Library of
Connection Elements
ABB Procontic T200

ABB Procontic
Programming System

907 PB 360, PB 361, PB 362

Block Library Expansion

ABB Schalt-
und Steuerungstechnik
Detailed descriptions of the connection elements

Contents

The connection elements from the block library expansions 907 PB 360, PB 361 and PB 362 have been integrated into the programming system 907 PC 332.

The following table of contents shows the CEs with their page numbers.
For instructions for use see volume 7.

<table>
<thead>
<tr>
<th>CE name</th>
<th>Function</th>
<th>Page</th>
<th>from CE library</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADR</td>
<td>Store absolute address</td>
<td>1</td>
<td>907 PB 361</td>
</tr>
<tr>
<td>IDLB</td>
<td>Read indirect, bit</td>
<td>2</td>
<td>907 PB 361/362</td>
</tr>
<tr>
<td>IDS8</td>
<td>Write indirect, bit</td>
<td>3</td>
<td>907 PB 361/362</td>
</tr>
<tr>
<td>PID-PARA</td>
<td>Operate controller</td>
<td>4</td>
<td>907 PB 361</td>
</tr>
<tr>
<td>PID-RUN</td>
<td>Control one cycle</td>
<td>6</td>
<td>907 PB 361</td>
</tr>
<tr>
<td>PID-VERW</td>
<td>Initialize, manage controller</td>
<td>7</td>
<td>907 PB 361</td>
</tr>
<tr>
<td>UD_SEND</td>
<td>Function block for the timer module 07 UD 60</td>
<td>10</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>UD_EMPF</td>
<td>Function block for the timer module 07 UD 60</td>
<td>13</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>UD_ANZ</td>
<td>Function block for the timer module 07 UD 60</td>
<td>16</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>AMELD</td>
<td>Change annunciator, word</td>
<td>19</td>
<td>907 PB 360</td>
</tr>
<tr>
<td>BMELD</td>
<td>Change annunciator, bit</td>
<td>21</td>
<td>907 PB 360</td>
</tr>
<tr>
<td>COPY</td>
<td>Copy a memory area</td>
<td>23</td>
<td>907 PB 360</td>
</tr>
<tr>
<td>FIFOB</td>
<td>FIFO, bit</td>
<td>24</td>
<td>907 PB 360</td>
</tr>
<tr>
<td>FIFOW</td>
<td>FOFO, word</td>
<td>26</td>
<td>907 PB 360</td>
</tr>
<tr>
<td>FKG</td>
<td>Function generator</td>
<td>28</td>
<td>907 PB 361</td>
</tr>
<tr>
<td>KT_DRU</td>
<td>Output of a text page with variables (07 KT 60)</td>
<td>30</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>KT_INI</td>
<td>Initialization of text processor 07 KT 60</td>
<td>33</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>KT_RD</td>
<td>Block for general text input 07 KT 60</td>
<td>35</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>KT_WR</td>
<td>Block for general text output 07 KT 60</td>
<td>38</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>KT_ZUST</td>
<td>Block for interrogating 07 KT 60 condition register</td>
<td>41</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>LIFO</td>
<td>LIFO, word</td>
<td>44</td>
<td>907 PB 360</td>
</tr>
<tr>
<td>LIZU</td>
<td>List allocator</td>
<td>46</td>
<td>907 PB 360</td>
</tr>
<tr>
<td>PI</td>
<td>Proportional–integral controller</td>
<td>48</td>
<td>907 PB 361</td>
</tr>
<tr>
<td>PO_RD</td>
<td>Read communication module 07 PO 60</td>
<td>50</td>
<td>907 PB 362</td>
</tr>
<tr>
<td>PO_WR</td>
<td>Write communication block 07 PO 60</td>
<td>57</td>
<td>907 PB 362</td>
</tr>
</tbody>
</table>
The absolute address of the variable VARI is stored in the word flag MW.

Parameters

<table>
<thead>
<tr>
<th>VARI</th>
<th>WORD</th>
<th>EW, AW, MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>MW</td>
<td>WORD</td>
<td>MW, AW</td>
</tr>
</tbody>
</table>

CE Data

Runtime:
- 07 ZE 60
- 07 ZE 61
- 07 ZE 62
- 07 ZE 63

Basic runtime:
- 13 μs
- 13 μs
- 16 μs
- 13 μs

Additional runtime:
- 0
- 0
- 0
- 0

Output updating: yes

Available as of: ABB Procontic T200 with 07 ZE 6x R201, 907 PC 332 and 907 PB 361

Description

The CE "ADR" is only needed in connection with the PID controller. It is called within the PID controller's CEs.
The value of a binary variable, whose address is calculated by adding the offset (OFFS) to the basic address (BADR), is read and allocated to the target address (ZADR).

Addressing outside of the flag range is not reported.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>SPECIAL</td>
<td>Block number</td>
</tr>
<tr>
<td>FREI</td>
<td>BINARY</td>
<td>Enable</td>
</tr>
<tr>
<td>OFFS</td>
<td>WORD</td>
<td>Offset</td>
</tr>
<tr>
<td>BADR</td>
<td>BINARY</td>
<td>Base address</td>
</tr>
<tr>
<td>ZADR</td>
<td>BINARY</td>
<td>Target address</td>
</tr>
</tbody>
</table>

CE Data

<table>
<thead>
<tr>
<th>Runtime:</th>
<th>07 ZE 60</th>
<th>07 ZE 61/63</th>
<th>07 ZE 62</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic runtime:</td>
<td>15 μs</td>
<td>15 μs</td>
<td>8 μs</td>
</tr>
<tr>
<td>Additional runtime:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Output updating: yes

Memory space occupied once when CE is called: 5 words with 32 bits each

Available as of: ABB Procontic T200 / 907 PC 332 / 907 PB 361

Description

If the enable condition FREI (status = 1) is fulfilled, the value of a binary variable whose address is calculated by adding the offset (OFFS) to the basic address (BADR) is read and is allocated to the target address (ZADR).

Addressing outside of the flag range is not reported.
The value of a binary variable (QADR) is written to the variable whose address is calculated by adding the distance (DIST) to the basic address (BADR). The flag M 127,04 is set if addressing is incorrect.

<table>
<thead>
<tr>
<th>FBD/LD</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDSB</td>
<td>FREI</td>
</tr>
<tr>
<td>BN</td>
<td>=BA BN</td>
</tr>
<tr>
<td>FREI</td>
<td>FB 047 IDSB</td>
</tr>
<tr>
<td>QADR</td>
<td>EP QADR</td>
</tr>
<tr>
<td>DIST</td>
<td>EP DIST</td>
</tr>
<tr>
<td>BADR</td>
<td>EPA BADR</td>
</tr>
</tbody>
</table>

**Parameters**

- **BN**: SPECIAL
- **FREI**: BINARY
- **QADR**: BINARY
- **DIST**: WORD
- **BADR**: BINARY

BN xxx
E, E', A, A', M, M'
#B0, #B1, S, T, Z
E, E', A, A', M, M'
S, T, Z
#H, #W, MW, MW'
A, A', M, M', S, T, Z

Block number
Enable
Source address
Distance
Base address

**CE Data**

Runtime:
- Basic runtime:
- Additional runtime:
Output updating:
Memory space occupied once when CE is called:
Available as of:

<table>
<thead>
<tr>
<th></th>
<th>07 ZE 60</th>
<th>07 ZE 61/63</th>
<th>07 ZE 62</th>
</tr>
</thead>
<tbody>
<tr>
<td>26 µs</td>
<td>26 µs</td>
<td>13 µs</td>
<td></td>
</tr>
<tr>
<td>none</td>
<td>17 words with 32 bits each</td>
<td>ABB Proconic T200 / 907 PC 332 / 907 PB 361</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

The value of a binary variable (QADR) is written to the variable whose address is calculated by adding the distance (DIST) to the basic address (BADR). The flag M 127,04 is set if addressing is incorrect.
Description

The connection element PID–PARA contains all control parameters. A separate CE is required for each controller. By way of this CE's parameters, the controller is operated and the status is displayed.

The output parameters contain status displays and an error message.

The program contains a separate CE for each controller. In the main program, it must be placed before the CE PID–VERW.

The controller is assigned parameters and operated by way of the CE PID–PARA. The CE must be run through in each cycle to ensure that a change in the parameters can be recognized by the central unit.

see CE IL Definition in the programming system
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREI</td>
<td>BINARY</td>
<td>Controller enabling, control block is active</td>
</tr>
<tr>
<td>MWRe</td>
<td>WORD</td>
<td>Start address of the word flag range</td>
</tr>
<tr>
<td>MBRe</td>
<td>BINARY</td>
<td>Start address of the bit flag range</td>
</tr>
<tr>
<td>TZ=T</td>
<td>WORD</td>
<td>Sampling time, an integral multiple of the time-controlled block’s interval</td>
</tr>
<tr>
<td>KP</td>
<td>WORD</td>
<td>Proportionality factor</td>
</tr>
<tr>
<td>TN/T</td>
<td>WORD</td>
<td>Integral-action time, scaled to the sampling time</td>
</tr>
<tr>
<td>TV/T</td>
<td>WORD</td>
<td>Derivative action time, scaled to the sampling time</td>
</tr>
<tr>
<td>T1/T</td>
<td>WORD</td>
<td>Returning time, scaled to the sampling time</td>
</tr>
<tr>
<td>OG</td>
<td>WORD</td>
<td>Upper limit</td>
</tr>
<tr>
<td>UG</td>
<td>WORD</td>
<td>Lower limit</td>
</tr>
<tr>
<td>INIT</td>
<td>WORD</td>
<td>Initial value</td>
</tr>
<tr>
<td>w</td>
<td>WORD</td>
<td>Reference variable (setpoint)</td>
</tr>
<tr>
<td>x</td>
<td>WORD</td>
<td>Controlled variable (actual value)</td>
</tr>
<tr>
<td>y</td>
<td>WORD</td>
<td>Output (manipulated variable)</td>
</tr>
<tr>
<td>STOS</td>
<td>BINARY</td>
<td>Bumpy or bumpless setting to initial value</td>
</tr>
<tr>
<td>AEND</td>
<td>BINARY</td>
<td>A change in the parameters TZ, KP, TN, TV and T1 is not accepted until a positive edge appears at AEND.</td>
</tr>
</tbody>
</table>

| S        | BINARY | E, E', M, M', #B0, #B1, A, A'                                              |
| R        | BINARY | E, E', M, M', #B0, #B1, A, A'                                              |
| DT1      | BINARY | E, E', M, M', #B0, #B1, A, A'                                              |
| RUN      | BINARY | M, M', A, A'                                                                |
| CALC     | BINARY | M, M', A, A'                                                                |
| PARA     | BINARY | M, M', A, A'                                                                |
| y=OG     | BINARY | M, M', A, A'                                                                |
| y=UG     | BINARY | M, M', A, A'                                                                |
| Fehl     | BINARY | M, M', A, A'                                                                |

### CE Data

<table>
<thead>
<tr>
<th>Runtime:</th>
<th>07 ZE 60</th>
<th>07 ZE 61/63</th>
<th>07 ZE 62</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic runtime:</td>
<td>220 µs</td>
<td>220 µs</td>
<td>150 µs</td>
</tr>
<tr>
<td>Additional runtime:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output updating:</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Available as of: ABB Proconic T200 with 07 ZE 6x R201, 907 PC332 and 907 PB 361
The connection element PID–RUN controls execution and calculation of one control cycle.

It only has input parameters. These are the start address for the management table "Feld" and the start address of each controller’s flag range "Re0" – "Re63".

**Parameters**

<table>
<thead>
<tr>
<th>Feld</th>
<th>WORD</th>
<th>MW, MW’, AW, AW’</th>
<th>Start address of the management table; Length of the table: 5 + n (n = number of controllers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re</td>
<td>WORD</td>
<td>MW, MW’, AW, AW’</td>
<td>Start address of each controller’s word flag range; Size of the range: 48 words per controller</td>
</tr>
</tbody>
</table>

**CE Data**

<table>
<thead>
<tr>
<th>Runtime:</th>
<th>07 ZE 60</th>
<th>07 ZE 61/63</th>
<th>07 ZE 62</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic runtime (1 controller): 3.9 ms</td>
<td>3.9 ms</td>
<td>2.1 ms</td>
<td></td>
</tr>
<tr>
<td>Additional runtime (per controller): 30 μs</td>
<td>30 μs</td>
<td>15 μs</td>
<td></td>
</tr>
</tbody>
</table>

Output updating: yes

Available as of: ABB Procontic T200 with 07 ZE 6x R201, 907 PC 332 and 907 PB 361

**Description**

The CE occurs once only in the program. It must be called in a time-controlled block.

The input parameter "Feld" is the first word flag for the management table; compare PID–VERW. The parameter Re is capable of duplication. The parameters Re0 – Re063 each contain the first word flag of the corresponding controller’s parameter list. This list is defined with the CE "PID–PARA". The parameters Feld and Re0 must be identical with the identically named parameters of PID–VERW. The CE has no output parameters.

The blocks PID1 (FB 044) and PID2 (FB 045) are called within the CE.
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frei</td>
<td>BINARY</td>
<td>E, E', M, M', #B0, #B1, A, A', S</td>
</tr>
<tr>
<td>ANZ</td>
<td>WORD</td>
<td>EW, EW', MW, MW', #W, AW, AW'</td>
</tr>
<tr>
<td>Feld</td>
<td>WORD</td>
<td>MW, MW', AW, AW'</td>
</tr>
<tr>
<td>Re0</td>
<td>WORD</td>
<td>MW, MW', AW, AW'</td>
</tr>
<tr>
<td>ERR0</td>
<td>WORD</td>
<td>MW, MW', AW, AW'</td>
</tr>
<tr>
<td>ERR1</td>
<td>WORD</td>
<td>MW, MW', AW, AW'</td>
</tr>
<tr>
<td>ERR2</td>
<td>WORD</td>
<td>MW, MW', AW, AW'</td>
</tr>
<tr>
<td>STAT</td>
<td>WORD</td>
<td>MW, MW', AW, AW'</td>
</tr>
</tbody>
</table>

CE Data

Runtime:
- Basic runtime (1 controller):
- Additional runtime (per controller):
The complete runtime occurs once only in the first cycle.

Output updating: yes

Available as of: ABB Procontic T200 with 07 ZE 6x R201, 907 PC 332 and 907 PB 361
Description

The connection element PID—VERW initializes and manages the controllers incorporated in the program. The essential input parameters are the number of controllers and the start address of each controller's flag range.

The output parameters contain error messages and a status display.

The CE occurs once only in the program. In the main program, it must be placed after the CE PID—PARA. The CE must only be enabled once. Therefore, the enable input must be assigned the flag M 126.03. This flag has a 1-signal in the first cycle only.

The input parameter ANZ contains the number of controllers used in the program. The parameter "Feld" is the first word flag for the management table. This table (field) consists of 5 words (16 bits) + 1 word for each controller. That is to say, a maximum flag range of 69 word flags is needed for 64 controllers. The first 5 flags contain the 3 error numbers, the status and the number of controllers. The parameter Re is capable of duplication. The parameters Re0 — Re63 each contain the first word flag of the corresponding controller's parameter list. This list is defined with the CE "PID—PARA".

The output parameters ERR0, ERR1, ERR2 and STAT contain error messages and a status display.

