Contents

1 System technology of the basic units

1.1 Introduction

The basic units 07 KT 95, 07 KT 96, 07 KT 97 and 07 KT 98 are programmed using the programming system 907 AC 1131. For this purpose, the new RT operating system is installed on the basic units.

The basic units 07 KR 91, 07 KT 92, 07 KT 93 and 07 KT 94 use a different operating system (EBS operating system). Therefore they **cannot** be programmed using the programming system 907 AC 1131.

Note:
This documentation is applicable for all categories of the basic units 07 KT 95, 07 KT 96, 07 KT 97 and 07 KT 98. When the term 07 KT 97 is given in the text, this text is also applicable for 07 KT 95, 07 KT 96 and 07 KT 98. Texts which are exclusively applicable for 07 KT 97 are expressly mentioned by a note.

1.2 Operands of the basic units

All the operands which are supported by 907 AC 1131 are described in the documentation of the programming system 907 AC 1131. In this documentation the "address" operands (i.e. %I – for inputs, %Q – for outputs and %M – for addressable flags) which are mentioned in 907 AC 1131 are described in detail in the following.

All previous operands of the EBS operating system of the controllers 07 KR 91, 07 KT 92, 07 KT 93 and 07 KT 94 were imported into the 907 AC 1131. How to call the operands in 907 AC 1131 is described in the following. In the hardware documentation the previous designators are still used.

In the 907 AC 1131 all operands can be called bitwise (X), byte wise (B) and word wise (W). The double word operands can be called bitwise (X), byte wise (B), word wise (W) and double word wise (D).

Declaration of operands:

The declaration of the operands is done as following:

Symbol AT address : Type [:= initialization value]; (* comment *)

[.] - optional

CAUTION:
For multitasking the binary inputs and outputs for every task are byte wise cycle consistent, i.e. for instance the inputs E62,00-62,07 for task 1 and E62,08-E62,15 for task 2. This is not significant for programs with only one task.

Note:
With the software 907 AC 1131, Importfiles *.exp are automatically installed in the Library subdirectory. In this import files, important operands are already declared (see chapters Inputs, Outputs, System constants, system flags).

1.2.1 Inputs / %I area

Binary inputs:

E 00,00-E 61,15	**Binary inputs, CS31 remote modules:**	
%IW000-%IW061	%IB0-%IB123	%IX000.0-%IX061.15
E 62,00-E 63,15	**Binary inputs of the basic unit 07 KT 97**	
%IW062-%IW063	%IB124-%IB127	%IX062.0-%IX063.15

Note:
The binary inputs E 63,00-E 63,07 / %IX63.0-%IX63.7 are the designators for the 8 inputs of the binary I/Os.

E 64,00-E 64,07	**Binary inputs of the basic unit 07 KT 97** **(developed from the analog inputs EW 06,00...EW 6,07)**	
%IW064	%IB128	%IX064.0-%IX064.7
E 65,00-E 99,15	**reserved**	
%IW065-%IW099	%IB130-%IB199	%IX065.0-%IX065.15
E 100,00-E 163,15	**reserved**	
%IW100-%IW163	%IB200-%IB327	%IX163.0-%IX163.15
E 200,00-E 263,15	**reserved**	
%IW200-%IW263	%IB400-%IB527	%IX263.0-%IX263.15

Analog inputs:

EW 00,00-EW 05,15	**Analog inputs, CS31 remote modules**	
%IW1000.0-%IW1005.15	%IB1000.0.0-%IB1005.15.1	%IX1000.0.0-%IX1005.15.15
EW 06,00-EW 06,07	**Analog inputs of the basic unit 07 KT 97**	
%IW1006.0-%IW1006.07	%IB1006.0.0-%IB1006.07.1	%IX1006.0.0-%IX1006.07.15
EW 07,00-EW 07,14	**reserved**	
%IW1007.0-%IW1007.14	%IB1007.0.0-%IB1007.14.1	%IX1007.0.0-%IX1007.14.15
EW 07,15	**Status word for CS31 system bus**	
%IW1007.15	%IB1007.15.0-%IB1007.15.1	%IX1007.15.0-%IX1007.15.15
EW 08,00-EW 15,15	**Analog inputs, CS31 remote modules**	
%IW1008.0-%IW1015.15	%IB1008.0.0-%IB1015.15.1	%IX1008.0.0-%IX1015.15.15
EW 16,00-EW 34,15	**reserved**	
%IW1016.0-%IW1034.15	%IB1016.0.0-%IB1034.15.1	%IX1016.0.0-%IX1034.15.15
EW 100,00-EW 107,15	**reserved**	
%IW1100.0-%IW1107.15	%IB1100.0.0-%IB1107.15.1	%IX1100.0.0-%IX1007.15.15
EW 200,00-EW 207,15	**reserved**	
%IW1200.0-%IW1207.15	%IB1200.0.0-%IB1207.15.1	%IX1200.0.0-%IX1207.15.15

Inputs line 1 and line 2 (PROFIBUS):

Line 1:		
%IW1.0-%IW1.1792	%IB1.0-%IB1.3583	%IX1.0.0.-%IX1.1792.15
Line 2:		
%IW2.0-%IW2.1792	%IB2.0-%IB2.3583	%IX2.0.0.-%IX2.1792.15

Note:
Line 1 and line 2 were not existent before.

Examples for the declaration of inputs:

The declaration of the operands is done as following:

Symbol AT address : Type [:= initialization value]; (* comment *)

[.] - optional

Binary input E 62,00:
E62_00_Input0 AT %IX62.0 : BOOL; (* This is input 0 *)

Reading the binary inputs E62,00-E62,07 as a byte:
EB62_0_Byte0 AT %IB124 : BYTE; (* Input byte 62/0 *)

Reading the binary inputs E62,08-E62,15 as a byte:
EB62_1_Byte1 AT %IB125 : BYTE; (* Input byte 62/1 *)

Reading the binary inputs E62,00-E62,15 as a word:
EW62_Word0 AT %IW62 : WORD; (* Input word 62 *)

Analog input EW06,00:
EW06_00_Ana0 AT %IW1006.0 : INT; (* This is analog input 0 *)

Calling bit 15 of the analog input EW06,00:
EW06_00_Bit15 AT %IX1006.0.15 : BOOL; (* Bit 15 of EW06,00 *)

Import file for inputs and outputs:

For the declaration of the inputs and outputs of the basic units, there are pre-produced import files available:

KT95_onb.exp	Onboard inputs and outputs of the basic unit 07 KT 95
KT96_onb.exp	Onboard inputs and outputs of the basic unit 07 KT 96
KT97_onb.exp	Onboard inputs and outputs of the basic unit 07 KT 97
KT97_dea.exp	Binary onboard inputs and outputs of the basic unit 07 KT 97
KT98_onb.exp	Onboard inputs and outputs of the basic unit 07 KT 98

The files are installed into the Library subdirectory of the software 907 AC 1131. In a project, they are inserted by means of the Resources menu into the object „Global Variables“ with „Project / Options / Import“.

Configuration of the binary inputs of the basic unit 07 KT 97:

The configuration of the binary inputs of the basic units 07 KT 97 is carried out with the system constants KW85,00 / %MW3085.0, see chapter ‚System constants‘.

Configuration and measuring ranges of the analog inputs of the basic unit 07 KT 97:

The configuration of the analog inputs of the basic unit 07 KT 97 is carried out with the system constants KW86,00 / %MW1086.0..KW86,07 / %MW3086.7. In the chapter ‚System constants‘ you can find the description of the configuration and selection of the measuring ranges of the analog input channels.

1.2.2 Outputs / %Q area

Binary outputs:

A 00,00-A 61,15	**Binary outputs, CS31 remote modules**	
%QW000-%QW061	%QB0-%QB123	%QX000.0-%QX061.15
A 62,00-A 63,07 :	**Binary outputs of the basic unit 07 KT 97**	
%QW062-%QW063	%QB124-%QB126	%QX062.0-%QX063.07
A 62,00	**High-speed counter with direct output of "counter zero-crossing" after activation**	
%QW062	%QB124	%QX062.0
A 64,00-A 99,15	**reserved**	
%QW064-%QW099	%QB128-%QB199	%QX064.0-%QX099.15
A 100,00-A 163,15	**reserved**	
%QW100-%QW163	%QB200-%QB327	%QX100.0-%QX163.15
A 200,00-A 263,15	**reserved**	
%QW200-%QW263	%QB400-%QB527	%QX200.0-%QX263.15

Analog outputs:

AW 00,00-AW 05,15	**Analog outputs, CS31 remote modules**	
%QW1000.0-%QW1005.15	%QB1000.0.0-%QB1005.15.1	%QX1000.0.0-%QX1005.15.15
AW 06,00-AW 06,03	**Analog outputs of the basic unit 07 KT 97**	
%QW1006.0-%QW1006.03	%QB1006.0.0-%QB1006.03.1	%QX1006.0.0-%QX1006.03.15
AW 07,00-AW 07,15	**reserved**	
%QW1007.0-%QW1007.15	%QB1007.0.0-%QB1007.15.1	%QX1007.0.0-%QX1007.15.15
AW 08,00-AW 15,15	**Analog outputs, CS31 remote modules**	
%QW1008.0-%QW1015.15	%QB1008.0.0-%QB1015.15.1	%QX1008.0.0-%QX1015.15.15
AW 16,00-AW 34,15	**reserved**	
%QW1016.0-%QW1034.15	%QB1016.0.0-%QB1034.15.1	%QX1016.0.0-%QX1034.15.15
AW 100,00-AW 107,15	**reserved**	
%QW1100.0-%QW1107.15	%QB1100.0.0-%QB1107.15.1	%QX1100.0.0-%QX1107.15.15
AW 200,00-AW 207,15	**reserved**	
%QW1200.0-%QW1207.15	%QB1200.0.0-%QB1207.15.1	%QX1200.0.0-%QX1207.15.15

Outputs line 1 and line 2 (PROFIBUS):

Line 1:		
%QW1.0-%QW1.1792	%QB1.0-%QB1.3583	%QX1.0.0.-%QX1.1792.15
Line 2:		
%QW2.0-%QW2.1792	%QB2.0-%QB2.3583	%QX2.0.0.-%QX2.1792.15

Note:
Line 1 and line 2 were not existent before.

Examples for the declaration of outputs:

The declaration of the operands is done as following:

Symbol AT address : Type [:= initialization value]; (* comment *)

[.] - optional

Binary output A 62,00:
A62_00_Output0 AT %QX62.0 : BOOL; (* This is output 0 *)

Analog output AW06,00:
AW06_00_Ana0 AT %QW1006.0 : INT; (* This is analog output 0 *)

Output of binary outputs A62,00-A62,07 as a byte:
AB62_0_Byte0 AT %QB124 : BYTE; (* Output byte 62/0 *)

Output of binary outputs A62,08-A62,15 as a byte:
AB62_1_Byte1 AT %QB125 : BYTE; (* Output byte 62/1 *)

Output of binary outputs A62,00-A62,15 as a word:
AW62_Word0 AT %QW62 : WORD; (* Output word 62 *)

Import file for inputs and outputs:

For the declaration of the inputs and outputs of the basic units, there are pre-produced import files available:

KT95_onb.exp	Onboard inputs and outputs of the basic unit 07 KT 95
KT96_onb.exp	Onboard inputs and outputs of the basic unit 07 KT 96
KT97_onb.exp	Onboard inputs and outputs of the basic unit 07 KT 97
KT97_dea.exp	Binary onboard inputs and outputs of the basic unit 07 KT 97
KT98_onb.exp	Onboard inputs and outputs of the basic unit 07 KT 98

The files are installed into the Library subdirectory of the software 907 AC 1131. In a project, they are inserted by means of the Resources menu into the object „Global Variables“ with „Project / Options / Import“.

Configuration and measuring ranges of the analog outputs of the basic unit 07 KT 97:

The configuration of the analog outputs of the basic units 07 KT 97 is carried out with the system constants KW88,00 / %MW1088.0..KW88,03 / %MW3088.3. In the chapter ‚System constants‘ you can find the description of the configuration and the measuring ranges of the analog output channels.

1.2.3 Operands in the %M area

Flags:

M 000,00-M 254,15 — **Binary flags**
%MW000-%MW254 %MB000.0-%MB254.1 %MX000.0-%MX254.15

M 255,00-M 255,15 — **System flags**
%MW255 %MB255.0-%MB255.1 %MX255.0-%MX255.15

M 256,00-M 279,15 — **System flags / reserved**
%MW256-%MW279 %MB256.0-%MB279.1 %MX256.0-%MX279.15

M 280,00-M 511,15 — **Binary flags**
%MW280-%MW511 %MB280.0-%MB511.1 %MX280.0-%MX511.15

Word flags:

MW 000,00-MW 253,15 — **Word flags**
%MW1000.0-%MW1253.15 %MB1000.0.0-%MB1253.15.1 %MX1000.0.0-%MX1253.15.15

MW 254,00-MW 255,15 — **Error message**
%MW1254.0-%MW1255.15 %MB1254.0.0-%MB1255.15.1 %MX1254.0.0-%MX1255.15.15

MW 256,00-MW 259,15 — **System flags / reserved**
%MW1256.0-%MW1259.15 %MB1256.0.0-%MB1259.15.1 %MX1256.0.0-%MX1259.15.15

MW 260,00-MW 511,15 — **User area**
%MW1260.0-%MW1511.15 %MB1260.0.0-%MB1511.15.1 %MX1260.0.0-%MX1511.15.15

Double word flags:

MD 00,00-MD 63,15 — **Double word flags**
%MD2000.0-%MD2063.15 %MW2000.0.0-%MW2063.15.1
%MB2000.0.0-%MB2063.15.3 %MX2000.0.0-%MX2063.15.31

Word constants:

KW 00,00-KW 00,15 — **System constants**
%MW3000.0-%MW3000.15 %MB3000.0.0-%MB3000.15.1 %MX3000.0.0-%MX3000.15.15

KW 01,00-KW 79,15 — **Word constants**
%MW3001.0-%MW3079.15 %MB3001.0.0-%MB3079.15.1 %MX3001.0.0-%MX3079.15.15

KW 80,00-KW 89,15 — **System constants**
%MW3080.0-%MW3089.15 %MB3080.0.0-%MB3089.15.1 %MX3080.0.0-%MX3089.15.15

Double word constants

KD 00,00 — **System constant**
%MD4000.0 %MW4000.0.0-%MW4000.00.1
%MB4000.0.0-%MB4000.00.3 %MX4000.0.0-%MX4000.00.31

KD 00,01-KD 23,15 — **Double word constants**
%MD4000.1-%MD4023.15 %MW4000.1.0-%MW4023.15.1
%MB4000.1.0-%MB4023.15.3 %MX4000.1.0-%MX4023.15.31

Steps:

S 000,00-S 255,15 — **Steps (for transferring projects from 907 PC 331)**
%MW5000-%MW5255 %MB5000.0-%MB5255.1 %MX5000.0-%MX5255.15

1.2.4 Finding out absolute addresses of operands with the address operator ADR

The address operator ADR is described in the documentation of the programming system 907 AC 1131 (refer to volume 5). In this documentation only the peculiarities applicable for bit operands are described.

The address operator ADR provides the address of an operand in one double word DWORD (i.e. 32 bits). The address operator provides the address of the first byte of a variable (byte address). For the free definable variables, variables of the type BOOL are stored as byte. In the addressable flag area %M, for the inputs %I and the outputs %Q, variables of the type BOOL are stored as real bits. Therefore, the address operator ADR provides in this case the variable address in addition to the bit number.

Standard format (not addressable for BOOL, BYTE, INT, WORD, DINT, DWORD, ...)

Bit	Assignment
00...27	Variable address (bits 00..27)
28	Variable identifier 0 – non addressable variable or addressable non-boolean variable 1 – addressable boolean variable
29..31	Variable address (bits 29..31)

The bits 27..31 of the address are always FALSE.

Special format for addressable binary variables of the type BOOL (%MX, %IX, %QX)

Bit	Assignment
00...03	Bit number The stated bit number may have values of 0...15dez (0...Fhex). If the variable address is used, the bit number may only range from 0...7 (size of one byte), i.e. for a bit number of ≥8, the value 8 must be subtracted in order to obtain 0...7.
04...27	Variable address of the byte (bits 00..23)
28	Variable identifier 0 – non addressable variable or addressable non-boolean variable 1 – addressable boolean variable
29..31	Variable address (bits 24..31)

The bits 27..31 of the address are always FALSE.

