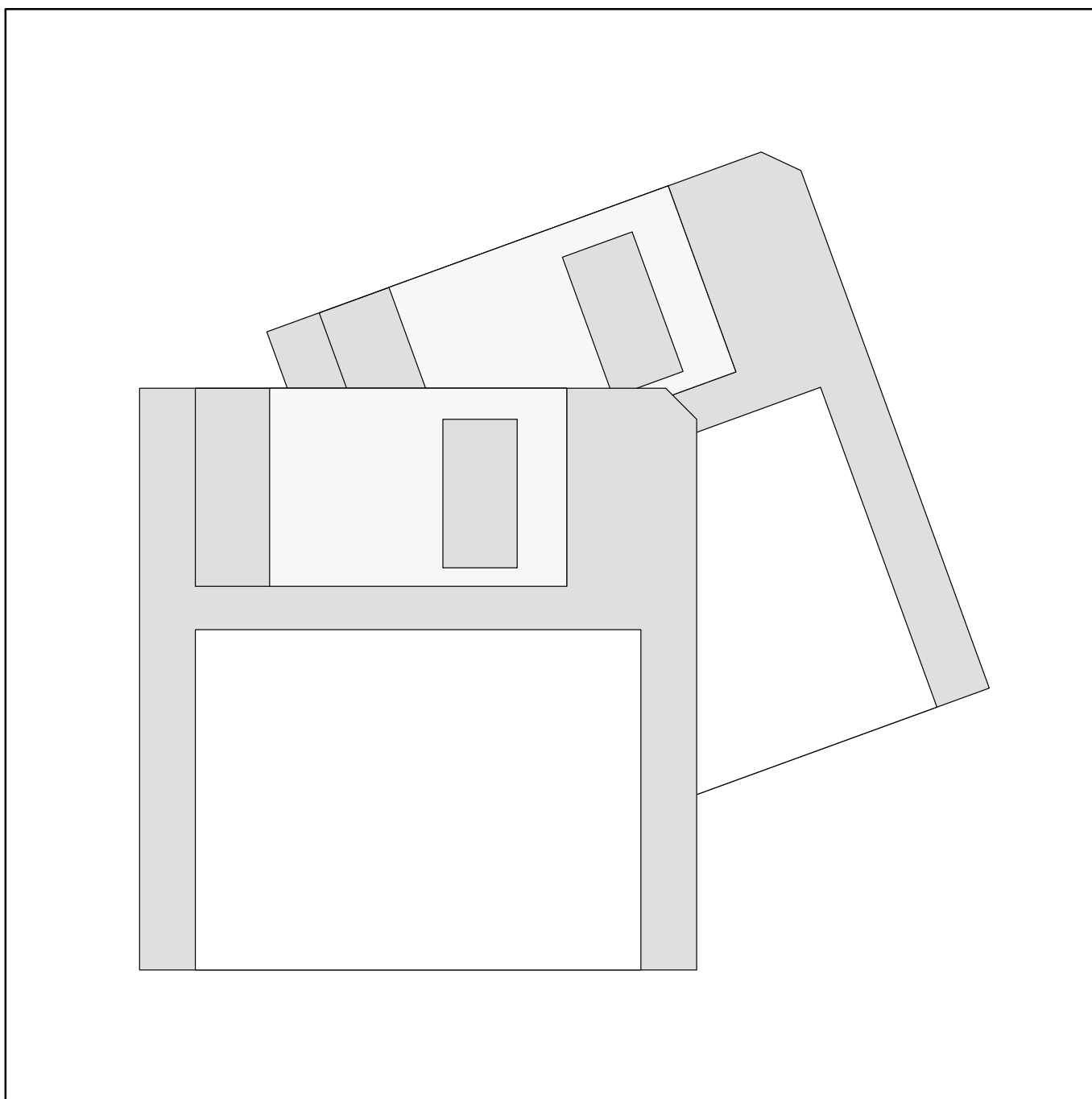


907 PC 331
Programming and Test Software



Contents

1	Operands Series 90 (07 KR 91, 07 KT 92, 07 KT 93, 07 KT 94), Serie 30 (07 KR 31, 07 KT 31) and 07 KP 62	
	(variables and constants)	1– 1
1.1	Freely available variables and constants	1– 1
1.2	System constants / diagnosis flags / CS31 status (overview)	1– 2
1.3	System constants / setting of operating modes	1– 3
2	PLC language repertoire	2– 1
2.1	Operators for configuring sentences (instruction list according to DIN 19239)	2– 1
2.2	Block calls	2– 1
2.3	NOP	2– 2
2.4	Jumps	2– 2
2.5	Labels	2– 2
2.6	Program end	2– 2
2.7	Configuration of time values	2– 2
2.8	Data types / formats	2– 2
2.9	Operand identifiers	2– 3
2.10	Numbering the operands (variables)	2– 3
2.11	Step chains	2– 3