The block PID0 (FB 043) is called within the CE.
Function blocks for the timer module 07 UD 60 R1

Using the function blocks:

- UD_SEND
- UD_EMPF
- UD_ANZ

It is possible to execute all commands of the timer module (except 12H and 22H). The function blocks execute a handshake between the timer module and central unit automatically.

Observe the following when using the function blocks:

- Only ever one block may be active at one time (FREI = 1).
- If one block is activated (not applicable to UD_ANZ), the FREI input must then retain a continuous 1 signal until the RDY output has a 1 signal.
- The function block's error number FeNR is not identical with the error code. The error number FeNR is a form of representing the error bits in the status word, where:

  FeNR = 0  No error
  FeNR = 1  Battery error (BTE bit set)
  FeNR = 2  Data backup invalid (DTE bit set)
  FeNR = 4  Average error (CMDE bit set)
  FeNR = 8  Critical error (MDLE bit set)

FeNR may have other numbers if several error bits are set simultaneously. In the event of an error, however, the error code is transferred from the timer module to the central processing unit as a data word (word 1). That is to say, in the event of an error, the data read by the timer module must be interpreted as the error code. In addition, the ERR output is set in the event of an error.
All commands of the timer module involved in data output (ZE -> UD) can be executed with the function block UD_SEND (FB040).

The following commands can be executed with the function block UD_SEND:

- **02 DEZ / 02 H** Set timer mode;
- **03 DEZ / 03 H** Freeze, enable output register;
- **04 DEZ / 04 H** Disable/enable timer No. x;
- **08 DEZ / 08 H** Timer No. x, delete alarm bit;
- **09 DEZ / 09 H** Stop/enable internal timer;
- **10 DEZ / 0A H** Set seconds counter to 00;
- **15 DEZ / 0F H** Delete error message/display;
- **16 DEZ / 10 H** Set time;
- **17 DEZ / 11 H** Set date;
- **20 DEZ / 14 H** Set timer No. x.

The command 12H (set date and time) cannot be executed with the function block UD_SEND because 7 data words have to be output for this command, thus requiring 2 handshake cycles in the send direction and use of the "Further send data" bit. Omission of this command, however, does not restrict the functional scope of the 07 UD 60 R1 because the command 12H can be replaced completely by the commands 10H and 11H.

### Parameters

<table>
<thead>
<tr>
<th>FREI</th>
<th>Binary</th>
<th>E, E', A, A', M, M' #B0, #B1, S, T, Z</th>
<th>Block enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>UD_A</td>
<td>Address</td>
<td>E W</td>
<td>Slot address of the 07 UD 60 (EW x.yy.00)</td>
</tr>
<tr>
<td>HM_A</td>
<td>Address</td>
<td>M W</td>
<td>Auxiliary flag</td>
</tr>
<tr>
<td>INST</td>
<td>Word</td>
<td>#H, #W, MW, MW'</td>
<td>Command code</td>
</tr>
<tr>
<td>ANZ</td>
<td>Word</td>
<td>#H, #W, MW, MW'</td>
<td>Number of valid data</td>
</tr>
<tr>
<td>DW1</td>
<td>Word</td>
<td>#H, #W, MW, MW'</td>
<td>Send data word 1</td>
</tr>
<tr>
<td>DW2</td>
<td>Word</td>
<td>#H, #W, MW, MW'</td>
<td>Send data word 2</td>
</tr>
<tr>
<td>DW3</td>
<td>Word</td>
<td>#H, #W, MW, MW'</td>
<td>Send data word 3</td>
</tr>
<tr>
<td>ALM</td>
<td>Word</td>
<td>MW, MW'</td>
<td>Timer alarm</td>
</tr>
<tr>
<td>RDY</td>
<td>Binary</td>
<td>A, A', M, M'</td>
<td>Block processed</td>
</tr>
<tr>
<td>ERR</td>
<td>Binary</td>
<td>A, A', M, M'</td>
<td>Processing faulty</td>
</tr>
<tr>
<td>FeNR</td>
<td>Word</td>
<td>AW, AW', MW, MW'</td>
<td>Error number</td>
</tr>
</tbody>
</table>

### CE data

- **Runtime:**
  - 07 ZE 60: 49.7 μs
  - 07 ZE 61: 49.7 μs
  - 07 ZE 62: 27.5 μs
  - Additional operating time: 168 μs
  - 168 μs
  - 88.8 μs

- **Output updating:**
  - yes

- **Memory allocated once when called:**
  - 132 words of 32 bits each

- **Available as of:**
  - ABB Proconic T200 / 907 PC 332 / 907 PB 362
Description

All commands of the timer module involving output of data (ZE -> UD) can be executed with the function block UD_SEND (FB040).

The function block is executed by a positive edge at the FREI input of the CE, whereby FREI must retain a 1 signal until the RDY output (block processed completely) also has a 1 signal. A UD block may only be activated when no other block is currently being processed by the timer module O7 UD 60 R1 (FREI of a different block has a 1 signal, but RDY of the block still has a 0 signal). Before execution of a UD block, it is also expedient to check whether an error has occurred in the timer module (e.g. via FeNR of the block UD_ANZ; FeNR = 0 -> no error).

In the event of an error, troubleshooting measures must first of all be taken before a new command can be executed correctly.

The hardware address (slot) of the timer module must be entered as a word input with the channel number 0 at the input UD_A, e.g.:

Timer module in the basic configuration, slot No. 5
UD_A = EW 005,00

A word flag must be entered at the input HM_A as an auxiliary flag denoting internal statuses within the block.

This word flag must not be used again in the program.

The command code (hex constant, word constant or word flag) of the command to be executed must be entered at the input INST.

The input ANZ, which must be allocated a word constant, hex constant or a word flag, defines how many of the subsequent send data words contain valid data (ANZ = 0 ... 3).

The inputs DW1 ... DW3 serve the purpose of input of the send data, whereby attention must be paid to the fact that the send data has a byte structure. All send data words not needed to execute the required command must be assigned the value 0 (word constant or hex constant).

The input ALM must be assigned a word flag. After execution of the command (RDY = 1), this word flag indicates the current alarm state of the 7 timers. Each single bit of the 7 least significant bits of this word flag is allocated to one timer (timer 1 -> bit 0, LSB, ... timer 7 -> bit 6).

The output RDY indicates whether the command to be executed has been processed. All outputs have a 0 signal as long as the FREI input also has a 0 signal. However, if the block is enabled (Frel = 1), the output RDY is set to 1 after the command has been executed completely by the timer module. The RDY output retains its 1 signal until the FREI input is reset.

The ERR output indicates whether an error has occurred during execution of the command by the timer module. The 4 error bits of the status word are checked (all bits 0 -> no error, one or several bits 1 -> error). The indication at the ERR output is valid as soon as the RDY output is set. The ERR indication is cancelled when the FREI input is reset.

The FeNR output is a word output and indicates the affiliated error number whenever ERR = 1. In this case, the error number consists of the 4 error bits of the status word (STE -> bit 0, LSB, DTE -> bit 1, CMDE -> bit 2, MDLE -> bit 3, all remaining bits = 0). The indication at the FeNR output is valid as soon as the RDY output is set. The FeNR indication is cancelled (reset to 0) if the FREI input is reset.
Example

FBD

| E 000,00 | UD_SEND |
| EW 005,00 | FREI |
| MW 000,00 | UD_A |
| #H 02 | INST |
| #W 1 | ANZ |
| #W 0 | DW1 |
| #W 0 | DW2 |
| MW 000,01 | DW3 |
| ALM | RDY |
| | ERR |
| | FeNR |

IL

1 E 000,00
=BA 000
FB 040 UD_SEND
EPA EW 005,00
EPA MW 000,00
EP #H 02
EP #W 1
EP #W 0
EP #W 0
EPA MW 001,00
AP A 001,00
AP A 001,01
AP AW 002,00
All commands of the timer module involving reception of data (UD -> ZE) can be executed with the function block UD_EMPF (FB041).

The following commands can be executed with the function block UD_EMPF:

- 01 DEZ / 01 H  Check data transfer (ZE <-> UD)
- 32 DEZ / 20 H  Read time;
- 33 DEZ / 21 H  Read date;
- 36 DEZ / 24 H  Read timer setpoint of timer No. x
- 37 DEZ / 25 H  Read actual timer value of timer No. x

The command 22H (read date and time) cannot be executed with the function block UD_EMPF because 7 data words have to be read for this command, thus requiring 2 handshake cycles in the receive direction and use of the "Further receive data" bit. Omission of this command, however, does not involve any restriction of the scope of functions of 07 UD 60 R1 because the command 22H can be replaced completely by the commands 20H and 21H.

### Parameters

<table>
<thead>
<tr>
<th>FREI</th>
<th>BINARY</th>
<th>E, E', A, A', M, M'</th>
</tr>
</thead>
<tbody>
<tr>
<td>UD_A</td>
<td>ADDRESS</td>
<td>EW</td>
</tr>
<tr>
<td>HM_A</td>
<td>ADDRESS</td>
<td>MW</td>
</tr>
<tr>
<td>INST</td>
<td>WORD</td>
<td>#H, #W, MW, MW'</td>
</tr>
<tr>
<td>ANZ</td>
<td>WORD</td>
<td>#H, #W, MW, MW'</td>
</tr>
<tr>
<td>T_NR</td>
<td>WORD</td>
<td>#H, #W, MW, MW'</td>
</tr>
<tr>
<td>DW</td>
<td>WORD</td>
<td>#H, #W, MW, MW'</td>
</tr>
<tr>
<td>ALM</td>
<td>WORD</td>
<td>MW, MW'</td>
</tr>
<tr>
<td>RDY</td>
<td>BINARY</td>
<td>A, A', M, M'</td>
</tr>
<tr>
<td>ERR</td>
<td>BINARY</td>
<td>A, A', M, M'</td>
</tr>
<tr>
<td>FeNR</td>
<td>WORD</td>
<td>AW, AW', MW, MW'</td>
</tr>
</tbody>
</table>

- **Block enable**
- **Slot address of the 07 UD 60** (EW x.yy.00)
- **Auxiliary flag**
- **Command code**
- **Number of valid data**
- **Timer number**
- **1st receive data word**
- **Timer alarm**
- **Block processed**
- **Processing faulty**
- **Error number**

### CE data

**Runtime:**
- 07 ZE 60
- 07 ZE 61
- 07 ZE 62

**Basic operating time:**
- 50.2 μs
- 50.2 μs
- 26.2 μs

**Additional operating time:**
- 249.4 μs
- 249.4 μs
- 137.7 μs

**Output updating:**
- yes

**Memory allocated once when called:**
- 195 words of 32 bits each

**Available as of:**
- ABB Proconic T200 / 907 PC 332 / 907 PB 362
Description
All commands of the timer module involving reading of data (UD \textgreater{} ZE) can be executed with the function block UD_EMPF (FB041).

The function block is executed by a positive edge at the FREI input of the CE, but FREI must retain a 1 signal until the RDY output (block processed completely) also has changed to 1 signal. A UD block may only be activated when no other block is currently being processed by the timer module 07 UD 60 R1 (FREI of a different block has a 1 signal, but RDY of the block still has a 0 signal). Before execution of a UD block, it is also expedient to check whether an error has occurred in the timer module (e. g. via FeNR of the block UD_ANZ: FeNR = 0 \rightarrow no error). In the event of an error, error recovery measures must first of all be taken before a new command can be executed correctly.

The hardware address (slot) of the timer module must be entered as a word input with the channel number 0 at the input UD_A, e.g.:

\[ \text{Timer module in the basic configuration, slot No. 5 UD_A = EW 005,00} \]

A word flag must be entered at the input HM_A as an auxiliary flag denoting internal statuses within the block. This word flag must not be used again in the program.

The command code (hex constant, word constant or word flag) of the command to be executed must be entered at the input INST.

The input ANZ, which must be allocated a word constant, hex constant or a word flag, defines how many of the received data words contain valid data (ANZ = 0 ... 3).

The first word flag of a linear flag field consisting of three word flags must be entered at the input DW. The data words read are stored in these three word flags. Even if it is intended to read less than 3 data words, a flag field consisting of 3 word flags must always be reserved for the data to be read.

The input ALM must be assigned a word flag. After execution of the command (RDY = 1), this word flag indicates the current alarm state of the 7 timers. Each single bit of the 7 least significant bits of this word flag is allocated to one timer (timer 1 \rightarrow bit 0, LSB, ... timer 7 \rightarrow bit 6).

The output RDY indicates whether the command to be executed has been processed. All outputs have a 0 signal as long as the FREI input also has a 0 signal. However, if the block is enabled (FREI = 1), the output RDY is set to 1 after the command has been executed completely by the timer module. The RDY output retains its 1 signal until the FREI input is reset.

The ERR output indicates whether an error has occurred during execution of the command by the timer module. The 4 error bits of the status word are checked (all bits 0 \rightarrow no error; one or several bits 1 \rightarrow error). The indication at the ERR output is valid as soon as the RDY output is set. The ERR indication is cancelled when the FREI input is reset.

The FeNR output is a word output and indicates the affiliated error number whenever ERR = 1. In this case, the error number consists of the 4 error bits of the status word (BTE \rightarrow bit 0, LSB, DTE \rightarrow bit 1, CMDE \rightarrow bit 2, MDLE \rightarrow bit 3, all remaining bits = 0). The indication at the FeNR output is valid as soon as the RDY output is set. The FeNR indication is cancelled (reset to 0) if the FREI input is reset.
Example

FBD

<table>
<thead>
<tr>
<th></th>
<th>FBD</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>E 000,00</td>
<td>UD_EMPF</td>
<td></td>
</tr>
<tr>
<td>EW 005,00</td>
<td>FREI</td>
<td></td>
</tr>
<tr>
<td>MW 000,00</td>
<td>UD_A</td>
<td></td>
</tr>
<tr>
<td>#H 20</td>
<td>HM_A</td>
<td></td>
</tr>
<tr>
<td>#W 2</td>
<td>INST</td>
<td></td>
</tr>
<tr>
<td>#W 1</td>
<td>ANZ</td>
<td></td>
</tr>
<tr>
<td>MW 004,00</td>
<td>T_NR</td>
<td></td>
</tr>
<tr>
<td>MW 004,03</td>
<td>DW</td>
<td></td>
</tr>
<tr>
<td>RDY</td>
<td>A 001,00</td>
<td></td>
</tr>
<tr>
<td>ERR</td>
<td>A 001,01</td>
<td></td>
</tr>
<tr>
<td>FeNR</td>
<td>AW 002,02</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>E 000,00</td>
</tr>
<tr>
<td>=</td>
<td>BA 000</td>
</tr>
<tr>
<td>FB</td>
<td>041</td>
</tr>
<tr>
<td>EPA</td>
<td>EW 005,00</td>
</tr>
<tr>
<td>EPA</td>
<td>MW 000,00</td>
</tr>
<tr>
<td>EP</td>
<td>#H 20</td>
</tr>
<tr>
<td>EP</td>
<td>#W 2</td>
</tr>
<tr>
<td>EP</td>
<td>#W 1</td>
</tr>
<tr>
<td>EPA</td>
<td>MW 004,00</td>
</tr>
<tr>
<td>EPA</td>
<td>MW 004,03</td>
</tr>
<tr>
<td>AP</td>
<td>A 001,00</td>
</tr>
<tr>
<td>AP</td>
<td>A 001,01</td>
</tr>
<tr>
<td>AP</td>
<td>AP 002,02</td>
</tr>
</tbody>
</table>
The function block UD_ANZ is a pure display block. The data words (time/date) output by the timer module when no command is being executed are read by this block, are split into bytes and are displayed at the outputs (JAHR (YEAR), MON, TAG (DAY), STD (HR), MIN, SEC) after BCD -> binary conversion.

