Product Specification

RobotWare Options

3HAC 9218-1/Rev.2
BaseWare OS 4.0

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Technology Products AB, Robotics. ABB Automation Technology Products AB, Robotics assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Technology Products AB, Robotics be liable for incidental or consequential damages arising from use of this document or of the software and hardware described in this document.

Additional copies of this document may be obtained from ABB Automation Technology Products AB, Robotics at its then current charge.

Article number: 3HAC 9218-1 Rev.2
Issue: BaseWare OS 4.0

ABB Automation Technology Products AB, Robotics
SE-721 68 Västerås
Sweden

CONTENTS

1 Introduction

RobotWare is a family of software products from ABB Automation Technology Product AB, Robotics, designed to make you more productive and lower your cost of owning and operating a robot.

ABB Automation Technology Product AB, Robotics has invested many man-years into the development of these products and they represent knowledge and experience based on several thousand robot installations.

Within the RobotWare family there are three classes of products:

BaseWare OS - This is the operating system of the robot and constitutes the kernel of the RobotWare family. BaseWare OS provides all the necessary features for fundamental robot programming and operation. It is an inherent part of the robot but can be provided separately for upgrading purposes.
For the description of BaseWare OS, see Product Specification S4Cplus.

BaseWare Options - These products are options that run on top of BaseWare OS of the robot. They represent functionality for robot users that need additional functionality, for example run multitasking, transfer information from file to robot, communicate with a PC, perform advanced motion tasks etc.

ProcessWare - ProcessWare products are designed for specific process applications like welding, gluing and painting. They are primarily designed to improve the process result and to simplify installation and programming of applications. These products also run on top of BaseWare OS.

2 BaseWare Options

[544] Absolute Accuracy

Absolute Accuracy (AbsAcc) is a calibration concept, which ensures a TCP absolute accuracy of better than ±1 mm in the entire working range.

The user is supplied with robot calibration data (error parameter file) and a certificate that shows the performance (Birth Certificate).

The difference between an ideal robot and a real robot can be typically 8 mm, resulting from mechanical tolerances and deflection in the robot structure. Absolute Accuracy option is integrated in the controller algorithms for compensation of this difference, and does not need external equipment or calculation.

Features

- Compensation of mechanical tolerances.
- Compensation of deflection due to load (tool, object and equipment on arm).

Applications

Any application where Absolute Accuracy is needed to facilitate:

- Exchangeability of robots
- Off-line programming with minimum touch-up.
- On-line programming with accurate linear movement as well as accurate reorientation of tool
- Re-use of programs between applications

Controller algorithms

Inherent mechanical tolerances and deflection due to load in the robot structure decrease the robot’s absolute accuracy. Practical compensation of such errors is a complex and highly non-linear problem. The ABB solution is to compensate positions internally in the controller, resulting in a defined and measurable robot TCP (Tool Center Point) accuracy. A generic robot control model is used for each robot family and robot individuals are described by a set of error parameters, generated during calibration at ABB Automation Technology Products, Robotics. Accuracy of each robot will be ascertained and verified through the “Birth Certificate” which statistically describes the robot accuracy in a large sample of robot positions.

[544] Absolute Accuracy

Performance

Once the Absolute Accuracy parameter file is loaded and activated, the robot can be used.

Absolute Accuracy is active in:

- Motion function based on robtarget (MoveJ, MoveL, MoveC and ModPos)
- Reorientation jogging
- Linear jogging (no online compensation as the user defines the physical location, but absolute coordinates are determined for the active pose and shown in jogging window)
- Tool definition (4, 5, 6 point tool definition, room fixed TCP, stationary tool)
- Workobject definition

Absolute Accuracy is inactive in:

- Motion function based on a jointtarget (MoveAbsJ). Independent joint
- Joint based jogging
- External axes
- Track motion
- Any feature not listed in “Absolute Accuracy is active in”

For joint based motions, switching to the jogging window and selecting a cartesian jog mode (Linear, Reorient) will show the correct absolute coordinates. Similarly creation of a robtarget at a point taught by joint based motion will be absolutely accurate.

Requirements

Each Absolute Accuracy robot is shipped with an error parameter file that is unique to that robot. This file must be loaded into the controller and subsequently activated in order to use Absolute Accuracy. Absolute Accuracy functionality may also be deactivated. Both actions require a cabinet restart.

Supported robot types

Please contact your local ABB office in order to get the latest list of supported robot types.

RAPID instructions included in this option

No specific RAPID instructions are included.

[541] Load Identification and Collision Detection (LidCode)

This option is available for the following robot families: IRB 140, IRB 1400, IRB 2400, IRB 4400, IRB 6400 (not 640) IRB 7600 and for external manipulators IRBP-L, IRBP-K, IRBP-R and IRBP-A.
Load identification is not available for the hanging variants of IRB 1400 and IRB 2400 robots.

LidCode contains two very useful features:

Load Identification

To manually calculate or measure the load parameters accurately can be very difficult and time consuming. Operating a robot with inaccurate load parameters can have a detrimental influence on cycle time and path accuracy.

With LidCode, the robot can carry out accurate identification of the complete load data (mass, centre of gravity, and three inertia components). If applicable, tool load and payload are handled separately.

The identification procedure consists of limited predefined movements of axes 3, 5 and 6 during approximately three minutes. The starting point of the identification motion pattern can be chosen by the user so that collisions are avoided.

The accuracy achieved is normally better than 5%.

Collision Detection

Abnormal torque levels on any robot axis (not external axes) are detected and will cause the robot to stop quickly and thereafter back off to relieve forces between the robot and environment.

Tuning is normally not required, but the sensitivity can be changed from Rapid or manually (the supervision can even be switched off completely). This may be necessary when strong process forces are acting on the robot.

The sensitivity (with default tuning) is comparable to the mechanical alternative (mechanical clutch) and in most cases much better. In addition, LidCode has the advantages of no added stick-out and weight, no need for connection to the e-stop circuit, no wear, the automatic backing off after collision and, finally, the adjustable tuning.

Two system outputs reflect the activation and the trig status of the function.

RAPID instructions included in this option

MotionSup	Changing the sensitivity of the collision detection or activating/deactivating the function.
ParIdRobValid	Checking that identification is available for a specific robot type.

ParIdPosValid	Checking that the current position is OK for identification.
LoadId	Performing identification.
MechUnitLoad	Definition of payload for external mechanical units.

[542] ScreenViewer

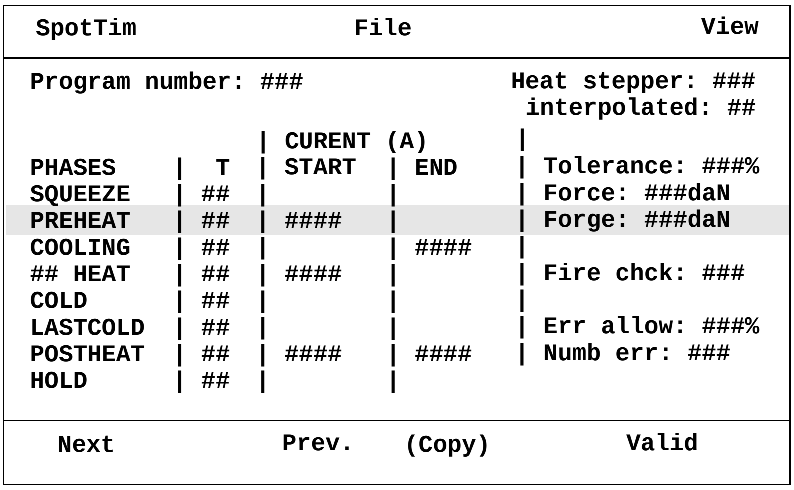
This option adds a user window to display user defined screens with advanced display functions. The user window can be displayed at any time, regardless of the execution state of the RAPID programs.