1 Operands Series 90 (07 KR 91, 07 KT 92, 07 KT 93, 07 KT 94), Series 30 (07 KR 31, 07 KT 31) and 07 KP 62 (variables and constants)

1.1 Freely available variables and constants

Inputs Series 90

E 00,00...E 61,15	: Binary inputs, remote modules
E 62,00...E 63,03	: Binary inputs of the central unit 07 KR 91
E 62,00...E 62,11	: Binary inputs of the central unit 07 KT 92
E 62,00...E 63,07	: Binary inputs of the central unit 07 KT 93
E 63,14 and E 63,15	: High-speed binary inputs ($T_D = 0.02\text{ms}$), signal is identical to E 62,00 and E 62,01
E 63,13	: High-speed counter, interrogation of "Zero crossing"

EW 00,00...EW 05,15	: Analog inputs, remote modules
EW 06,00...EW 06,03	: Analog inputs of the central unit 07 KT 92
EW 06,15	: High-speed counter, interrogation of the counter content
EW 07,00...EW 07,07	: reserved
EW 07,08...EW 07,14	: Reading of the real-time clock
EW 07,15	: Status for CS31 system bus, clock, battery

Inputs Series 30

E 00,00...E 61,15	: Binary inputs, remote modules
E 62,00...E 62,11	: Binary inputs of the central unit 07 KR 31
E 62,00...E 62,11	: Binary inputs of the central unit 07 KT 31
E 63,14	: High-speed binary inputs ($T_D = 0.02\text{ms}$), signal is identical to E 62,00
E 63,13	: High-speed counter, interrogation of "Zero crossing"

EW 00,00...EW 05,15	: Analog inputs, remote modules
EW 08,00...EW 15,15	: Analog inputs, remote modules
EW 06,15	: High-speed counter, interrogation of the counter content
EW 07,00...EW 07,07	: reserved
EW 07,08...EW 07,14	: Reading of the real-time clock
EW 07,15	: Status for CS31 system bus, clock, battery

Inputs 07 KP 62

The module 07 KP 62 has no process I/Os.

EW 00,04...EW 00,07	: Inputs for communication with the ABB Procontic T200
---------------------	--

Outputs Series 90

A 00,00...A 61,15	: Binary outputs, remote modules
A 62,00...A 62,11	: Binary outputs of the central unit 07 KR 91
A 62,00...A 62,07	: Binary outputs of the central unit 07 KT 92
A 62,00...A 62,15	: Binary outputs of the central unit 07 KT 93
A 62,00	: High-speed counter, direct output of "Counter zero crossing" after activation
A 63,13...A 63,15	: High-speed counter, enabling, activation of E 62,01 and A 62,00, adoption of start value

AW 00,00...AW 05,15	: Analog outputs, remote modules
AW 06,00	: Analog outputs of the central unit 07 KT 92 ($-10\text{V}...+10\text{V}$)
AW 06,15	: High-speed counter, "Start value"

Outputs Series 30

A 00,00...A 61,15	: Binary outputs, remote modules
A 62,00...A 62,11	: Binary outputs of the central unit 07 KR 31
A 62,00...A 62,11	: Binary outputs of the central unit 07 KT 31
A 63,15	: High-speed counter, adoption of start value

AW 00,00...AW 05,15	: Analog outputs, remote modules
AW 06,15	: High-speed counter, "Start value"

Outputs 07 KP 62

The module 07 KP 62 has no process I/Os.