The addresses, supplied by the address operator, can serve as inputs for those function blocks, which require absolute addresses (such as COPY, WOS, WOL). If these blocks must be used for internal variables, it must be made sure that the variables have addresses succeeding one another. This is performed by declaration of ARRAYS and STRINGS. The blocks with address inputs consider the peculiarities for the bit operands.

Examples for absolute addresses:

In a program, the following variables may be declared:

```
PROGRAM ADR_ST
VAR
    byZaehler           : BYTE;    (* Cycle counter (internal variable) *)
    dwADR0              : DWORD;          (* Address of byZaehler *)
    dwADR1              : DWORD;          (* Address of DUMMY_M *)
    dwADR2              : DWORD;          (* Address of M0_0 *)
    dwADR3              : DWORD;          (* Address of E62_0 *)
    dwADR4              : DWORD;          (* Address of A63_0 *)
    dwADR5              : DWORD;          (* Address of S0_0 *)
    dwADR6              : DWORD;          (* Address of DUMMY_MW *)
    dwADR7              : DWORD;          (* Address of MW0_0 *)
    dwADR8              : DWORD;          (* Address of EW62_0 *)
    dwADR9              : DWORD;          (* Address of AW63_0 *)
    dwADR10             : DWORD;          (* Address of SW0 )
    dwADR11             : DWORD;          (* Address of the1st Bytes of the ARRAY
                                          DUMMY_byARRAY *)
    byQB124             AT %QB124 : BYTE;       (* A62,00..A62,07 as Byte *)
    M0_15               AT %MX0.15 : BOOL;      (* M00,15 *)
    E62_5               AT %IX62.5 : BOOL;      (* E62,05 *)
    A63_7               AT %QX63.7 : BOOL;      (* A63,07 *)
    S0_2                AT %MX5000.2 : BOOL;    (* S00,02 *)
    MW0_0               AT %MW0 : INT;   (* M00,00..M00,15 as Word *)
    EW62_0              AT %IW62 : INT;  (* E62,00..E62,15 as Word *)
    AW63_0              AT %QW63 : INT; (* A63,00..A63,15 as Word *)
    SW0                 AT %MW5000 : INT;       (* S00,00..S00,15 as Word *)
    DUMMY_M             : BOOL;    (* internal variable BOOL *)
    DUMMY_MW            : INT;     (* internal variable INT )
    DUMMY_byARRAY       : ARRAY[0..99] OF BYTE; (* ARRAY of 100 Byte internal *)
END_VAR

(* In the program part, the addresses of the variables are allocated to the variables ADRx:)
byZaehler := byZaehler + 1;
byQB124 := byZaehler;      (* byZaehler sent to the outputs A62,00..A62,07 *)
dwADR0 := ADR(byZaehler);
dwADR1 := ADR(DUMMY_M);
dwADR2 := ADR(M0_15);
dwADR3 := ADR(E62_5);
dwADR4 := ADR(A63_7);
dwADR5 := ADR(S0_2);
dwADR6 := ADR(DUMMY_MW);
dwADR7 := ADR(MW0_0);
dwADR8 := ADR(EW62_0);
dwADR9 := ADR(AW63_0);
dwADR10 := ADR(SW0);
dwADR11 := ADR(DUMMY_byARRAY[0]);
```

The following addresses are supplied:

Variable	Address on basic unit		Remarks
	07 KT 95..97	07 KT 98	
ByZaehler	16#001C7090	16#00603090	Byte address
DUMMY_M	16#001C70C1	16#006030C1	Byte address
M0_15	16#11B8001F	16#3800001F	Bit address→Mask: 16#EFFFFFF0 / Bit-No. 15
E62_5	16#12F807C5	16#171007C5	Bit address→Mask: 16#EFFFFFF0 / Bit-No. 5
A63_7	16#12FC07E7	16#171407E7	Bit address→Mask: 16#EFFFFFF0 / Bit-No. 7
S0_2	16#11BE5402	16#38065402	Bit address→Mask: 16#EFFFFFF0 / Bit-No. 2
DUMMY_MW	16#001C70C2	16#006030C2	Byte address
MW0_0	16#001B8000	16#02800000	Byte address
EW62_0	16#002F807C	16#0071007C	Byte address
AW63_0	16#002FC07E	16#0071407E	Byte address
SW0	16#001BE540	16#02806540	Byte address
DUMMY_byARRAY	16#001C70C4	16#006030C4	Byte address

Evaluation of the addresses:

In the example, the flag M00_00 is addressed with **M0_15** as Bit and the flags M00_00...M00_15 are addressed with **MW0_0** as Words. If the address is represented bitwise, the connection described above can be realized:

PLC	Variable	Address HEX	Address BIN
		Bit:	31 28 27 15 4 3 0
07 KT 97	**M0_15:**	16#11B8 001F	2#000 1 0001 1011 1000 0000 0000 0001 1111
Bit 28:=TRUE → Mask:		16#EFFF FFF0	2#111 0 1110 1111 1111 1111 1111 1111 0000
Address (*16):		16#01B8 0010	2#000 0 0001 1011 1000 0000 0000 0001 0000
Byte address:		**16#001B 8001**	2#000 0 0000 0001 1011 1000 0000 0000 0001

Bit 0..3: 1111 → **Bit number := 15**

07 KT 97	**MW0_0:**	16#001B 8000	2#000 0 0000 0001 1011 1000 0000 0000 0000

Bit 28 FALSE → Byte address corresponds to the supplied address

07 KT 98	**M0_15:**	16#3800 001F	2#001 1 1000 0000 0000 0000 0000 0001 1111
Bit 28:=TRUE → Mask:		16#EFFF FFF0	2#111 0 1111 1111 1111 1111 1111 1111 0000
Address (*16):		16#2800 0010	2#001 0 1000 0000 0000 0000 0000 0001 0000
Byte address:		**16#0280 0001**	2#000 0 0010 1000 0000 0000 0000 0000 0001

Bit 0..3: 1111 → **Bit number := 15**

07 KT 98	**MW0_0:**	16#0280 0000	2#001 0 1000 0000 0000 0000 0000 0000 0000

Bit 28 FALSE → Byte address corresponds to the supplied address

If the found addresses are compared to each other, it can be seen that the addresses differ between the basic units 07 KT 97 and 07 KT 98. However, using the address operator ADR, the user must not take care for the absolute addresses.

As it can be seen, the address operator supplies the same byte address for both the bit-wise and the word-wise access to the same memory. With the 07 KT 97, for MW0_0 the byte address 16#001B8000 is supplied and for M0_15 the byte address 16#001B8001 is supplied. This value is correct, because M0_15 is located on the byte of the flags M00_08..M00_15, i.e. one byte higher than M00_00..M00_07. With MW0_0, the supplied address of M00_00..M00_07 is correct.

1.2.5 System constants

The constants
KW 00,00 - KW 00,15 / %MW3000.0 - %MW3000.15 and
KW 80,00 - KW89,15 / %MW3080.0 - %MW3089.15
are reserved for the usage as system constants. It is also not allowed to use constants from this areas which are not yet seized for other purposes.

The system constants are declared in the export file Sy_const.exp. When creating a new project using "File / New", these files are automatically stored in the "Resources" menu and there under the "Global variables" object. Otherwise they can be read in with "Project / Import".

System constants overview:

KW 00,00 : %MW3000.0	Setting of PLC application mode, (Standalone PLC, Master PLC, Slave PLC)
KW 00,01 : %MW3000.1	Initialization: Binary flags area
KW 00,02 : %MW3000.2	Initialization: Word flags area
KW 00,03 : %MW3000.3	Initialization: Double word flags area
KW 00,04 : %MW3000.4	Initialization: Step chain flags area (project adoption)
KW 00,05 : %MW3000.5	**without function** (for EBS: Initialization of history values)
KW 00,06 : %MW3000.6	**without function** (for EBS: Application mode of serial interface COM1)
KW 00,07 : %MW3000.7	PLC reaction to class 3 errors
KW 00,08 : %MW3000.8	**without function** (for EBS: Overload/Short-circuit 07 KT 92/93)
KW 00,09 : %MW3000.9	**without function** (for EBS: Starting-up the CS31 system)
KW 00,10 : %MW3000.10	Transmit area size of the slave PLC
KW 00,11 : %MW3000.11	Receive area size of the slave PLC
KW 00,12 : %MW3000.12	Automatically initiated warm start after FK2 errors
KW 00,15 : %MW3000.15	Deactivation of oscillators on M 255,00-M 255,06
KW 80,00 : %MW3080.0	**new:** Monitoring of the task with a cycle time > 0
KW 80,01 : %MW3080.1	**new:** Monitoring of the task with a cycle time = 0
KW 85,00 : %MW3085.0	Configuration of the binary inputs E62,00-E62,15
KW 85,01 : %MW3085.1	Configuration of the binary inputs E63,00-E63,15
KW 85,02 : %MW3085.2	Configuration of the operating modes for the high-speed counter
KW 86,00.: %MW3086.0	Configuration of the analog input EW 06,00
KW 86,01 : %MW3086.1	Configuration of the analog input EW 06,01
KW 86,02 : %MW3086.2	Configuration of the analog input EW 06,02
KW 86,03 : %MW3086.3	Configuration of the analog input EW 06,03
KW 86,04 : %MW3086.4	Configuration of the analog input EW 06,04
KW 86,05 : %MW3086.5	Configuration of the analog input EW 06,05
KW 86,06 : %MW3086.6	Configuration of the analog input EW 06,06
KW 86,07 : %MW3086.7	Configuration of the analog input EW 06,07
KW 88,00 : %MW3088.0	Configuration of the analog output AW 06,00
KW 88,01 : %MW3088.1	Configuration of the analog output AW 06,01
KW 88,02 : %MW3088.2	Configuration of the analog output AW 06,02
KW 88,03 : %MW3088.3	Configuration of the analog output AW 06,03
KD 00,00 : %MD4000.0	**without function** (for EBS: Setting the cycle time)

KW 00,00 / %MW3000.0:
PLC application mode: Master PLC, slave PLC or standalone PLC

Meaning of the initialization value of the constants:

– Master PLC on CS31 system bus:	-1 (FFFF_H)
– Standalone PLC:	-2 (FFFE_H)
– Slave PLC on CS31 system bus:	CS31 module addresses 0-61 and 100-115
– Range of values:	-2, -1, 0-61, 100-115
– Default value:	-2 (standalone PLC)

Examples:

Declaration for a standalone PLC:
KW00_00_MAST_SLV AT %MW3000.0 : INT := -2; (* Standalone PLC *)

Declaration for a master PLC on the CS31 system bus:
KW00_00_MAST_SLV AT %MW3000.0 : INT := -1; (* Master PLC *)

Declaration for slave PLC no. 5:
KW00_00_MAST_SLV AT %MW3000.0 : INT := 5; (* Slave PLC *)

CAUTION:
Changing the PLC application mode is carried out in three steps:
1. Changing the system constant KW 00,00 in the PLC
2. Create boot project (flash user program)
3. Activating the new PLC application mode by
- initiating a warm start (refer to section 1.2.1 "Terms") or by
- initiating a cold start (refer to section 1.2.1 "Terms").

For the operating mode "Slave PLC on the CS31 system bus" the following applies (refer also to chapter 1.5.6 - Intelligent I/O remote modules as slaves on the CS31 system bus):
- The value ranges for the address are 0..61 and 100..115. The highest permissible address depends on the size of both the set sending area and the set receiving area. The greater these two areas are chosen, the smaller is the highest permissible address.
- The slave basic unit can be used on the CS31 system bus in both the digital and the word area. When operated in word area, the sending and receiving data are on the channels 0..7 or 8...15. The selection is performed together with the address setting
  KW00_00 / %MW3000.0 := 0..5; 8..15 → channels 8..15
  KW00_00 / %MW3000.0 := 100..105; 108..115 → channels 8..15
  By adding 100 to the address, the upper channel range of 8...15 is configured.

KW 00,01 / %MW3000.1:
Initialization of binary flags

Meaning of the initialization value **n** of the constants:

→Initialized binary flag areas (set to 0)

n = 0 (Default)	M 000,00-M 511,15 / %MX0.0-%MX511.15
n = 1-511	M n,00-M 511,15 / %MXn.0-%MX511.15
n < 0, n > 511	M 255,10-M 511,15 / %MX255.10-%MX511.15

Example:

KW 00,01 := 52 and the battery is available:
KW00_01_INIT_M AT %MW3000.1 : INT := 52; (* Initialization of binary flag *)

Flags initialized with 0:	M 52,00-M 511,15 / %MX52.0-%MX511.15
Flags buffered:	M 00,00-M 51,15 / %MX0.0-%MX51.15

KW 00,02 / %MW3000.2:
Initialization of word flags

Meaning of the initialization value **n** of the constants:

→Initialized word flag areas (set to 0)

n = 0 (Default)	MW 000,00-MW 511,15 / %MW1000.0-%MW1511.15
n = 1-511	MW n,00-MW 511,15 / %MW1000+n.0-%MW1511.15
n < 0, n > 511	no initialization

Example:

KW 00,02 := 52 and the battery is available:
KW00_02_INIT_MW AT %MW3000.2 : INT := 52; (* Initialization of word flag *)

Flags initialized with 0:	MW 52,00-MW 511,15 / %MW1052.0-%MW1511.15
Flags buffered:	MW 00,00-MW 51,15 / %MW1000.0-%MW1051.15

KW 00,03 / %MW3000.3:
Initialization of double word flags

Meaning of the initialization value **n** of the constants:

→Initialized double word flag areas (set to 0)

n = 0 (Default)	MD 00,00-MD 63,15 / %MD2000.0-%MD2063.15
n = 1-63	MD n,00-MD 63,15 / %MD2000+n.0-%MD2063.15
n < 0, n > 63	no initialization

Example:

KW 00,03 := 52 and the battery is available:
KW00_03_INIT_MD AT %MW3000.3 : INT := 52; (* Initialization of double word flag *)

Flags initialized with 0:	MD 52,00-MD 63,15 / %MD2052.0-%MD2063.15
Flags buffered:	MD 00,00-MD 51,15 / %MD2000.0-%MD2051.15

KW 00,04 / %MW3000.4:
Initialization of step chains

Meaning of the initialization value **n** of the constants:

→Initialized step chains (set to step 0)

n = 0 (Default)	S 00,00-S 255,15 / %MW5000-%MW5255
n = 1-255	S n,00-S 255,15 / %MW5000+n-%MW5255
n < 0, n > 255	no initialization

Example:

KW 00,04 := 52 and the battery is available:
KW00_04_INIT_S AT %MW3000.4 : INT := 52; (* Initialization of step chains *)

Flags initialized with 0:	S 52,00-S 255,15 / %MW5052-%MW5255
Flags buffered:	S 00,00-S 51,15 / %MW5000-%MW5051

Note:
The step flags S 00,00-S 255,15 / %MW5000-%MW5255 are required for transferring projects which were created using 907 PC 331. The step flags **can not** be used as in the EBS operating system.

KW 00,07 / %MW3000.7:
PLC reaction to class 3 errors

Meaning of the initialization value **n** of the constants:

n = 0 (Default)	Errors are only reported
n < 0, >0	Errors are reported and the PLC program is aborted

Changing the system constant takes immediately effect.

Example:

```
KW 00,07 := 0
KW00_07_FK3_REAK AT %MW3000.7 : INT := 0; (* FK3 errors are only reported *)
```

KW 00,10 / %MW3000.10:
Transmit area size of the slave PLC

Meaning of the initialization value **n** of the constants:

The slave PLC may either be used on the CS31 system bus in the binary area or in the word area. The binary values are each transmitted byte-by-byte where the number of bytes or words which are sent from the slave to the master PLC can be set.

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

Changing the system constant takes effect:

– on the next warm start or cold start

Example:

```
KW 00,10 := 0
KW00_10_SLV_SEND AT %MW3000.10 : INT := 0; (* 4 byte data transfer *)
```

KW 00,11 / %MW3000.11:
Receive area size of the slave PLC

Meaning of the initialization value **n** of the constants:

The slave PLC may either be used on the CS31 system bus in the binary area or in the word area. The binary values are each transmitted byte-by-byte where the number of bytes or words which are sent from the slave to the master PLC can be set.

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

Changing the system constant takes effect:

– on the next warm start or cold start

Example:

```
KW 00,11 := 0
KW00_11_SLV_REC AT %MW3000.11 : INT := 0; (* 4 byte data transfer *)
```

Note:

The default setting is as following:

Binary area, sending 4 bytes and receiving 4 bytes.

This is achieved using the default combination KW 00,10 = KW 00,11 = 0. The planned combination KW 00,10 = KW 00,11 = 4 has the same effect as the default combination. The combination KW 00,10 = KW 00,11 = 100 (sending 0 words and receiving 0 words) is not allowed.