These date/time values must only be used for pure display purposes and not for control tasks because mutual consistency of the individual data words cannot be guaranteed. Only ever data read by the timer module by means of the read commands (see UD_EMPF) may be used for control tasks.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>FREI</th>
<th>E, E', A, A', M, M'</th>
</tr>
</thead>
<tbody>
<tr>
<td>UD_A</td>
<td>ADDRESS</td>
<td>EW #B0, #B1, S, T, Z</td>
</tr>
<tr>
<td>FeNR</td>
<td>WORD</td>
<td>AW, AW', MW, MW'</td>
</tr>
<tr>
<td>ALM</td>
<td>WORD</td>
<td>AW, AW', MW, MW'</td>
</tr>
<tr>
<td>JAHR</td>
<td>WORD</td>
<td>AW, AW', MW, MW'</td>
</tr>
<tr>
<td>MON</td>
<td>WORD</td>
<td>AW, AW', MW, MW'</td>
</tr>
<tr>
<td>STD</td>
<td>WORD</td>
<td>AW, AW', MW, MW'</td>
</tr>
<tr>
<td>MIN</td>
<td>WORD</td>
<td>AW, AW', MW, MW'</td>
</tr>
<tr>
<td>SEC</td>
<td>WORD</td>
<td>AW, AW', MW, MW'</td>
</tr>
</tbody>
</table>

Block enable: FREI = BA 000
Slot address of the 07 UD 60 (EW x.yy.00)
Error number: FeNR
Timer alarm word: ALM
Date / year: JAHR
Date / month: MON
Date / day: STD
Time / hours: MIN
Time / minutes: SEC
Time / seconds: SEC

CE data
Runtime (when FREI = 1): 07 ZE 60 07 ZE 61 07 ZE 62
Basic operating time: 82,6 μs 82,6 μs 48,6 μs
Additional operating time: --- --- ---
Output updating: yes
Memory allocated once when called: 74 words of 32 bits each
Available as of: ABB Procontic T200 / 907 PC 332 / 907 PB 362
Description
When currently no command has to be executed, the timer module 07 UD 60 R1 outputs the date and time constantly (monitor mode). By means of the function block UD_ANZ (FB042), this data is read from the T200 bus, the data words are split into bytes and these bytes are output individually after BCD → BIN conversion. Thus the user of the timer module can easily make use of the current date/time for display purposes and for checks. The data determined with the function block UD_ANZ must not be used for control tasks because the individual data words are not updated synchronously in monitor mode, i.e. consistency of the displayed data is not guaranteed. Therefore, date/time values must only be used for control purposes if they have been read by using the command code from the 07 UD 60 R1.

The data (date/time) read by the timer module is displayed at the output of the function block provided a 1 signal is present at the FREI input. If there is 0 signal at the FREI input, the status displayed last is retained until the block UD_ANZ is enabled again or until the supply voltage is deactivated. If a different function block (UD_SEND, UD_EMPF) is to be executed, it is expedient to reset the FREI input of the block UD_ANZ while a different command is being processed. If a different function block is being executed while the FREI input of the block UD_ANZ has a 1 signal, no error will appear during execution of the command, but the date/time displayed by UD_ANZ may have invalid values while the command is being executed.

The hardware address (slot) of the timer module must be entered at the input UD_A as a word input with the channel number 0, e.g.:

Timer module in the basic configuration, slot No. 5
UD_A = EW 005,00

The FeNR output is a word output and indicates the error number of any error that has occurred. In this case, the error number consists of the 4 error bits of the status word (BTE → bit 0, LSB, DTE → bit 1, CMDE → bit 2, MDLE → bit 3, all remaining bits = 0).

The output ALM must be assigned a word flag. This word flag indicates the current alarm status of the 7 timers. Each single bit of the 7 least significant bits of this word flag is allocated to one timer (timer 1 → bit 0, LSB, ... timer 7 → bit 6).

The JAHR output indicates the year number (2 least significant positions of the Gregorian calendar) of the date set in the timer module.

The MON output indicates the month of the date set in the timer module.

The TAG output indicates the day of the date set on the timer module.

The STD output indicates the number of hours (12 or 24 hour system) of the current time in the timer module.

The MIN output indicates the number of minutes of the current time in the timer module.

The SEC output indicates the number of seconds of the current time in the timer module.
Example

FBD

IL

E 000,00
EW 005,00
UD_ANZ
FREI
UD_A
FeNR  MW 000,00
ALM  MW 001,00
JAHR  AW 002,00
MON  AW 002,01
TAG  AW 002,02
STD  AW 002,03
MIN  AW 002,04
SEC  AW 002,05

!  E 000,00
=BA  000
FB 042
EPA  EW 005,00
AP  MW 000,00
AP  MW 001,00
AP  AW 002,00
AP  AW 002,01
AP  AW 002,02
AP  AW 002,03
AP  AW 002,04
AP  AW 002,05
Function: Change annunciatior, word

Brief description

This block checks word values (16 bits) for changes. Each time this block is called, the current input value (actual value) is compared with the historical value (previous value) from the previous PLC program cycle. If these values do not match, the current actual value will be output.

FBD

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREI</td>
<td>Block enable</td>
</tr>
<tr>
<td>R</td>
<td>Block reset</td>
</tr>
<tr>
<td>An</td>
<td>Number of values to be monitored</td>
</tr>
<tr>
<td>NEU</td>
<td>Current input values</td>
</tr>
<tr>
<td>ALT</td>
<td>Historical values</td>
</tr>
<tr>
<td>W+01</td>
<td>Auxiliary word flag</td>
</tr>
<tr>
<td>NR</td>
<td>Number of the value that has changed</td>
</tr>
<tr>
<td>A</td>
<td>Actual value of the variable that has changed</td>
</tr>
<tr>
<td>AND</td>
<td>1 signal in the event of change</td>
</tr>
</tbody>
</table>

CM data

Basic runtime: 40 μs
Additional runtime: 40 μs + 22 μs for n value pairs: n * 26 μs n * 14 μs
Updating of outputs: YES
Memory allocated once when called: 131 words of 32 bits each
Available as of: ABB Proconic 1200
907 PC 321 & 907 PB 360

Description

- FREI: Block enable
  - FREI = 1 → A 1 signal at the FREI input enables the block to be processed.
    In the first cycle after the block has been enabled, all new values are copied to the previous values.
    The outputs NR, A and AND are additionally set to the value 0.
    In all further cycles, the new values are compared with the previous values as long as the block is enabled.
  - FREI = 0 → No change of the new values is detected
R BINARY Reset of the block
- R = 1 --> The outputs NR, A and AND are set to the value 0.
- R = 0 --> This enables the block to be monitored.

#n WORD Number of values to be monitored.
No monitoring takes place if #n is specified.
The number is limited to 255.

NEU WORD Parameter specifying the first variable
of the new values.
All new values are located one after the other.
A total of #n variables is assigned.

ALT WORD Parameter specifying the first variable of the previous values.
All previous values are located one after the other.
A total of #n variables is assigned.

W=01 WORD Parameter specifying the first auxiliary word flag.
A total of 2 consecutive variables are assigned.
These must no longer be used in the remaining
program.

The following states can be interrogated at the outputs:

NR WORD Number of the variable that has changed.
- One change is detected in each cycle.
The number last found remains unchanged if no
change is detected.

A WORD Output of the new value if a variable has changed.
- The value last found remains unchanged if no change has been
detected.

AND BINARY AND is set if a change has been detected
during the last PLC cycle. For each change, this
flag remains set for one PLC cycle only.

Example

<table>
<thead>
<tr>
<th>AMELD</th>
<th>FB 0027 AMELD</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0.00,00 --- FREI</td>
<td>EP E0.00,00</td>
</tr>
<tr>
<td>E0.00,01 --- R</td>
<td>EP E0.00,01</td>
</tr>
<tr>
<td>#W5 --- NEU</td>
<td>EP #W5</td>
</tr>
<tr>
<td>MW0000,00 --- ALT</td>
<td>EPA MW0000,00</td>
</tr>
<tr>
<td>MW0001,00 --- W=01</td>
<td>EPA MW0001,00</td>
</tr>
<tr>
<td>MW0002,00 --- NR</td>
<td>EPA MW0002,00</td>
</tr>
<tr>
<td>MW0003,00 --- A</td>
<td>EPA MW0003,00</td>
</tr>
<tr>
<td>MW0004,00 --- AND</td>
<td>EPA MW0004,00</td>
</tr>
<tr>
<td>MW0005,00 ---</td>
<td>EPA MW0005,00</td>
</tr>
<tr>
<td>AM</td>
<td>EPA MW0006,00</td>
</tr>
<tr>
<td>AP</td>
<td>EPA MW0007,00</td>
</tr>
</tbody>
</table>

The flags MW 000,00 ... MW 004,00 are assigned the new values.
The flags MW 001,00 ... MW 005,00 are assigned the previous values.
The flags MW 002,00 and MW 002,01 are assigned as auxiliary flags.
Function: Change annunciator, bit

Brief description

This block checks binary values for changes. Each time this block is called, the current input value (actual value) is compared with the historical value (previous value) from the previous PLC program cycle. The current actual value is output if these two values do not agree.

FREI  IL

<table>
<thead>
<tr>
<th>FREE 000</th>
<th>BMEED 000</th>
<th>NR 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB 0028</td>
<td>FB 0028</td>
<td></td>
</tr>
</tbody>
</table>

Parameters

FREE   BINARY  E,E',A,A',M,M',  Block enable
        #B0,#B1,S,T,Z
R      BINARY  E,E',A,A',M,M',  Block reset
        #B0,#B1,S,T,Z

#n WORD  EW,EW',AW,AW',  Number of values to be monitored
        MW,MW',MW'
NEU    BINARY  E,E',A,A',M,M'  Current input values
        1st variable
ALT    BINARY  A,A',M,M'  Historical values
        1st variable
W+01   WORD   AW,AW',MW,MW'  Auxiliary word flag
NR      WORD   AW,AW',MW,MW'  Number of the value that has changed
A      BINARY  A,A',M,M',S    Actual value of the variable that has changed
        T,Z
AND    BINARY  A,A',M,M',S    1 signal in the event of change
        T,Z

CE data

<table>
<thead>
<tr>
<th>07 2E 60</th>
<th>07 2E 61</th>
<th>07 2E 62</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 µs</td>
<td>60 µs</td>
<td>22 µs</td>
</tr>
</tbody>
</table>

Additional runtime: -  -  -

for n value pairs: n * 28 µs  n * 28 µs  n * 15 µs

Updating of outputs: YES

Memory allocated once when called: 132 words per 32 bits each

Available as of: ABB Procontrol T200
                907 PC 321 & 907 PB 360

Description

FREE  BINARY  Block enable
- FREE = 1  -> A 1 signal at the FREE input enables the block to be processed.

In the first cycle after the block has been enabled, all new values are copied to the previous values. The outputs NR, A and AND are additionally set to the value 0.

In all further cycles, the new values are compared with the previous values as long as the block is enabled.

- FREE = 0  -> No change of the new values is detected
R BINARY Reset of the block
- R = 1 --> The outputs NR, A and AND are set to the value 0.
- R = 0 --> This enables the block to be monitored.

#n WORD Number of values to be monitored.
- No monitoring takes place if 0 is specified.
- The number is limited to 255.

NEU BINARY Parameter specifying the first variable of the new values.
- All new values are located after the other.
- A total of #n variables is assigned.

ALT BINARY Parameter specifying the first variable of the previous values.
- All previous values are located after the other.
- A total of #n variables is assigned.

W+01 WORD Parameter specifying the first auxiliary word flag.
- A total of 2 consecutive variables is assigned.
- These must no longer be used in the remaining program.

The following states can be interrogated at the outputs:

NR WORD Number of the variable that has changed.
- One change is detected in each cycle.
- The number last found remains unchanged if no change is detected.

A BINARY Output of the new value if a variable has changed.
- The value last found remains unchanged if no change has been detected.

AND BINARY AND is set if a change has been detected during the last PLC cycle. For each change, this flag remains set for one PLC cycle only.

Example

```
| E0.00,00 | FREI | FB 002B | BMELD |
| E0.00,01 | R    | EP E0.00,00 |
| #n      | NEU  | EP E0.00,01 |
| #0000,00| ALT  | EP #b    |
| #0001,00| ALTW002,00,01 | EPA #0000,00 |
| W002,00 | NR   | EPA #0002,02 |
| A      | M0004,00 | EPA M0002,00 |
| AND   | M0005,00 | EPA M0002,02 |
```

The flags MW 000,00 ... MW 000,04 are assigned the new values.
The flags MW 001,00 ... MW 001,04 are assigned the previous values.
The flags MW 002,00 and MW 002,01 are assigned as auxiliary flags.
Function: COPY

Brief description

The function block COPY copies a memory area specified by the source address and its length to a different memory area. The number of words that can be copied is limited to 255. No processing is executed if 0 is specified.

FBD

<table>
<thead>
<tr>
<th>COPY</th>
<th>IBA 000 NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUE</td>
<td>FB 0019 COPY</td>
</tr>
<tr>
<td>ANZ</td>
<td>EPA PP 000 QUE</td>
</tr>
<tr>
<td>ZIEL</td>
<td>EP PP 000 ANZ</td>
</tr>
<tr>
<td></td>
<td>EPA PP 000 ZIEL</td>
</tr>
</tbody>
</table>

Parameters

QUE WORD EW,EW',AW,AW',MW,MW'  Source start address
ANZ WORD EW,EW',AW,AW',MW,MW',MW Number of values to be copied
ZIEL WORD AW,AW',MW,MW' Target start address

CE data

07 ZE 60  07 ZE 61  07 ZE 62
Basic runtime: 19 µs 19 µs 10 µs
Additional runtime: 11.5 µs 11.5 µs 6.2 µs

Updating outputs: YES

Mersoy allocated once when called: 16 words of 32 bits each

Available as of: ABB Procontic 1200
907 PC 321 & 907 PC 360

Description

QUE WORD 1st source address
- The 1st source address of the memory area to be copied is specified here.

ANZ WORD Number
- Number of values to be copied

ZIEL WORD 1st target address
- The 1st target address of the memory area to be copied is specified here.

Example

<table>
<thead>
<tr>
<th>COPY</th>
<th>IBA 000 NR.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MW0000,00 QUE</td>
<td>FB 0019 COPY</td>
</tr>
<tr>
<td>MW5 ANZ</td>
<td>EPA MW0000,00</td>
</tr>
<tr>
<td>MW0005,00 ZIEL</td>
<td>EP MW5</td>
</tr>
<tr>
<td>EPA MW0005,00</td>
<td></td>
</tr>
</tbody>
</table>

The flags MW 0000,00 ... MW 0000,04 are copied to MW 0005,00 ... MW 0005,04.
Function: FIFO, bit

Brief description

This block is a buffer for binary data from which the
data written first is also read first.