User defined screens

The user defined screens are composed of:

- A fixed background with a size of 12 lines of 40 characters each. These characters can be ASCII and/or horizontal or vertical strokes (for underlining, separating or framing).
- 1 to 5 function keys.
- 1 to 4 pop-up menus containing from 1 to 10 choices.
- 1 to 30 display and input fields defined by:
  - Their position and size.
  - Their type (display, input).
  - Their display format (integer, decimal, binary, hexadecimal, text).
  - A possible boundary with minimum and maximum limits.

Example of a user defined screen. The ### represent the fields.



```
 SpotTim              File                  View

Program number: ###             Heat stepper: ###
                                 interpolated: ##
                | CURENT (A)        |
PHASES     |  T | START   | END     | Tolerance: ###%
SQUEEZE    | ## |         |         | Force: ###daN
PREHEAT    | ## | ####    |         | Forge: ###daN
COOLING    | ## |         | ####    |
## HEAT    | ## | ####    |         | Fire chck: ###
COLD       | ## |         |         |
LASTCOLD   | ## |         |         | Err allow: ###%
POSTHEAT   | ## | ####    | ####    | Numb err: ###
HOLD       | ## |         |

   Next            Prev.    (Copy)         Valid
```

Advanced Display functions

The user defined screens run independently of the RAPID programs.

Some events occur on a screen (new screen displayed, menu choice selected, function key pressed, field modified, ...). A list of user screen commands can be associated with any of these events, then when the event occurs, the command list will be executed.

A screen event can occur

- When a new screen is displayed (to initialize the screen contents).
- After a chosen interval (to refresh a screen).
- When a menu choice or a function key is selected (to execute a specific action, or change the screen).
- When a new value is entered in a field, or when a new field is selected (to execute some specific action).

The commands that can be executed on screen events are

- Reading/writing RAPID or I/O data.
- Reading/writing fields contents.
- Arithmetical (+, -, /, *, div) or logical (AND, OR, NOT, XOR) operations on the data read.
- Comparing data read (=, <, >) and carrying out a command or not, depending on the comparison result.
- Displaying a different screen.

Capacities

The user screens can be grouped in a screen package file under a specific name. Up to 8 packages can be loaded at the same time.

A certain amount of memory (approx. 50 kbytes) is reserved for loading these screen packages.

- The screen package to be displayed is selected using the far right hand menu “View” (which shows a list of the screen packages installed).

ScreenMaker

ScreenMaker is a complete tool for creating and editing screens for the ScreenViewer on desktop computers running Windows 95/98 or Windows NT.

See ScreenMaker Product Specification.

[532] Multitasking

Up to 10 programs (tasks) can be executed in parallel with the normal robot program.

- These additional tasks start automatically at power on and will continue until the robot is powered off, i.e. even when the main process has been stopped and in manual mode.
- They are programmed using standard RAPID instructions, except for motion instructions.
- They can be programmed to carry out various activities in manual or automatic mode, and depending on whether or not the main process is running.
- Communication between tasks is carried out via I/O or global data.
- Priorities can be set between the processes.

Examples of applications:

- The robot is continuously monitoring certain signals even when the robot program has stopped, thus taking over the job traditionally allocated to a PLC.
- An operator dialogue is required at the same time as the robot is doing, for example, welding. By putting this operator dialogue into a background task, the operator can specify input data for the next work cycle without having to stop the robot.
- The robot is controlling a piece of external equipment in parallel with the normal program execution.

Performance

When the various processes are programmed in the correct way, no performance problems will normally occur:

- When the priorities for the various processes are correctly set, the normal program execution of the robot will not be affected.
- Because monitoring is implemented via interrupts (instead of checking conditions at regular intervals), processor time is required only when something actually happens.
- All input and output signals are accessible for each process.

Note that the response time of Multitasking does not match that of a PLC. Multitasking is primary intended for less demanding tasks. The normal response time is about 5 ms, but in the worst cases, e.g. when the processor is computing new movements, it can be up to 120 ms.

The available program memory can be divided up arbitrarily between the processes. However, each process in addition to the main process will reduce the total memory, see Product Specification S4Cplus.

[531] Advanced Motion

Contains functions that offer the following possibilities:

- Resetting the work area for an axis.
- Independent movements.
- Contour tracking.
- Coordinated motion with external manipulators.

Resetting the work area for an axis

The current position of a rotating axis can be adjusted a number of complete turns without having to make any movements.

Examples of applications:

- When polishing, a large work area is sometimes needed on the robot axis 4 or axis 6 in order to be able to carry out final polishing without stopping. Assume that the axis has rotated 3 turns, for example. It can now be reset using this function, without having to physically rotate it back again. Obviously this will reduce cycle times.
- When arc welding, the work object is often fitted to a rotating external axis. If this axis is rotated more than one turn during welding, the cycle time can be reduced because it is not necessary to rotate the axis back between welding cycles.

Coordinated motion with multi-axis manipulators

Coordinated motion with multi-axis manipulators or robot carriers (gantries) requires the Advanced Motion option. Note that simultaneous coordination with several single axis manipulators, e.g. track motion and workpiece manipulator, does not require Advanced Motion.

Note! There is a built-in general method for defining the geometry for a manipulator comprising two rotating axes (see User's Guide, *Calibration*). For other types of manipulators/robot carriers, comprising up to six linear and/or rotating axes, a special configuration file is needed. Please contact your nearest local ABB office.

Contour tracking

Path corrections can be made in the path coordinate system. These corrections will take effect immediately, also during movement between two positions. The path corrections must be entered from within the program. An interrupt or multitasking is therefore required to activate the correction during motion.

Example of application:

- A sensor is used to define the robot input for path correction during motion. The input can be defined via an analog input, a serial channel or similar. Multitasking or interrupts are used to read this information at specific intervals. Based on the input value, the path can then be adjusted.

Independent movements

A linear or rotating axis can be run independently of the other axes in the robot system. The independent movement can be programmed as an absolute or relative position. A continuous movement with a specific speed can also be programmed.

Examples of applications:

- A robot is working with two different stations (external axes). First, a work object located at station 1 is welded. When this operation is completed, station 1 is moved to a position where it is easy to change the work object and at the same time the robot welds the work object at station 2. Station 1 is moved independently of the robot's movement, which simplifies programming and reduces the cycle time.
- The work object is located on an external axis that rotates continuously at a constant speed. In the mean time, the robot sprays plasma, for example, on the work object. When this is finished the work area is reset for the external axis in order to shorten the cycle time.

Friction Compensation

During low speed (10-100 mm/s) cutting of fine profiles, in particular small circles, a friction effect, typically in the form of approximately 0.5 mm "bumps", can be noted. Advanced Motion offers a possibility of compensating for these frictional effects. Typically a 0.5 mm "bump" can be reduced to about 0.1 mm. This, however, requires careful tuning of the friction level (see User's Guide for tuning procedure). Note that even with careful tuning, there is no guarantee that "perfect" paths can always be generated.

For the IRB 6400 family of robots, no significant effects can be expected by applying Friction Compensation.

RAPID instructions and functions included in this option

IndReset	Resetting the work area for an axis
IndAMove	Running an axis independently to an absolute position
IndDMove	Running an axis independently for a specified distance
IndRMove	Running an axis independently to a position within one revolution, without taking into consideration the number of turns the axis had rotated earlier
IndCMove	Running an axis continuously in independent mode
IndInpos	Checking whether or not an independent axis has reached the programmed position
IndSpeed	Checking whether or not an independent axis has reached the programmed speed
CorrCon	Activating path correction
CorrWrite	Changing path correction
CorrRead	Read current path correction
CorrDiscon	Deactivating path correction
CorrClear	Removes all correction generators

[530] Advanced Functions

Includes functions making the following possible:

- Information transfer via serial channels or files.
- Setting an output at a specific position.
- Checking signal value at a specific position.
- Executing a routine at a specific position.
- Defining forbidden areas within the robot´s working space.
- Automatic setting of output when the robot is in a user-defined area.
- Robot motion in an error handler or trap routine, e.g. during automatic error handling.
- Cross connections with logical conditions.
- Interrupts from analog input or output signals.