AW 00,00...AW 00,03 : Outputs for communication with the ABB Procontic T200

Internal operands Series 90

M 00,00...M 255,09 : Binary flags

S 00,00...S 127,15 : Steps

K 00,00...K 00,01 : Binary constants

MW 00,00...MW 253,15 : Word flags

KW 01,00...KW 39,15 : Word constants

MD 00,00...MD 31,15 : Double word flags

KD 00,01...KD 07,15 : Double word constants

Internal operands Series 30

M 00,00...M 21,15 : Binary flags

M 230,00...M 239,15 : Binary flags

S 00,00...S 15,15 : Steps

K 00,00...K 00,01 : Binary constants

MW 00,00...MW 05,15 : Word flags

MW 230,00...MW 239,15 : Word flags

KW 01,00...KW 07,15 : Word constants

MD 00,00...MD 01,15 : Double word flags

KD 00,01...KD 01,15 : Double word constants

Time values for time functions

KD yy,xx : Time values for time functions such as ESV, ASV etc. are configured as *double word constants* or as

MD yy,xx : *double word flags*. Only integral multiples of 5 ms are permitted.

1.2 System constants / diagnosis flags / CS31 status (overview)

Setting the operating modes

The constants KW 00,00...KW 00,15 are reserved as system constants. Even the constants KW 00,12...KW 00,15 which are not used yet may *under no circumstances* be used for other purposes.

Series 30, 40, 50, 90

KW 00,00 : Setting the PLC operating modes, (Stand-alone PLC, Master PLC, Slave PLC)

KW 00,01 : Initialization: bit flag area

KW 00,02 : Initialization: word flag area

KW 00,03 : Initialization: double word flag area

KW 00,04 : Initialization: step chain flag area

KW 00,05 : Initialization: historical values

KW 00,06 : Application modes of the serial interface COM 1

KW 00,07 : PLC reaction to class 3 errors

KW 00,08 : PLC reaction to an overload/short-circuit at the transistor outputs A 62,00...A 62,07 (A 62,15)
(07 KT 92 and 07 KT 93 only)

KW 00,09 : Minimum number of the remote modules included in the CS31 system bus cycle

KW 00,10 : Size of the transmitting area of the slave PLC

KW 00,11 : Size of the receiving area of the slave PLC

KW 00,12 : Automatic warm start after an FK2 error

07 KP 62

Only the system constants KW 00,01 ... KW 00,07 apply to the 07 KP 62 module.

Setting the cycle time

KD 00,00 : The cycle time of the PLC program is preset with this constant. The cycle time is given in the unit of measurement milliseconds. Only integral multiples of 5 ms are permitted.

Error diagnosis flags

Summation error display : M 255,10 indicates, that the PLC has detected an error

Fatal error , FK1 : M 255,11 = 1 i.e. error detected, detailed information in MW 254,00...MW 254,07

Serious error, FK2 : M 255,12 = 1 i.e. error detected, detailed information in MW 254,08...MW 254,15

Light error , FK3 : M 255,13 = 1 i.e. error detected, detailed information in MW 255,00...MW 255,07

Warning, FK4 : M 255,14 = 1 i.e. error detected, detailed information in MW 255,08...MW 255,15

First-cycle detection

M 255,15

This binary flag can be used for detection of the *first* program cycle after a program start. It is always set to "zero" after each program start, independent of the initialization instructions given by the system constants. If this flag is read by the user program and then set to "1", it can be found out whether or not the user program has been restarted.

CS31 status word

(only Series 30, 50, 90 as CS31 bus master)

EW 07,15

Bit 0 = 1 : No class 2 error present.

Bit 1 = 1 : PLC has been adopted into the CS31 bus cycle (only relevant if used as a slave).

Bit 2 = 1 : Time and date are valid.

Bit 3 = 1 : Battery is effective.

Bit 4...7 : Not used.

Bit 8..15 : Maximum number of modules on the CS31 system bus, found out until now (only relevant if used as a master).

1.3 System constants / Setting of operating modes

● Definitions

At first, the definitions used with the setting of operating modes are explained:

- Cold start and
- Warm start

Cold start

- All of the RAM memories are tested and deleted.
- If there is *no user program* in the Flash–EPROM, the default values are set to all of the system constants (identical to the factory settings).
- If there is *a user program* in the Flash–EPROM, this program is loaded into the RAM including the system constants.
- The operating modes given by the system constants are set.
- The CS31 system bus is initialized again (only when used as a master on the CS31 system bus).

Performing a cold start

- Power OFF/ON, if there is *no* backup battery or
- Command KALT <CR> in terminal mode (see volume 7.3) or
- Menu field "Cold start" in the programming system

Warm start

- All of the RAM memories, with the exception of the program memory and the operand memory (flags), are tested and deleted.
- If there is *a user program* in the Flash–EPROM, this program is loaded into the RAM including the system constants.
- The operating modes given by the system constants are set.
- The CS31 system bus is initialized again (only when used as a master on the CS31 system bus).