FBD

Parameters

LADE BINARY E,E',A',A',M',M', LOAD FIFO
S,T,Z
LESE BINARY E,E',A',A',M',M', Read FIFO
S,T,Z
R BINARY E,E',A',A',M',M', Reset the block with a
S,T,Z
E BINARY E,E',A',A',M',M', Value to be read
S,T,Z
#L WORD EM,EW,AW,AW', Length of the FIFO
MW,MW',MW
ANF BINARY A,A',M',M', Initial address of the FIFO
W+05 WORD AW,AW',MW,MW' Auxiliary word flag
A BINARY A,A',M',M', Output value
FST WORD AW,AW',MW,MW' Occupancy display
L BINARY A,A',M',M',S, Empty display
T,Z
V BINARY A,A',M',M',S, Full display
T,Z

CR data

07 2E 60 07 2E 61 07 2E 62
Basic runtime: 151 μs 151 μs 85 μs
Additional runtime: - - -

Updating of outputs: YES

Memory assigned once when called: 124 words of 32 bits each

Available as of: ABB Proconic T200
907 PC 321 & 907 PB 360

Description

LADE BINARY Load FIFO
- LADE = 1 -> The value located at the input E is read
  into the FIFO.
  If the load signal is applied for more than
  1 cycle, the value E is read in during each
  cycle until the FIFO signals that it is full.
- LADE = 0 -> No value is read.

LESE BINARY Read FIFO
- LESE = 1 -> The value addressed by the read pointer is
  output at the output A.
  If the read signal is applied for more
  than 1 cycle, a value is read out during
  each cycle until the FIFO signals that it is empty.
- LESE = 0 -> No value is read.
R BINARY. Reset of the block
- R = 1 → A D/I edge resets the FIFO. Values previously
  read are no longer available.
- R = 0 → No change of the FIFO's contents

E BINARY. New value to be read

#L WORD. Length of the FIFO;
O is not allowed as an input.

ANF BINARY. Initial address of the FIFO. #L variables are
assigned as of this address.

W+0S WORD. Parameter specifying the first auxiliary word flag.
A total of 6 consecutive variables is assigned.
These must no longer be used in the remaining
program.

The following statuses can be interrogated at the outputs:

A BINARY. The value addressed by the read pointer is output at the output A.
- The value is applied for one PLC cycle

FST WORD. FIFO occupancy display

L BINARY. FIFO empty display
- No values can be read if L=1

V BINARY. FIFO full display
- No values can be read if V=1

Loading and reading:
If a load and read signal are applied simultaneously,
the value located at E is read and the value addressed
by the read pointer is output through the output A.
The occupancy is not changed. The output LEER remains 1
if the FIFO is empty.

Reset and loading:
If a reset and load signal are applied simultaneously, the
RESET is executed first, followed by LOADING.

Reset, loading and reading:
If a reset, a load and a read signal are applied simultaneously,
the RESET will be executed first.
The value located at E will then be shifted directly
through to A.
The occupancy will be set to 0.

Example

\[
\begin{array}{c|c}
\text{FIFO} & 18A \ 000 \ \text{NR.} \\
\hline
\text{MO000,00—LAD} & EP \ \text{MO000,00} \\
\text{MO000,01—LESE} & EP \ \text{MO000,01} \\
\text{ MO000,02—R} & EP \ \text{MO000,02} \\
\text{MO010,00—E} & EP \ \text{MO010,00} \\
\text{ #W5—ANF} & EP \ #W5 \\
\text{MW0004,00—W+0S} & EP \ \text{MW0004,00} \\
\text{ MW0005,00—FST} & \text{FPA} \ \text{MW0005,00} \\
\text{ MW0005,01—L} & \text{EPA} \ \text{MW0005,01} \\
\text{ MW0005,03—V} & \text{AP} \ \text{MW0005,03} \\
\text{ MW0005,04—AP} & \text{AP} \ \text{MW0005,04}
\end{array}
\]

The flags MW 003,00 ... MW 003,04 are assigned the values read.
The flags MW 004,00 ... MW 004,05 are assigned as auxiliary flags.
Function: FIFO word

Brief description

This block is a buffer for word data from which the data written first is also read first.

Parameters

LADe BINARY E,E',A,A',M,M', LOAD FIFO
S,T,Z
LESe BINARY E,E',A,A',M,M', Read FIFO
S,T,Z
R BINARY E,E',A,A',M,M', Reset the block with a
S,T,Z 0/1 edge
E WORD EW,EW',AW,AW', Value to be read
MW,MW',MW
WL WORD EW,EW',AW,AW', Length of the FIFO
MW,MW',MW
ANF WORD AW,AW',MW,MW' Initial address of the FIFO
W05 WORD AW,AW',MW,MW' Auxiliary word flag
A WORD AW,AW',MW,MW' Output value
FST WORD AW,AW',MW,MW' Occupancy display
L BINARY A,A',M,M',S, Empty display
T,Z
V BINARY A,A',M,M',S, Full display
T,Z

CE data

07 2E 60 07 2E 61 07 2E 62
Basic runtime: 151 μs 151 μs 83 μs
Additional runtime: - - -

Updating of outputs: YES

Memory assigned once when called: 124 words of 32 bits each

Available as of: ABB Proconic 1200
907 PC 321 & 907 PB 360

Description

LADe BINARY Load FIFO
- LADe = 1 --> The value located at the input E is read
  into the FIFO.
  If the load signal is applied for more than
  1 cycle, the value E is read in during each
  cycle until the FIFO signals that it is full.
- LADe = 0 --> No value is read.

LESe BINARY Read FIFO
- LESe = 1 --> The value addressed by the read pointer is
  output at the output A.
  If the read signal is applied for more
  than 1 cycle, a value is read out during
  each cycle until the FIFO signals that it is empty.
- LESe = 0 --> No value is read.
R BINARY Reset of the block
- R = 1 --> A 0/1 edge resets the FIFO. Values previously
  read are no longer available.
- R = 0 --> No change of the FIFO's contents

E WORD New value to be read

#L WORD Length of the FIFO;
0 is not allowed as an input.

ANF WORD Initial address of the FIFO. #L variables are
assigned as of this address.

W#05 WORD Parameter specifying the first auxiliary word flag.
A total of 6 consecutive variables is assigned.
These must no longer be used in the remaining
program.

The following statuses can be interrogated at the outputs:

A WORD The value addressed by the read pointer is output at the output A.
- The value is applied for one PLC cycle

FST WORD FIFO occupancy display

L BINARY FIFO empty display
- No values can be read if L=1

V BINARY FIFO full display
- No values can be read if V=1

Loading and reading:
If a load and read signal are applied simultaneously,
the value located at E is read and the value addressed
by the read pointer is output through the output A.
The occupancy is not changed. The output LEER remains 1
if the FIFO is empty.

Reset and loading:
If a reset and load signal are applied simultaneously, the
RESET is executed first, followed by LOADING.

Reset, loading and reading:
If a reset, a load and a read signal are applied simultaneously,
the RESET will be executed first.
The value located at E will then be shifted directly
through to A.
The occupancy will be set to 0.

Example

```
#FIFOW
#MO000,00-- LADE
#MO000,01-- LENE
#MO000,02-- R
#MO000,03-- E
#MO003,00-- ANF
#MO004,00-- W#05
FST--#MO005,00
L--#MO005,01
V--#MO005,04
```

The flags #M 03,00 ... #M 03,04 are assigned the values read.
The flags #M 04,00 ... #M 04,05 are assigned as auxiliary flags.
Function: Function generator
Call in FBD: FKG

Brief description:
This block calculates a polygon on the basis of the specified
interpolation points and outputs the y value pertaining to
each x value.

Parameters:
- x WORD EW,EW',MW,MW',MW', Input value
  AW,AW',TW,TW'
- n WORD #W
  Number of interpolation
  point pairs
- Xi ADDRESS MW,MW'
  Initial address of the flag
  area for the x values
- Yi ADDRESS MW,MW'
  Initial address of the flag
  area for the y values
- y WORD AW,AW',MW,MW'
  Output value

CE data:
07 ZE 60 07 ZE 61 07 ZE 62
Basic runtime: 2551 µs 2551 µs 1326 µs
Additional runtime: 290 µs 290 µs 166 µs
The additional runtime must be added once for each 2 to the
power n interpolation points. That is to say:
4 interpolation points : 1 x additional runtime = 2841 / 1472 µs
8 interpolation points : 3 x additional runtime = 3421 / 1764 µs
16 interpolation points : 4 x additional runtime = 3711 / 1910 µs
32 interpolation points : 5 x additional runtime = 4001 / 2056 µs

Updating of outputs: YES
Memory allocated once when called: 344 words of 32 bits each
Available as of: ABB Proconit 7200
907 PC 321 and
block library expansion 907 PB 361

Description:
A total of n coordinate points can be specified in an X-Y
coordinate system.
Linear interpolation takes place between these points.
The resulting polygon represents the relationship between
the input quantity X and the output quantity Y.
The following applies:
1. The X coordinates must be located at the following addresses:
   e.g. X1 = MW 00,00  X2 = MW 00,01  X3 = MW 00,02 etc.
2. The Y coordinates must be located at the following addresses:
   e.g. Y1 = MW 03,00  Y2 = MW 03,01  Y3 = MW 03,02 etc.
3. The coordinates must be sorted as follows:
   (from left to right in a coordinate system)
4. The maximum X or Y distance between two consecutive points is 32767
; The following also applies:
; y = X1 for x < Xn
; y = Yn for x > Xn
; The number of coordinates must be specified by a direct constant
; at the input n.

; Example

<table>
<thead>
<tr>
<th>FBD</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!BA 00</td>
</tr>
<tr>
<td></td>
<td>FB 014 FKG</td>
</tr>
<tr>
<td></td>
<td>EP MW 000,00</td>
</tr>
<tr>
<td></td>
<td>EP MW 10</td>
</tr>
<tr>
<td></td>
<td>EPA MW 001,00</td>
</tr>
<tr>
<td></td>
<td>EPA MW 002,00</td>
</tr>
<tr>
<td></td>
<td>AP MW 000,01</td>
</tr>
</tbody>
</table>

MW 000,00  x
# W 10  n
MW 001,00  X1
MW 002,00  Y1

MW 000,01
Function: Output of a text page with variables

Brief description

KT_DRU is a block for data/text output with the 07 KT 60. Using this block up to 15 variables can be read into the KT 60 internal text memory; in addition, a page designated with PAGE (number of a text page of the KT 60) can be output (together with all associated variables) via the serial interface.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREI</td>
<td>Block enable</td>
</tr>
<tr>
<td>KT_A</td>
<td>Slot address of 07 KT 60 (EW x.yy.00)</td>
</tr>
<tr>
<td>STAT</td>
<td>Status register, identical for all blocks on a 07 KT 60</td>
</tr>
<tr>
<td>HM_A</td>
<td>Auxiliary flag, internal block states, output parameters</td>
</tr>
<tr>
<td>AN_V</td>
<td>Number of variables to be output (pair of values: variable number / data word)</td>
</tr>
<tr>
<td>MW_A</td>
<td>Beginning of a linear flag field for data transfer</td>
</tr>
<tr>
<td>PAGE</td>
<td>Page number of KT text RAM page to be output</td>
</tr>
<tr>
<td>RDY</td>
<td>Block processing ended</td>
</tr>
<tr>
<td>ERR</td>
<td>Error during block processing</td>
</tr>
<tr>
<td>FeNR</td>
<td>Error number</td>
</tr>
</tbody>
</table>

CR data

Basic operating time: 60.3 μs  60.3 μs  32.1 μs
Max. operating time: 233.3 μs  233.3 μs  124.7 μs

Output updating: yes

Memory occupied at call: 233 words of 32 bits each

Available with: ABB Proconic 1200
907 PC 332 and
Block library expansion 907 P8 362

Description

Using the block "KT_DRU" (FB33) it is possible to output a maximum of 15 variables (word values) into the text memory of the 07 KT 60. In addition, a KT 60 text page (page number must be entered on the PAGE input of the CE), together with all variables assigned to it, is output via the serial interface (command 1204).

The output data (variable number, variable/constant) must be transferred in a linear (uninterrupted, consecutive, ascending) flag field (MW_A).
A positive edge at the input FREI causes the function block "KT_DRU" to be executed exactly once. If the addressed 07 KT 60 is occupied with another task, the command is executed after conclusion of the current task. After processing of the command, the output RDY is set =1. This also the case if a fault occurs during execution of the command. In the case of a fault, RDY is still set =1 and as well as this the ERR output is also set.

In addition, the number of the fault that has occurred can be read at output FENR.

The input FREI must remain at 1-signal until RDY is signalled.

If the input FREI is reset too early, i.e. before RDY =1, then the internal state register STAT can show any status.

After the abortion of a command in this manner, the state register STAT must be cleared before a KT block can be started again (e.g. controller STOP -> RUN, or assignment #W 0 = STAT).

The output parameters (RDY, ERR and FENR) only exhibit valid states when:

a) the block has been completely processed and

b) as long as input FREI =1.

If the input FREI is reset, RDY and ERR are reset and FENR is set =0.

KT_A is the module address (slot address) of the 07 KT 60 text processor used (channel number always 00).

e.g. Slot 5 --> KT_A corresponds to --> EW 005,00 subrack number =0, channel number =00

STAT is a word flag that is identical for all blocks assigned to one 07 KT 60.

In the state register STAT, internal function block states are stored; in addition STAT coordinates the processing of blocks, so that with several blocks active (FREI =1), the 07 KT 60 blocks are processed individually one after the other.

HM_A is a word flag that is used as register for internal block states. HM_A is assigned to only one function block at a time.

The beginning of the linear flag field must be entered on input MW_A. The structure of the field must be as follows:

Example:

MEANING:                  EXAMPLE:

MW_A.0  Variable number of KT 60  #W 23

MW_A.1  Data word whose status is assigned to the above variable

MW_A.2  Variable number of KT 60  #W 24

MW_A.3  Data word #H3333

MW_A.n-1  Variable number of KT 60  MW0.14,12

MW_A.n  Data word MW0.14,13

Maximum of 15 pairs of values (variable number / data word)

The number of variables to be output must be entered on input AN_V.

AN_V:  0 - 15; when AN_V =0 -> page output with old variable values.

The number of the 07 KT 60 text page which, together with all variables assigned to it, is to be output must be entered on the input PAGE (constant or word flag).
Example

FB0

| E 0.00,00 | FRE1 |
| E20.05,00 | KT_A |
| M20.00,00 | STAT |
| M20.00,01 | NM_A |
| #W 3 | AN_V |
| M20.01,00 | MW_A |
| #W 0 | PAGE |

| R DY | A 0.01,00 |
| ERR | A 0.01,01 |
| F = NR | MW0.00,15 |

IL

| MBA 00 |
| FB 33 KT_DRU |
| EP E 0.00,00 |
| EPA MW0.05,00 |
| EPA MW0.00,00 |
| EPA MW0.00,01 |
| EP #W 3 |
| EPA MW0.01,00 |
| EP #W 0 |
| AP A 0.01,00 |
| AP A 0.01,01 |
| AP MW0.00,15 |
! Function: Initialization of text processor 07 KT 60
!
! Brief description:
! This block is used for initialization of the text processor 07 KT 60.