Transferring information via serial channels

Data in the form of character strings, numeric values or binary information can be transferred between the robot and other peripheral equipment, e.g. a PC, bar code reader, or another robot. Information is transferred via an RS232 or RS485 serial channel.

Examples of applications:

- Printout of production statistics on a printer connected to the robot.
- Reading part numbers from a bar code reader with a serial interface.
- Transferring data between the robot and a PC.

The transfer is controlled entirely from the robot’s work program. When it is required to control the transfer from a PC, use the option *RAP Communication* or *FactoryWare Interface*.

Data transfer via files

Data in the form of character strings, numerical values or binary information can be written to or read from files on a diskette or other type of mass storage/memory.

Examples of applications:

- Storing production statistics on a diskette or flashdisk. This information can then be read and processed by an ordinary PC.
- The robot’s production is controlled by a file. This file may have been created in a PC, stored on a diskette, and read by the robot at a later time.

Fixed position output

The value of an output (digital, analog or a group of digitals) can be ordered to change at a certain distance before or after a programmed position. The output will then change at the same place every time, irrespective of the robot’s speed.

Consideration can also be given to time delays in the process equipment. By specifying this time delay, the output is set at the corresponding time before the robot reaches the specified position.

The distance can also be specified as a certain time before the programmed position. This time (max. 500 ms) must be within the deceleration time when approaching that position.

Examples of applications:

- Handling press work, to provide a safe signalling system between the robot and the press, which will reduce cycle times. Just as the robot leaves the press, an output is set that starts the press.
- Starting and finishing process equipment. When using this function, the start will always occur at the same position irrespective of the speed. For dispensing and sealing, see *DispenseWare*.

Fixed position IO check

The value of an input/output signal (digital, analog or group) can be checked at a certain distance before or after a programmed position. The signal will then be checked at the same place every time, irrespective of the robot's speed.

The distance can also be specified as a certain time (max 500 ms) before the programmed position.

The data being checked is compared with a certain programmed value and if the comparison is false, the robot will stop and an interrupt routine will be executed. In the interrupt routine appropriate error handling can be executed.

Examples of applications:

- A robot is used for extraction of parts from a die casting machine. Before entering the machine the robot can check, in the fly, if the gate is open. If not, the robot will stop and, in the interrupt routine, wait for the gate to open.

Fixed position procedure call

A procedure call can be carried out when the robot passes the middle of a corner zone. The position will remain the same, irrespective of the robot’s speed.

Example of application:

- In the press example above, it may be necessary to check a number of logical conditions before setting the output that starts the press. A procedure which takes care of the complete press start operation is called at a position just outside the press.

World Zones

A spherical, cylindrical or cubical volume can be defined within the working space. When the robot reaches this volume it will either set an output or stop with the error message “Outside working range”, both during program execution and when the robot is jogged into this area. The areas, which are defined in the world coordinate system, can be automatically activated at start-up or activated/deactivated from within the program.

Examples of applications:

- A volume is defining the home position of the robot.
  When the robot is started from a PLC, the PLC will check that the robot is inside the home volume, i.e. the corresponding output is set.
- The volume is defining where peripheral equipment is located within the working space of the robot.
  This ensures that the robot cannot be moved into this volume.
- A robot is working inside a box.
  By defining the outside of the box as a forbidden area, the robot cannot run into the walls of the box.
- Handshaking between two robots both working in the same working space.
  When one of the robots enters the common working space, it sets an output and after that enters only when the corresponding output from the other robot is reset.

Movements in interrupt routines and error handlers

This function makes it possible to temporarily interrupt a movement which is in progress and then start a new movement which is independent of the first one. The robot stores information about the original movement path which allows it to be resumed later.

Examples of applications:

- Cleaning the welding gun when a welding fault occurs. When a welding fault occurs, there is normally a jump to the program’s error handler. The welding movement in progress can be stored and the robot is ordered to the cleaning position so that the nozzle can be cleaned. The welding process can then be restarted, with the correct parameters, at the position where the welding fault occurred. This is all automatic, without any need to call the operator. (This requires options *ArcWare* or *ArcWare Plus*.)
- Via an input, the robot can be ordered to interrupt program execution and go to a service position, for example. When program execution is later restarted (manually or automatically) the robot resumes the interrupted movement.

Cross-connections with logical conditions

Logical conditions for digital input and output signals can be defined in the robot’s system parameters using AND, OR and NOT. Functionality similar to that of a PLC can be obtained in this way.

Example:

- Output 1 = Input 2 AND Output 5.
- Input 3 = Output 7 OR NOT Output 8.

Examples of applications:

- Program execution to be interrupted when both inputs 3 and 4 become high.
- A register is to be incremented when input 5 is set, but only when output 5=1 and input 3=0.

Interrupts from analog input or output signals

An interrupt can be generated if an analog input (or output) signal falls within or outside a specified interval.

RAPID instructions and functions included in this option

Open	Opens a file or serial channel
Close	Closes a file or serial channel
Write	Writes to a character-based file or serial channel
WriteBin	Writes to a binary file or serial channel
WriteStrBin	Writes a string to a binary serial channel
ReadNum	Reads a number from a file or serial channel
ReadStr	Reads a string from a file or serial channel
ReadBin	Reads from a binary file or serial channel
Rewind	Rewind file position
WriteAnyBin	Write data to a binary serial channel or file
ReadAnyBin	Read data from a binary serial channel or file
ReadStrBin	Read a string from a binary serial channel or file
ClearIOBuff	Clear input buffer of a serial channel
WZBoxDef	Define a box shaped world zone
WZCylDef	Define a cylinder shaped world zone
WZLimSup	Activate world zone limit supervision
WZSphDef	Define a sphere shaped world zone
WZDOSet	Activate world zone to set digital output
WZDisable	Deactivate world zone supervision
WZEnable	Activate world zone supervision
WZFree	Erase world zone supervision
StorePath	Stores the path when an interrupt or error occurs
RestoPath	Restores the path after an interrupt/error
TriggC	Position fix output/interrupt during circular movement
TriggL	Position fix output/interrupt during linear movement
TriggJ	Position fix output/interrupt during joint movement
TriggIO	Definition of trigger conditions for one output
TriggEquip	Definition of trigger conditions for process equipment with time delay
TriggCheckIO	Definition of trigger condition for check of signal value
TriggInt	Definition of trigger conditions for an interrupt
MoveCSync	Position fix procedure call during circular movement
MoveLSync	Position fix procedure call during linear movement

MoveJSync	Position fix procedure call during joint movement
ISignalAI	Interrupts from analog input signal
ISignalAO	Interrupts from analog output signal

[537] Developer’s Function

This option is intended to be used by application developers requiring more advanced functions than normally available for an end user. The package includes a detailed reference manual on the RAPID language kernel and a number of instruction and function groups useful for different application development as listed below.
The groups are:

- Bit Functions
- Data Search Functions
- RAPID Support Functions
- Power Failure Functions
- Trigg Functions
- File Operation Functions

RAPID Kernel Reference Manual

The manual describes the RAPID language syntax and semantics in detail concerning the kernel, i.e. all general language elements which are not used to control robot or other equipment. In addition to this the manual includes descriptions on:

- Built-in Routines
- Built-in Data Objects
- Built-in Objects
- Intertask Objects
- Text Files
- Storage allocation for RAPID objects

Bit Functions

This is a package for handling, i.e. setting, reading and clearing, individual bits in a byte. The instructions/functions are:

byte	Data type for a byte data
BitSet	Set a specified bit in a byte
BitClear	Clear a specified bit in a byte
BitCheck	Check if a specified bit in a byte is set
BitAnd	Logical bitwise AND operation on byte
BitOr	Logical bitwise OR operation on byte
BitXOr	Logical bitwise XOR operation on byte
BitNeg	Logical bitwise NEGATION operation on byte
BitLSh	Logical bitwise LEFT SHIFT operation on byte
BitRSh	Logical bitwise RIGHT SHIFT operation on byte

Data Search Functions

With these functions it is possible to search all data in a RAPID program, where the name or the data type is given as a text string. This might be useful for instance in the following examples:

- A common problem is to check if a data with a certain name is declared in the system, and in such case what is its value, e.g.a robtarget
- Another problem is to list all variables of a certain datatype, which are declared in the system, and write their values on the screen, e.g. all weld data.