Performing a warm start

- Power OFF/ON, if there is a backup battery or
- Command WARM <CR> in terminal mode (see volume 7.3) or
- Menu field "Release PLC mode" in the programming system

● Operating mode: Master PLC, Slave PLC or Stand-alone PLC

- Absolute identifier: KW 0,0
- Symbolic identifier: MAST_SLV
- Meaning of the value of the constants:
 - Master PLC at the CS31 system bus –1 (FFFF_H)
 - Stand-alone PLC –2 (FFFE_H)
 - Slave PLC at the CS 31 system bus module address CS31 module addresses 0...61
- Range of values: –2, –1, 0...61
- Default value: –2 (Stand-alone)

Important!

The change of the PLC operation mode is carried out in three steps:

1. Change system constant KW 0,0 in the PLC
2. Save PLC program in the Flash EPROM
3. Activate new PLC operating mode with the following steps:
 - Call menu point "Release PLC mode" in the ABB programming and test system or
 - perform a warm start or
 - perform a cold start.

● Back-up of data areas

Back-up of data areas, i.e. saving of data during power OFF/ON, is only feasible with built-in battery. The following data can be backed, completely or partly:

- Binary flags
- Word flags
- Double word flags
- Step chains
- Historical values

In order to back-up certain data, they have to be excluded from initialization to 0.

● Initialization of data areas

During *program start*, that data areas are initialized to 0 partly or completely, that are defined by system constants. The initialization works as shown in the following table.

If no battery is effective or if the system constants are in their default values (factory settings), all of the above mentioned data areas are completely set to 0 after power OFF/ON.

Conditions, —> Action	Flags, step chains, and historical values which are initialized (set to 0)
No battery available, —> Power ON	all
Menu item —> Abort —> Cold start	all
Battery effective, RUN/ STOP switch to RUN, —> Power ON	according to the va- lues of the system constants (see below)
RUN/STOP switch, —> RUN	
Menu item —> Abort —> Start	

Initialization: Binary flags

- Absolute identifier: KW 00,01
- Symbolic identifier: INIT_M

Value n of the system constant KW 00,01	Binary flag areas which are initialized (set to 0)
n = 0 (default)	M 000,00...M 255,15
n = 1...255	M n,00...M 255,15
n < 0, n > 255	M 255,10...M 255,15

- Example: KW 00,01 = 52
 Initialized is: M 52,00...M 255,15
 Backed is: M 00,00...M 51,15
 Precondition: Battery is available

Initialization: Word flags

- Absolute identifier: KW 00,02
- Symbolic identifier: INIT_MW

Value n of the system constant KW 00,02	Word flag areas which are initialized (set to 0)
n = 0 (default)	MW 000,00...MW 253,15
n = 1...253	MW n,00...MW 253,15
n < 0, n > 253	no initialization

Initialization: Double word flags

- Absolute identifier: KW 00,03
- Symbolic identifier: INIT_MD

Value n of the system constant KW 00,03	Double word flag areas which are initialized (set to 0)
n = 0 (default)	MD 00,00...MD 31,15
n = 1...31	MD n,00...MD 31,15
n < 0, n > 31	no initialization

Initialization: Step chains

- Absolute identifier: KW 00,04
- Symbolic identifier: INIT_S

Value n of the system constant KW 00,04	Step chain areas which are initialized (set to 0)
n = 0 (default)	S 000,00...S 127,15
n = 1...127	S n,00...S 127,15
n < 0, n > 127	no initialization

Initialization: Historical values

- Absolute identifier: KW 00,05
- Symbolic identifier: INIT_VW

Value n of the system constant KW 00,05	Historical values which are initialized (set to 0)
n = 0 (default)	Initialization of all historical values
n < 0, n > 0	no initialization

● **Application mode: Serial interface COM1**

- Absolute identifier: KW 0,6
- Symbolic identifier: MODE_SST
- Default value: 0

Table: Conditions for the settings of the operating modes of the COM1 interface

RUN/STOP switch	System constant KW00,06	System cable/device	Mode set by this
STOP	x	x	Active
RUN	1	x	Active
	2	x	Passive
	0, <0, >2	07 SK 90	Active
		07 SK91, TCZ	Passive

x: without effect

- A change of this system constant becomes effective:
 - immediately

For Series 30, 40, 50 refer to their own descriptions.