When used in the word area the unused upper 8 channels of the address can be seized by an analog module, if necessary.

KW 00,12 / %MW3000.12:
Automatically initiated warm start after FK2 errors

Meaning of the initialization value **n** of the constants:

n = 0 (Default)	No automatically initiated warm start after FK2 errors
n = 1	Automatically initiated warm start after FK2 errors

Changing the system constant takes effect after the next warm start.

Example:

KW 00,12 := 0
KW00_12_SYSTEM AT %MW3000.12 : INT := 0; (* no autom. initiated warm start *)

KW 00,15 / %MW3000.15:
Configuration of the oscillators

Meaning of the initialization value **n** of the constants:

- n = 0 (Default)	Oscillators at M 255,00 to M 255,06 active
- n = 1	Oscillators changed to M 256,00 to M 256,06 active
- n = 2 or 3	No oscillators present

Example:

KW 00,15= 52 and the battery is available:
KW00_15_BLINK AT %MW3000.15 : INT := 0; (* Oscillators active *)

KW 80,00 / %MW3080.0:
Monitoring of the task with a cycle time > 0

Meaning of the initialization value **n** of the constants:

- n = 0 (Default)	FK2 error after the watchdog exceeded the cycle time three times or exceeded the triple cycle time.
- n > 0	A pause period of 10 % of the cycle time is performed after the task is finished if the watchdog exceeded the cycle time three times or if a FK2 error occurred after the watchdog exceeded the value of n [in ms].
– Range of values:	0-32767 ms
– Default value:	0

Example:

KW 80,00= 1000 and the task is entered to the task configuration together with the cycle time:
KW80_00_TASK_ZYKL AT %MW3080.0 : INT := 1000; (* FK2 after time period *)

Cycle time violations are entered to the system flags MW259,1-MW259,3 / %MW1259.1-%MW1259.3.

KW 80,01 / %MW3080.1:
Monitoring of the task with a cycle time = 0

Meaning of the initialization value **n** of the constants:

- n = 0 (Default)	The watchdog initiates a FK2 error if a time period of 10 sec is exceeded.
- n > 0	The watchdog initiates a FK2 error if the cycle time [ms], corresponding to the value of n, is exceeded.
– Range of values:	0-32767 ms
– Default value:	0

Example:

KW 80,00= 1000 and no task or a cycle time of 0 is entered to the task configuration:
KW80_01_TASK_ZYKL0 AT %MW3080.1 : INT := 1000; (* FK2 after time period *)

KW85,00 / %MW3085.0:
Configuration of the binary inputs E62,00-E62,15:

Meaning of the initialization value **n** of the constants:

Bit	00	configures input	E 62,00 / %IX62.0
	01		E 62,01 / %IX62.1
	...		...
	14		E 62,14 / %IX62.14
	15		E 62,15 / %IX62.15
Bit = 0		Input delay	= 7 ms (default value)
Bit = 1		Input delay	= 1 ms
Default value:		0 (all inputs with an input delay of 7 ms)	

Example:

KW 85,00= 255: E 62,00-E 62,07 – 1 ms and E 62,08-E 62,15 – 7 ms
KW85_00 AT %MW3085.0 : INT := 255; (* Input delay E62,00-E62,15 *)

KW85,01 / %MW3085.1:
Configuration of the binary inputs E63,00-E63,15:

Meaning of the initialization value **n** of the constants:

Bit	00	configures input	E 63,00 / %IX63.0
	01		E 63,01 / %IX63.1
	...		...
	14		E 63,14 / %IX63.14
	15		E 63,15 / %IX63.15
Bit = 0		Input delay	= 7 ms (default value)
Bit = 1		Input delay	= 1 ms
Default value:		0 (all inputs with an input delay of 7 ms)	

Example:

KW 85,01= 255: E 63,00-E 63,07 – 1 ms and E 63,08-E 63,15 – 7 ms
KW85_01 AT %MW3085.1 : INT := 255; (* Input delay E 63,00-E 63,15 *)

Note:
The binary inputs E 63,00-E 63,07 / %IX63.0-%IX63.7 are the designators for the 8 inputs of the binary I/Os.

KW85,02 / %MW3085.2:
Configuration of the operating mode for the high-speed counter:

Meaning of the initialization value **n** of the constants:	
- 00	No counter (default value)
- 01	Mode 1, one count-up counter
- 02	Mode 2, one count-up counter with release input
- 03	Mode 3, two count-up/count-down counters
- 04	Mode 4, two count-up/count-down counters, the second counter counts the pulses on the falling edge
- 05	Mode 5, one count-up/count-down counter with dynamic set input; setting a rising edge
- 06	Mode 6, one count-up/count-down counter with dynamic set input; setting a falling edge
- 07	Mode 7, one count-up/count-down counter with directional discriminator
The high byte is configured with 0.	
- Default value:	0 (no counter)

Example:

KW 85,02= 1: Mode 1, one count-up counter
KW85_02_COUNTER AT %MW3085.2 : INT := 1; (* Configuration high-speed counter *)

KW86,0x /%MW3086.x:
Configuration of the analog inputs:

KW 86,00 / %MW3086.0 configures analog input EW 6,00 / %IW1006.0.
KW 86,01 / %MW3086.1 configures analog input EW 6,01 / %IW1006.1.
KW 86,02 / %MW3086.2 configures analog input EW 6,02 / %IW1006.2.
KW 86,03 / %MW3086.3 configures analog input EW 6,03 / %IW1006.3.
KW 86,04 / %MW3086.4 configures analog input EW 6,04 / %IW1006.4.
KW 86,05 / %MW3086.5 configures analog input EW 6,05 / %IW1006.5.
KW 86,06 / %MW3086.6 configures analog input EW 6,06 / %IW1006.6.
KW 86,07 / %MW3086.7 configures analog input EW 6,07 / %IW1006.7.

Meaning of the initialization value **n** of the constant (hex value in the low byte):

- 00_H Analog input 0-10 V (default value)
- 01_H not used
- 02_H Digital input with an input delay of 7 ms
- 03_H Analog input 0...20 mA
- 04_H Analog input 4...20 mA
- 05_H Analog input –10...+10 V
- 06_H Analog input 0...5 V
- 07_H Analog input –5...+5 V
- 08_H Analog input Pt100 with two-wire connection –50...+400°C
- 09_H Analog input Pt100 with three-wire connection –50...+400°C*
- $0A_H$ Analog input 0...10 V differential input *
- $0B_H$ Analog input -10...+10 V differential input *
- $0C_H$ Analog input 0...5 V differential input *
- $0D_H$ Analog input -5...+5 V differential input *
- $0E_H$ Analog input Pt100 with two-wire connection –50...+70°C
- $0F_H$ Analog input Pt100 with three-wire connection –50...+70°C*

Hex value in the high byte. Meaning of the bits **15 14 13 12 11 10 09 08**:

- 00_H Plausibility monitoring and cut-wire and short-circuit monitoring (default value)
- 01_H Cut-wire and short-circuit monitoring (default value)
- 02_H Plausibility monitoring
- 03_H No monitoring

Refer to "Measuring ranges of the analog input channels" for more detailed information about monitoring.

Changing the system constant takes effect:
– on the next warm start or cold start

*) For the operating modes "Pt100 with three-wire connection" and all configurations with differential input always two adjacent analog inputs belong together (e.g. EW06,00 and EW06,01).
In this case both inputs are configured according to the desired operating mode. The lower address has to be the even address (EW06,00) and the next higher address (EW06,01) has to be the odd address.
The converted analog value is available on the higher address (EW06,01).

Examples

a) EW06,00 / %IW1006.0 as an analog input ±10V, standard monitoring

Configuration in KW86,00:
High byte: 00_H = Standard monitoring
Low byte: 05H = ±10V
KW86,00: 0005H = +5 decimal
KW86_00_Ana0 AT %MW3086.0 : INT := 5; (* Configuration EW06,00 *)

b) EW06,02 / %IW1006.2 (even) and EW06,03 / %IW1006.3 (next higher) as analog input Pt100, three-wire, -50...+400°C, only cut-wire and short-circuit monitoring

Configuration in KW86,02 / %MW3086.2:
High byte: 01_H = Cut-wire and short-circuit monitoring
Low byte: 09_H = Pt100, three-wire, -50...+400°C
KW86,02: 0109_H = +265 decimal
KW86_02_Ana2 AT %MW3086.2 : INT := 265; (* Configuration EW06,02 *)

Configuration in KW86,03 / %MW3086.3:
High byte: 01_H = Cut-wire and short-circuit monitoring
Low byte: 09_H = Pt100, three-wire, -50...+400°C
KW86,03: 0109_H = +265 decimal
KW86_03_Ana3 AT %MW3086.3 : INT := 265; (* Configuration EW06,03 *)

The measured value is available on **EW06,03 / %IW1006.3**.

Measuring ranges of the analog input channels:

Resolution in the PLC system:

The measured values are converted with a resolution of 12 bits, i.e. 11 bits plus sign for voltage and 12 bits without sign for current. The ranges 0..5 V and ±5 V are converted with 10 bits plus sign.

Examples:

Measuring range	Bereich der Zahlendarstellung
-10 V..0..10 V	$-32760_D..0..32760_D$
	$8008_H..0000..7FF8_H$
0..20 mA	$0..32760_D$
	$0000..7FF8_H$

Further details can be found in volume 2, chapter 5.1 „General information for the use of analog inputs and outputs“

In order to make sure that **unused input channels** have a defined 0V level, they may be shorted to AGND. Unused inputs must be configured with **"unused".**

The relationship between the analog input channels and the converted numbers is illustrated in the following figures.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–100 %	50 %	25 %	12.5 %	6.25 %	3.13 %	1.56 %	0.78 %	0.39 %	0.20 %	0.10 %	0.05 %	0.02 %	0	0	0
	Sign															
±10 V	–10V	5V	2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5 mV	2mV	0	0	0
± 5 V	–5V	2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5mV	2mV	1mV	0	0	0
0...10 V		5V	2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5mV	2mV	0	0	0
0...5 V		2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5mV	2mV	1mV	0	0	0
0...20 mA		10mA	5mA	2.5mA	1.25mA	625µA	313µA	156µA	78µA	39µA	20µA	10µA	5µA	0	0	0
4...20 mA		8mA	4mA	2mA	1mA	500µA	250µA	125µA	62.5µA	31.3µA	16µA	8µA	4µA	+ 4 mA offset		
Bit values	–32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Measuring ranges ±10 V, 0...10 V — 11 bits resolution plus sign,
Measuring ranges ±5 V, 0...5 V — 10 bits resolution plus sign,
Measuring ranges 0...20 mA, 4...20 mA — 12 bits resolution without sign,

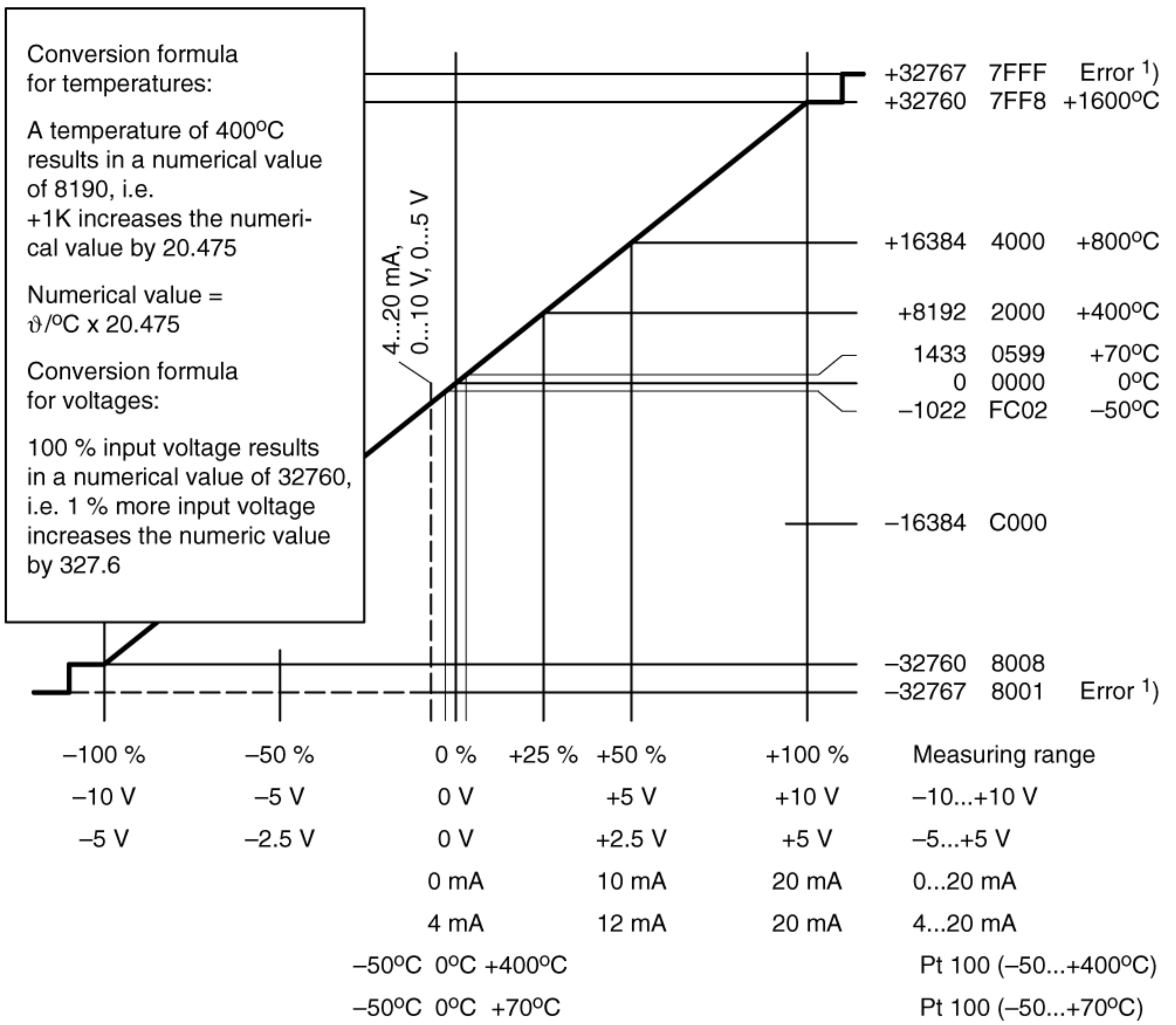
the value range of –100...+100 % corresponds to the numbers 8008_H...$7FF8_H$ (–32760...+32760),
range overflow: $7FFF_H$ (32767), range underflow: 8001_H (–32767)
open–circuit 4...20 mA: 8001_H (–32767)

Remark: Independent of the resolution, all numbers are represented with 12 bits plus sign. Because of the results of internal calculations, all these bits can appear.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–100 %	50 %	25 %	12.5 %	6.25 %	3.13 %	1.56 %	0.78 %	0.39 %	0.20 %	0.10 %	0.05 %	0.02 %	0.01 %	0.005 %	0
	Sign															
Pt100	–1600°C	800°C	400°C	200°C	100°C	50°C	25°C	12.5°C	6.25°C	3.13°C	1.56°C	0.78°C	0.39°C	0.2°C	0.1°C	0
Bit values	–32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

The measuring range for Pt100 is 11 bits plus sign,
the value range of –50...+400 °C corresponds to the numbers $FC02_H$...$1FFE_H$ (–1022...+8190),
the value range of –50...+70 °C corresponds to the numbers $FC02_H$...0599_H (–1022...+1433),
range overflow / open–circuit: $7FFF_H$ (32767),
range underflow / short–circuit of the sensor: 8001_H (–32767)

Relationship between the measured values and the positions of the bits in the 16-bit word



The following is valid for the platinum resistance thermometers:

Pt100 = Platinum 100 Ω @ 0 °C Measuring range –50...+400°C
Measuring range –50...+70°C

The dashed line is valid for the measuring ranges 4...20 mA, 0...10 V and 0...5 V. If in these ranges the measuring value is 2...3 % lower than the lower limit, this is evaluated as a range underflow.

Relationship between measuring value and converted numerical value, voltage input, current input, temperature input

1) In case of a configured plausibility monitoring, range overflow and range underflow by 2...3 % result in an error message (FK4, error number 10).
Independent of the configured monitoring, the error values +32767 and -32767 are always generated in case of range underflow and range overflow. Exception: In the measuring range of 0...20 mA only the **range overflow** is detected.