The following instructions/functions are included in the package:

SetDataSearch	Define the search criteria
GetNextSym	Search next data and get its name as a string
GetDataVal	Get the value of a data, specified with a string for the name
SetDataVal	Set the value of a data, specified with a string for the name

RAPID Support Functions

This package includes a number of miscellaneous instructions etc., which are used in application development.

User defined data types	This will make it possible to create your own data types, like a record definition
AliasIO	Instruction used to define a signal of any type with an alias (alternative) name. The instruction can be used to make generic modules work together with site specific I/O, without changing the program code.
ArgName	Function used inside a routine to get the name of a data object, which is referenced as argument in the call of the routine. The name is given as a string. The function can also be used to convert the identifier of a data into a string.
BookErrNo	Instruction used to book a new RAPID system error number. This should be used to avoid error number conflicts if different generic modules are combined in a system.
TextTabGet	Function used to get the text table number of a user defined text table during runtime.
TextGet	Function used to get a text string from the system text tables (installed at cold start).
TextTabInstall	Instruction used to install a text table in the system.
TextTabFreeToUse	Function to test whether the text table name (text resource string) is free to use.
IsSysId	Function used to test the system identity.
SetSysData	Instruction which will activates the specified system data (tool or workobject). With this instruction it is possible to change the current active tool or workobject.

IsStopStateEvent	Function which will return information about the movement of the Program Pointer (PP).
ReadCfgData	Read system configuration data.
WriteCfgData	Write system configuration data.

Power Failure Functions

The package is used to get I/O signal values before power failure and to reset them at power on. The following instructions are included and are normally used in the power on event routine:

PFIOResto	Restore the values of all digital output signals.
PFDOVal	Get the value of the specified digital output signal at the time for power failure.
PFGOVal	Get the value of the specified digital output group at the time for power failure.
PFRestart	Check if path has been interrupted.

Trigg Functions

TriggSpeed	Instruction to define conditions and actions for control of an analog output signal with an output value proportional to the actual TCP speed. Note that this instruction must be used in combination with a TriggL/C/J instruction (see [530] Advanced Functions).
StepBwdPath	Instruction used to move backward on its path in a RESTART event routine.
TriggStopProc	Generation of restart data at program stop or emergency stop.

File Operation Functions

The package includes instructions and functions to work with directories and files on mass memory like floppy disc, flash disc or hard disc. It can be used when creating application packages, using RAPID, where RAPID programs and modules should be loaded or stored. It can also be used to search for all files in different directories and e.g. list them on the teach pendant.

The following instructions and functions are available:

dir	Datatype for variables referencing a directory
MakeDir	Create a new directory
OpenDir	Open a directory to read the underlaying files or subdirectories
CloseDir	Close a directory
RemoveDir	Delete a directory
ReadDir	Read next object in a directory, file or subdirectory
RemoveFile	Delete a file
IsFile	Check the type of a file

FileSize	Get the size of a file
FSSize	Get the size of a file system

[558] Discrete Application

Discrete Applications Platform (DAP) is a software platform for time critical applications, where certain actions shall be performed at specific robot positions.

Target users are advanced application software engineers and system integrators, e.g. for spot welding, drilling, measuring, quality control. The main advantages are achieved in the following areas:

- Development time
- Program execution time.
- RAPID- program memory needed
- Similar “look and feel” between applications
- Tested kernel software

Features

- Specialized RAPID instructions and datatypes.
- A single instruction for motion and process execution.
- Combination of fine point positioning with execution of up to 4 parallel processes.
- Specialized process for monitoring of external process devices, like spot welding controllers.
- Supports encapsulation of the process and motion, in shell-routines provided to the end-user.
- The Advanced Functions option is included in DAP.

Application

Creation of software for advanced applications with a discrete behaviour, such as spot welding, drilling, measuring, quality control.

Performance

C-code kernel and RAPID calls.
The DAP platform is designed to have an internal kernel, administrating the fast and quality secured process sequence skeleton. The kernel calls RAPID routines, which are prepared by the application writer to fulfil the specific tasks. The application developer regulates the degree of flexibility of the end-user.

Requirements

There are no other requirements than S4CPlus cabinet and BaseWare.

Rapid instructions included in this option

See RAPID Discrete Application Platform User’s Guide.

[540] Conveyor Tracking

Conveyor Tracking (also called Line Tracking) is the function whereby the robot follows a work object which is mounted on a moving conveyor. While tracking the conveyor, the programmed TCP speed relative to the work object will be maintained, even when the conveyor speed is changing slowly.

Note that hardware components for measuring the conveyor position are also necessary for this function. Please refer to the Product Specification for your robot.

Conveyor Tracking provides the following features:

- A conveyor can be defined as either linear or circular.
- It is possible to have four conveyors connected simultaneously and to switch between tracking the one or the other.
- Up to 254 objects can reside in an object queue which can be manipulated by RAPID instructions.
- It is possible to define a start window in which an object must be before tracking can start.
- A maximum tracking distance may be specified.
- If the robot is mounted on a parallel track motion, then the system can be configured such that the track will follow the conveyor and maintain the relative position to the conveyor.
- Tracking of a conveyor can be activated “on the fly”, i.e. it is not necessary to stop in a fine point.

Performance

At 150 mm/s constant conveyor speed, the TCP will stay within ±2 mm of the path as seen with no conveyor motion. When the robot is stationary relative to the conveyor, the TCP will remain within 0.7 mm of the intended position.

These values are valid as long as the robot is within its dynamic limits with the added conveyor motion and they require accurate conveyor calibration.

RAPID instructions included in this option

WaitWObj	Connects to a work object in the start window
DropWObj	Disconnects from the current object

[533] Electronically Linked Motors

This option is used to make master/slave configurations of motors, which are defined as external axes. The main application is to replace mechanical driving shafts of Gantry machines, but the option can be used to control any other set of motors as well.

Features

- Up to 4 master motors.
- Up to 11 motors total (masters and followers).
- Jogging and calibration routines.
- Replacement of mechanical driving shafts.
- Arm/Motor position available on the TPU.
- Possibility to activate/deactivate link during process.
- Automatic calibration at startup.

Application

Gantry machines: to replace mechanical driving shafts.

Requirements

There are no software or hardware requirements for this option.

Performance

- When jogging, the electronically linked motors will follow the master motor
- Calibration – running follower motors independent of the master - is performed through a RAPID calibration program, to ensure high personnel safety
- At startup, a routine will automatically set the master- and follower motors at the start position, through a safe maneuver

RAPID instruction included in this option

There are no RAPID instructions included in this option.

[547] Sensor Synchronization

Sensor Synchronization adjusts the robot speed to an external moving device (e.g. a press or conveyor) with the help of a sensor. This option simplifies programming and improves productivity of any loading /unloading application since it provides automatic sensor status check and speed adjustment.

The robot TCP speed will be adjusted in correlation to the sensor output so that the robot will reach the programmed robtargets at the same time as the external device reaches their programmed positions. The synchronization is started/stopped with a new instruction, SyncToSensor, combined with movement instructions (fine points or corner zones).

Note that hardware components for measuring the sensor output are needed for this function. The same hardware as for Conveyor Tracking is used: encoder and canbus boards. Please refer to the Product Specification for your robot.

Features

- Up to 4 sensors/robot.
- "On-the-fly" activation.
- Valid for any type of movement.
- RAPID access to sensor and queue data.
- Object queue: the same functionality as conveyor tracking.