● **SPS reaction to class 3 errors**

- Absolute identifier: KW 0,7
- Symbolic identifier: FK3_REAK
- Meaning of the value of the constant:
 - Just output error: 0
 - Output error and abort PLC program: <0, >0
- Range of values: <0, =0, >0
- Default value: 0
i.e. just output error
- A change of this system constant becomes effective:
 - immediately

● **PLC reaction to the occurrence of an overload/short circuit at the direct transistor outputs A 62,00...A 62,07...A 62,15 (07 KT 92 and 07 KT 93 only)**

- Absolute identifier: KW 0,8
- Symbolic identifier: ÜLAST_REAK
- Meaning of the value of the constant:
 - Overloaded output is switched on again automatically: 0
 - Overloaded output is **not** switched on again automatically: 1

If another value than 0 or 1 is given, the PLC selects the standard setting "automatic reset"

- Range of values: 0, 1
- Default value: 0
i.e. the overloaded output is switched on again automatically by the PLC.

Important!

The change of the PLC reaction on an overload/short-circuit is carried out in three steps:

1. Change system constant KW 0,8 in the PLC
2. Save PLC program in Flash EPROM
3. Activate new PLC operating mode with the following steps:
 - perform a warm start or
 - perform a cold start.

● **Initialization of the AC31 system after power ON, warm start or cold start**

- Absolute identifier: KW 0,9
- Symbolic identifier: HOCHFAHR
- This system constant is only effective if the central unit is configured as a bus master.
- Meaning of the value of the constants:

The user program is started.

The central unit takes no notice of initialization of the CS31 remote modules and their adoption into the CS31 bus cycle: =0

The user program is not started until at least *n* remote modules have been initialized and adopted into the CS31 bus cycle: =+n

The user program is started. It does not handle the process inputs and outputs until at least *n* remote modules have been initialized and adopted into the CS31 bus cycle. However, the CS31 status information in EW 07,15 is available as early as with the program start. This is also valid for the dual port RAM image of the two high-speed inputs at terminals 05 and 06: =-n

In Series 30, 40, 50 not available.

- Range of values: -31...+31
- Default value: 0
i.e. the user program is started immediately.

A change of this system constant becomes effective:

- with the next warm start or
- with the next cold start.

● Size of the transmitting area of the slave PLC

- Absolute identifier: KW 00,10
- Symbolic identifier: SLV_SEND
- Meaning of the value of the constants:

The slave PLC can be used at the CS31 system bus *either* in the binary area *or* in the word area. The binary values are transferred byte by byte. It is possible to set the number of bytes (or words) which are to be sent from the slave PLC to the master PLC.

- For use in the binary area:
Transmitting: 0...15 bytes 0...15
- For use in the word area:
Transmitting 0...8 words 100...108
- Default value: 0
- Range of values: 0...15 and 100...108

A change of this system constant becomes effective:

- with the next warm start or
- with the next cold start.

Note:

The default setting

- in the binary area is:
 - transmit 4 bytes and
 - receive 4 bytes.

This is defined by the default combination KW 00,10 = KW 00,11 = 0.

The configured combination KW 00,10 = KW 00,11 = 4 has the same result as the default combination.

The combination

KW 00,10 = KW 00,11 = 100
is inadmissible! It would mean:
Transmit 0 words and receive 0 words.

When employed in the word area, the unused higher 8 channels of the address can be used by an analog modul (no KR/KT).

● Size of the receiving area of the slave PLC

- Absolute identifier: KW 00,11
- Symbolic identifier: SLV_REC
- Meaning of the value of the constants:

The slave PLC can be used at the CS31 system bus *either* in

the binary area *or* in the word area. It is possible to set the number of bytes (or words) which are to be received by the slave PLC from the master PLC.

- For use in the binary area:
Receiving: 0...15 bytes 0...15
- For use in the word area:
Receiving: 0...8 words 100...108
- Default value: 0
- Range of values: 0...15 and 100...108

A change of this system constant becomes effective:

- with the next warm start or
- with the next cold start.

Note:

The default setting

- in the binary area is:
 - transmit 4 bytes and
 - receive 4 bytes.

This is defined by the default combination KW 00,10 = KW 00,11 = 0.

The configured combination KW 00,10 = KW 00,11 = 4 has the same result as the default combination.