Measuring ranges ±10 V / ±5 V / 0..10 V / 0..5 V

Input voltages that exceed the measuring range cause the overflow numerical value of +32767. If the measured value is below the range, the underflow numerical value of -32767 is set.
The input impedance is > 100 kΩ.

Measuring range 4..20 mA (passive-type 2-pole sensors)

Input voltages that exceed the measuring range cause the overflow numerical value of +32767. If the measured value is below the range, the underflow numerical value of -32767 is set.

The input impedance is ca. 330 Ω. The current input has a self-protecting mechanism. If the input current gets too high, the shunt is switched off and the value for range overflow is generated. About every second, the unit tries to switch on the shunt again. In this way the correct measurement will succeed after the current has reached a normal value again.

The trigger of the self-protecting mechanism is displayed by the red LED Ovl as long as the overload is present. In the PLC system an error message is then stored (FK4, error number 4).

The open-circuit monitoring begins below ca. 3 mA. The value of the range underflow is stored. If the open-circuit monitoring is configured, the open-circuit event is displayed by the red LED Ovl as long as it is present. In the PLC system an error message is stored (FK4, error number 9).

Measuring range 0..20 mA (active-type sensors with external supply voltage)

Input voltages that exceed the measuring range cause the overflow numerical value of +32767. If the measured value is below the range, the underflow numerical value of -32767 is set.

The input impedance is ca. 330 Ω. The current input has a self-protecting mechanism. If the input current gets too high, the shunt is switched off and the value for range overflow is generated. About every second, the unit tries to switch on the shunt again. In this way the correct measurement will succeed after the current has reached a normal value again.

The trigger of the self-protecting mechanism is displayed by the red LED Ovl as long as the overload is present. In the PLC system an error message is then stored (FK4, error number 4).

Measuring ranges ±10 V / ±5 V / 0..10 V / 0..5 V as differential inputs

Differential inputs are very useful, if analog sensors are used which are remotely non-isolated (e.g. the minus terminal is remotely earthed).

Since the earthing potential is not exactly the same as AGND1, it has to be measured bipolar in order to compensate measuring errors. Additionally, in case of single-pole configuration, AGND1 would be connected directly to the remote earth potential. This would cause inadmissible (and possibly dangerous) earthing loops.

In all configurations using **differential inputs** two adjacent analog inputs belong together (e.g.. EW06,00 / %IW1006.0 and EW06,01 / %IW1006.1).

The measured value is calculated by subtraction. The value of the channel with the lower address is subtracted from the value of the channel with the higher address.

The converted measured value is available on the odd address (e.g. EW06,01 / %IW10006.1).

Important:
The common mode input voltage range equals the measuring range of the single channel. I.e. that the signals, related to AGND, at the two involved inputs must not exceed this measuring range.

Input voltages that exceed the measuring range cause the overflow numerical value of +32767. If the measured value is below the range, the underflow numerical value of -32767 is set.

Measuring range –50 °C..+400 °C and –50 °C..+70 °C with Pt100 as temperature sensor in 2-wire configuration

When resistance thermometers are used, a constant current must flow through the measuring resistor in order to create the necessary voltage drop for the evaluation. For this purpose, the basic unit 07 KT 97 provides a constant current sink, which is multiplexed to the 8 analog channels.

Depending on the configured operating mode, the measured value is assigned linearly as follows:

Range	assigned numerical value range	
–50 °C..400 °C	-1022..+8190	($FC02_H$..$1FFE_H$)
–50 °C..70 °C	-1022..+1433	($FC02_H$..0599_H)

The basic unit linearizes the Pt100 characteristic.

Temperatures, that exceed the measuring range cause the overflow numerical value of +32767. If the measured value is below the range, the underflow numerical value of -32767 is set.

In case of a detected open-circuit the numerical value of +32767 is set. If the sensor is short-circuited, the numerical value of -32767 is set.

If the open-circuit or short-circuit monitoring is configured, the detected error is displayed by the red LED Ovl as long as it is present. In the PLC system an error message is stored (FK4, error number 9).

In order to avoid error messages with unused analog inputs, it is useful, **not** to configure these channels for Pt100.

Measuring ranges –50 °C..+400 °C and –50 °C..+70 °C with Pt100 as temperature sensor in 3-wire configuration

In the operating mode "Pt100 in 3-wire configuration" two adjacent analog inputs belong together (e.g. EW06,00 / %IW1006.0 and EW06,01 / %IW1006.1).

For configuration, both inputs must be configured to the desired operating mode.

The constant current of the one channel flows through the Pt100 resistance sensor, the constant current of the other channel through one of the wires.

The basic unit calculates the measuring value from the two voltage drops and stores it under the odd address (e.g. EW06,01 / %IW1006.1).

Depending on the configured operating mode, the measured value is assigned linearly as follows:

Range	assigned numerical value range	
–50 °C..400 °C	-1022..+8190	($FC02_H$..$1FFE_H$)
–50 °C..70 °C	-1022..+1433	($FC02_H$..0599_H)

The basic unit linearizes the Pt100 characteristic.

Temperatures, that exceed the measuring range cause the overflow numerical value of +32767. If the measured value is below the range, the underflow numerical value of -32767 is set.

In case of a detected open-circuit the numerical value of +32767 is set. If the sensor is short-circuited, the numerical value of -32767 is set.

If the open-circuit or short-circuit monitoring is configured, the detected error is displayed by the red LED Ovl as long as it is present. In the PLC system an error message is stored (FK4, error number 9).

In order to avoid error messages with unused analog inputs, it is useful, **not** to configure these channels for Pt100.

Use of analog inputs as digital inputs

Several (or all) analog inputs can be configured as digital inputs. When doing so, they evaluate input voltages higher than ca. +7 V as signal TRUE.

The input signal delay is 7 ms. It cannot be changed by configuration. The inputs are not electrically isolated.

KW88,0x /%MW3088.x:
Configuration of the analog outputs:

KW 88,00 / %MW3088.0 configures analog output AW 6,00 / %QW1006.0.
KW 88,01 / %MW3088.1 configures analog output AW 6,01 / %QW1006.1.
KW 88,02 / %MW3088.2 configures analog output AW 6,02 / %QW1006.2.
KW 88,03 / %MW3088.3 configures analog output AW 6,03 / %QW1006.3.

Meaning of the initialization value **n** of the constant (hex value in the low byte):

- 00_H Analog output ±10 V (default value)
- 01_H not used
- 02_H Analog output 0...20mA
- 03_H Analog output 4...20 mA

Hex value in the high byte. Meaning of the bits **15 14 13 12 11 10 09 08**:

No significance, reserved, can be configured with 00_H.

Changing the system constant takes effect:
– on the next warm start or cold start

Measuring ranges of the analog outputs:

Resolution in the control system:
All analog output values are converted with a resolution of 12 bits, i.e. either 11 bits plus sign or 12 bits without sign.

Examples:

Range of numerical values	Output value
-32760D..0..32760D	-10 V..+10 V
8008H..0000..7FF8H	
0..32760D	0..20 mA
0000..7FF8H	

Further details can be found in volume 2, chapter 5.1 „General information for the use of analog inputs and outputs“.

Unused output channels may be left unconnected. They are configured as **"unused"**.

The relationship between numerical values and the output analog signals is illustrated in the following figures.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	−100 %	50 %	25 %	12.5 %	6.25 %	3.13 %	1.56 %	0.78 %	0.39 %	0.20 %	0.10 %	0.05 %	0.02 %	0	0	0
	Sign															
±10 V	−10V	5V	2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5 mV	2mV	0	0	0
± 5 V	−5V	2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5mV	2mV	1mV	0	0	0
0...10 V		5V	2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5mV	2mV	0	0	0
0...5 V		2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5mV	2mV	1mV	0	0	0
0...20 mA		10mA	5mA	2.5mA	1.25mA	625µA	313µA	156µA	78µA	39µA	20µA	10µA	5µA	0	0	0
4...20 mA		8mA	4mA	2mA	1mA	500µA	250µA	125µA	62.5µA	31.3µA	16µA	8µA	4µA	+ 4 mA offset		
Bit values	−32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Measuring ranges ±10 V, 0...10 V — **11 bits resolution plus sign,**
Measuring ranges ±5 V, 0...5 V — **10 bits resolution plus sign,**
Measuring ranges 0...20 mA, 4...20 mA — **12 bits resolution without sign,**

the value range of –100...+100 % corresponds to the numbers 8008_H...$7FF8_H$ (–32760...+32760),
range overflow: $7FFF_H$ (32767), range underflow: 8001_H (–32767)
open–circuit 4...20 mA: 8001_H (–32767)

Remark: Independent of the resolution, all numbers are represented with 12 bits plus sign. Because of the results of internal calculations, all these bits can appear.

Relationship between numerical value and analog value

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	−100 %	50 %	25 %	12.5 %	6.25 %	3.13 %	1.56 %	0.78 %	0.39 %	0.20 %	0.10 %	0.05 %	0.02 %	0	0	0
	Sign															
±10 V	−10V	5V	2.5V	1.25V	625mV	313mV	156mV	78mV	39mV	20mV	10mV	5 mV	2mV	0	0	0
0...20 mA		10mA	5mA	2.5mA	1.25mA	625µA	313µA	156µA	78µA	39µA	20µA	10µA	5µA	0	0	0
4...20 mA		8mA	4mA	2mA	1mA	500µA	250µA	125µA	62.5µA	31.3µA	16µA	8µA	4µA	+ 4 mA offset		
Bitwerte	−32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Measuring range ±10 V — **11 bits resolution plus sign,**
Measuring ranges 0...20 mA, 4...20 mA — **12 bits resolution without sign,**

the value range of –100...+100 % corresponds to the numerical values 8008_H...$7FF8_H$ (–32760...+32760),
Range overflow: $7FFF_H$ (32767), Range underflow: 8001_H (–32767)

Relationship between the output values and the position of the bits in the 16-bit word

1.2.6 System and diagnosis flags

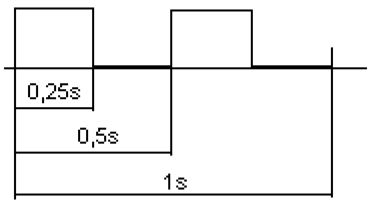
The following areas are reserved for system flags and system flag words. It is also not allowed to use flags or flag words from this areas which are not yet seized for other purposes.

Flags:	M255,00 – M279,15	%MX255.0 - %MX279.15
Flag words:	MW254,00 – MW259,15	%MW1254.0 - %MW1259.15

The system and diagnosis flags are declared in the export files Fehler_M.exp and Auslastung.exp. When a new project is being created with "File / New", these files are automatically stored in the "Resources" menu under the "Global variables" object. Otherwise they can be read in with "Project / Import".

M255,0-M255,6 /%MX255.0-%MX255.6:
Oscillators:

M 255,00 : %MX255.0	Oscillator approx. 2 Hz
M 255,01 : %MX255.1	Oscillator approx. 1 Hz
M 255,02 : %MX255.2	Oscillator approx. 0.5 Hz
M 255,03 : %MX255.3	Oscillator with a period duration of approx. 1 minute (1/64 Hz)
M 255,04 : %MX255.4	Oscillator with a period duration of approx. 8 seconds (1/8 Hz)
M 255,05 : %MX255.5	Oscillator approx. 4 Hz
M 255,06 : %MX255.6	Oscillator approx. 8 Hz



Example: Oscillator 2 Hz (= 2 periods per second)

Caution:
The oscillators are only active if the system constant is KW 00,15 / %MW3000.15 = 0 (see system constant KW 00,15 / %MW3000.15).

M255,9 /%MX255.9:
Interrupt from external networking interface:

M 255,9 : %MX255.9 Interrupt from the external networking interface

A flag value of TRUE indicates that an interrupt was initiated by the external networking interface. The value must be set back to FALSE by the user.

M255,10-M255,14 /%MX255.10-%MX255.14:
Error messages from the PLC:

M 255,10 : %MX255.10	Sum error message, indicates that the PLC has detected an error
M 255,11 : %MX255.11	Error message FK1, fatal error, detailed information in MW 254,00-MW 254,07 / %MW1254.0-%MW1254.7
M 255,12 : %MX255.12	Error message FK2, serious error, detailed information in MW 254,08-MW 254,15 / %MW1254.8-%MW1254.15
M 255,13 : %MX255.13	Error message FK3, light error, detailed information in MW 255,00-MW 255,07 / %MW1255.0-%MW1255.7
M 255,14 : %MX255.14	Error message FK4, warning, detailed information in MW 255,08-MW 255,15 / %MW1255.8-%MW1255.15

Refer to chapter 1.6 "Diagnosis" for a detailed description of the error messages.

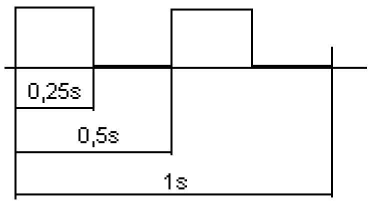
M255,15 /%MX255.15:
"Restart" detection:

M 255,15 : %MX255.15	"Restart" detection (detection of the first cycle)

This binary flag can be used to detect the first program cycle after a program start. It is always initialized to "0" on each start of the user program, independent of the initialization settings predetermined by the system constant. Interrogating this flag in the user program and subsequently setting it to "1", allows to determine whether the user program was restarted or not.

M256,0-M256,6 /%MX256.0-%MX256.6:
Oscillators:

M 256,00 : %MX256.0	Oscillator approx. 2 Hz
M 256,01 : %MX256.1	Oscillator approx. 1 Hz
M 256,02 : %MX256.2	Oscillator approx. 0,5 Hz
M 256,03 : %MX256.3	Oscillator with a period duration of approx. 1 minute (1/64 Hz)
M 256,04 : %MX256.4	Oscillator with a period duration of approx. 8 seconds (1/8 Hz)
M 256,05 : %MX256.5	Oscillator approx. 4 Hz
M 256,06 : %MX256.6	Oscillator approx. 8 Hz



Example: Oscillator 2 Hz (= 2 periods per second)

Caution:
The oscillators are only active if the system constant is KW 00,15 / %MW3000.15 = 1 (see system constant KW 00,15 / %MW3000.15).

MW254,00-MW255,15 /%MW1254.00-%MW1255.15:
Error messages from the PLC:

The detailed messages of the error flags are contained in the area MW254,00-MW255,15 / %MW1254.0-%MW1255.15. Refer to chapter 1.6 "Diagnosis" for a detailed description of the error messages.

MW259,00 /%MW1259.0:
Capacity utilization of the PLC:

The flag word MW259,0 / %MW1259.0 displays the current capacity utilization of the PLC in %. Refer to chapter 1.3 "Processing times" for more detailed information.

MW259,01-MW259,03 /%MW1259.1-%MW1259.3:
Cycle time exceeding indication of the PLC:

Cycle time exceedings of the tasks are indicated in the flag words MW259,1-MW259,3 / %MW1259.1-%MW1259.3. Refer to chapter 1.3 "Processing times" for more detailed information.

MW 259,1 : %MW1259.1	Number of the task which exceeded the cycle time
MW 259,2 : %MW1259.2	Time in ms that calling of the task was performed too late
MW 259,3 : %MW1259.3	Number of missed task calls (cumulative counter for all PLC user tasks)

EW 07,15 /%IW1007.15:
CS31 status word:

Bit 0 = 1 :	No CS31 error of class 2 present
Bit 1 = 1 :	PLC is added to the CS31 cycle (only relevant for use as slave)
Bit 2 = 1 :	Time and date are valid
Bit 3 = 1 :	Battery available
Bit 4...7 :	Not used
Bit 8..15 :	Maximum number of modules which were detected on the CS31 system bus up to the current point in time (only relevant for use as slave)

1.3 Starting the controller

1.3.1 Terms

Cold start:

- All RAM memory modules are tested and erased.
- If there is no user program present in the Flash EPROM, all system constants are set to the default values (corresponds to the settings on delivery).
- If there is a user program present in the Flash EPROM, it is loaded into the RAM together with the system constants.
- The operating modes predetermined by the system constants are set.
- The CS31 system bus is reinitialized.

Initiating a cold start:

- Voltage ON/OFF if no battery is available or
- menu item "Online/Cold start" in the programming system.

Warm start:

- All RAM memory modules are tested and erased except of the buffered operand areas and the RETAIN variables.
- If there is a user program present in the Flash EPROM, it is loaded into the RAM together with the system constants.
- The operating modes predetermined by the system constants are set.
- The CS31 system bus is reinitialized.

Initiating a warm start:

- Voltage ON/OFF if the battery is available or
- menu item "Online/Reset" in the programming system.