Applications

Press synchronization
"Side robot"or "Top_Robot " (1 plane work robot NOFAC), paint application

Performance

The TCP will stay within ±50ms delay of the teached sensor position with linear sensor and constant sensor speed.

Rapid instructions included in this option

SyncToSensor	Start/stop synchronization
WaitSensor	Connect to an object in the start window
DropSensor	Disconnect from current object

[539] Sensor Interface

The Sensor Interface option can be used to integrate sensor equipment for adaptive control, like path correction or process tuning. The option includes a driver for serial communication with the sensor system using a specific link protocol (RTP1) and a specific application protocol (LTAPP). The communication link makes it easy to exchange data between the robot controller and the sensor system, using predefined numbers for different data like x,y,z offset values, gap between sheets, time stamp etc.

Features

- Interrupt routines, based on sensor data changes.
- Read/write sensor data from/to sensors using RAPID functions.
- Store/retrieve sensor data as a block to/from a mass memory.
- Seam tracking functionality, when combined with option Advanced Motion, based on using the contour tracking (path correction) functionality.

Application

In any application where it is wanted to read/control a sensor during execution, and to react on changes in certain data, like path offset or process supervisory data, thus making adaptive seam tracking and process control possible.

Requirements

External sensors communicating with the robot controller via serial links.

RAPID instruction included in this option

IVarValue	Used to order and enable an interrupt when the value of a variable accessed via the serial sensor interface has been changed
ReadBlock	Used to read a block of data from a device connected to the serial sensor interface
ReadVar	Used to read a variable from a device connected to the serial sensor interface
WriteBlock	Used to write a block of data to a device connected to the serial sensor interface
WriteVar	Used to write a variable to a device connected to the serial sensor interface

[561] Servo Tool Control

The Servo Tool Control is a general and flexible software platform for controlling an integrated servo tool from S4CPlus. For additional features, like control of external processes, or control of several ServoGuns in parallel, please refer to the option SpotWare Servo.

Target users are advanced system integrators who want to develop customer specific application software, such as spotwelding packages.

As a “quick-start”, the option includes an example code package. This package can be used as a base for application development.

Features

- Position control (gap).
- Force control.
- Dynamic and kinematic model (tool configured as external axis).
- Example code package.

Application

Spot Welding with Servo Guns: The option provides advanced control functionality for Servo Guns. Communication with Weld timers and other process control functionality needs to be implemented outside this option. For a total spot welding package, please refer to the option SpotWare Servo.

Performance

The tool is configured as an external axis, which ensures optimal performance, regarding path following and speed. (Dynamic and kinematic model.)

The option Servo Tool Change can be added to the system in order to allow a switch between two or more servo tool, which will then utilize the same drive unit and measurement board.

Requirements

- Motion parameter file
  A specific servo tool parameter file has to be installed in the controller, for each servo tool. The parameter file is optimized for each system, concerning system behaviour and motion/process performance.
- Drive Module & Measurement board
  See User’s Guide External Axes.

Rapid instructions included in this option

STClose	Close a Servo Tool with a predefined force and thickness
STOpen	Open a Servo Tool
STCalib	Calibrate a Servo Tool
STTune	Tune motion parameters for a Servo Tool
STTuneReset	Reset tuned motion parameters
STIsClosed	Test if a Servo Tool is closed
STIsOpen	Test if a Servo Tool is open
STCalcTorque	Calculate the motor torque for a Servo Tool
STCalcForce	Calculate the programmable force for a Servo Tool

[631] Servo Tool Change

Servo Tool Change enables an on-line change of tools (external axes), for a certain drive- and measurement system. The control is switched between the axes by switching the motor cables from one servo motor to another. The switch is performed on-line, during production.

The main advantages are:

- Flexibility in the production process
  One robot handles several tools
- Minimized equipment
  A single drive-measurement system shared by many tools

Features

- On-line change of tools.
- Up to 8 different tools.

Application

Servo gun changing; Robot held servo guns, designed for different reach and weld forces, equipped with different brands and sizes of servo motors, may be held and operated by a robot, switching from one servo gun to another.

Servo Tool Change can be used as an independent option, or as an addition to the SpotWare Servo or Servo Toool Control options.

Requirements

Servo Tool Change requires a mechanical wrist interface, a *Tool Changer.*

A MOC service parameter, *Disconnect deactive* = YES (Types: Measurement channel), must be set for each tool (external axis) used with this function.

Performance

When switching tools, the following steps are performed (switching from Axis 1 to Axis 2):

- Axis 1 is deactivated using the RAPID instruction *DeactUnit*.
- Axis 1 is disconnected from the motor cables.
- Axis 2 is connected to the motor cables.
- Axis 2 is activated using RAPID instruction *ActUnit*.

After activation, Axis 2 is ready to run.

The motor position at the moment of deactivation of one axis is saved and restored next time the axis is activated. Note: The motor position must not change more than half a motor revolution, when the axis is disconnected. In SpotWare Servo, there is a calibration routine, which handles larger position changes.

RAPID instructions included in this option

There are no specific RAPID instructions included in this option.

[535] RAP Communication

This option is required for all communication with a superior computer, where none of the WebWare products are used. It includes the same functionality described for the option Factory Ware Interface.

It also works for the WebWare products. There is no difference from the FactoryWare Interface (except that the price is higher).

Note that both FactoryWare Interface and RAP Communication can be installed simultaneously.

[543] Ethernet Services

FTP

This option includes the same functionality as described for Ethernet Services NFS except that the protocol used for remote mounted disc functionality is FTP.

The aspect of authorization differs between NFS and FTP.

Examples of applications:

- All programs for the robot are stored in the PC. When a new part is to be produced, i.e. a new program is to be loaded, the program can be read directly from the hard disk of the PC. This is done by a manual command from the teach pendant or an instruction in the program. If the option RAP Communication or FactoryWare Interface is used, it can also be done by a command from the PC (without using the ramdisk as intermediate storage).
- Several robots are connected to a PC via Ethernet. The control program and the user programs for all the robots are stored on the PC. A software update or a program backup can easily be executed from the PC.

NFS

Information in mass storage, e.g. the hard disk in a PC, can be read directly from the robot using the NFS protocol. The robot control program can also be booted via Ethernet instead of using diskettes. This requires Ethernet hardware in the robot.

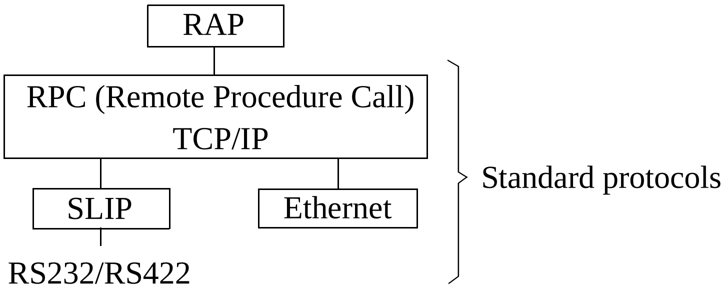
[534] FactoryWare Interface

This option enables the robot system to communicate with a PC. FactoryWare Interface serves as a run-time license for WebWare, i.e. the PC does not require any license protection when executing a WebWare based application.

Factory Ware Interface includes the Robot Application Protocol (RAP). The Robot Application Protocol is used for computer communication. The following functions are supported:

- Start and stop program execution
- Transfer programs to/from the robot
- Transfer system parameters to/from the robot
- Transfer files to/from the robot
- Read the robot status
- Read and write data
- Read and write output signals
- Read input signals
- Read error messages
- Change robot mode
- Read logs

RAP communication is available both for serial links and network, as illustrated by the figure below.



Examples of applications:

- Production is controlled from a superior computer. Information about the robot status is displayed by the computer. Program execution is started and stopped from the computer, etc.
- Transferring programs and parameters between the robot and a PC. When many different programs are used in the robot, the computer helps in keeping track of them and by doing back-ups.