The combination

KW 00,10 = KW 00,11 = 100
is inadmissible! It would mean:
Transmit 0 words and receive 0 words.

When employed in the word area, the unused higher 8 channels of the address can be used by an analog modul (no KR/KT).

● Automatic warm start after an FK2 error

- Absolute identifier: KW 00,12
- Symbolic identifier: SYSTEM
- By means of the system constant KW 00,12 an automatic warm start can be configured after an FK2 error:

Bit 0 of KW 00,12 = 0: no automatic warm start
Bit 0 von Kw 00,12 = 1: automatic warm start

The bits 1...15 of KW 00,12 have to be 0.

In the default setting KW 00,12 = 0 the module 07 KT 92 **R202/262** has the same behaviour as the module 07 KT 92 **R101** (no warm start after an FK2 error).

A change of this system constant becomes effective:

- with the next warm start.

- **PLC cycle time**

- Absolute identifier: KD 0,0
- Symbolic identifier: ZYKL_ZEIT
- Meaning of the value of the constants:
The PLC program is processed cyclically in the time intervals stated by the set cycle time. The entries are made in the unit of measurement [ms]. The smallest cycle time that can be entered is 5 ms. Only integral multiples of 5 ms are permissible.

- Range of values: ≥ 5

- Default value: 10

A change of this system constant becomes effective:

- with the next program start.

2 PLC language repertoire

A user program comprises the following:

- Sentences (equational notation) and/or S block calls.

The sentences consist of individual instructions according to DIN 19239.

Sentences and block calls can be mixed as required. The end of the program is marked with !PE.

Possibilities for entering programs:

- Using the ABB Procontic programming system, the PLC program can be created on a very user-friendly programming surface, transferred to the PLC and tested.
- The language objects stated in this chapter and the blocks described in the block library can, however, also be entered directly into the PLC as an instruction list using a terminal or the ABB service device (TCZ) and tested.

2.1 Operators for configuring sentences (instruction list according to DIN 19239)

Sentence start operators for Boolean sentences

! Start of sentence
!N Start of sentence with negation

Boolean operators

& AND
&N AND with negation
/ OR
/N OR with negation

Sentence start operators for arithmetical sentences

! Start of sentence
!- Start of sentence with negation

Arithmetical operators

+ PLUS
- MINUS
* MULTIPLIED by
*- MULTIPLIED by with negation
: DIVIDED by
:- DIVIDED by with negation

Brackets

(OPEN BRACKET
(N OPEN BRACKET with negation (binary)
(- OPEN BRACKET with negation (arithmetic)
) CLOSE BRACKET

The maximum nesting depth is 15.

Comparative operators

Note:

If not an individual variable but an entire expression is to the left and/or right of the comparative operator, it must be placed inside brackets in each case.

Example:

!(EW0,0+MW0,0*KW0,5) > (MW5,8+MW3,4) = A0,0

Comparative expressions must *not* be combined with Boolean expressions in one sentence. For this purpose, intermediate flags must be configured.

Example:

- correct:

!MW 0,0 > MW 0,1 = M 5,5
!M 5,5 & M 0,1 = A 0,0

- incorrect:

!MW 0,0 > MW 0,1 & M 0,1 = A 0,0 or
!(MW 0,0 > MW 0,1)
& M 0,0 = A 0,0

> Greater than
>- Greater than with negation
< Less than
<- Less than with negation
>= Greater than or equal to
>=- Greater than or equal to with negation
=? Equal to
=?- Equal to with negation
<> Unequal to
<>- Unequal to with negation
<= Less than or equal to
<=- Less than or equal to with negation

Allocations

= Equal to
=N Equal to with negation (binary)
=- Equal to with negation (arithmetic)
=SM Set flag
=RM Reset flag
=S Set step
=PE Conditional program end

2.2 Block calls

The blocks available and their call instruction are described in the list of connection elements.

The call in the instruction list has the following form:

!BA Number
Name

whereby:

!BA ::= Keyword according to DIN 19239
Number ::= 0...999, Number of the call (optional)
Name ::= Name of the block to be called

2.3 NOP

The user program may contain NOPs.

NOP (non-described memory cell)

Restrictions:

- Block calls: There must be *no* NOPs within block calls.
- Sentences: NOPs may be within sentences but, in addition, only directly in front of an operator. They must *not* be in place of an operator (e.g. flag).

