
multi-view, 327
N
networks
  cables, 55
  camera settings, 56
  configuring, 54
  configuring controller, 54, 58
  configuring vision, 55
  controller, 54
  Ethernet card, 54
  switches, 55
  typical settings, controller, 54
  typical settings, vision, 58
  vision prerequisites, 55
NextItmTgtType, instruction, 346
noncnvwobjdata, 371
O
offline simulation
definition, 18
options
  PickMaster, 42
P
PatMax
  advanced model settings, 327
pausing
  robot
    restarting, 315
  PickMaster Host
definition, 17
  PickMaster Operator
definition, 17
  PickMaster PowerPac
definition, 17
  PickMaster Runtime
definition, 17
pick rate, 315
PMHook.dll, 149
PMHook.xml, 149
position available signal, 117, 120, 174–175
positions
  bad accuracy, 408
  used twice, 409
precision time protocol, PTP, 40
PROC.cfg, system parameters, 62
Process
  system parameters, 62
product
  height, 326
Q
QStartItmSrc, instruction, 348
QStopItmSrc, instruction, 349
queue idle signal, 117, 120, 174–175
quiet shut down, 42
R
recipe
definition, 17
restarting
  emergency stop, 315
  MultiMove, 315
  robot, 315
resuming
  robot, 315
RGB
description, 317
RIS
definition, 17
log, 42
status messages, 42
robot
  not moving, 407
  starting, 315
  stopping, pausing, 315
robot controller
I/O boards, 52
settings, 60
robot states, 315
RobotWare, 60

S
saturation
description, 317
selection
data, data type, 363
sensor chip, 27
signals
configuring, 172
connecting, 52
conveyor start/stop, 117, 174–175
position available, 117, 120, 174–175
predefined, 174–175
queue idle, 117, 120, 174–175
strobe, 117, 120, 174–175
trig, 117, 120, 174–175
SimplePosGenHook.cs, 149
Simulated camera, 179
six axes robot configuration, 62
sortdata, data type, 366
sourcedata, 370
standards
IEEE 1588, 40
StartSig, 52
status messages
log, 42
strobe signal, 117, 120, 174–175
sub inspection models
blob, 307
caliper, 310
different, 312
geometric PatMax, 307
histogram, 307
switches
network, 55
SyncSeparation filter distance, 409
system parameters
about, 60
counts_per_meter, 197, 220, 228, 252
I/O boards, 52

I/O connections, 52
PROC.cfg, 62
six axes robot configuration, 62

T
time synchronization service, 40
trigger signal, 117, 120, 174–175
troubleshooting, 399

U
UDP/IP, 40
User Hook
creating, 150
dll, 157

V
vision
color, 317
coloring height methods, 327
Vision Interface, 149
vision models
blob, 297
caliper, 310
different, 312
gemetric model PatMax, 289
PatMax, 289
vision system
Gigabit Ethernet, 24
maximum number of cameras, 24
requirements, 24

W
Warning
4326, 403
4327, 403
4328, 404
4329, 404
white balance
calibrating, 319
work area
definition, 17

Z
zoom, 285