<table>
<thead>
<tr>
<th>PHD</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>KT_INI</td>
<td>IBA 00</td>
</tr>
<tr>
<td>FREI</td>
<td>FB 035</td>
</tr>
<tr>
<td>KT_A</td>
<td>KT 00</td>
</tr>
<tr>
<td>STAT</td>
<td>EP 09</td>
</tr>
<tr>
<td>MM_A</td>
<td>EPA STAT</td>
</tr>
<tr>
<td>RDY</td>
<td>EP MM_A</td>
</tr>
<tr>
<td></td>
<td>AP RDT</td>
</tr>
</tbody>
</table>

Parameters:

- KT_A: ADDRESS EW
- STAT: ADDRESS MM,MM'
- MM_A: ADDRESS MM,MM'
- RDY: BINARY A,A',M,M'

Block enable
Slot address of 07 KT 60 (EW.x.yy,00)
Status register, identical for all blocks on a 07 KT 60
Auxiliary flag, internal block states, output parameters
Initialization ended

CPU data:

- Basic operating time: 40.8 µs 40.8 µs 22.2 µs
- Max. operating time: 80.2 µs 80.2 µs 43.0 µs

Output updating: yes
Memory occupied at call: 92 words of 32 bits each
Available with: ABB Proconic T200
907 PC 332 and
Block library expansion 907 PB 362

Description:

Using the block "KT_INI" the text processor 07 KT 60 can be initialized. Initialization of the text processor must take place after switching on voltage or it can be invoked at any time to reset a KT error which has occurred or to establish a defined operational state.

A positive edge at the input FREI causes the function block "KT_INI" to be performed exactly once. If the addressed 07 KT 60 is occupied with another task, initialization is carried out after the current task is finished. After a successful initialization the output RDY is set = 1. The input FREI must remain at 1-signal until RDY is signalled. If the input FREI is reset too early, i.e. before RDY = 1, then the internal state register STAT can show any status.

After the abortion of a command in this manner the state register STAT must first be cleared, before a KT block can be started again (e.g.: controller STOP -> RUN or assignment EW 0 = STAT). The output parameter RDY only shows a 1-signal when:
- a) the block has been completely processed
- b) as long as input FREI = 1.
If input FREI is reset, then RDY is reset.

KT_A is the device address (slot address) of the 07 KT 60 text processor used (channel number always 00).
E.g.: slots 4 and 5 --> KT_A correspond to --> EW 005,00
; STAT is a word flag that is identical for all blocks assigned to one 07 KT 60.
; In the state register STAT, internal function block states are stored; in addition STAT coordinates the processing of blocks, so that with several blocks active (FREI =1), the 07 KT 60 blocks are processed individually one after the other.
; HM_A is a word flag that is used as register for internal block states. HM_A is assigned to only one function block at a time.

; Example

<table>
<thead>
<tr>
<th>FBD</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>E 000,00</td>
<td>IBA 00</td>
</tr>
<tr>
<td>EW 005,00</td>
<td>FB 035 KT_INIT</td>
</tr>
<tr>
<td>MW 000,00</td>
<td>EPA E 000,00</td>
</tr>
<tr>
<td>MW 000,01</td>
<td>EPA M 000,00</td>
</tr>
<tr>
<td>HM_A</td>
<td>EPA M 000,01</td>
</tr>
<tr>
<td></td>
<td>RDY A 001,00</td>
</tr>
<tr>
<td></td>
<td>AP A 001,00</td>
</tr>
</tbody>
</table>
; function: Block for general text input
;
; brief description
;
; Using the function block "KT_RD" (FB32) it is possible to read a
text string with a maximum length of 64 words from the text
memory of the 07 KT 60. The word values read are stored in a
linear word flag field and are thus accessible to theZE.

<table>
<thead>
<tr>
<th>FRED</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>KT_RD</td>
<td>IBA 00</td>
</tr>
<tr>
<td>FREI</td>
<td>FB 32 KT</td>
</tr>
<tr>
<td>KT_A</td>
<td>KT_A</td>
</tr>
<tr>
<td>STAT</td>
<td>STAT</td>
</tr>
<tr>
<td>MK_A</td>
<td>MK_A</td>
</tr>
<tr>
<td>AN_V</td>
<td>AN_V</td>
</tr>
<tr>
<td>RDY</td>
<td>AP RDY</td>
</tr>
<tr>
<td>ERR</td>
<td>AP ERR</td>
</tr>
<tr>
<td>FENR</td>
<td>AP FENR</td>
</tr>
</tbody>
</table>

Parameters

FREI: BINARY #E,E',M,M',ABO
Number: Block enable

KT_A: ADDRESS #B1,A,A',S,T,Z
Slot address of
07 KT 60 (EW x.yy,00)

STAT: ADDRESS #MW,MW'
Status register; identical for
all blocks on a 07 KT 60

MK_A: ADDRESS #MW,MW'
Auxiliary flag, internal block
states, output parameters

AN_V: WORD #MW,MW,MW'
Beginning of a linear flag-
field for data transfer

RDY: BINARY #A,A',M,M'
Number of data words to be
read

ERR: BINARY #A,A',M,M'
Block processing ended

FENR: WORD #MW,MW,MW'
Error during block
processing

CE data

Basic operating time: 67.6 μs 67.6 μs 36.4 μs
Max. operating time: 304.3 μs 304.3 μs 160.9 μs

Output updating: yes

Memory occupied at call: 264 words of 32 bits each

Available with: ABB ProconLine T200
907 PC 332 and
Block library expansion 907 PB 362

Description

Using the block "KT_RD" (FB32) it is possible to read a maximum
of 64 data words from the text RAM of the 07 KT 60 and to store
them in a linear word flag field.
For this purpose the 07 KT 60 command 1100 is always used.

The parameters which identify the beginning of the data block
to be read (Page/Line/Column) and the placeholders (word flags)
for the data words read (number = AN_V) must be located in a
linear flag field (uninterrupted, consecutive, ascending). The
first word flag of the field must be entered on input MW_A of
the CE.

A positive edge at the input FREI causes the function block
"KT_RD" to be executed exactly once. If the addressed 07 KT 60
is occupied with another task, the command is executed after
completion of the current task. After processing of the command,
the output RDY is set =1. This also the case if a fault occurs
during execution of the command. In the case of a fault, RDY is
still set =1 and as well as this the ERR output is also set =1.
In addition, the number of the fault that has occurred can be read at output FeNr. The input FREI must remain at 1-signal until RDY is signalled. If the input FREI is reset too early, i.e. before RDY = 1, then the internal state register STAT can show any status. After the abortion of a command in this manner, the state register STAT must be cleared before a KT block can be started again (e.g. controller STOP -> RUN, or assignment MW 0 = STAT). The output parameters (RDY, ERR and FeNr) only exhibit valid states when:

- The block has been completely processed and
- as long as input FREI = 1.

If the input FREI is reset, RDY and ERR are reset and FeNr is set = 0.

KT_A is the module address (slot address) of the 07 KT 60 text processor used (channel number always 00).
E.g.: Slot 5 --> KT_A corresponds to --> EW 005,00

STAT is a word flag that is identical for all blocks assigned to a 07 KT 60.
In the state register STAT, internal function block states are stored; in addition STAT coordinates the processing of blocks, so that with several blocks active (FREI = 1), the 07 KT 60 blocks are processed individually one after the other.

HM_A is a word flag that is used as register for internal block states. HM_A is assigned to only one function block at a time.

The number of data words to be read (AN_V : 1 - 64) must be entered on input AN_V.

The beginning of the linear flag field must be entered on input MW_A. The structure of the field must be as follows:

**MEANING:**

- **MW_A.0**
  - No. of the data sequence
  - Is required internally by function block KT_60.
- **MW_A.1**
  - PAGE
  - Page number of the 07 KT 60 text RAM from which the data are to be read.
- **MW_A.2**
  - Line number on the selected page of the 07 KT 60 text RAM at which data output is to begin.
- **MW_A.3**
  - COLUMN
  - Column number on the selected page line of the 07 KT 60 text RAM at which data output is to begin.
- **MW_A.4**
  - Data word 1
- **MW_A.5**
  - Data word 2
- -
- -
- **MW_A.n**
  - Data word n

The following apply generally:

- Length of linear flag field: AN_V + 4
- First data word read is in: MW_A + 4
- Last data word read is in: MW_A + 4 + (AN_V - 1)
- The counting method of the word flags should be noted:

  \[ MW \ 0.12,15 \ + 1 = MW \ 0.13,00 \]
Example:
MW A (beginning of field): MW 0.06,07
AN_V (number n): 11
This results in:
length of linear flag field: 11 + 4 = 15 word flags
1st data word in flag: MW 0.06,07 + 4 = MW 0.06,11
11th data word in flag: MW 0.06,07 + 14 = MW 0.07,05

Example

<table>
<thead>
<tr>
<th>FBD</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>KT_RD</td>
<td>IBA GO</td>
</tr>
<tr>
<td>E 0.00,00</td>
<td>FB 32 KT_RD</td>
</tr>
<tr>
<td>EW0.05,00</td>
<td>EP E 0.00,00</td>
</tr>
<tr>
<td>MW0.00,00</td>
<td>EPA EW0.05,00</td>
</tr>
<tr>
<td>MW0.00,01</td>
<td>EPA MW0.00,00</td>
</tr>
<tr>
<td>MW0.01,00</td>
<td>EPA MW0.01,00</td>
</tr>
<tr>
<td>MW 11</td>
<td>EP MW 11</td>
</tr>
<tr>
<td>AN_V</td>
<td>AP A 0.01,00</td>
</tr>
<tr>
<td>Rdy</td>
<td>AP A 0.01,01</td>
</tr>
<tr>
<td>ERR</td>
<td>AP A 0.01,01</td>
</tr>
<tr>
<td>FeNR</td>
<td>AP MW0.00,15</td>
</tr>
</tbody>
</table>
Function: Block for general text output

Brief description:

KT_WR is a block for data and text output using the 07 KT 60. Any transmit command (number of received data = 0) can be used with as many data words as desired.

<table>
<thead>
<tr>
<th>FB 36</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>KT_WR</td>
<td>18A 00</td>
</tr>
<tr>
<td>FREI</td>
<td>FB 36 KT_WR</td>
</tr>
<tr>
<td>KT_A</td>
<td>EP FREI</td>
</tr>
<tr>
<td>STAT</td>
<td>EPA KT_A</td>
</tr>
<tr>
<td>MM_A</td>
<td>EPA STAT</td>
</tr>
<tr>
<td>MW_A</td>
<td>EPA MM_A</td>
</tr>
<tr>
<td>AN_V</td>
<td>EPA MW_A</td>
</tr>
<tr>
<td>RDY</td>
<td>EP AN_V</td>
</tr>
<tr>
<td>ERR</td>
<td>AP RDY</td>
</tr>
<tr>
<td>FeNR</td>
<td>AP ERR</td>
</tr>
</tbody>
</table>

Parameters:

**FREI** BINARY E,E',M,M',#RD Block enable

**KT_A** ADDRESS EW Slot address of 07 KT 60 (EW.x.yy,00)

**STAT** ADDRESS Mw, Mw' Status register, identical for all blocks on an 07 KT 60

**HM_A** ADDRESS Mw, Mw' Auxiliary flag, internal block states, output parameters

**MW_A** ADDRESS Mw, Mw' Beginning of a linear flag field for data transfer

**AN_V** WORD #w,Mw,Mw' Length of linear flag field (number of variables)

**RDY** BINARY A,A',M,M' Block ended

**ERR** BINARY A,A',M,M' Error in block execution

**FeNR** WORD Aw, Aw', Mw, Mw' Error number

CR data

<table>
<thead>
<tr>
<th>07 ZE 60</th>
<th>07 ZE 61</th>
<th>07 ZE 62</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.0 µs</td>
<td>61.0 µs</td>
<td>33.4 µs</td>
</tr>
</tbody>
</table>

Output updating: yes

Memory occupied at call: 259 words of 32 bits each

Available with:

- ABB Proconic T200
- 907 PC 332 and block library expansion 907 PB 362

Description:

Using the block "KT_WR" (FB 36) it is possible to execute all transmit commands of the 07 KT 60 (any desired number of transmitted data, no received data).

Brief description of the 07 KT 60 commands permitted when using function block KT_WR:

0 - 299 Output of variables to the 07 KT 60 text RAM. Per KT_WR block, one variable, identified by the variable no. (0-299), can be written to the location in the 07 KT 60 text RAM defined in EDITOR mode.

1101 Output of a data string with a maximum length of 64 words from the central processing unit to the text RAM (target address: Page/Line/Column) of the 07 KT 60.
1200 Command to the 07 KT 60 to receive a text string of defined length via the serial interface (printer, terminal etc.) and to store it in the 07 KT 60 text RAM (Page/Line/Column).

1201 Command to the 07 KT 60 to read a text string of defined length (Page/Line/Column/Number) from the KT 60 text RAM and to output it via the serial interface.

1202 Command to the 07 KT 60 to receive a text string (up to CR or 80 characters) via the serial interface and to store it in the 07 KT 60 text RAM (Page/Line/Column).

1203 Command to the 07 KT 60 to read a text string (beginning: Page/Line/Column; end: CR) from the 07 KT 60 text RAM and to output it via the serial interface to a predetermined location (Line2/Column2; not with printer connected!) in the display.

1204 Command to the 07 KT 60 to output an entire page complete with all variables contained in it via the serial interface.

1205 Command to the 07 KT 60 to output the ASCII code for form feed (DC H) via the serial interface.

The output parameters (number of data, command code) and the output data must be transferred in a linear (uninterrupted consecutive, ascending) flag field (MW_A).

A positive edge at the input FREI causes the function block "KT_WRP" to be executed exactly once. If the addressed 07 KT 60 is occupied with another task, the command is executed after conclusion of the current task. After processing of the command, the output RDY is set =1. This is also the case if a fault occurs during execution of the command. In the case of a fault, RDY is still set =1 and as well as this the ERR output is also set =1. In addition, the number of the fault that has occurred can be read at output FenR.

The input FREI must remain at 1-signal until RDY is signalled.

If the input FREI is reset too early, i.e. before RDY =1,

then the internal state register STAT can show any status.

After the abortion of a command in this manner, the state register STAT must be cleared before a KT block can be started again (e.g.: controller STOP -> RUN, or assignment MW 0 = STAT).

The output parameters RDY and ERR only exhibit a 1-signal and error number FenR is only valid when:

a) the block has been completely processed

and

b) as long as input FREI =1.

If the input FREI is reset, RDY and ERR are reset and FenR is is set =0.

KT_A is the module address (Slot address) of the 07 KT 60 text processor used (channel number always 00).

e.g.: Slot 5 -> KT_A corresponds to -> EW 005,00

STAT is a word flag that is identical for all blocks assigned to one 07 KT 60.

In the state register STAT, internal function block states are stored; in addition, STAT coordinates the processing of blocks, so that with several blocks active (FREI =1), the 07 KT 60 blocks are processed individually one after the other.