NOPs may be positioned between sentences/blocks without restrictions.

Example:

- correct:

```
NOP NOP ! M 0,0 NOP & M 0,1 NOP = A 0,2 NOP
NOP
NOP
!BA 123
  ESV
  E 0,0
  KD 0,5
  A 12,13
NOP
```

- incorrect:

```
!NOP & NOP M 0,1 = NOP A 0,2
!BA 123
  ESV
  NOP
  E 0,0
  KD 0,5
  A 12,13
```

2.4 Jumps

Jumps enable the linear processing of a program to be interrupted. The jumps are realized by the function block SPBM (see volume 7, CE discription).

A jump always take place forwards to the next label, with the label No. given in the function block SPBM.

2.5 Labels

Labels are used as jump targets for forward jumps for the blocks JUMP and LZB (run number block).

They are characterized by:

MA Number, with number 0 ... 999

The same label may also be used as a jump target for several jump blocks and/or several run number blocks.

In addition, a label with a certain No. may be used as often as required in the program. Ensure that a jump/run number block always takes place to the next label arising in the program with the label No. given in the jump block / run number block.

2.6 Program end

Absolute

The absolute end of the program is always marked by !PE

Conditional

The conditional program end is marked by:

=PE

E.g.: !M 00,00 & M 00,01 = PE

2.7 Configuration of time values

Real-time data is always configured as indirect double word constants or as double word flags. Real-time data should be given directly in milliseconds, and it is dealt with, e.g., cycle times (KD 0,0) or time values of timers (ESV, ASV, VVZ, etc.). Only integral multiples of the basic time of 5 ms are permissible as the entered time values. Other values given are rounded off by the PLC to values divisible by 5 ms.

2.8 Data types / formats

The PLC processes the following data types and data formats:

- BINARY: Boolean algebra
- INTEGER WORD (16 bits): Integer arithmetic
Figure range: -32767 (8001_H) ... +32767 (7FFF_H)
- INTEGER DOUBLE WORD (32 bits):
for presetting time values for
 - timers and
 - cycle times of PLC programs and for use in double word blocks.

Figure range: -2 147 483 647...+2 147 483 647
(8000 0001_H) ...(7FFF FFFF_H).

Only positive figure values may be given for time values.

Figure range

Word range:

- lower limit: 8001_H –32767
- upper limit: 7FFF_H +32767
- inadmissible value: 8000_H //

The value 8000_H (–32768) is *outside* of the figure range of the two's complement arithmetic, and is neither produced nor correctly processed by the PLC. If this *forbidden* value is created due to

- bit manipulations by the user
- reading from outside the PLC or
- an indirect word constant in the PLC,

a negation or subtraction must *on no account* be applied to this value.

A permissible value is generated again by means of an allocation (=), addition (+), multiplication(*) or division (:).

When the allocation (=) is used, the *forbidden* value 8000_H (–32768) is corrected to the allowed value 8001_H (–32767).

Double word range:

- lower limit: 8000 0001_H –2 147 483 647
- upper limit: 7FFF FFFF_H +2 147 483 647
- inadmissible value: 8000 0000_H //

All of the blocks for the double word arithmetic check the values to be processed for admissibility. If an inadmissible value occurs as a result, it is corrected to the permissible value 8000 0001_H (–2 147 483 647).

2.9 Operand identifiers

The PLC processes the following operand identifiers (DIN 19239):

M	BINARY flag
S	BINARY step
MW	WORD flag
MD	DOUBLE WORD flag
K	BINARY constant (indirect)
KW	WORD constant (indirect)
KD	DOUBLE WORD constant (indirect)
E	BINARY input
EW	WORD input
A	BINARY output
AW	WORD output
#	Direct constant, decimal (–32768...32767) *
#H	Direct constant, hexadecimal (0...FFFF _H) *

*) Direct constants are only permissible for certain blocks (see also block catalogue). They are *not* admissible in instructions for binary and word processing.

2.10 Numbering the operands (variables)

The syntax of the operands is in accordance with DIN 19239 and has the following structure:

operand :: OPERAND IDENTIFIER
 FORMAT IDENTIFIER
 NUMBER

Example:

MW	123,15
A	11,11
E	04,08
EW	10,01
KD	00,00
S	127,15

The following are available:

- OPERAND IDENTIFIERS
 - E Input
 - A Output
 - M Flag
 - S Step
 - K Indirect constant
- FORMAT IDENTIFIERS
 - Binary, no identifier
 - W Word
 - D Double word (only for indirect constants and flags, i.e. *not* for inputs and outputs)
- NUMBER
 - GROUP NUMBER, CHANNEL NUMBER

The number of the operand comprises a group number and a channel number. Group number and channel number are separated by a comma. The GROUP NUMBER has the range of 0..max. The maximum group number is dependent on the OPERAND IDENTIFIER and on the FORMAT IDENTIFIER (see below).