Buffering of data areas:

The buffering of data, i.e. the maintenance of data after voltage ON/OFF, is only possible when the battery is available. The following can be buffered completely or in parts:

- Binary flags
- Word flags
- Double word flags
- Step chains (for transferring projects generated with 907 PC 331)
- RETAIN variable

In order to buffer particular data these data must be excluded from the initialization process.

1.3.2 Initialization and backup of operands

The initialization with 0 or to the initial value is performed with the program start.

For the operands in the % area, the areas (area parts) to be backed are set by system constants (see section "System constants"). If no battery is available or the system constants correspond to the delivery settings (default values) the data areas mentioned above are completely set to 0 after voltage ON/OFF.

If internal variables are to be backed, then they must be marked as „VAR_RETAIN“ with the declaration.

Note:
The internal variables only retain their order of sequence in case of an Online Change. If the program is translated, the order of sequence can change and (consequently) the backed values are wrong.

The following table shows an overview of the initialization of the variables:

Action	VAR	VAR := value	VAR_ RETAIN	VAR_ RETAIN := value	%M	%M := value	%M buffered with KW0,x	%M buffered with KW0,x := value
Without battery								
Voltage ON	0	value	0	value	0	0	0	value
STOP→RUN	0	value	unchanged	unchanged	0	0	unchanged	value
STOP→START	0	value	unchanged	unchanged	0	0	unchanged	value
Download	0	value	unchanged	unchanged	0	0	unchanged	value
Online change	unchanged	unchanged	unchanged	unchanged	unchanged	unchanged	unchanged	unchanged
Reset (warm start)	0	value	0	value	0	0	0	value
Cold start	0	value	0	value	0	0	0	value
Battery available								
	0	value	unchanged	unchanged	0	0	unchanged	value
STOP→RUN	0	value	unchanged	unchanged	0	0	unchanged	value
STOP→START	0	value	unchanged	unchanged	0	0	unchanged	value
Download	0	value	unchanged	unchanged	0	0	unchanged	value
Online change	unchanged	unchanged	unchanged	unchanged	unchanged	unchanged	unchanged	unchanged
Reset (warm start)	0	value	unchanged	unchanged	0	0	unchanged	value
Cold start	0	value	0	value	0	0	0	value

Note:
- The area of the constants KW00,00-KW89,15 / %MW3000.0-%MW3089.15 and the double word constants KD00,00-KD23,25 / %MD4000.0-%MD4023.15 is stored in the Flash EPROM together with the user program and initialized with the initialization value when the controller is started.

1.4 Processing times

The most important times for the use of the basic unit 07 KT 97 with or without connected remote modules are as follows:

- The reaction time is the time between a signal transition at the input terminal and the signal response at the output terminal.
  For binary signals, the reaction time consists of the input delay, the cycle time of the program processing and the bus transmission time, if the system is expanded by remote modules.

- The cycle time determines the time intervals after which the processor restarts the execution of the user program.
  The cycle time has to be specified by the user. It should be greater than the program processing time of the user program plus the data transfer times and the related waiting times.
  The cycle time is also the time base for some time-controlled functions, such as for the PID controller.

- The program processing time is the net time for processing the user program.

1.4.1 Program processing time

Instructions	07 KT 95..97	07 KT 98
• **Binary instructions of the type:**		
!M /M &M =M !NM /NM &NM =NM Processing time for 1000 instructions:	1.22 ms	0.32 ms
!M /M &M =SM !NM /NM &NM =RM Processing time for 1000 instructions:	1.46 ms	0.52 ms
• **Word instructions of the type:**		
!MW +MW -MW =MW !-MW -MW +MW =-MW Processing time for 1000 instructions:	3.37 ms	0.90 ms
!MW *MW :MW =MW !-MW *-MW :-MW =-MW Processing time for 1000 instructions:	4.78 ms	1.14 ms
• **Mixed instructions:**		
- 65 % binary: !, /, &, = - 20 % word: !, +, -, = - 15 % word: !, *, :, = Processing time for 1000 instructions:	2.18 ms	0.48 ms

1.4.2 Set cycle time

It is assumed that the processor always gets access in a moment with a worst-case condition.

The cycle time is stored in the task configuration and can be selected in steps of 1 ms. If the selected cycle time is too short, the processor will not be able to fulfil the tasks assigned to it every cycle. This will result in a time delay.

If this lack of time becomes too large over several cycles, the processor aborts the program execution and outputs an error (FK2).

Using some function blocks, such as the PID controller, the error-free execution depends on an exact timing sequence. Make sure that there is a larger time reserve.

The correct setting of the cycle time should be checked using the following procedure:

- Loading the user program into the basic unit.
- If the operating mode has been changed from standalone to bus master: Voltage ON or menu item "Reset" in the programming software.
- Interrogation of the capacity utilization using the menu item "Online/Global Variable/Utilization".
- Changing the cycle time until the capacity utilization is below 80 %.

When setting the cycle time the following values are to be taken into account:

- Block copy time: Time for copying the input signals from the DPR to the I/O image:
  binary inputs (E):
    basic run time 28 µs + 3 µs per group of inputs (E)
    + 21 µs for input (E) via the CS31 bus
  analog inputs (EW):
    basic run time 20...50 µs + 6 µs per group of inputs (EW)
  2 interface interrupts: 2 * 50 ms
- Time for copying the input signals of the user task from the I/O image to the image memory
- Program processing time
- Time for copying the output signals of the user task from the I/O image to the image memory
- Block copy time: Time for copying the output signals from the I/O image into the DPR:
  binary outputs (A):
    basic run time 33 µs +5 µs per group of outputs (A)
    +30 ms for output (A) via the CS31 bus
  analog outputs (AW):
    basic run time 20...32 µs + 6 µs per group of outputs (AW)
  2 interface interrupts: 2 * 50 ms
- When using the setting Master or Slave, one interrupt from the CS31 bus must be added for each CS31 bus cycle (36 µs...330 µs):
    basic run time 36 µs + 8 µs per group of inputs (EW)
    + 8 µs per group of outputs (AW) + 30 µs for E + 40 µs for A
- Receiving/Sending interrupts from ARCNET telegrams within the cycle time

- Receiving/Sending interrupts from the serial interface within the cycle time
- Task changes
- Runtime of the watchdog task

1.4.3 Reaction time for binary signals

Direct inputs and outputs:

- Delay of the binary inputs (E), configurable 7 ms / 1 ms
- Cycle time of the I/O processor, approx. 1 ms
- 2 * PLC cycle time (adjustable by the user)
- Cycle time of the I/O processor, approx. 1 ms
- Delay of the binary outputs (A) (negligible)

Inputs and outputs via the CS31 bus:

- Delay of the binary remote modules, usually 8 ms
- Basic time of CS31 bus (2 ms) + 2 * Sum of the bus transmission times of the remote modules (refer to volume 2 for the bus transmission times of the remote modules).
- Interrupt CS31 bus (see cycle time: 36 µs...330 µs)
- 2 * PLC cycle time (adjustable by the user)
- Basic time of CS31 bus (2 ms) + Sum of the bus transmission times of the remote modules
- Interrupt CS31 bus (see cycle time: 36 µs...330 µs)
- Delay of the binary outputs (A), usually < 1 ms

1.4.4 Cycle time monitoring

In order to monitor the cycle time and in particular to prevent infinite loops, the two system constants KW80,0 / %MW3080.0 and KW80,1 / %MW3080.1 are available. The mode of action is described in the following.

There are two different cases for an user program which runs with a set cycle time:

- The program end is reached but the cycle time is not met. In this case the task is able to monitor itself.

- The program end is not reached ("infinite loop"). The task is then stopped by the watchdog task. The watchdog task works with an interval of 200 ms. That means that this monitoring has an "inaccuracy" of 200 ms.

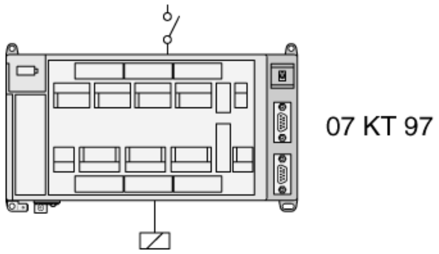
Also an user program with a cycle time of 0 can be configured. This is the default setting with PLC_PRG and without task configuration. A cycle time of 0 means that the program runs "as fast as possible" and there is no definite time that has to be met. This program is only monitored by the watchdog.

If a cycle time error occurs, an FK2 error with the error number 131 is initiated. The detailed information of the FK2 error contains the task ID. All user programs go into the STOP state. In order to restart the programs, a warm start or a cold start can be performed.

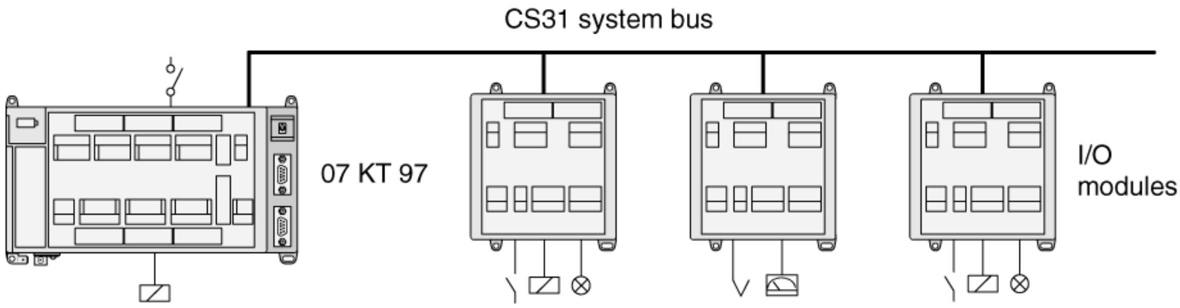
KW80,0 [ms] Monitoring of the task with a cycle time > 0	**Cycle time violation**	**Infinite loop**
0	Behaviour "like EBS"; the FK2 error is initiated after the cycle time is "violated" three times.	The watchdog initiates the FK2 error if the triple cycle time is exceeded.
>0	The task is paused after the PE for a period of 10 % of the set cycle time if the cycle time was violated three times. (During this time other PLC programs and operating system functions can be processed). Cycle time violations are entered to the following system flags: %MW1259,1 Task ID %MW1259,2 Error in ms %MW1259,3 Number of faulty cycles (cumulative counter for all PLC user tasks) The task returns to the set cycle time as soon as possible.	The watchdog initiates the FK2 error if the cycle time is exceeded by the value set in KW80,00 / %MW3080.0.
KW80,1 [ms] Monitoring of the task with a cycle time = 0	**Infinite loop**	
0	The watchdog initiates the FK2 error after 10 s.	
>0	The watchdog initiates the FK2 error if the cycle time set in KW80,00 / %MW3080.0 is exceeded.	

1.5 Addressing with 07 KT 97 as bus master

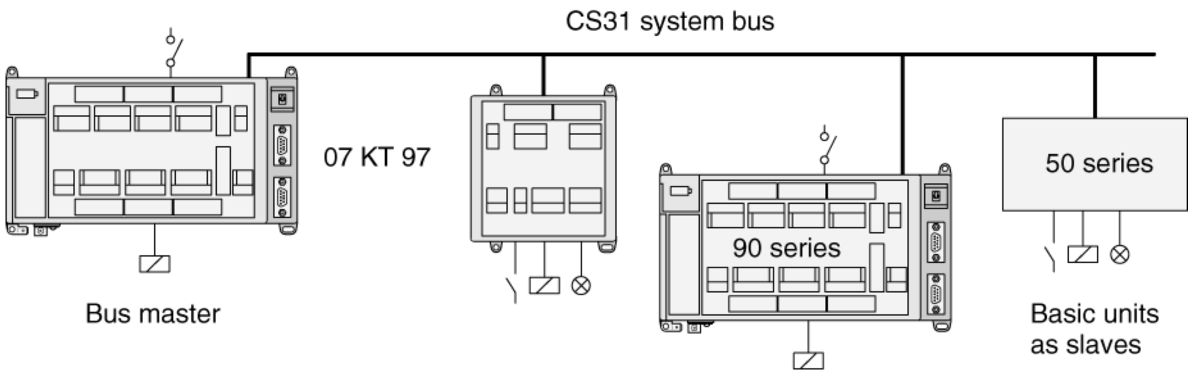
1.5.1 Introduction / Structure examples with 07 KT 97 as bus master



Example 1: 07 KT 97 as standalone PLC



Example 2: 07 KT 97 as bus master on the CS31 system bus; only I/O modules are used as remote modules



Example 3: 07 KT 97 as bus master and as slaves on the CS31 system bus; basic units of the series 50 and 90 as slaves, I/O modules

Regardless of the address ranges, the following modules can be connected to a CS31 system bus:

- max. 1 bus master
- max. 31 remote modules / slaves

Further restrictions result from the address range of the basic unit 07 KT 97:

- max. 28 analog input modules
- max. 28 analog output modules
- max. 31 binary input modules
- max. 31 binary output modules

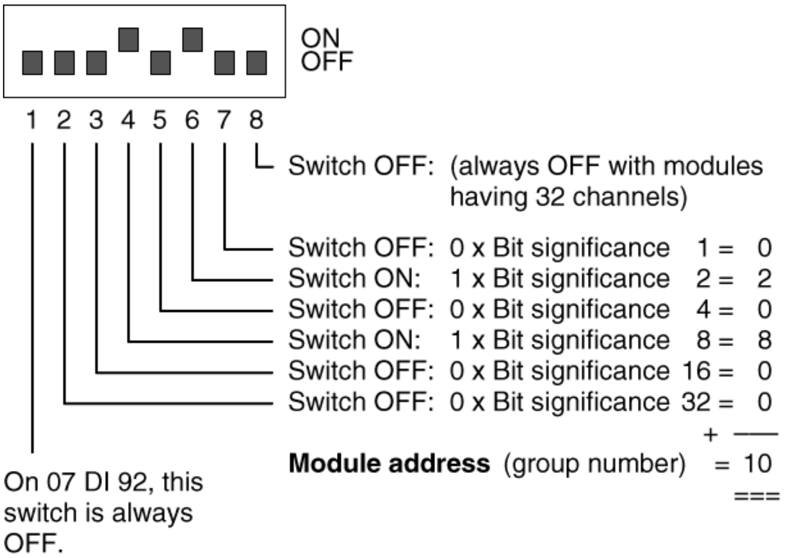
Depending on the structure of the installation and the type of the remote modules used, there may be further restrictions.
Refer to chapter 1.4.2 for recommended addresses.

Structure of the input and output addresses in the remote modules

To enable the basic unit to correctly access the inputs and outputs, an address must be set for every module on the bus.

As an example, the binary input module **07 DI 92** is introduced here.

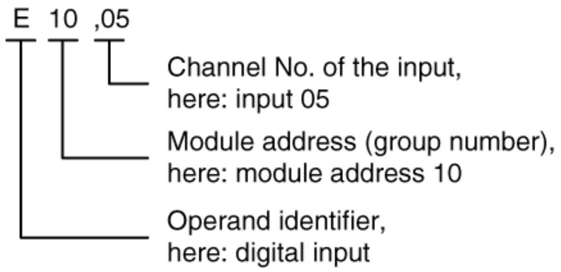
The address setting is done with the DIL switch located under the slide cover on the right hand side of the module housing (meaning of the 8 switches, see below).



Example:

Setting of the module address (group number) on the digital input module 07 DI 92 (variable identifier E)

The bus master basic unit reads the input signals as operands. The hardware designator and the complete address of the input signal have the following structure:



In the programming system 907 AC 1131 the input declaration is performed as follows:

E10_05_switch1 AT %IX10.5 : BOOL; (* Switch 1 *)

Any symbolic designator (here: E10_05_switch1) and comment can be used.

The following table shows the input channel allocations which result from the use of the central unit 07 KT 97 as bus master.