RAPID instruction included in this option

SCWrite	Sends a message to the computer (using RAP)

[271] Interbus Configuration Tool

The Interbus Configuration Tool is used to configure the communication channels of the DSQC 512 board. (See ‘I/O Interfaces’, in the S4Cplus Product Specification.)

The tool consists of standard PC software. The tool creates a bus configuration, which is used by the controller.

[270] Profibus DP Configuration Tool

The Profibus Configuration Tool is used to configure the master channel of the Profibus DP DSQC 510 board. (See ‘I/O Interfaces’, in the S4Cplus Product Specification.)

The tool consists of standard PC software. The tool creates a bus configuration, which is used in the robot controller.

Note: This tool is NOT needed for configuration and use of other channels than the master channel of the DSQC 510 board.

[538] I/O Plus

I/O Plus enables the S4Cplus to use non-ABB I/O units. The following units are supported:

- Wago modules with DeviceNet fieldbus coupler, item 750-306 revision 3.
- Lutze IP67 module DIOPLEX-LS-DN 16E 744-215 revision 2 (16 digital input signals).
- Lutze IP67 module DIOPLEX-LS-DN 8E/8A 744-221 revision 1 (8 digital input signals and 8 digital output signals).

For more information on any of these units, please contact the supplier.

The communication between these units and S4Cplus has been verified (this does not, however, guarantee the internal functionality and quality of the units). Configuration data for the units is included.

In I/O Plus there is also support for a so-called “Welder”. This is a project specific spot welding timer, and is not intended for general use.

In addition to the above units, the I/O Plus “Generic Driver” also opens up the possibility to use other digital I/O units that conform with the DeviceNet specification. ABB does not assume any responsibility for the functionality or quality of such units. The user must provide the appropriate configuration data.

I/O Plus also opens up the use of the second DeviceNet channel named CAN2, the configuration of the second channel is automatic if you have I/O Plus.

I/O Plus also opens up the DeviceNet Slave functionality, which allow the S4Cplus controller to act as a slave unit towards another DeviceNet master, the configuration data for the slave unit is included.

3 ProcessWare

[551] ArcWare

ArcWare comprises a large number of dedicated arc welding functions, which make the robot well suited for arc welding. It is a simple yet powerful program since both the positioning of the robot and the process control and monitoring are handled in one and the same instruction.

I/O signals, timing sequences and weld error actions can be easily configured to meet the requirements of a specific installation.

ArcWare functions

A few examples of some useful functions are given below.

Adaptation to different equipment

The robot can handle different types of weld controllers and other welding equipment. Normally communication with the welding controller uses parallel signals but a serial interface is also available.

Advanced process control

Voltage, wire feed rate, and other process data can be controlled individually for each weld or part of a weld. The process data can be changed at the start and finish of a welding process in such a way that the best process result is achieved.

Testing the program

When testing a program, welding, weaving or weld guiding can all be blocked. This provides a way of testing the robot program without having the welding equipment connected.

Automatic weld retry

A function that can be configured to order one or more automatic weld retries after a process fault.

Weaving

The robot can implement a number of different weaving patterns up to 10 Hz depending on robot type. These can be used to fill the weld properly and in the best possible way. Weaving movement can also be ordered at the start of the weld in order to facilitate the initial striking of the arc.

Wire burnback and rollback

These are functions used to prevent the welding wire sticking to the work object.

Fine adjustment during program execution

The welding speed, wire feed rate, voltage and weaving can all be adjusted whilst welding is in progress. This makes trimming of the process much easier because the result can be seen immediately on the current weld. This can be done in both manual and automatic mode.

Seam finding and tracking

Seam finding and tracking can be implemented using a number of different types of sensors. Please contact your nearest local ABB office for more information.

Interface signals

The following process signals are, if installed, handled automatically by ArcWare. The robot can also support dedicated signals for workpiece manipulators and sensors.

Digital outputs	**Description**
Power on/off	Turns weld on or off
Gas on/off	Turns gas on or off
Wire feed on/off	Turns wire feed on or off
Wire feed direction	Feeds wire forward/backward
Weld error	Weld error
Error information	Digital outputs for error identification
Weld program number	Parallel port for selection of program number, or 3-bit pulse port for selection of program number, or Serial CAN/Devicenet communication

Digital inputs	**Description**
Arc OK	Arc established; starts weld motion
Voltage OK	Weld voltage supervision
Current OK	Weld current supervision
Water OK	Water supply supervision
Gas OK	Gas supply supervision
Wire feed OK	Wire supply supervision
Manual wire feed	Manual command for wire feed
Weld inhibit	Blocks the welding process
Weave inhibit	Blocks the weaving process
Stop process	Stops/inhibits execution of arc welding instructions
Wirestick error	Wirestick supervision
Supervision inhibit	Program execution without supervision
Torch collision	Torch collision supervision

Analog outputs	**Description**
Voltage	Weld voltage
Wire feed	Velocity of wire feed
Current	Weld current
Voltage adjustment	Voltage synergic line amplification
Current adjustment	Current synergic line amplification

Analog inputs (cont.)	**Description (cont.)**
Voltage	Weld voltage measurement for monitoring and supervision
Current	Weld current measurement for monitoring and supervision

RAPID instructions included in this option

ArcL	Arc welding with linear movement
ArcC	Arc welding with circular movement
ArcKill	Aborts the process and is intended to be used in error handlers
ArcRefresh	Updates the weld references to new values

[552] ArcWare Plus

ArcWare Plus contains the following functionality:

- ArcWare, see previous chapter.
- Arc data monitoring.
  Arc data monitoring with adapted RAPID instructions for process supervision. The function predicts weld errors.
- Contour tracking during welding.
  Path corrections during welding, i.e. when executing ArcL or ArcC instructions, can be made relative to the path using external sensors like Serial Weld Guide or Laser Track. Such corrections will take effect immediately, also during movement between two positions. The correction data are sent from the sensor system to the controller using a serial link and will automatically affect the path through built in functionality.
  Please note, that this option is compulsory for Serial Weld Guide systems (AWC) or Laser Track systems (M-Spot 90).
- Contour tracking in normal movements (path corrections)
  Path corrections can also be activated when running normal movements like MoveL using specific RAPID path correction instructions.
  This functionality is also a part of option Advanced Motion, see this option for more information.
- Adaptive process control for e.g. sensors like LaserTrak and Serial Weld Guide systems. The sensor can for instance provide the robot system with changes in the shape of the seam. These values can then be used to adapt the process parameters, like voltage or wire feed, to the current shape. (See option Sensor Interface for more information)

RAPID instructions and functions included in this option

CorrCon	Activating path correction
CorrWrite	Changing path correction
CorrRead	Read current path correction
CorrDiscon	Deactivating path correction
CorrClear	Removes all correction generators
SpcCon	Activates statistical process supervision
SpcWrite	Provides the controller with values for statistical process supervision
SpcDump	Dumps statistical process supervision data to a file or on a serial channel
SpcRead	Reads statistical process supervision information
SpcDiscon	Deactivates statistical process supervision
IVarValue	Orders a variable interrupt
ReadBlock	Read a block of data from the sensor device
ReadVar	Read a variable from the sensor device
WriteBlock	Write a block of data to the sensor device
WriteVar	Write a variable to the sensor device

[556] Arcitec

This option is intended to be used in combination with the Arcitec power sources.
It shall only be ordered by the supplying unit of Arcitec.

The package is a special software, used together with the ArcWare package, to be able to control not only the robot program but also the set up, configuration and programming of the power source. Thus the robot teach pendant will be used for programming and tuning both the robot and the power source.

The package also includes a special aid for easy welding programming, i.e. the synergic function. This means that there is a pre-programmed relationsship between the wire feed rate and all other data components in the power source, making it easy to control the entire welding process, just by tuning the wire feed rate.

[553] SpotWare

The Spotweld options are general and flexible software platforms for creation of **customized** and **easy to use** function packages for different types of spotweld systems and process equipments.