The CHANNEL NUMBER has the range of 0..15. After the CHANNEL NUMBER 15 comes the CHANNEL NUMBER 0 again, whereby the GROUP NUMBER is incremented by 1.

2.11 Step chains

Functionality of the step chain

Step chains are used for clearly controlling processes, whose execution is characterized by sequential execution steps. In this case, the activation of an execution step is determined by the status of its predecessor **and** by an additional enabling condition (transition).

Length of the step chain

A step chain consists of 16 steps (Sxxx,00 ... Sxxx,15).

Number of step chains

Depending on the PLC type there are up to 128 step chains available depending on the PLC type in the user program (S 000,yy ... S 127,yy).

Number of steps

There are up to 2048 steps available in the user program (S 000,00 ... S 127,15).

Idle state of a step chain

The idle state of a step chain is characterized by the fact that the first step of the chain (Sxxx,00) is *set*.

The idle state of each chain is set:

- basically when cold-starting the PLC
- when starting the PLC program according to the setting of the system constant KW 00,04.

Setting the step

The chain is set using an allocation.

It remains set until *another* step is set in the *same* chain.

Example: !M 1,5 & NE 3,3 = S11,11

Automatic resetting of the previous step

An important feature of the step chain is that *only one* step can be set in *one* chain. I.e. all other steps in this chain are automatically reset whenever a step is set.

Step operand in sentences

The step is handled like a binary operand in the sentences of the user program. The step can therefore be both in the interrogation part and in the allocation part of the sentence.

Example: !(E 0,0 / M 7,13) & S 0,3 = M 1,1

Unlike a binary flag, the following instructions are *impermissible* in a step. They are not accepted by the PLC if entered:

- = NS xxx,yy
- = SS xxx,yy
- = RS xxx,yy

These allocations could lead to inadmissible statuses of step chains (no step is set or several steps are set).

Step operand in blocks: Important!

In blocks, the step must *only* be at inputs and *not* at outputs as otherwise the previous step is not automatically reset.

Example: !BA 123 "Block number"
ESV "ON-delay type"
S 1,1 "Input"
KD 0,3 "Delay time"
M 0,2 "Output"; there must be *no* step operand at the output

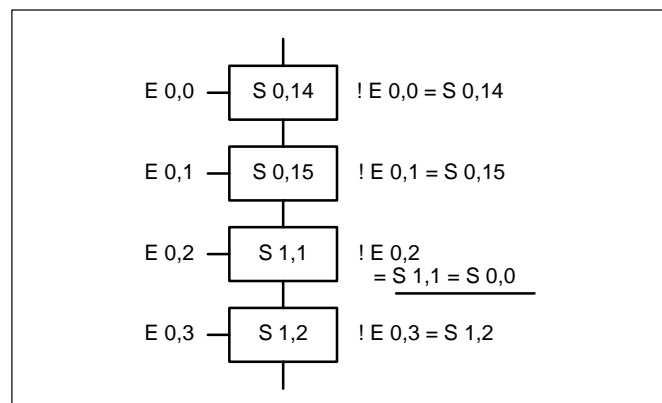
If it is worked with the function block diagram or with the extended instruction list in the 907 PC 331 programming and test system, this regulation will be fulfilled automatically by the programming system.

Joining step chains

The step chains are completely independent of one another.

If more than 16 steps are needed within a PLC program, several step chains can be connected in series. When passing from one step chain to another, the step chain which is left must be switched to the idle state (0-step).

Example: Joining step chain 1 with step chain 2





Printed on chlorine-free bleached paper

ABB Schalt- und Steuerungstechnik GmbH
Eppelheimer Straße 82 Postfach 10 50 09
D-69123 Heidelberg D-69040 Heidelberg

Telephone +49 6221 777-0
Telefax +49 6221 777-111
E-Mail desst.help@de.abb.com
Internet <http://www.abb-sst.de>

Printed in the Federal Republic of Germany (08.99)