HM_A is a word flag that is used as register for internal block states. HM_A is assigned to only one function block at a time.
; On the input MW_A the beginning of a linear flag field with the
; following structure must be entered:
;
; MEANING: EXAMPLE:
;
; MW_A.0 No. of the data sequence MW0.04,00
; (f836 internal)
; MW_A.1 Number of data transmitted MW0.04,01
; command + data words
; (2E -> KT)
; MW_A.2 Number of data received: MW0.04,02
; here always = 0 (KT -> 2E)
; MW_A.3 Command identifier MW0.04,03
;
; MW_A.4 Data word 1 MW0.04,04
;
; MW_A.5 Data word 2 MW0.04,05
; " " "
;
; MW_A.n Data word n MW0.04, n
;
; On the input AN_V the overall length (number) of the MW_A
; flag field must be entered (above example: n=1).
;
; Example
;
; FBD IL
;
; E 0.00,00 FREI
; EW0.05,00 KT_A
; MW0.00,00 STAT
; MW0.00,01 MW_A
; MW0.01,00 MW_A
; MW 4 AN_V
; RDI A 0.01,00 AP A 0.01,00
; ERR A 0.01,01 AP A 0.01,01
; FENR MW0.00,15 AP MW0.00,15
;
Function: Block for interrogating 07 KT 60 condition register

Brief description:
KT_ZUST is a block for interrogating the internal state of the 07 KT 60. 1 data word is transmitted (command to interrogate a KT 60 state register) and 1 data word is received (status of corresponding KT 60 state register).

Parameters:

FREI binary E, E', M, M', QB0 #81, A, A', S, T, Z Block enable
KT_A address EW Slot address of 07 KT 60 (EW x, y, z, 00)
STAT address MW, MW' Status register; identical for all blocks on a 07 KT 60
HM_A address MW, MW' Auxiliary flag, internal block states, output parameters
INST word #1, MW, MW' Command to the 07 KT 60 to output a state word
MW_Z address MW, MW' Word flag in which the status reply of the 07 KT 60 is output.
RDY binary A, A', M, M' Block processing finished
ERR binary A, A', M, M' Error during block processing
FenR word MW, MW', MW, MW' Error number

CR data
07 ZE 60 07 ZE 61 07 ZE 62
Basic operating time: 57.8 μs 57.8 μs 30.8 μs
Max. operating time: 191.1 μs 191.1 μs 101.1 μs

Output updating: yes
Memory occupied at call: 166 words of 32 bits each
Available with: ABB Procomit T200
907 PC 332 and Block library expansion 907 PB 362

Description:
Using the block "KT_ZUST" (FB34) it is possible to interrogate the internal state values of the 07 KT 60.
To do this, the command code must be entered on input INST and the status reply of the 07 KT 60 is stored in the word flag entered on the input MW_Z.
A positive edge at the input FREI causes the function block "KT_ZUST" to be executed exactly once. If the address 07 KT 60 is occupied with another task, the command is executed after conclusion of the current task.
After processing of the command, the output RDY is set =1. This also the case if a fault occurs during execution of the command. In the case of a fault, RDY is still set =1 and as well as this the ERR output is also set =1. In addition, the number of the fault that has occurred can be read at output FenR.
The input FREI must remain at 1-signal until RDY is signalled.
If the input FREI is reset too early, i.e. before RDY =1,
then the internal state register STAT can show any status.
After the abortion of a command in this manner, the state
register STAT must be cleared before a KT block can be started
again (e.g.: controller STOP -> RUN, or assignment ZU 0 = STAT).
The output parameters (RDY, ERR and FeNR) only exhibit valid
status values when:
a) the block has been completely processed
and
b) as long as input FREI =1.
If the input FREI is reset, RDY and ERR are reset and FeNR is
set =0.

KT_A is the module address (Slot address) of the 07 KT 60 text
processor used (channel number always 00),
e.g.: Slot 5 --> KT_A corresponds to --> EW 005,00

STAT is a word flag that is identical for all blocks assigned
to one 07 KT 60.
In the state register STAT, internal function block states are
stored; in addition STAT coordinates the processing of blocks,
so that with several blocks active (FREI =1), the 07 KT 60 blocks
are processed individually one after the other.

HM_A is a word flag that is used as register for internal block
states. HM_A is assigned to only one function block at a time.

MU_Z is the input for a word flag. The status read from the
07 KT 60 is output on this word flag.

INST is a word input (word flag, constant) on which the command
code for the status interrogation desired must be entered.

Brief description of the KT 60 commands permitted when using
function block KT_ZUST:

1000 Command status register
This command checks the status of the 07 KT 60
command memory. The KT status read is stored
in the word flag which is entered on the STAT
input of the KT_ZUST.
Before giving the 07 KT 60 a command to output
data (commands: 1200 - 1204), the status of
the command memory must be interrogated to see
whether the 07 KT 60 is free for a new
command.

Status = 0000 H: Command memory empty, com-
mand can be carried out.
Status = 00FF H: (corresponds to 255 decimal)
Command memory full, com-
mand cannot be carried out.

1001 Status of serial communication
(07 KT 60 <-> terminal)
This command checks the status of serial
communication between the 07 KT 60 and a
terminal (printer) connected to the serial
interface.
The status determined is stored in the word
flag which is entered on the STAT input of
the KT_ZUST.

Status = 0000 H: Communication running.
Status = 0100 H: (corresponds to 256 decimal)
Communication ended
normally.
Status = 0102 H: (corresponds to 258 decimal)
Communication error
Status = 0103 H: (corresponds to 259 decimal)
Time-out error
Status of 1200 BUS communication
(07 KT 60 <-> ZE)
This command checks the status of BUS
communication between the 07 KT 60 and the T200
central unit.
The status determined is stored in the word
flag which is entered on the STAT input of
KT_ZUST.

Status = 0000 H: Communication ended
normally.

Status = 0001 H: Data transfer error

Status = 0003 H: Time-out error

Status = 0005 H: Data error

Status = 0006 H: Overflow of 07 KT 60
command memory.

Status = 0007 H: Parameter erroneous.

Reset communications error, clear KT 60 error
display.
This command resets communications errors
which have occurred (see command codes: 1001,
1002) and clears the 7-segment display of the
07 KT 60.
The input STAT of the CE's KT_ZUST must be
occupied by a word flag, but no status value
is assigned (with command 1010) to this
word flag (dummy).

Example

FBD

IL

```
KT_ZUST
E 0.00,00  FREI
EWO.05,00  KTA
MW0.00,00  STAT
MW0.00,01  RMA
MW 1010  INST
MW0.01,00  MW_2
RDY  A 0.01,00
ERR  A 0.01,01
FenR  MW0.00,15
```

```
IBA 00
FB 34 KT_ZUST
EP  E 0.00,00
EPA  EWO.05,00
EPA  MW0.00,00
EPA  MW0.00,01
EP  MW 1010
EPA  MW0.01,00
AP  A 0.01,00
AP  A 0.01,01
AP  MW0.00,15
```
Function: LIFO, word

Brief description

This block is a stack for word data from which the data written last is read first.

<table>
<thead>
<tr>
<th>RWB</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>!BA 000 NR.</td>
<td></td>
</tr>
<tr>
<td>FB 0030 LIFO</td>
<td></td>
</tr>
<tr>
<td>EP PP 000 LASE</td>
<td></td>
</tr>
<tr>
<td>EP PP 000 R</td>
<td></td>
</tr>
<tr>
<td>EP PP 000 E</td>
<td></td>
</tr>
<tr>
<td>EP PP 000 #L</td>
<td></td>
</tr>
<tr>
<td>EPA PP 000 ANF</td>
<td></td>
</tr>
<tr>
<td>EPA PP 000 W+04</td>
<td></td>
</tr>
<tr>
<td>AP PP 000 A</td>
<td></td>
</tr>
<tr>
<td>AP PP 000 FST</td>
<td></td>
</tr>
<tr>
<td>AP PP 000 L</td>
<td></td>
</tr>
<tr>
<td>AP PP 000 V</td>
<td></td>
</tr>
</tbody>
</table>

Parameters

<table>
<thead>
<tr>
<th>LADE</th>
<th>BINARY E,E',A,A',M,M', Load LIFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>LESE</td>
<td>BINARY E,E',A,A',M,M', Read LIFO</td>
</tr>
<tr>
<td>R</td>
<td>BINARY E,E',A,A',M,M', Reset the block with a</td>
</tr>
<tr>
<td></td>
<td>S,T,Z 0/1 edge</td>
</tr>
<tr>
<td>E</td>
<td>WORD EW,EW',AW,AW', Value to be read</td>
</tr>
<tr>
<td>#L</td>
<td>WORD EW,EW',AW,AW', Length of the LIFO</td>
</tr>
<tr>
<td>ANF</td>
<td>WORD AW,AW',MW,MW'</td>
</tr>
<tr>
<td>W+04</td>
<td>WORD AW,AW',MW,MW'</td>
</tr>
<tr>
<td>A</td>
<td>WORD AW,AW',MW,MW'</td>
</tr>
<tr>
<td>FST</td>
<td>WORD AW,AW',MW,MW'</td>
</tr>
<tr>
<td>L</td>
<td>BINARY A,A',M,M',S, Empty display</td>
</tr>
<tr>
<td>V</td>
<td>BINARY A,A',M,M',S, Full display</td>
</tr>
</tbody>
</table>

CE data

07 E 60
07 E 61
07 E 62

Basic runtime: 119 µs
Additional runtime: 74 µs

Updating the outputs: YES

Memory allocated once when called: 108 words at 32 bits each

Available as of: ABB Procentic 1200
907 PC 321 & 907 PB 360

Description

LADE BINARY Load LIFO
- LADE = 1 --> The value located at the input E is read into the LIFO.
  If the load signal is applied for more than 1 cycle, the value E is read in during each cycle until the LIFO signals "Full".
- LADE = 0 --> No value is read.

LESE BINARY Read LIFO
- LESE = 1 --> The value read first is output through the output A.
  If the read signal is applied for more than one cycle, a value is read out during each cycle until the LIFO signals "Empty".
- LESE = 0 --> No value is read out.
R BINARY Reset of the block
- R = 1 --> A 0/1 edge resets the LIFO.
  Values read previously are no longer available.
- R = 0 --> No change of the LIFO's contents

E WORD New value to be read

#L WORD Length of the LIFO;
  0 is not allowed as an input.

ANF WORD Initial address of the LIFO;
  #L variables are assigned as of this address.

W+04 WORD Parameter specifying the first auxiliary word flag.
  5 consecutive variables are assigned.
  These must no longer be used in the remaining program.

The following statuses can be interrogated at the outputs:

A WORD Output of the first value read when LESE=1
  The value is present for one PLC cycle

FST WORD Occupancy display of the LIFO

L BINARY Empty display of the LIFO
  No read out is possible if L=1

V BINARY Full display of the LIFO
  Values cannot be read in if V=1

Loading and reading:
If a load and read signal are applied simultaneously,
the value located at E is shifted directly through to A.
The occupancy is not changed. The output LEER remains 1
if the LIFO is empty.

Reset and load:
If a reset and load signal are present simultaneously,
the RESET is executed first, followed by LOADING.

Reset and loading and reading:
If a reset, a load and a read signal are applied simultaneously,
the RESET will be executed first.
The value located at E will then be shifted directly through to A.
The occupancy will be set to 0.

Example

```
 IBA 000 NR.
 FB 0030 LIFO
 EP M0000,00
 EP M0000,01
 EP M0000,02
 EP M0000,04
 EP M0000,05
 EP #W
 M0000,000,00
 M0000,01 LESE
 M0000,02 R
 M0000,03 E
 M0000,04 ANF
 M0000,05 W+04 A
 M0000,05,01 EPA M0000,02
 M0000,05,03 EPA M0000,04
 M0000,05,04 EPA M0000,05
 M0000,05,00
 M0000,05,01
 M0000,05,02
 M0000,05,03
 M0000,05,04

The flags M0000,00 ... M0000,04 are assigned the values read.
The flags M0000,00 ... M0000,04 are assigned as auxiliary flags.
```
Function: List allocator

Brief description

This block has a list of word variables at its input. With a list pointer, it selects a word value out of the list and outputs it through its output.

The list pointer is specified at the input ZEIG.

The list pointer starts with the value 0, i.e., it points to the initial value of the list.

The number of word variables is specified at the input \( n \).

The first word variable of the list is specified at the input \( A=\#E \). The word variables of the list use up \( \#n \) variables. These must be located in successive order.

Whether the list pointer is located within the valid area is indicated at the output \( A=\#E \).

If the list pointer is located within the valid area, the value of the word variable to which the pointer is pointing will be output through the output \( A \).

Program:

```
    IL
    LIZU  !8A 000 NO.
    ZEIG  FB 0031 LIZU
    #n  A=\#E
    #E  A
    EPA PP 000 #n
    EPA PP 000 #E
    EPA PP 000 A
    AP PP 000 A=#E
```

Parameters:

- **ZEIG** WORD: \( EW, EW', AW, AW' \) List pointer
- **\( \#n \)** WORD: \( EW, EW', AW, AW' \) Number of word variables
- **\( \#E \)** WORD: \( AW, AW', MW, MW' \) First variable of the list
- **A=\#E** BINARY: \( A, A', M, M', S \) Pointer is valid
- **A** WORD: \( AW, AW', MW, MW' \) Selected word variable

CE data:

- **07 ZE 60**
- **07 ZE 61**
- **07 ZE 62**

Basic runtime: 34 μs 34 μs 19 μs
Additional runtime: - - -

Updating outputs: YES

Memory allocated once when called: 25 words of 32 bits each

Available as of: ABB Proconic 1200
- 907 PC 321 & 907 PB 360

Description:

- **ZEIG** WORD: List pointer
  - Whether the value is greater than/equal to 0 and less than \( \#n \) is monitored.
  - If the pointer is valid, a 1 signal is output through the output \( A=\#E \) and the corresponding value is output through the output \( A \).
- **\( \#n \)** WORD: Number of list values
- **\( \#E \)** WORD: 1st variable of the list
  - All variables must be located in consecutive order.
  - A total of \( \#n \) variables is needed.
- The following statuses can be interrogated at the outputs:
  - **A=\#E** BINARY: Pointer is within the valid range.
  - **A** WORD: Output of the variables to which the list pointer is pointing.
; Example

; LIZU
; M$000,00 ZE1G
; M$0003,00 A=#$0000,00
; M$0003,00 A="$0000,00"

; The flags MW 0003,00 ... MW xxxxx,xx are used as a list
; depending on #n (in this case $W$).
Function: Proportional-integral controller

Call in FBD: PI

Description

PI controller for a control loop with set and reset input.

FBD

```
  PI          # BA 00
  w          # FB 012 PI
  x          # EP w
  KP         # EP KP
  TN/T       # EP TN/T
  S          # EP S
  INIT       # EP INIT
  R          # EP R
  MW         # EPA MW
  y          # AP y
```

Parameters

w WORD EW,W,M,W,MW,MW' Command variable input
x WORD EW,E,M,M,W,MW,MW' Controlled variable input
KP WORD EW,E,M,W,MW,MW' Proportional coefficient input
TN/T WORD EW,E,M,W,MW,MW' Reset time input
S BINARY E,EW,E,M,M',#B1,A,A',S,T,2 Set to initial value input
INIT WORD EW,E,M,W,MW,MW' Input for initial value
R BINARY E,E,M,M',#B1,A,A',S,T,2 Reset to 0 input
MW ADDRESS MW,MW' Flag area start address
y WORD AW,AW',M,W,MW' Correcting variable output

CK data

Basic runtime:
- S and R = 0: 2877 μs 2877 μs 1498 μs
- S or R = 1: 2932 μs 2932 μs 1527 μs

Additional runtime: - - -

Updating of outputs: YES

Memory allocated once when called: 303 words of 32 bits each

Available as of: ABB Prosonic 1200
907 PC 332 and
Block library expansion 907 PB 361

Description

The PI controller modifies its output y (correcting variable) until the input x (controlled variable) is equal to the input w (command variable).