Terminal	Input	Address	Terminal	Input	Address
5	E n,00	%Ixn.00	30	E n+1,00	%Ixn+1.00
6	E n,01	%Ixn.01	31	E n+1,01	%Ixn+1.01
7	E n,02	%Ixn.02	32	E n+1,02	%Ixn+1.02
8	E n,03	%Ixn.03	33	E n+1,03	%Ixn+1.03
9	E n,04	%Ixn.04	34	E n+1,04	%Ixn+1.04
10	E n,05	%Ixn.05	35	E n+1,05	%Ixn+1.05
11	E n,06	%Ixn.06	36	E n+1,06	%Ixn+1.06
12	E n,07	%Ixn.07	37	E n+1,07	%Ixn+1.07
15	E n,08	%Ixn.08	40	E n+1,08	%Ixn+1.08
16	E n,09	%Ixn.09	41	E n+1,09	%Ixn+1.09
17	E n,10	%Ixn.10	42	E n+1,10	%Ixn+1.10
18	E n,11	%Ixn.11	43	E n+1,11	%Ixn+1.11
19	E n,12	%Ixn.12	44	E n+1,12	%Ixn+1.12
20	E n,13	%Ixn.13	45	E n+1,13	%Ixn+1.13
21	E n,14	%Ixn.14	46	E n+1,14	%Ixn+1.14
22	E n,15	%Ixn.15	47	E n+1,15	%Ixn+1.15

07 DI 92: Addresses of the 32 input channels

n: Module address. Can be set with the switches 2...7 on the address DIL switch. Recommended module addresses for the use of 07 KT 97 as bus master: 08, 10, 12.....60 (even-numbered addresses)

The module seizes 2 addresses for inputs on the CS31 system bus.

The switches 1 and 8 of the address DIL switch must be set to OFF.

Note:
Some other modules may have a more or less different address setting. See the following chapters.

1.5.2 Recommended module addresses on the CS 31 system bus

The standard addressing has the purpose of

- simplifying and schematizing the setting of addresses on the CS31 system bus and
- simplifying the diagnosis and troubleshooting.

The standard addressing makes sure that there will be no address overlappings even for modules with a higher amount of data.

Recommendation:

- Assign a specific module address for each module / each slave basic unit. That means the giving up of the possibility of a double assignment of module addresses by binary and analog modules.
- Module addresses 8, 10, ..., 58, 60 for binary remote modules and basic units (only even-numbered addresses). Refer also to chapter 1.4.6 "Intelligent I/O remote modules (basic units) as slave on the CS31 system bus".
- Module addresses 0...5 and 8...15 for analog remote modules.

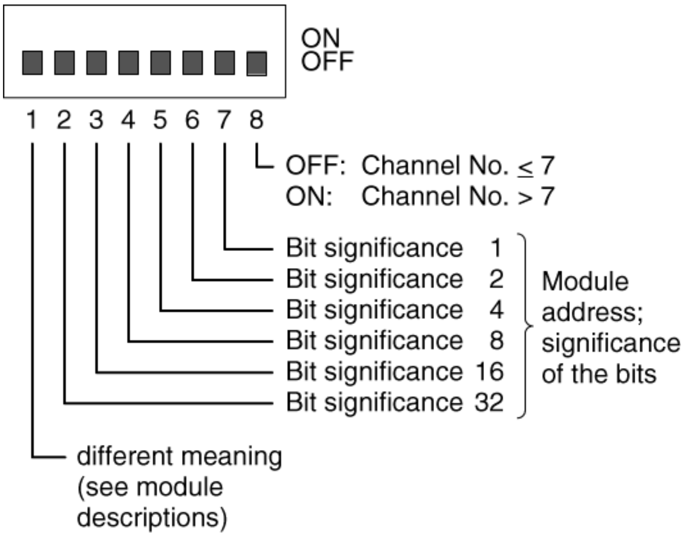
1.5.3 Address settings of the individual modules

Setting of the address switch for binary modules

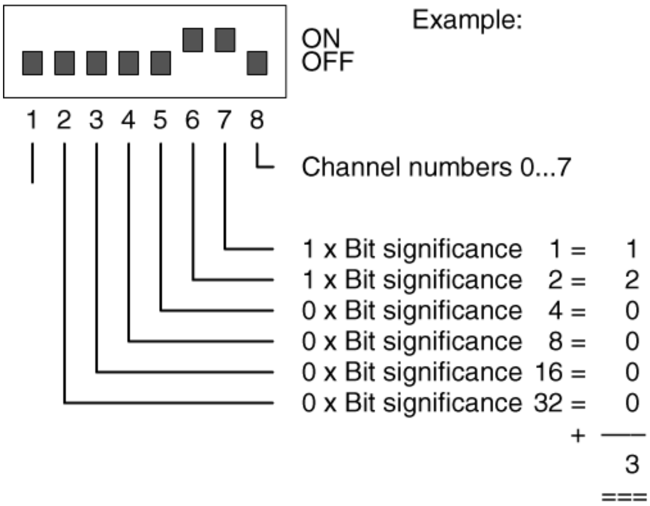
The possible module address range for the basic unit 07 KT 97 is:

0...61

The meaning of the individual switches is as follows:



Example for a binary module: Module address 3



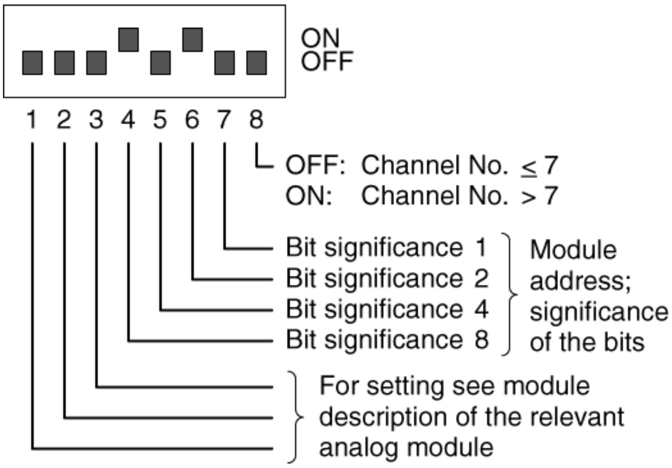
Setting of the address switch for analog modules

When using the basic unit 07 KT 97 as bus master, the possible module address range for analog modules is:

0...5 and 8...15

The analog representation in the 07 KT 97 is fixed. The AC31 analog modules provide appropriate values. For more information concerning the analog representation, refer to the description of the analog modules "General information for the use of analog modules" in chapter "Analog modules" in volume 2.

The meaning of the individual switches is as follows:



1.5.4 07 KT 97 used as standalone basic unit

If the basic unit 07 KT 97 is to be used without the CS31 system bus connected, perform the following setting when programming the user program:

System constant KW 00,00 / %MW3000.0 := -2
KW00_00_Mast_Slav AT %MW3000.0 : INT := -2; (* Standalone *)

This value is set on delivery.

1.5.5 07 KT 97 used as bus master basic unit

If remote modules (slaves) are connected to the basic unit 07 KT 97 via the CS31 system bus, proceed as follows:

1. Change the system constant: KW 00,00 = -1
   KW00_00_Mast_Slav AT %MW3000.0 : INT := -1; (* Master *)
2. Save the PLC program in the Flash EPROM.
3. Activate the new PLC mode by:
   - Selecting the menu item "Online/Reset" (warm start) in the ABB programming and test system.
   - Selecting the menu item "Online/Cold start" in the ABB programming and test system.
   - Voltage "ON".

1.5.6 Intelligent I/O remote modules (basic units) as slave on the CS31 system bus

The basic unit 07 KT 97 can also be used as a slave on the CS31 system bus.

The basic unit 07 KT 97 may be used in the binary as well as in the word area.

The address values can range from 0 to 61 or 100 to 115. The maximum permissible address depends on the size of the set receive and transmit area. The larger you choose the receive or transmit area, the smaller is the maximum permissible address (see examples 1...4).

The slave basic unit may either be used in the binary area or in the word area on the CS31 system bus. If used in the word area the transmit or receive data are located on the channels 0...7 or 8...15, which is selected together with the address setting:

KW 00,00 / %MW3000.0 := 0...5; 8...15 -> Channels 0...7

KW 00,00 / %MW3000.0 := 100...105; 108...115 -> Channels 8...15

The upper channel range of 8...15 is configured by adding the value of 100 to the address.

In order to switch over to the "slave mode", proceed as follows:

1. Change the system constant: KW 00,00 / %MW3000.0 := 0...61
   KW00_00_Mast_Slav AT %MW3000.0 : INT := 0...61; (* Slave *)
2. Save the user program in the Flash EPROM.
3. Activate the new PLC mode by:
   - Selecting the menu item "Online/Reset" (warm start) in the ABB programming and test system.
   - Selecting the menu item "Online/Cold start" in the ABB programming and test system.
   - Voltage "ON".

There is no direct access to the inputs and outputs of the slave basic unit via the CS31 system bus. The communication between master and slave is performed using input and output operands.

All the master data are consistently transferred to the slave and all the slave data are consistently transferred to the master.

The slave basic unit may either be used in the binary area or in the word area on the CS31 system. Using the two system constants KW 00,10 / %MW3000.10 and KW 00,11 / %MW3000.11 the transmit and the receive area of the slave can be adapted to the application specific requirements (see also chapter "System constants").

You can set:

- the size of the transmit and receive area and
- the mode of operation of the slave (in the binary or the word area).

Default condition:

If the basic units 07 KT 97 are switched over to the "slave mode", they behave on the CS31 system bus like input and output modules with 32 inputs and 32 outputs.

This means that the default setting of the transmit and receive area is within the binary area of the master and their size is 32 bits (4 bytes) each.

Example 1:

Default configuration of the slave (binary area):

KW00_10_SLV_SEND AT %MW3000.10 : INT := 0; (* 4 bytes – data transmission *)
KW 00,10 = 0: Slave transmit area: 4 bytes (4 bytes * 8 channels = 32 digital outputs (A))

KW00_11_SLV_REC AT %MW3000.11 : INT := 0; (* 4 bytes – data transmission *)
KW 00,11 = 0: Slave receive area: 4 bytes (4 bytes * 8 channels = 32 digital inputs (E))

Note:
The default configuration is the same as the configuration
KW 00,10 / %MW3000.10 = KW 00,11 / %MW3000.11 := 4

07 KT 97 as bus master	⟷	07 KT 97 as slave with: KW 00,10 / %MW3000.10 := 0 or 4 KW 00,11 / %MW3000.11 := 0 or 4
Receiving or transmitting with E/A operands (I/O operands)		Transmitting or receiving with E/A operands (I/O operands)
E n ,00 / %IXn .0	⟵	A 00,00 / %QX0.0
:		:
E n ,15 / %IXn .15		A 00,15 / %QX0.15
E n+1,00 / %IXn+1.15		A 01,00 / %QX1.0
:		:
E n+1,15 / %IXn+1.15	⟵	A 01,15 / %QX1.15
A n ,00 / %QXn .0	⟶	E 00,00 / %IX0.0
:		:
A n ,15 / %QXn .15		E 00,15 / %IX0.15
A n+1,00 / %QXn+1.0		E 01,00 / %IX1.0
:		:
A n+1,15 / %QXn+1.15	⟶	E 01,15 / %IX1.15

n: Module address of the slave basic unit. For this example: $0 \le n \le 60$

For the slave address of n = 12 the following applies for example:
The output signal A 00,00 / %QX0.0 of the 07 KT 97 used as slave is the input signal E 12,00 / %IX12.0 for the 07 KT 97 used as bus master.

Example 2:

Configuration of the slave for the binary area:

KW00_10_SLV_SEND AT %MW3000.10 : INT := 15; (* 15 bytes – data transmission *)
KW 00,10 = 15: Slave transmit area: 15 bytes (15 bytes * 8 channels = 120 digital outputs (A))

KW00_11_SLV_REC AT %MW3000.11 : INT := 6; (* 6 bytes – data transmission *)
KW 00,11 = 6: Slave receive area: 6 bytes (6 bytes * 8 channels = 48 digital inputs (E))

07 KT 97 as bus master Receiving or transmitting with E/A operands (I/O operands)	⟵⟶	07 KT 97 as slave with: KW 00,10 / %MW3000.10 := 15 KW 00,11 / %MW3000.11 := 6 Transmitting or receiving with E/A operands (I/O operands)
E n ,00 / %IXn .0	⟵	A 00,00 / %QX0.0
:		:
E n ,15 / %IXn .15		A 00,15 / %QX0.15
:		:
E n+7,00 / %IXn+7.15		A 07,00 / %QX7.0
:		:
E n+7,07 / %IXn+7.7	⟵	A 07,07 / %QX7.7
A n ,00 / %QXn .0	⟶	E 00,00 / %IX0.0
:		:
A n ,15 / %QXn .15		E 00,15 / %IX0.15
:		:
A n+2,00 / %QXn+2.0		E 02,00 / %IX2.0
:		:
A n+2,15 / %QXn+2.15	⟶	E 02,15 / %IX2.15

Notes:

The upper 8 input channels of the address n+7

E n+7,08 / %IXn+7.8...E n+7,15 / %IXn+7.15

can be seized on the CS31 system bus with an other binary 8-bit input module (excluding KR/KT).

The output channels beginning from the address n+3

A n+3,00 / %QXn+3.00...A n+7,15 / %QXn+7.15

can be seized on the CS31 system bus with other binary output modules (including KR/KT).

n: Module address of the slave basic unit. For this example: $0 \leq n \leq 54$

For the slave address of n = 12 the following applies for example:
The output signal A 00,00 / %QX0.0 of the 07 KT 97 used as slave is the input signal E 12,00 / %IX12.0 for the 07 KT 97 used as bus master.

Example 3:

Configuration of the slave for the word area:

KW00_10_SLV_SEND AT %MW3000.10 : INT := 101; (* 1 word – data transmission *)
KW 00,10 = 101: Slave transmit area: 1 word (1 word = 1 word output)

KW00_11_SLV_REC AT %MW3000.11 : INT := 108; (* 8 words – data transmission *)
KW 00,11 = 108: Slave receive area: 8 words (8 words = 8 word inputs)

07 KT 97 as bus master Receiving or transmitting with EW/AW operands	⟷	07 KT 97 as slave with: KW 00,10 / %MW3000.10 := 101 KW 00,11 / %MW3000.11 := 108 Transmitting or receiving with EW/AW operands
EW n,00 / %IW1000+n.0	⟵	AW 00,00 / %QW1000.0
AW n,00 / %QW1000+n.0 : AW n,07 / %QW1000+n.7	⟶	EW 00,00 / %IW1000.0 : EW 00,07 / %IW1000.7

Note:
The upper 8 channels of the address n

EW n,08 / %IW1000+n.8...EW n,15 / %IW1000+n.15 and
AW n,08 / %QW1000+n.8...AW n,15 / %QW1000+n.15

can be seized on the CS31 system bus with another module, for instance.

n: Module address of the slave basic unit. For this example: $0 \le n \le 5$ and $8 \le n \le 15$

For the slave address of n = 4 the following applies for example:
The output signal AW 00,00 / %QW1000.0 of the 07 KT 97 used as slave is the input signal EW 04,00 / %IW1004.0 for the 07 KT 97 used as bus master.

Example 4:

Configuration of the slave for the word area:

KW00_10_SLV_SEND AT %MW3000.10 : INT := 101; (* 1 word – data transmission *)
KW 00,10 = 101: Slave transmit area: 1 word (1 word = 1 word output)

KW00_11_SLV_REC AT %MW3000.11 : INT := 108; (* 8 words – data transmission *)
KW 00,11 = 108: Slave receive area: 8 words (8 words = 8 word inputs)

07 KT 97 as bus master Receiving or transmitting with EW/AW operands	⟵⟶	07 KT 97 as slave with: KW 00,10 / %MW3000.10 := 101 KW 00,11 / %MW3000.11 := 108 Transmitting or receiving with EW/AW operands
EW n,08 / %IW1000+n.8	⟵	AW 00,00 / %QW1000.0
AW n,08 / %QW1000+n.8	⟶	EW 00,00 / %IW1000.0
:		:
AW n,15 / %QW1000+n.15		EW 00,07 / %IW1000.7

Note:
The lower 8 channels of the address n

EW n,00 / %IW1000+n.0...EW n,07 / %IW1000+n.7 and
AW n,00 / %QW1000+n.0...AW n,07 / %QW1000+n.7

can be seized on the CS31 system bus with another module, for instance.

n: Module address of the slave basic unit. For this example: $100 \leq n \leq 105$ and $108 \leq n \leq 115$

For the slave address of n = 104 the following applies for example:
The output signal AW 00,00 / %QW1000.0 of the 07 KT 97 used as slave is the input signal EW 04,08 / %IW1004.8 for the 07 KT 97 used as bus master.

1.5.7 Special modules used as slave on the CS 31 system bus

Festo valve island / installation island

The Festo valve island and the Festo installation island behave on the CS31 system bus like binary input and output modules. Refer to the Festo documentation for the scope of seized data.

1.5.8 Complex structure examples including addresses

Categorization of the modules with respect to the I/O terminals

There are two main module types:

- **Binary modules:** These modules are controlled by means of binary I/O operands (E or A). The basic units 07 KT 97 also belong to this type, if they are used as slaves.
- **Analog modules:** These modules are controlled by means of word I/O operands (EW or AW). The basic unit 07 KT 97 also belong to this type as well as the high-speed counter ICSF 08 D1 which receives its preset data as word data, for example.