The **SpotWare** option is used for sequential welding with one or several pneumatic gun equipments. If welding with several pneumatic guns at the same time is desired then the **SpotWare Plus** option has to be used instead.

The SpotWare option provides dedicated spotweld instructions for fast and accurate positioning combined with gun manipulation, process start and supervision of the weld equipment.

Communication with the welding equipment is normally carried out by means of digital inputs and outputs but a serial interface is also available for some type of weld timers.

It should be noted that the SpotWare options are general and can be extensively customized. They have a default “ready to use” functionality directly after install but it is intended that some configuration data, RAPID data and RAPID routines has to be changed during the customizing.

SpotWare features

Some examples of useful functions are given below:

- Fast and accurate positioning using the unique QuickMove and TrueMove concept.
- Gun pre-closing.
- Quick start after a weld.
- Handling of an on/off gun with two strokes.
- Dual/single gun.
- Manual actions for welding and gun control.
- Simulation possibilities for test purposes.
- Reverse execution with gun control.
- Spot counters.
- User-defined supervision and error recovery. Weld error recovery with automatic rewelding.
- User-defined continuous supervision of the weld equipment, such as weld current signal and water cooling start. Note: This feature requires the MultiTasking option.
- Wide customizing possibilities.

Principles of SpotWare

The SpotWare functions will be controlled by separate internal program processes, which will run in parallel. For instance the robot movements, the continuous supervision and the spot welding will be handled in different independent processes. This means that if for instance the program execution and thus the robot movements is stopped, then the welding and supervision will continue until they come to a well defined process stop. For example, the welding process will carry on and finish the weld and open the gun, although the program has been stopped during the weld phase.

For well defined points in the welding sequence and movements, calls to user routines offer adaptations to the plant environment. A number of predefined parameters are also available to shape the behaviour of the SpotWare instructions.

Programming principles

Both the robot movement and the control of the spot weld equipment are embedded in the basic spot weld instructions *SpotL* and *SpotJ*.

The spot welding process is specified by:

- Spotdata: spot weld process data
- Gundata: spot weld equipment data
- The system modules SWUSRC and SWUSRF: RAPID routines and global data for changing of process and test behaviour.
- System parameters: the I/O Signal configuration.

Spot welding instructions

Instruction	Used to:
SpotL	Control the motion, gun closure/opening and the welding process. Move the TCP along a linear path and perform a spot welding at the end position.
SpotJ	Control the motion, gun closure/opening and the welding process. Move the TCP along a non-linear path and perform a spot welding at the end position.

Spot welding data

Data type	Used to define:
spotdata	The spot weld process
gundata	The spot weld equipment

[554] SpotWare Plus

The **SpotWare Plus** package provides support for sequential welding with one or several pneumatic on/off gun equipments, as the **SpotWare** package, but also welding and full individual monitoring of up to four separate gun equipments at the same time.

SpotWare Plus features

The SpotWare Plus package contains the same features as SpotWare
but with following feature in addition:

- Possibility to weld with up to four guns at the same time.

Principles of SpotWare Plus

As in SpotWare the spotweld functions will be controlled by separate internal program processes, which will run in parallel. For instance the robot movements, the continuous supervision and each spotweld process will be handled in different independent program processes. This means that if for instance the program execution and thus the robot movements is stopped, then the weld processes and supervision will continue until they come to a well defined process stop. For example, the welding processes will carry on and finish the welds and open the guns, although the program has been stopped during the weld phase.

For well defined points in the welding sequence, calls to user routines offer adaptations to the plant environment. A number of predefined parameters are also available to shape the behaviour of the SpotWare instruction.

The opening and closing of the guns are always executed by RAPID routines. These gun routines may be changed from the simple on/off default functionality to a more complex gun control and they may contain additional gun supervision.

SpotWarePlus is based on the DAP (Discrete Application Platform).

Programming principles

Both the robot movement and control of up to four spot weld processes are embedded in the basic spot weld instructions for multiple welding, *SpotML* and *SpotMJ*.

Each spot welding process is specified by:

- Spotmdata: spot weld process data
- Gunmdata: spot weld equipment data
- The system modules SWUSRF and SWUSRC: RAPID routines and global data for customizing purposes and data for changing of process and test behaviour.
- System parameters: the I/O Signal configuration.

Spot welding instructions

Instruction	Used to:
SpotML	Control the motion, gun closure/opening and 1 - 4 welding processes. Move the TCP along a linear path and perform spot welding with 1 - 4 gun equipments at the end position.
SpotMJ	Control the motion, gun closure/opening and 1 - 4 welding processes. Move the TCP along a non-linear path and perform spot welding with 1 - 4 gun equipments at the end position.

Spot welding data

Data type	Used to define:
spotmdata	The spot weld process
gunmdata	The spot weld equipment

[625] SpotWare Servo

The Spotweld options are general and flexible software platforms for creation of **customized** and **easy to use** function packages for different types of spotweld systems and process equipments.

The **SpotWare Servo** option is used for sequential welding with one or two servo gun equipments. If also welding with two servo guns at the same time is desired then the **SpotWare Servo Plus** option has to be used instead.

The SpotWareServo option provides dedicated spotweld instructions for fast and accurate positioning combined with gun manipulation, process start and supervision of the different gun equipments.

Communication with the welding equipment is carried out by means of digital inputs and outputs.

It should be noted that the SpotWare options are general and can be extensively customized. They have a default “ready to use” functionality directly after install but it is intended that some configuration data, RAPID data and RAPID routines has to be changed during the customizing.

SpotWare Servo features

The SpotWare Servo package contains the following features:

- Fast and accurate positioning using the unique QuickMove and TrueMove concept.
- Gun pre-closing, i.e having the gun closing synchronized with weld position.
- Gun equalizing, i.e. having the gun “floating” around the weld position.
- Constant tip force during welding.
- Manual actions for welding and gun control.
- Several simulation possibilities for test purposes.
- Reverse execution with gun control.
- Weld error recovery with automatic rewelding.
- User-defined supervision and error recovery.
- User-defined autonomous supervision, such as weld current signal and water cooling start.
- Wide customizing possibilities.
- Default “ready to use” functionality directly after install.
- Detecting of missing or improper plates.
- Gun calibration functions.
- Spot counters and tip wear data for each used gun.

- Fast switch between two servo guns with a tool changer. Note: This feature requires the Servo Tool Change option.

Principles of SpotWare Servo

The SpotWare functions will be controlled by separate internal program processes, which will run in parallel. For instance the robot movements, the continuous supervision and the spotwelding will be handled in different independent processes. This means that if for instance the program execution and thus the robot movements is stopped, then the welding and supervision will continue until they come to a well defined process stop. For example, the welding process will carry on and finish the weld and open the gun, although the program has been stopped during the weld phase.

For well defined points in the welding sequence and movements, calls to user routines offer adaptations to the plant environment. A number of predefined parameters are also available to shape the behaviour of the SpotWare instructions.

Programming principles

Both the robot movement and the control of the spot weld equipment are embedded in the basic spot weld instructions *SpotL* and *SpotJ*.

The spot welding process is specified by:

- Spotdata: spot weld process data
- Gundata: spot weld equipment data
- The system modules SWDEFINE and SWDEFUSR: RAPID routines and global data for customizing purposes e.g. adaptations for a specific process equipment.
- The system module SWUSER: RAPID routines and global data for changing of process and test behaviour.
- System parameters: the I/O Signal configuration and the Manipulator configuration.

Spot welding instructions

Instruction	Used to:
SpotL	Control the motion, gun closure/opening and the welding process. Move the TCP along a linear path and perform a spot welding at the end position.
SpotJ	Control the motion, gun closure/opening and the welding process. Move the TCP along a non-linear path and perform a spot welding at the end position.
SetForce	Close the gun a predefined time then open the gun.
CalibL	Calibrate the gun during linear movement to the programmed position.
CalibJ	Calibrate the gun during non-linear movement to the programmed position.
Calibrate	Calibrate the gun in current position without movement.
STTune	Tune motion parameters for the servo gun.
STTuneReset	Reset tuned motion parameters for the servo gun.