KP WORD Proportional coefficient input
The proportional coefficient is specified as a % at the input KP.
e.g. 50 % = #W00050, 25 % = #W00025 etc.

TN/T WORD Reset time input
The reset time TN must be scaled to the PLC cycle time
and must be specified at the input TN/T (0 < TN/T <= 32767).

S BINARY Set to initial value input
The output y (correcting variable) can be set by
a 1 signal at the binary input S to the initial value
at the word input INIT

R BINARY Reset to 0 input
The output y (correcting variable) can be set
by a 1 signal at the binary input R to the value 0
MW ADDRESS Initial address of flag area
2 word flags (YALT and REST) are reserved. Both addresses must
be located one after the other. The first address is specified
at the input MW.

Y WORD Correcting variable output
The 1 component (YALT) must be limited externally if
the controller's output is to be limited.

The simple rectangle rule is used for integration.

Example

FBDO

I1

MW 000,00 \rightarrow w
MW 000,01 \rightarrow x
MW 000,02 \rightarrow KP
MW 000,03 \rightarrow TN/T
M 000,00 \rightarrow S
# W 100 \rightarrow INIT
M 000,01 \rightarrow R
MW 010,00 \rightarrow MW
Y \rightarrow MW 000,04

LAB 00
FBO 012 PI
EP MW 000,00
EP MW 000,01
EP MW 000,02
EP MW 000,03
EP M 000,00
EP # W 100
EP M 000,01
EPA MW 010,00
AP MW 000,04
;function: READ COMMUNICATION MODULE 07 Po 60
;
;brief description
;
;This block serves to read a single-axis positioning controller 07 Po 60.
The communication block Po_WR is needed to write into it.
The actual position is indicated as a double word;
only possible with 907 PC 32

;FBD
;
; IL

PO RD
FREI
PO-A
MW'A
KoEr
Bet0
Acbe
End-
End-
Scab
Pose
Saua
StaQ
Stzu
Scau
Verw
MFkt
FLA1
FLA2
FLA3
BTFL
Pols
FeNr
RAW1
RAW2
RAB1
RAB2

;Parameters
;
; FREI  BINARY E,AM,M',MB  Block enable
; PO-A  Address AW
; MW'A  Address MW'
; KoEr  BINARY A,M,M'  Communication error
; Bet0  WORD AM,M,W,MW  Acknowledgment mode
; Acbe  BINARY A,M,M'  Axis ready
; End-  BINARY A,M,M'  End position for the positive direction
; End-  BINARY A,M,M'  End position for the negative direction
; Scab  BINARY A,M,M'  Max. following error exceeded
; Pose  BINARY A,M,M'  Position reached
; Saua  BINARY A,M,M'  Set executed
; StaQ  BINARY A,M,M'  Start acknowledgment for NC set
; Stzu  BINARY A,M,M'  Start status
; Scau  BINARY A,M,M'  Loop executed
; Verw  BINARY A,M,M'  Dwell time
; MFkt  WORD AW,MW,MW'  Machine function
; FLA1  BINARY A,M,M'  FLAG1
; FLA2  BINARY A,M,M'  FLAG2
; FLA3  BINARY A,M,M'  FLAG3
; BTFL  WORD AW,MW,MW'  8 section flags
; Pols  DOUBLE AD,MD,MD'  Actual position of the axis in
; WORD  absolute dimensions
; FeNr  WORD AW,MW,MW'  Error number
; RAW1  WORD AW,MW,MW'  Reserve output word 1
; RAW2  WORD AW,MW,MW'  Reserve output word 2
; RAB1  BINARY A,M,M'  Reserve output bit 1
; RAB2  BINARY A,M,M'  Reserve output bit 2
CE data

Runtime:
Function block PO_RD in IL: 800 µs 370 µs
Basic runtime of the CE: 1000 µs 480 µs

Updating of outputs: yes if KoEr = 0

Available as of: ABB Proconic T200
907 PC 332 and
block library extension 907 PB 362

Memory allocated once when called: 200 words of 32 bits each

Number of flags from the
MW area allocated: 22 beginning with the start address specified at MW'A
The CE additionally allocates flags MW'359,00 and 359,01.

Important:
The 07 PO 60 I/O addresses must only be addressed by the program
via the blocks PO_RD and PO_WR. Different addressing in the program will lead
to communication errors.
Using PO_RD and PO_WR, the same variables must be allocated to
the inputs 'FREI', 'PO-A', 'MW'A', and 'KoEr'.

Description

This block serves to control a single axis positioning controller
07 PO 60 and also requires the communication block PO_WR for this purpose.
The block PO_RD reads the current values of the 07 PO 60 bus outputs
and provides them to the PO_RD block outputs. The currently valid
image of the 07 PO 60 bus outputs is obtained in this way.

Exception: After the block call of PO_RD KoEr = 1.
In this case, the values of the block outputs do not agree
with the 07 PO 60 bus outputs.

Refer to the 07 PO 60 system description for further information.
The diskette 907 PB 361 also contains an example program
(PO60BE.P) which shows use of the connection elements.

FREI BINARY: Block enable
FREI = 0 -- No communication of the blocks PO_RD and PO_WR
with the corresponding 07 PO 60 module takes place.
-- Block output 'KoEr'=1 (see application notes).
FREI = 1 -- The block is enabled.

PO-A ADDRESS: Address of (AW x.xx,04)
Address of the first output (AW x.xx,04) of 07 PO 60

MW'A ADDRESS: Start address of (MW'xxx,xx)
Address of the first word flag (MW'xxx,xx) of the flag
area reserved for communication (22 word flags from
the area MW'xxx,xx)

KoEr BINARY: Communication error
KoEr = 1 -- Communication to this 07 PO 60 has not yet been
set up or is disturbed. It may also be interrupted
by FREI = 0 (see application notes).
KoEr = 0 -- The 07 PO 60 can be addressed correctly via the bus.
The current image of the 07 PO 60 outputs is available
at the PO_RD outputs.

BetQ WORD: Indicates the current mode of the 07 PO 60.
Value range 0 - 3.
0 -- No external mode
1 -- External Automatic Single Set
2 -- External Automatic Follow Up Set
3 -- External mode

Acbe BINARY: Axis ready
Acbe = 0 -- Positioning of the 07 PO 60 not possible
in external mode or interrupted.
Acbe = 1 -- Positioning of the 07 PO 60 allowed in
external mode or not interrupted.
END+ BINARY: End position for the positive direction
    End+ = 1 --> End position for the positive direction travelled to.
    End+ = 0 --> End position for the positive direction not travelled to.

END- BINARY: End position for the negative direction
    End- = 1 --> End position for the negative direction travelled to.
    End- = 0 --> End position for the negative direction not travelled to.

Scab BINARY: Maximum following error exceeded.
    Scab = 1 --> Maximum following error has been exceeded.
    Scab = 0 --> Maximum following error has not been exceeded.

Pose BINARY: Position reached
    Pose = 1 --> Axis within the target window
    Pose = 0 --> Axis not within the target window

Sauw BINARY: Set executed
    Sauw = 1 --> after connection of the supply voltage
             after execution of an NC set
             after aborting of a positioning set with
             07 PO 60, enable initiator = 0, in enable
             mode 2 or 3.
    Sauw = 0 --> after starting an NC set

StaQ BINARY: Start acknowledgment
    After starting of an NC set, StaQ is set to 1 for
    one PLC cycle. With an NC set having an execution
    time less than the cycle time, this signal and the 'Sauw'
    signal can be used to determine that the NC set has
    been executed.

St2u BINARY: Start status
    In the external mode, 'St2u' reflects the 07 PO 60's internal
    status of the start signal 'Star' in order to synchronize
    set execution in high-speed PLC programs (see example
    program PO60BSP* on the delivered 907 PO 362 diskette)

Scau BINARY: Loop executed
    Execution of a loop comprising several NC sets
    is defined in a loop set.
    Scau = 0 --> Loop set has been called and the loop is still
             being executed.
    Scau = 1 --> No loop set is called or the loop has been
             executed.

Verwa BINARY: Dwelling time elapsed
    Verwa = 0 --> Indicates 'NC set has been started'
    Verwa = 1 --> Indicates 'NC set executed and dwelling
                  time elapsed'

Mfkt WORD: Machine function
    Value range: 0-999

FLA1 BINARY: Programmed flag position 1 travelled over.
    FLA1 = 1 --> Programmed flag position 1 travelled over.
    FLA1 = 0 --> Programmed flag position 1 not travelled over.

FLA2 BINARY: Programmed flag position 2 travelled over.
    FLA2 = 1 --> Programmed flag position 2 travelled over.
    FLA2 = 0 --> Programmed flag position 2 not travelled over.

FLA3 BINARY: Programmed flag position 3 travelled over.
    FLA3 = 1 --> Programmed flag position 3 travelled over.
    FLA3 = 0 --> Programmed flag position 3 not travelled over.

BTFL WORD: 8 section flags
    After 1/8 of the set's travel distance in each case,
    one section flag is set to 1.

    Bit 0 --> Section flag 1 travelled over.
    Bit 1 --> Section flag 2 travelled over.
    Bit 2 --> Section flag 3 travelled over.
    Bit 3 --> Section flag 4 travelled over.
    Bit 4 --> Section flag 5 travelled over.
    Bit 5 --> Section flag 6 travelled over.
    Bit 6 --> Section flag 7 travelled over.
    Bit 7 --> Section flag 8 travelled over.

Pols DOUBLE WORD: Actual position value of the axis in absolute dimensions.
FeNr WORD: Error number
  Indication of the 07 PO 60 error number (e.g.: ‘31’ -->
  travel to software limit switch)
  New error numbers overwrite the previous ones.
  The error number OFFFH signifies an unidentifiable
  error.
  After power on, the error number is set to 0.
  Note: Delete the error number through input ‘Fbbo’=1
  on PO_WR

RAW1 WORD: Reserve word is read by 07 PO 60.

RAW2 WORD: Reserve word is read by 07 PO 60.

RAB1 BIN: Reserve bit is read by 07 PO 60.

RAB2 BIN: Reserve bit is read by 07 PO 60.

Example

FBD

IL

E 0,00,00 FRE1
; MW 04,04 PO-A
; MW 010,00 KOER
  A 0,02,00
  MW 120,02
  M 011,01
  M 011,02
  M 011,03
  M 011,04
  M 011,05
  M 011,06
  M 011,07
  M 011,15
  M 011,08
  M 011,09
  M 0120,00
  M 011,10
  M 011,11
  M 011,12
  MW 0120,01
  MW 0130,00
  MW 0121,00
  MW 0122,00
  MW 0122,01
  M 011,13
  M 011,14
  A 0,02,00

Application notes

- Arrangement of PO_Rd, PO_WR in the PLC program
  One PO_Rd and PO_WR block each is placed in the PLC program
  to control one 07 PO 60.
- Several PO_Rd/PO_WR block calls can be executed in one PLC
  program (in each case for other 07 PO 60 modules with different
  07 PO 60 addresses (PO-A)). The number depends on the chosen
  configuration.
- Each call of PO_WR must be placed after the PO_Rd call.
- The PLC evaluation program for a specific 07 PO 60 module should be
  placed after the pertinent PO_Rd block call and before the PO_WR
  block call.
- In long PLC programs, the PO_Rd and PO_WR blocks can also be
  arranged multiply in the program, with the same address at
  'PO-A' (but always in pairs).

PLC evaluation program for one 07 PO 60 module

- The PLC evaluation program for one 07 PO 60 module evaluates the
  07 PO 60 bus outputs, which are available as PO_Rd block outputs.
  After calling the PO_Rd block, the PO_Rd block outputs are the
  current image of the 07 PO 60 outputs (exceptions: 'KOER'=1)
  Depending on the status of these block outputs ('Acknowledgement mode',
  'Set executed',....) the user can operate the 07 PO 60 module
  via the PO_WR block inputs (e.g. mode select, start NC set,....)
Block enable ("FRE")

- Normally (no commissioning), the block enable should be set continuously for all P0_RD and P0_WR blocks.
- Resetting the block enable ("FRE" = 0) immediately suppresses access by "P0_RD" and "P0_WR" to the 07 P0 60 module allocated to them.
- Reset of the block enable of a P0 block always results in setting of the communication error ("KoEr") block output.

- Reset of block enable for one block pair comprising P0_RD and P0_WR does not affect communication between other block pairs and the 07 P0 60 modules allocated to them.
- Once the block output "KoEr" has been set and the corresponding block enable input has again assumed the value 1, the P0_RD block restores communication with the 07 P0 60 module and acknowledges connection establishment with a renewed 1 --> 0 changeover of the output signal "KoEr".

Evaluation of the block output "KoEr"

- The output "KoEr" = 1 after every PLC program start.
- The 1 --> 0 changeover of the "KoEr" output after the P0_RD block call acknowledges the PLC --> 07 P0 60 connection establishment.
- The "KoEr" output is set or reset both by the P0_RD and also by the P0_WR blocks. The values are transferred via the common flag area MW1xxx,xx ("MW1A").
- If the output "KoEr" is set again after acknowledgment of connection establishment, then communication with the affiliated 07 P0 60 module is disturbed (e.g. 07 P0 60 is disturbed, 07 P0 60 single board reset, block enable inhibited). In this case (prerequisite: block enable set), the P0_RD block again attempts to establish communication with the 07 P0 60 module and, after successful connection establishment, acknowledges this with a renewed 1 --> 0 changeover of the output "KoEr".
- The reaction to a single board reset of a 07 P0 60 module can be planned in the PLC program. Therefore, deactivation and renewed starting of the PLC program are not necessary.
- The user himself can plan evaluation of the "KoEr" signal and the desired reaction to this in his PLC program. This reaction generally depends on the momentary operation state and the system configuration etc.
- For example, the reaction may be emergency shutdown of the overall plant or the PLC program might only be placed in a specific state from which operation of the plant can be continued.

Diagnostic outputs

- Limiting input ("EiBe"):
  This output indicates that a value of an input planned for the P0_WR block lay outside of the permissible value range and was limited by the P0_WR block or set to the default before transferred to the 07 P0 60 module. The "EiBe" output is set in latching mode and can only be reset as follows:
  - By aborting and then restarting the PLC program
  - By resetting the flag planned at the output with =R 'Flag'.

Block output start acknowledgement ("StaQ"):

- At the block output 'Start Quit' = 1, the user recognizes that 07 P0 60 set execution has been started in the previous PLC cycle. The output assumes status 1 for only one PLC cycle. This also applies if set execution lasts longer than one PLC cycle.
- The block output 'StaQ' is not set to 1 if positioning is continued after aborting a positioning operation (e.g. by pressing the stop pushbutton on the operating unit 35 AB 50). A positioning operation aborted by pressing the stop pushbutton is continued by pressing the start pushbutton and additionally by a start signal 'Star' of the P0_WR block.

Problem when evaluating the block output 'Set executed' ("Saau"):

During the P0_WR block call, the planned PLC input flags are transferred to the 07 P0 60 and 07 P0 60 set execution is started, for instance. During the next PLC cycle, the user recognizes the start of set execution, e.g. by virtue of:
"Saau" = 0.
The user can recognize the end of set execution by virtue 
of the fact that 'Saau = 1'.
If, however, the set execution time is very short (less 
than the PLC cycle time), the output remains Saau = 1.
In this PLC cycle, the user can then recognize by virtue of the 
signal 'Stao = 1' that set execution has been started and has 
already been concluded.
It is therefore recommended to always evaluate both signals ('Saau' 
and 'Stao') in the PLC evaluation program (see also example program).