The following table shows an overview of the module types. These designations will be used in example 6.

Please note, that the configurable binary modules 07 DC 91 and 07 DC 92 behave differently according to the performed setting.

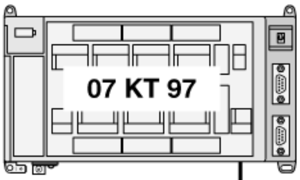
1.5.9 Module examples (slaves on the CS 31 system bus)

Module types, with regard to the I/O terminals	Module examples
Binary output modules with 32 outputs	07 DC 92 if switch 1 is set to ON
Binary modules with 16 inputs and 16 outputs	07 DC 91, if switch 1 is set to ON
Binary modules with 32 inputs and 32 outputs	07 DC 92 if switch 1 is set to OFF
Binary modules with 120 signals from and 120 signals to the CS31 system bus	07 KT 9x, 07 KR 51, 07 KT 51 as slave
Analog input modules with 8 inputs	07 AI 91
Analog output modules with 8 outputs	07 AC 91 configured as output module
Analog modules (word modules) with up to 8 inputs and 8 outputs	07 KT 9x, 07 KR 51, 07 KT 51, 07 AC 91

Examples for the assignment of module addresses

Example 5:

Bus master basic unit



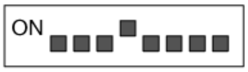
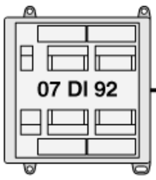
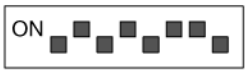
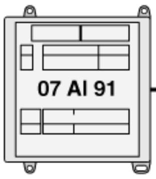
CS31 system bus

07 KT 97

E 62,00 / %IX62.0
:
E 62,15 / %IX62.15
E 63,00 / %IX63.0
:
E 63,15 / %IX63.15
A 62,00 / %QX62.0
:
A 62,15 / %QX62.15
A 63,00 / %QX63.0
:
A 63,07 / %QX63.7
EW 06,00 / %IW1006.0
:
EW 06,07 / %IW1006.7
AW 06,00 / %QW1006.0
:
AW 06,03 / %QW1006.3

Addresses of the inputs and outputs on the bus master basic units

Remote modules (slaves)

Address switch (DIL switch) on the module	Remote module	Address in the program of the master CPU	
 1 2 3 4 5 6 7 8 08	 32 digital inputs	E 08,00 / %IX8.0 : E 08,15 / %IX8.15 E 09,00 / %IX9.0 : E 09,15 / %IX9.15	1st digital input : 32nd digital input
 1 2 3 4 5 6 7 8 11 ≤ 7	 8 analog inputs	EW 11,00 / %IW1011.0 EW 11,01 / %IW1011.1 : EW 11,07 / %IW1011.7	1st analog input 2nd analog input : 8th analog input

Example 6:

See next page

07 KT 97

Address setting in the master:
KW 00,00 / %MW3000.0 = –1

E 62,00...E 62,15
E 63,00...E 63,15
A 62,00...A 62,15
A 63,00...A 63,07
EW 6,00...EW 6,07
AW 6,00...AW 6,03

Inputs / outputs on the basic unit

Address switch on the remote module	Remote module	Address in the program of the master CPU	Remarks
			– max. 31 slaves on the CS31 bus
ON 1 2 3 4 5 6 7 8 03 ≤ 7	07 AI 91 8 inputs analog E0 : E7	EW 03,00 %IW1003.0 : EW 03,07 %IW1003.7	– Permissible range of module addresses for analog modules: 0...5 and 8...15
ON 1 2 3 4 5 6 7 8 03 > 7	07 AI 91 8 inputs analog E0 : E7	EW 03,08 %IW1003.8 : EW 03,15 %IW1003.15	– 2 analog input modules with 8 channels each can be assigned to one address (16 channels together)
ON 1 2 3 4 5 6 7 8 09 ≤ 7	07 AC 91 8 outputs analog A0 : A7 8 inputs analog E0 : E7	AW 09,00 %QW1009.0 : AW 09,07 %QW1009.7 EW 09,00 %IW1009.0 : EW 09,07 %IW1009.7	– The same address (as for the analog input modules) may also be used for the connection of analog output modules (as shown to the left).
ON 1 2 3 4 5 6 7 8 12 ≤ 7	07 DI 92 32 inputs digital E00 : E31	E 12,00 %IX12.0 : E 12,15 %IX12.15 E 13,00 %IX13.0 : E 13,15 %IX13.15	– Permissible range of module addresses: for digital modules: 0...61, recommendation: 6...60 – The following might be done, but does not bring you any advantage: • using the same addresses for digital modules as for analog modules • collecting input and output modules under one address
07 KT 97: KW 0,0 / %MW3000.0 = 4 up to 120 digital inputs and up to 120 digital outputs or up to 8 word inputs and up to 8 word outputs		KR/KT in the digital range: E 04,00...E 11,07 / %IX4.0...%IX11.7 A 04,00...A 11,07 / %QX4.0...%QX11.7 or KR/KT in the word range: EW 04,00...EW 04,07 / %IW1004.0...%IW1004.7 AW 04,00...AW 04,07 / %QW1004.0...%QW1004.7	– Slave-KR/KT with 120 E and/or 120 A occupies the set **and** the following 7 addresses (only half of the 7th, though). For address 4 of the example: • Next free address for KT: Bit range: 12, word range: 5 • Max. settable KT address: Bit range: 54, word range: 5

1.6 I/O configuration

1.6.1 Purpose of the I/O configuration for I/O modules

Depending on the type of I/O module the following can be configured:

- For binary modules with combined I/O channels, these channels can be configured as "only inputs" or "only outputs".
- For analog modules, measuring ranges or output areas can be configured which are different from the factory setting.

Switching over of inputs and outputs, switching on the diagnosis functions and changing the measuring ranges and output areas is performed as follows, depending on the module type:

- Performing the I/O configuration via the CS31 system bus by means of the user program of the bus master basic unit.
- Setting of switches on the remote module.
- External wiring of the input/output module.

In some cases there is a relation between the settings made on the remote module and the information and diagnosis messages which can be interrogated at the remote module via the CS31 system bus. This relation is explained in the following chapters.

Do not perform an I/O configuration via the CS31 system bus if the factory setting is sufficient. Once an I/O configuration has been performed, it will remain stored in the corresponding I/O module until it is changed again by means of an I/O configuration process. Even in case of power OFF it will not be deleted.

1.6.2 Performing and reading the I/O configuration

There are the following possibilities for system structures with 07 KT 97 as bus master:

- Performing and reading the I/O configuration via the user program of the bus master basic unit 07 KT 97.
- Reading the I/O configuration from the remote modules.

Performing and reading the I/O configuration via the user program

The function block CS31CO is available for the I/O configuration of the modules. This function block is part of the programming software 907 AC 1131 and is described in the corresponding documentation.

Reading of I/O configuration and diagnosis data at the remote module

Reading the I/O configuration and the diagnosis information for an I/O terminal of a module will be shown in the following for the module ICSC 08 L1 as an example. The procedure is the same for all remote modules, only the type and the amount of diagnosis information are different. To read the information, the test button (4) and the LED displays (1) of the module have to be used.

When the test button is pressed for the first time the channel E/A0 (input/output 0) is selected: LED0 flashes. After releasing the button, the LEDs 0 to 7 display the diagnosis information of this channel for approx. 3 seconds.

The LEDs have the following meaning:

- 0 UE = Unit error
- 1 BE = Bus error
- 2 not used
- 3 CI/CO = Cut wire of inputs/outputs
- 4 OL = Overload
- 5 SC = Short-circuit
- 6 Configuration as output
- 7 Configuration as input. If the LEDs 6 and 7 light up simultaneously, the channel is configured as a combined input/output.

The meaning of the LEDs (2) is also printed in English on the front panel of the module.

The operation is repeated for the other channels each time the test button is pressed and released again.

After the last channel E/A7 (input/output 7) has been scanned, pressing the test button again causes a lamp test (LED test) to be performed. All 8 LEDs should light up. After releasing the button, the LEDs display the settings of the DIL switch on the plug-in base for approx. 5 seconds. LED 0 displays the setting of switch 1 (LEDs 0...7 are assigned to the switches 1...8).

All error messages are stored in the module and can only be deleted by pressing the test button for 10 seconds or by switching the power OFF and ON.

1.7 Diagnosis

1.7.1 Introduction

The diagnosis system of the 07 KT 97 is designed to ensure a quick and efficient troubleshooting. For this purpose the diagnosis system is organized as follows:

- "Vertically" in diagnosis, error flags, reactions, LED displays and acknowledgement. There are interrelations between the bus master basic unit and the remote modules: The basic unit reads the diagnosis data detected by the remote modules. An acknowledgement in the basic unit causes also the deletion of the error messages stored in the remote modules.
- "Horizontally" in 4 error classes corresponding to the severity of the errors.

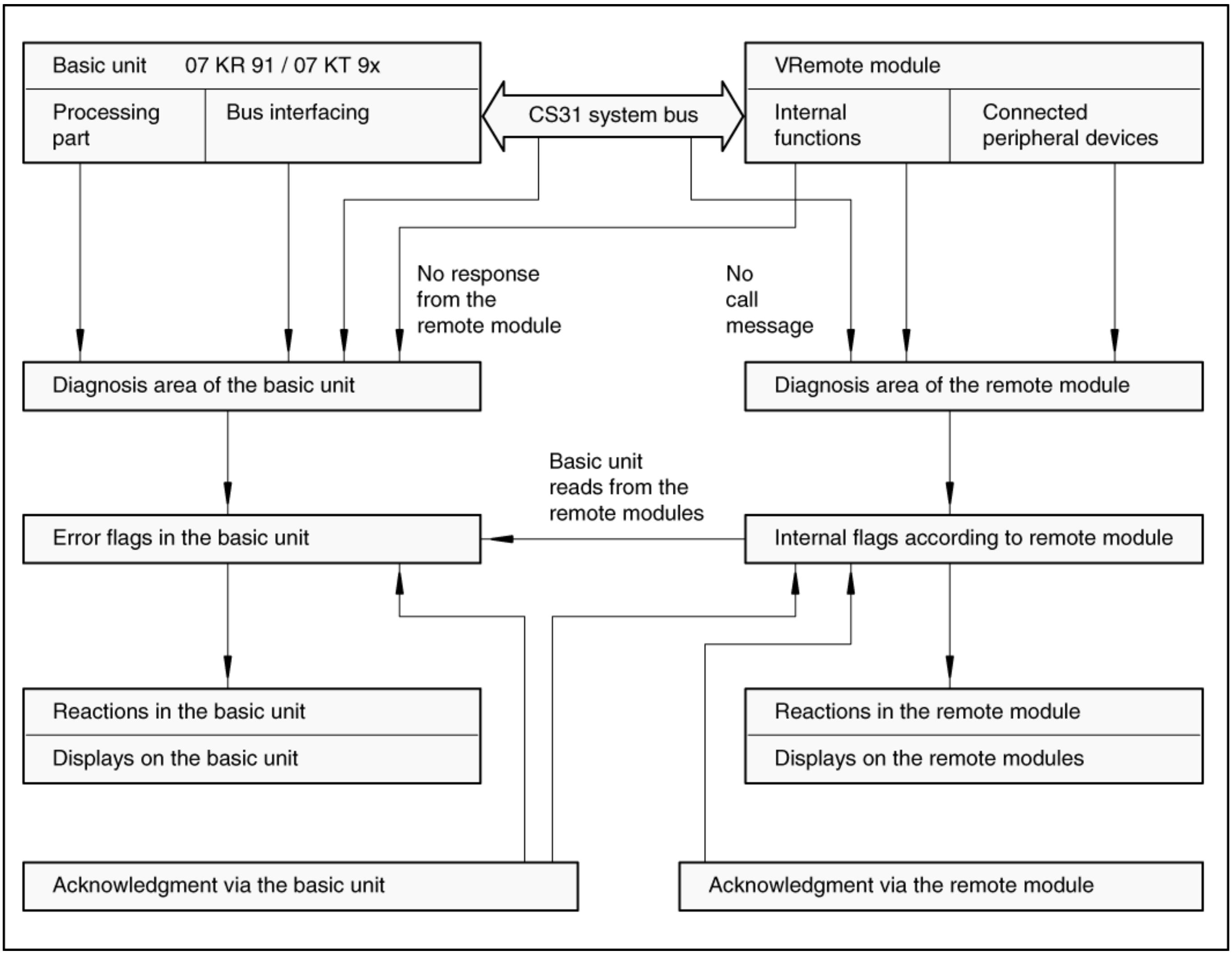
This concept is based on a system structure consisting of a bus master basic unit and several remote modules, and remote processors as well. The diagnosis system detects the follwoing errors:

- Errors in the bus master basic unit
- Errors on the CS31 system bus
- Errors in the remote modules
- Errors in the wiring of the remote modules on the process side

The troubleshooting is performed as follows:

- The LEDs on the basic unit 07 KT 97 give first hints. The errors detected by the remote modules are also displyed here.
- If these hints are not sufficient, the error flags have to be read out.
- The status register EW 07,15 / %IW1007.15 in the basic unit supplies additional information to be used for the diagnosis.
- The remote modules indicate errors occurring in their area. Detailed information can be obtained by pressing the test button on the modules.

1.7.2 Structure of the diagnosis



1.7.3 Troubleshooting by means of the LED displays on the basic unit

The LED displays on the front panel of the basic unit supply initial information on the errors which occurred:

- BA = CS31 bus processor active
- BE = Bus Error (error on the CS31 system bus)
- RE = Remote Unit Error (error in/on a remote unit)
- SE = Serial Unit Error (error in the CS31 bus interfacing of the basic unit)
- RUN = User program is running (no error)
- FK1 = Error class 1 (fatal error)
- FK2 = Error class 2 (serious error)
- FK3 = Error class 3 (light error)
- Supply = Supply voltage available
- Battery = Battery available / User program is being flashed
- Ovl = Overload or short-circuit on at least one direct binary output of the basic unit

If no LED lights up, the basic unit has not found any error. Exception: The LED Battery (battery is missing); the battery is only necessary for certain cases of application.
The LED Battery flashes during the process of writing the user program into the Flash memory.

LEDs for CS31 system bus and bus interfacing

LED / Basic unit	BA BE RE SE gn rd rd rd	Meaning	Remedy
A M S	✷ ○ ○ ○	Everything ok.	- -
A M S	○ ⊙ ⊙ ⊙	A fatal error occurred. The watchdog has switched off the CS31 bus. All outputs are OFF (0 state).	• Power OFF/ON. If unsuccessful the module is defective. • Evaluate the error flags.
A M S	✷ ○ ○ ✷	Dual-port RAM defective.	• Power OFF/ON. If unsuccessful the module is defective. • Evaluate the error flags.
A M S	○ ✷ ✷ ✷	Initialization phase after power ON or after cold start.	- -
M	✷ ✷ ○ ○	Master basic unit does not find any remote modules on the CS31 bus <u>after</u> power ON or after cold start.	• Install remote modules. • Check the CS31 bus line. • Check the supply voltage of the remote modules. • Evaluate the error flags.
M	✷ ○ ✷ ○	Error message from a remote module.	• Evaluate the error flags. • Check the remote modules.
M	✷ ✷ ✷ ○	One remote module can suddenly not be controlled by the master basic unit any more.	• Evaluate the error flags. • Check the supply voltage of the remote modules. • Check the CS31 bus line. • Check the remote module.
M	✷ ✷ ○ ○	There are at least 3 remote modules on the CS31 bus. Two of the remote modules can suddenly not be controlled by the master basic unit any more.	• Evaluate the error flags. • Check the supply voltage of the remote modules. • Check the CS31 bus line. • Check the remote module.
M	✷ ✷ ✷ ✷	There are at least 2 remote modules on the CS31 bus. Suddenly no remote module can be controlled by the master basic unit any more.	• Evaluate the error flags. • Check the supply voltage of the remote modules. • Check the CS31 bus line. • Check the remote module.
S	✷ ○ ⌘ ○	CS31 bus does not work.	• Check the CS31 bus line. • Check the master basic unit.