Spot welding data

Data type	Used to define:
spotdata	The spot weld process
gundata	The spot weld equipment
forcedata	The SetForce process
simdata	Simulation modes

[626] SpotWare Servo Plus

The **SpotWare Servo Plus** package provides support for sequential welding with one or several servo gun equipments, as the **SpotWare Servo** package, but also welding with two servo guns at the same time.

SpotWare Servo Plus features

The SpotWare Servo Plus package contains the same features as SpotWareServo but with following feature in addition:

- Possibility to weld with two servo guns at the same time.

Principles of SpotWare Servo Plus

As in SpotWare Servo the SpotWare functions will be controlled by separate internal program processes, which will run in parallel. For instance the robot movements, the continuous supervision and the spotwelding will be handled in different independent processes. This means that if for instance the program execution and thus the robot movements is stopped, then the weld processes and supervision will continue until they come to a well defined process stop. For example, the welding processes will carry on and finish the weld and open the guns, although the program has been stopped during the weld phase.

Programming principles

Both the robot movement and the control of the spot weld equipments are embedded in the basic spot weld instructions. *SpotL* and *SpotJ* are used for sequential welding. With *SpotML* or *SpotMJ* it is possible to weld with several guns simultaneously.

Each spot welding process is specified by:

- Spotdata: spot weld process data
- Gundata: spot weld equipment data
- The system modules SWDEFINE and SWDEFUSR: RAPID routines and global data for customizing purposes e.g. adaptations for a specific process equipment.
- The system module SWUSER: RAPID routines and global data for changing of process and test behaviour.
- System parameters: the I/O Signal configuration and the Manipulator configuration.

Spot welding instructions

The SpotWare Servo Plus package contains the same instructions as SpotWareServo plus following instructions in addition:

Instruction	Used to:
SpotML	Control the motion, gun closure/opening and 1 - 2 welding processes. Move the TCP along a linear path and perform spot welding with 1 - 2 gun equipments at the end position.
SpotMJ	Control the motion, gun closure/opening and 1 - 2 welding processes. Move the TCP along a non-linear path and perform spot welding with 1 - 2 gun equipments at the end position.

[569] DispenseWare

The DispenseWare package provides support for different types of dispensing processes such as gluing and sealing.

The DispenseWare application provides fast and accurate positioning combined with a flexible process control.

Communication with the dispensing equipment is carried out by means of digital and analog outputs.

DispenseWare is a package that can be extensively customized. The intention is that the user adapts some user data and routines to suit a specific dispensing equipment and the environmental situation.

Dispensing features

The DispenseWare package contains the following features:

- Fast and accurate positioning.
- Handling of on/off guns as well as proportional guns.
- Speed proportional or constant analog outputs.
- Up to five different guns can be handled simultaneously, controlled by 1 - 5 digital output signals (for gun on/off control) and 1 - 2 analog output signals (for flow control).
- Four different gun equipment, each controlled by 1 - 5 digital output signals and 1 - 2 analog output signals, can be handled in the same program.
- Possibility to use different anticipated times for the digital and analog signals.
- Possibility to use equipment delay compensation for the TCP speed proportional analog signals.
- Global or local flow rate correction factors.
- Dispensing instructions for both linear and circular paths.
- Dispensing in wet or dry mode.
- Wide opportunities of customizing the functionality to adapt to different types of dispensing equipment.
- Possibility to restart an interrupted dispense sequence.

Programming principles

Both the robot’s movement and the dispensing process control are embedded in the instructions, *DispL* and *DispC* respectively.

The gluing process is specified by:

- Bead specific dispensing data. See *Data types* - *beaddata*.
- Equipment specific dispensing data. See *Data types* - *equipdata*.
- RAPID routines and global data for customizing purposes. See *Predefined Data and Programs* - *System Module DPUSER.*
- The I/O configuration. See *System Parameters* - *DispenseWare*

Dispensing instructions

Instruction	Used to:
DispL	Move the TCP along a linear path and perform dispensing with the given data
DispC	Move the TCP along a circular path and perform dispensing with the given data

Dispensing data

Data type	Used to define:
beaddata	Dispensing data for the different beads.
equipdata	Dispensing data for the equipment in use.

[571] PalletWare

General

PalletWare is a ready-to-use software package for the S4Cplus controller, focused on palletizing. PalletWare imports data created with PalletWizard, the included off-line PC-tool, to execute the defined palletizing cycles. PalletWare has a predefined interface for connecting PLC (Programmable Logic Controller), which is the most common way to control external equipment such as infeeders, outfeeders and sensor equipment. The standard package includes software components such as priority and scheduling routines that are ready to use. The package also includes components that can be customized, e.g. grip tool control routines.

PalletWare supports system integrators who want to customize the system, by presenting a standardized interface.

What is included

Included in the PalletWare package is software components to control robot motion, and to communicate with the user and external equipment.

In order to minimize commissioning time, PalletWare is equipped with a standardized set of modules, referred to as the Standard Package Add-On. Advanced users can replace these modules with their own if special customizing is needed. PalletWare is also delivered with template user routines that handles the most common solutions.

PalletWare offers a standardized interface where to connect all external equipment such as infeeders, PLC, signal board, grip tool, sensor equipment etc. The package includes a set of predefined signals connected to a simulated board. The integrator has to implement the signal board and connect the signals. The configuration file for the signals has also to be modified depending on what type of board is used. Because of the large amount of signals, it is recommended to use a field bus such as for example Profibus.

PalletWare features

PalletWare offers for example following functionality:

- Multiplacing
- Parallel processing, up to 5 stations simultaneously
- Multistationary production
- User dialogue interface with Screen Viewer
- On-line tuning of geometrical data
- Safety functionality
- Prepared PLC interface
- Prepared MMI interface

- Tool control
- Standardized error handling
- Predefined user routines

Programming principles

PalletWare is added to the BaseWare system. PalletWare consists of predefined motion principles and communication routines for communicating with external equipment. It is not necessarily needed to implement any RAPID code. However, the system supports integrators for customizing by standardized functions and instructions.

Customizing PalletWare

PalletWare can be up-and-running without any need for implementing RAPID code, but it must be adapted to the current robot cell and its physical lay-out. For instance, there are a number of steps which are compulsory, e.g:

- Connect external equipment such as infeeders, tool, PLC etc., via the predefined interface.
- Connect safety equipment such as emergency stop, safety fences etc.
- Define tooldata if the tool does not match the templates
- Check the set-up in the configuration module PAL_CFG.
- Define/teach work objects to the stations
- Define/teach robtarget with tool in zero orientation
- Create and load pallet cycles with PalletWizard.

In addition to this, PalletWare offers a great deal of customizing possibilities for advanced users, e.g.

- Using tools with suction cups or mechanical gripper
- Sliding uppermost layer to gain pallet height
- Controlling orientation on infeeder
- Add or skip safety height movements between stations
- Set priority when working with several pallet cycles in parallel.
- Etc.

Pallet Wizard

Pallet Wizard is a complete and easy to use stand alone tool running on a PC under Windows 95/98 or Windows NT, for off line programming of palletizing or depalletizing processes. It is delivered as a part of the PalletWare option package.

In PalletWizard the complete cell with its different components like the products, the tools, the in/out feeders and pallet stations as well as the pallet cycles with the layers and the pattern descriptions can be defined.

PalletWizard offers for example the following features:

- Detailed On-line help
- Wizards for defining the products, tools, cell definition, station configurations, pallet composition and the pallet cycles
- Automatic pick- and place definition
- Automatic calculating of grip zones to be used for the tool
- Software based collision detection
- Library of predefined patterns

Several different pallet cycles can be combined into a production cycle and saved into a file, which can be downloaded to the robot. At the robot the operator can then select what specific pallet cycle to run and on which infeeder and pallet station.

4 Index