Allocation of the flag area used by PO_RD and PO_WR

PO status used by both blocks

Start address + 0

MW'xxx,00 = PO 60 status
M' xxx,00 = Init requested
M' xxx,01 = Axis Init
M' xxx,02 = Read PO
M' xxx,03 = Write PO
M' xxx,04 = Wait
M' xxx,05 = Input limiting in the past
M' xxx,06 = Free
M' xxx,07 = ERROR

Flags for PO_RD: These flags are written with the data of the selected 
07 PO 60 module by PO_RD.

Start address + 1

MW'xxx,01 = PO 60 status
M' xxx,16 = Axis ready
M' xxx,17 = End position +
M' xxx,18 = End position -
M' xxx,19 = Following error
M' xxx,20 = Position reached
M' xxx,21 = Set executed
M' xxx,22 = Start acknowledgement
M' xxx,23 = Loop executed
M' xxx,24 = Dwell time
M' xxx,25 = Mode acknowledgement 1
M' xxx,26 = Mode acknowledgement 2
M' xxx,27 = Start status

Start address + 2

MW'xxx+1,00 = M functions
M' xxx+1,13 = Flag 1
M' xxx+1,14 = Flag 2
M' xxx+1,15 = Flag 3

Start address + 3

MW'xxx+1,01 = Section flag
M' xxx+1,16 = Section flag 1
M' xxx+1,17 = Section flag 2
M' xxx+1,18 = Section flag 3
M' xxx+1,19 = Section flag 4
M' xxx+1,20 = Section flag 5
M' xxx+1,21 = Section flag 6
M' xxx+1,22 = Section flag 7
M' xxx+1,23 = Section flag 8

Start address + 4

MW'xxx+2,00 = Position value (low word)

Start address + 5

MW'xxx+2,01 = Position value (high word)

Start address + 6

MW'xxx+3,00 = Reserve word 1
; Start address + 7
; MW'xxx+3,01 = Reserve word 2
; Start address + 8
; MW'xxx+4,00 = Error number
; Start address + 9
; MW'xxx+4,01 = Reserve bits
; Flags for PO WR : The contents of these flags are written into
; the selected 07 PO 60 module by PO_WR.
; Start address + 10
; MW'xxx+5,00 = Set number
; Start address + 11
; MW'xxx+5,01 = Modes
; MW'xxx+5,16 = Mode selection 0
; MW'xxx+5,17 = Mode selection 1
; MW'xxx+5,18 = Start
; MW'xxx+5,19 = Teach in
; MW'xxx+5,20 = Enable mode bit 0
; MW'xxx+5,21 = Enable mode bit 1
; MW'xxx+5,22 = Delete error number
; Start address + 12
; MW'xxx+6,00 = User program segment
; Start address + 13
; MW'xxx+6,01 = Override selection
; Start address + 14
; MW'xxx+7,00 = Override value
; Start address + 15
; MW'xxx+7,01 = Length compensation selection
; Start address + 16
; MW'xxx+8,00 = Length compensation (low word)
; Start address + 17
; MW'xxx+8,01 = Length compensation (high word)
; Start address + 18
; MW'xxx+9,00 = Reserve word 1
; Start address + 19
; MW'xxx+9,01 = Reserve word 2
; Start address + 20
; MW'xxx+10,00= Reserve bit 1
; Start address + 21
; MW'xxx+10,01= Free
### Function:
WRITE COMMUNICATION BLOCK 07 PO 60

**Brief description**

This block serves to write into a 07 PO 60 module. The communication block PO RD is needed for reading. The length compensation is entered as a DOUBLE WORD value. Only possible with 907 PC 32.

<table>
<thead>
<tr>
<th>PAD</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBA</td>
<td>FB 026</td>
</tr>
<tr>
<td>EP</td>
<td>FREI &lt;Bit variable&gt;</td>
</tr>
<tr>
<td>EPA</td>
<td>PO-A &lt;AW x.xx,04&gt;</td>
</tr>
<tr>
<td>EPA</td>
<td>MU/A &lt;MU x.xx,xx&gt;</td>
</tr>
<tr>
<td>AP</td>
<td>KoEr &lt;Bit variable&gt;</td>
</tr>
</tbody>
</table>

#### Parameters:
- **FREI** BINARY E,A,M,M',#B: Block enable
- **PO-A** Address AW: Aw x.xx,04 start address of 07 PO 60
- **MU/A** Address MW: Start address of the flag area
- **BeAw** WORD EW,AW,MW,MW',#W: 22 word flags are allocated
- **SaAw** WORD EW,AW,MW,MW',#W: Start address of the flag area used in NC set selection of the 07 PO 60
- **AnPS** WORD EW,AW,MW,MW',#W: Selection of the 07 PO 60's user program segment
- **StEn** BINARY E,A,M,M',#B: Start of an NC set
- **TEn** BINARY E,A,M,M',#B: Start of an NC set
- **FeLo** BINARY E,A,M,M',#B: Delete error number
- **OvAw** WORD EW,AW,MW,MW',#W: Speed override selection
- **OvDe** WORD EW,AW,MW,MW',#W: External override value
- **LsAw** WORD EW,AW,MW,MW',#W: Length compensation selection
- **LsWe** DOUBLE ED,AD,MD,MD',#D: External length compensation value
- **WOWD** WORD EW,AW,MW,MW',#W: External length compensation value
- **FmMo** WORD EW,AW,MW,MW',#W: Enable node
- **ReH1** WORD EW,AW,MW,MW',#W: Reserve input word 1
- **ReH2** WORD EW,AW,MW,MW',#W: Reserve input word 2
- **ReB1** BINARY E,A,M,M',#B: Reserve input bit 1
- **ReB2** BINARY E,A,M,M',#B: Reserve input bit 2
- **KoEr** BINARY E,A,M,M': Communication error
- **ElBe** BINARY E,A,M,M': Input limiting, latching output

#### CE data
- **Runtime:** 07 ZE 60/61  07 ZE 62
- **Function block in IL:** 7.5 µs  348 µs
- **Basic runtime of the CE:** 1097 µs  541 µs

#### Updating of the outputs: yes
- **Memory allocated once when called:** 102 words of 32 bits each
- **Available as of:** ABB Precontrol 7200
  - 907 PC 332 and
  - block library expansion 907 PO 362

**Description**

This block serves to control a 07 PO 60 module. It also requires the PO RD communication block. The PO_WR block writes the inputs on the CE PO_WR to the 07 PO 60 bus inputs selected via the PO address.
Exception: After the block call of PO_WR KoEr = 1.
In this case, the values of the block outputs do not agree with the 07 PO 60 bus inputs.

FREI BINARY: Block enable
FREI = 0 --> No communication of the PO_RD and PO_WR blocks with the corresponding 07 PO 60 module takes place.
--> Block output 'KoEr' = 1' (see application notes).
FREI = 1 --> Communication takes place between the PO_RD and PO_WR blocks and the affiliated 07 PO 60 module.

PO-A Address: AW x.xxx,04 start address of the 07 PO 60
Address of the first output (AW x.xxx,04) of 07 PO 60

MWA Address: Start address of the flag area
Address of the first word flag (MWAxxx,xx) of the flag area reserved for communication.
(22 word flags from the area MWAxxx,xx)

BRAW WORD: Selection of the 07 PO 60's mode
Value range: 0-3
0 --> No external mode
1 --> External automatic single set
2 --> External automatic follow-up set
3 --> External mode
if 'BRAW' > 3 or 'BRAW' < 0
PO_WR sets the default: 'BRAW' = 0 and the block output 'EIBe' = 1 latches to 1

SAnR WORD: Selection of the NC set of the 07 PO 60.
Value range: 0-255
if 'SAnR' > 255 or 'SAnR' < 0
PO_WR sets the default: 'SAnR' = 255 and the block output 'EIBe' = 1 latches to 1

AMPS WORD: Specification of the user program segment for 07 PO 60
Value range: 0 ... 255
if 'AMPS' > 255 or 'AMPS' < 0
PO_WR sets the default: 'AMPS' = 0 and the block output 'EIBe' = 1 latches to 1

Star BINARY: Start an NC set

Teln BINARY: Acceptance of the momentary actual position as the position setpoint for the selected NC set.

FeLÖ BINARY: Deletion of an error number
An error number at the output 'FeNr' can be deleted with this input.
FeLÖ = 1 --> Deletion (setting to zero) of the error number at the output 'FeNr'.
Important: Continuous 1 signal --> no further error messages are possible.
FeLÖ = 0 --> No deletion of the error number.

DvAw WORD: Speed override selection
Value range: 0-3
if 'DvAw' > 3 or 'DvAw' < 0
PO_WR sets the default: 'DvAw' = 0 and the block output 'EIBe' latches to 1

DvWe WORD: External override value
Value range: 0-120
if 'DvWe' > 120 or 'DvWe' < 0
PO_WR sets the default: 'DvWe' = 120 and the block output 'EIBe' latches to 1

LdAw WORD: Length compensation selection
Value range: 0-4
if 'LdAw' > 4 or 'LdAw' < 0
PO_WR sets the default: 'LdAw' = 0 and the block output 'EIBe' latches to 1
LAWE DOUBLE WORD: External length compensation value
  Value range: -99999990 to +99999990
  The value is not limited. However, within the
  07 PO 60, the value is limited to
  - 9 999 999 to + 9 999 999.

FrMo WORD: Enable mode selection
  FrMo = 0 --> Selection of enable mode 1
  FrMo = 1 --> Selection of enable mode 2
  FrMo = 2 --> Selection of enable mode 3

REW1 WORD: Reserve word
  The reserve word is written to the 07 PO 60

REW2 WORD: Reserve word
  The reserve word is written to the 07 PO 60

REB1 BINARY: Reserve bit
  The reserve bit is written to the 07 PO 60

REB2 BINARY: Reserve bit
  The reserve bit is written to the 07 PO 60

KoEr BINARY: Communication with the affiliated 07 PO 60
  module is disturbed.
  KoEr = 1 --> Communication with the 07 PO 60 concerned
  has not yet been established or is disturbed.
  It may also be interrupted by FREI = 0 (see also
  application notes)
  KoEr = 0 --> The affiliated 07 PO 60 module can be addressed
  correctly. The current image of the 07 PO 60
  outputs is available at the PO_RD outputs.

EiBe BINARY: Diagnosis to the effect that a value concerning
  the axis lies or lay outside of the value range.
  EiBe = 1 --> Diagnosis to the effect that a value concerning
  the axis lies or lay outside of the value range.
  (See application notes for PO_RD)
  The output 'EiBe' can only be reset as follows:
  - by aborting and then restarting the PLC program.
  - by resetting the planned flag with = R <flag>
  EiBe = 0 --> The input values lie within the value range.

\example

FBD

IL

PO WR
FREI
PO-A
M0 010,00
M0 000,00
M0 000,01
M0 000,02
M0 003,00
M0 003,01
M0 003,02
M0 000,03
M0 000,04
M0 000,05
M0 001,00
M0 000,00
M0 000,06
M0 000,07
M0 003,03
M0 003,04

1BA
FB 026
EP E 0.00,00
EPA M0 04,06
EPA M0 010,00
AP A 0.02,00

Block application notes:
  See description of PO_RD
Allocation of the flag area used by PO_RD and PO_WR

- PO status used jointly by both blocks
- Start address = 0
  - MW\(\text{xxx},00\) = 07 PO 60 status
  - M\(\text{xxx},00\) = Init requested
  - M\(\text{xxx},01\) = Axis Init
  - M\(\text{xxx},02\) = Read PO
  - M\(\text{xxx},03\) = Write PO
  - M\(\text{xxx},04\) = Wait
  - M\(\text{xxx},05\) = Input limiting in the past
  - M\(\text{xxx},06\) = Free
  - M\(\text{xxx},07\) = ERROR

- Flags for PO_RD: These flags are written with the data of the selected 07 PO 60 module by PO_RD.
- Start address = 1
  - MW\(\text{xxx},01\) = 07 PO 60 status
  - M\(\text{xxx},16\) = Axis ready
  - M\(\text{xxx},17\) = End position +
  - M\(\text{xxx},18\) = End position -
  - M\(\text{xxx},19\) = Following error
  - M\(\text{xxx},20\) = Position reached
  - M\(\text{xxx},21\) = Set executed
  - M\(\text{xxx},22\) = Start acknowledgement
  - M\(\text{xxx},23\) = Loop executed
  - M\(\text{xxx},24\) = Dwell time
  - M\(\text{xxx},25\) = Acknowledgement mode 1
  - M\(\text{xxx},26\) = Acknowledgement mode 2
  - M\(\text{xxx},27\) = Start status
- Start address = 2
  - MW\(\text{xxx+1},00\) = M functions
  - M\(\text{xxx+1},13\) = Flag 1
  - M\(\text{xxx+1},14\) = Flag 2
  - M\(\text{xxx+1},15\) = Flag 3
- Start address = 3
  - MW\(\text{xxx+1},01\) = Section flag
  - M\(\text{xxx+1},16\) = Section flag 1
  - M\(\text{xxx+1},17\) = Section flag 2
  - M\(\text{xxx+1},18\) = Section flag 3
  - M\(\text{xxx+1},19\) = Section flag 4
  - M\(\text{xxx+1},20\) = Section flag 5
  - M\(\text{xxx+1},21\) = Section flag 6
  - M\(\text{xxx+1},22\) = Section flag 7
  - M\(\text{xxx+1},23\) = Section flag 8
- Start address = 4
  - MW\(\text{xxx+2},00\) = Position value (low word)
- Start address = 5
  - MW\(\text{xxx+2},01\) = Position value (high word)
- Start address = 6
  - MW\(\text{xxx+3},00\) = Reserve word 1
- Start address = 7
  - MW\(\text{xxx+3},01\) = Reserve word 2
- Start address = 8
  - MW\(\text{xxx+4},00\) = Error number
- Start address = 9
  - MW\(\text{xxx+4},01\) = Reserve bits
; Flags for PO_WR : The contents of these flags are written
; into the selected 07 PO 60 module by PO_WR.

; Start address + 10
; MW'xxx+5,00 = Set number
; Start address + 11
; MW'xxx+5,01 = Modes
; MW'xxx+5,16 = Mode selection 0
; MW'xxx+5,17 = Mode selection 1
; MW'xxx+5,18 = Start
; MW'xxx+5,19 = Teach In
; MW'xxx+5,20 = Enable mode bit 0
; MW'xxx+5,21 = Enable mode bit 1
; MW'xxx+5,22 = Delete error number

; Start address + 12
; MW'xxx+6,00 = User program segment
; Start address + 13
; MW'xxx+6,01 = Override selection
; Start address + 14
; MW'xxx+7,00 = Override value
; Start address + 15
; MW'xxx+7,01 = Length compensation selection
; Start address + 16
; MW'xxx+8,00 = Length compensation (low word)
; Start address + 17
; MW'xxx+8,01 = Length compensation (high word)
; Start address + 18
; MW'xxx+9,00 = Reserve word 1
; Start address + 19
; MW'xxx+9,01 = Reserve word 2
; Start address + 20
; MW'xxx+10,00 = Reserve bit 1
; Start address + 21
; MW'xxx+10,01 = Free