○ = LED off, ✷ = LED on, ⌘ = LED flashes, ⊙ LED on or off, gn = green, rd = red,
A = standalone, master or slave basic unit, M = master basic unit, S = slave basic unit

LEDs for user program and error display

LED / Basic unit	R U N Gn / F K 1 rd / F K 2 rd / F K 3 rd	Meaning	Remedy
A M S	✷ O O O	User program is running.	- -
A M S	✷ O O ✷	User program is running, but a light error occurred.	• Evaluate the error flags and eliminate the error.
A M S	O O O O	User program does not run.	• Start the user program.
A M S	O O O ✷	A light error occurred which caused the user program to be aborted automatically because the system constant KW00,07 is not equal to "0".	• Evaluate the error flags and eliminate the error.
A M S	O O ✷ O	A serious error occurred which caused the user program to be aborted automatically.	• Evaluate the error flags and eliminate the error, if possible.
A M S	O ✷ O O	A fatal error occurred. The user program cannot be started.	• Evaluate the error flags. • Power OFF/ON. If unsuccessful, the module is defective.
A M S	O O ✷ ✷	A light and a serious error occurred.	• Evaluate the error flags and eliminate the error, if possible.
A M S	⊙ ⌘ ⊙ ⊙	Power-fail.	• Power OFF/ON.
A M S	✷ ✷ ✷ ✷	Initialization phase, power ON, cold start, warm start.	- -
O = LED off, ✷ = LED on, ⌘ = LED flashes, ⊙ LED on or off, gn = green, rd = red, A = standalone, master or slave basic unit, M = master basic unit, S = slave basic unit			

LEDs for supply voltage and battery

LED / Basic unit	Supply gn	Battery rd	Meaning	Remedy
A M S	✷	O	Supply voltage available **and** battery is effective.	- -
A M S	✷	✷	Supply voltage available and battery is **not** effective.	- -
A M S	✷	⌘	Supply voltage available, user program is written into the Flash memory.	- -
A M S	O	O	Supply voltage is not available.	• Switch power ON. • Check the supply voltage.
O = LED off, ✷ = LED on, ⌘ = LED flashes, ⊙ LED on or off, gn = green, rd = red, A = standalone, master or slave basic unit, M = master basic unit, S = slave basic unit				

LEDs for overload or short-circuit on at least one direct binary output

LED / Basic unit	Ovl rd	Meaning	Remedy
A M S	✷	Overload or short-circuit on at least one of the direct binary outputs.	• Eliminate the overload or short-circuit.
A M S	O	No overload or short-circuit.	- -
O = LED off, ✷ = LED on, ⌘ = LED flashes, ⊙ LED on or off, gn = green, rd = red, A = standalone, master or slave basic unit, M = master basic unit, S = slave basic unit			

1.7.4 Troubleshooting on the remote modules

Diagnosis functions for the remote modules

The remote modules are equipped with a number of diagnosis functions. Some of these diagnosis functions become active only if they have been set by means of the I/O configuration.

Diagnosis, displays and messages to the basic unit						
Diagnosis function depending on the module a) always available b) if configured	 UE	 BE	 OL	 SC	 CI	 CO
Can be displayed on the module by pressing the test button [1]) a) Diagnosis messages	UE	BE	OL	SC	CI	CO
Available for bus master user program a) cyclic transmission	-	-	OL	SC	CI	CO
Module type						
ICSI 08 D1	●	●			●	
ICSI 16 D1	●	●			●	
ICSI 08 E1	●	●				
ICSI 16 E1	●	●				
ICSI 08 E4	●	●				
07 DI 92	●	●				
	●	●				
ICSO 08 R1	●	●				
ICSO 08 Y1	●	●	●	●		
ICSO 16 N1	●	●	●	●		
ICSK 20 F1	●	●				
ICSK 20 N1	●	●	●	●		
ICSC 08 L1	●	●	●	●	●	●
ICSC 16 L1	●	●	●	●	●	●
ICSF 08 D1	●	●				
ICDG 32 L1, 07 DC 91	●	●	●	●		
07 DC 92	●	●	●	●		
ICSE 08 A6	●	●				
ICSE 08 B5	●	●				
ICSA 08 B5						
ICSM 06 A6						
07 AI 91	●	●				
07 AC 91	●	●				
07 KR 31	●	●				
07 KT 31	●	●	●	●		
07 KR 91	●	●				
07 KT 92 / 07 KT 93 / 07 KT 94	●	●	●	●		
07 KT 95 / 07 KT 96 / 07 KT 97	●	●	●	●		

Explanations:

- ● Feature is completely available. Settings and errors can be interrogated from the involved channel.
- **(●)** Feature is partly available (restricted), see module description.
- **[1])** Refer to the module descriptions and the chapter "Reading I/O configuration and diagnosis data at the module" for information concerning the interrogation of settings and diagnosis data.

BE Bus error = Bus malfunction, always monitored. The module does not receive a call from the bus master.
This may have the following reasons:

- The CS31 system bus line is interrupted, short-circuited or wired with reversed polarity.
- The basic unit is no more set as bus master, see system constant KW 00,00 / %MW3000.0.

CI Cut Wire of Inputs = Open circuit (monitoring) at inputs, if configured. Each input circuit to be monitored has to be equipped with a resistor of 20...30 kΩ, e.g. in parallel to the signalling contact.

CO Cut Wire of Outputs = Open circuit (monitoring) at outputs, if configured. Each output expects a minimum load of approx. 40 mA when an ON signal is output.

OL Overload, is always monitored.

SC Short-circuit, is always monitored.

UE Unit Error = Internal error of the module (is always monitored as far as the internal processor can detect this).

Troubleshooting in the remote modules

If the remote module has detected an error, the LED (3) lights up.

The remote module supplies detailed error information via the 8 LEDs (1) if the test button is pressed; see also the module descriptions.

The procedure will be explained in the following for the module ICSC 08 L1 as an example.

When the test button is pressed for the first time the channel E/A0 (input/output 0) is selected: LED0 flashes. After releasing the button, the LEDs 0 to 7 display the diagnosis information of this channel for approx. 3 seconds.

The LEDs have the following meaning:

0 UE = Unit error
1 BE = Bus error
2 Not used
3 CI/CO = Cut wire of inputs/outputs
4 OL = Overload
5 SC = Short-circuit
6 Configuration as output
7 Configuration as input
If the LEDs 6 and 7 light up simultaneously, the channel is configured as combined input/output.

The meaning of the LEDs (2) is also printed in English on the front panel of the module.
The operation is repeated for the other channels each time the test button is pressed and released again.

After the last channel E/A7 (input/output 7) has been scanned, pressing the test button again causes a lamp test (LED test) to be performed. All 8 LEDs should light up. After releasing the button, the LEDs display the settings of the DIL switch on the plug-in base for approx. 5 seconds. LED 0 displays the setting of switch 1 (LEDs 0...7 are assigned to the switches 1...8).

All error messages are stored in the module and can only be deleted by pressing the test button for 10 seconds or by switching the power OFF and ON.

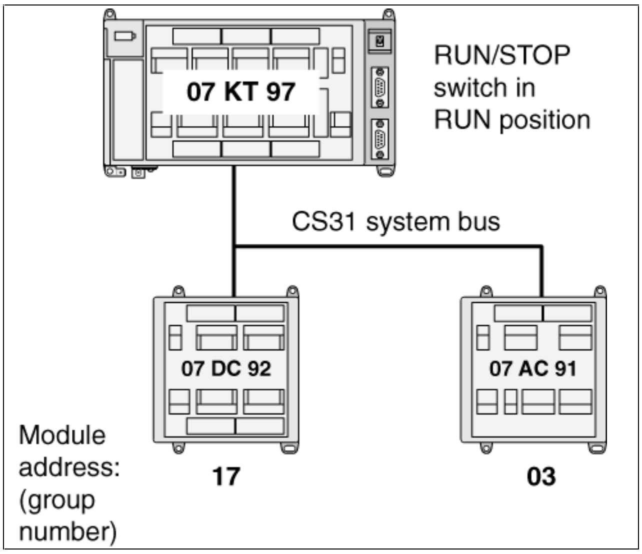
1.7.5 Acknowledgement of error messages in the remote modules

The remote modules store and display the error messages occurring there, independently of the basic unit. The error messages can be acknowledged:

- on the remote module by pressing the test button
- in the user program by means of the function block CS31QU (this also deletes the error message stored in the basic unit)

If the error has not been eliminated, the error message appears again.

1.7.6 Example of an error message



Errors which occurred:

The bus line to the remote module with the module address 3 has been broken during operation.

Error flags in the basic unit 07 KT 97:
It is assumed that the error flags have been set to 0 by acknowledgement/deletion before the error occurred. In the following only those error flags are listed the contents of which changes.

- M 255,10 / %MX255.10 = TRUE   Sum error message
- M 255,13 / %MX255.13 = TRUE   Error class message (FK3 error)
- MW 255,00 / %MW1255.0 = 15   Error identification: remote module is disconnected
- MW 255,01 / %MW1255.1 = 05   Module type: analog input and output
- MW 255,02 / %MW1255.2 = 03   Group number (module address)
- MW 255,03 / %MW1255.3 = 0
- MW 255,04 / %MW1255.4 = 0
- MW 255,05 / %MW1255.5 = 0
- MW 255,06 / %MW1255.6 = 0
- MW 255,07 / %MW1255.7 = 0

} not concerned, as well as all other error flags which have not been mentioned

LED displays on the bus master basic unit 07 KT 97:

- BA lights up ⇒ CS31 bus processor is active. The data communication with the remote module having the address 17 is continued.
- BE lights up ⇒ Bus Error, error on the CS31 system bus.
- RE lights up ⇒ Remote Unit Error, error on a remote module.
- SE off ⇒ Serial Unit Error, the CS31 bus interfacing in the basic unit works correctly
- RUN lights up
- FK3 lights up ⇒ light error

Reaction of the bus master basic unit 07 KT 97

The processing program and the bus operation continue running
(if KW 00,07 / %MW3000.7 := 0).
Reaction of the remote module 07 AC 91: All outputs turn to OFF (0 state).
Reaction of the remote module 07 DC 92: Data exchange with the bus master basic unit 07 KT 97.

Status word EW 07,15 in the basic unit 07 KT 97

- Bit 0	= 1	No class 2 error
- Bit 1	= 0	Only applicable for 07 KT 97 as slave
- Bit 2	= 1	Date and time of the real-time clock are valid
- Bit 3	= 1	Battery available
- Bit 4...7		Not used
- Bit 8...15	= x	Max. number of modules connected to the CS31 system bus which have been found since the last power-on operation. The number will not be changed by the error which has occurred in the meantime.

Acknowledgement of the error flags in the basic unit 07 KT 97

Eliminate the error before acknowledgement. Otherwise the error message will appear again.

The bit flags M 255,10 / %MX255.10 and M 255,13 / %MX255.13 can be acknowledged by the following:

- Power ON.
- Cold start (menu item in 907 AC 1131).
- Setting the RUN/STOP-switch to RUN.
- Overwriting the flag M 255,13 / %MX255.13 with "FALSE" (0) in the user program.
- Using the online function "Overwrite" in order to overwrite the flag M 255,13 / %MX255.13 with "FALSE" (0).
- Using the function block CS31QU in the user program. (The function block CS31QU is applicable only for errors which concern the CS31 system bus. It also deletes the error message in the remote module.)

The relevant LEDs turn off upon the acknowledgement.

The word flags MW 255,00...MW 255,07 / %MW1255.0...%MW1255.7 can only be deleted by overwriting them. They are overwritten by newly occurring errors.

Acknowledgement of the error flags in the remote module

- On the remote module by pressing the test button for a longer time.
- In the user program of the basic unit using the block CS31QU.

1.7.7 Error flags in the basic unit, error classification

The basic unit offers error messages for the user program which are classified into 4 error classes (FK1...FK4) according to their severity. The error messages are stored in error flags and can be used in the user program and be read by the programming system.

The following table gives an overview of the error flags:

Error class	**FK1 = Fatal error**	**FK2 = Serious error**	**FK3 = Light error**	**FK4 = Warning**
General feature of the error class, examples	Save operation of the operating system is no longer ensured. Error examples: - Checksum error in the operating system EPROM - Write-/read error while testing the operating system RAM	The operating system works correctly, but the error-free processing of the user program is not ensured. Error examples: - Checksum error in the operating system (Flash EPROM) - Write-/read error while testing the user program RAM	The choice whether the user program has to be aborted by the operating system or not depends on the application. The user decides which reactions are to be initiated. Error examples: - The Flash EPROM cannot be programmed - The dual-port RAM to the CS31 part (LED SE) is faulty - The remote module has failed (LED RE)	Errors which occur on peripheral devices or which will have an effect only in the future The user decides which reactions are to be initiated. Error examples: - Short-circuit on a remote module
Sum error message [1])	M 255,10 / %MX255.10			
Error class message (if 1, an error exists)	M 255,11 / %MX255.11	M 255,12 / %MX255.12	M 255,13 / %MX255.13	M 255,14 / %MX255.14
Error identification (word) [2]) Detailed info 1 (word) [2]) Detailed info 2 (word) [2]) Detailed info 3 (word) [2]) Detailed info 4 (word) [2]) Detailed info 5 (word) [2]) Detailed info 6 (word) [2]) Detailed info 7 (word) [2])	MW 254,00/%MW1254.0 MW 254,01/%MW1254.1 MW 254,02/%MW1254.2 MW 254,03/%MW1254.3 MW 254,04/%MW1254.4 MW 254,05/%MW1254.5 MW 254,06/%MW1254.6 MW 254,07/%MW1254.7	MW 254,08/%MW1254.8 MW 254,09/%MW1254.9 MW 254,10/%MW1254.10 MW 254,11/%MW1254.11 MW 254,12/%MW1254.12 MW 254,13/%MW1254.13 MW 254,14/%MW1254.14 MW 254,15/%MW1254.15	MW 255,00/%MW1255.0 MW 255,01/%MW1255.1 MW 255,02/%MW1255.2 MW 255,03/%MW1255.3 MW 255,04/%MW1255.4 MW 255,05/%MW1255.5 MW 255,06/%MW1255.6 MW 255,07/%MW1255.7	MW255,08/%MW1255.8 MW255,09/%MW1255.9 MW255,10/%MW1255.10 MW255,11/%MW1255.11 MW255,12/%MW1255.12 MW255,13/%MW1255.13 MW255,14/%MW1255.14 MW255,15/%MW1255.15
LED displays after initialization	FK1 lights up or LED RUN does not light up when the RUN/STOP switch is set to RUN	FK2 lights up or LED RUN does not light up when the RUN/STOP switch is set to RUN	FK3 lights up. In addition according to error type: LED BE (Bus Error) LED RE (Remote Unit Error) LED SE (Serial Unit Error)	LED RE (Remote Unit Error) lights up
Reaction when switching on the basic unit / Reaction during operation	All outputs remain FALSE or are set to FALSE. The programming system does not have access. **Caution:** Both processors of the basic unit monitor each other mutually, thus facilitating a powerful diagnosis. If the safety requirements are higher, use specially approved control systems.	All outputs remain set to FALSE or are set to FALSE. The programming system can get access. The user program is not started or is aborted.	You can choose in case of an error: - Just report the error: Evaluate the error flag M 255,13/%MX255.13 – Abort the user program: Set system constant KW 0,7/%MW1000.7 = 1 (FK3_REAK)	Evaluation of the error messages using the user program
Acknowledgement of the sum error message or the error class message	- Power ON - Cold start	- Power ON - Cold start	- Power ON / cold start - Set RUN/STOP switch to RUN - Start the program using 907 AC 1133 - Set M255,13 / %MX255.13 or M255,14 / %MX255.14 to FALSE - In case of CS31 error: function block CS31QU	

[1]) The sum error flag M 255,10 / %MX255.10 becomes TRUE, if at least one of the error class flags becomes 1.
If M 255,10 / %MX255.10 = FALSE the basic unit has not detected any error.
The sum error flag is automatically deleted when the error class flags are acknowledged.

[2]) The basic unit enters the most recent found error into the relevant word flag set for each error class. The entry is made at the end of the program cycle and remains unchanged during the next running program cycle. The word flags can only be acknowledged by overwriting them with "0".

1.7.8 Acknowledgement of error messages in the basic unit

Error messages remain stored and are displayed until they are acknowledged. The following applies:

- The sum error message, the error class messages (bit flags) and the relevant LEDs FK1, FK2 and FK3 are reset by power ON operation, for example. Refer to the previous chapters for further possibilities of resetting/acknowledging them.

- The error identification and the detailed information (word flag) are to be deleted by means of the user program or using the online function "Overwrite". They are also deleted when a cold start is performed or by a power-fail, if no backup battery is effective.

If an error has not been eliminated, the error message will appear again.

1.7.9 Additional diagnosis functions

Status word EW 07,15 / %IW1007.15

The following information are continuously updated in the status word EW 07,15 / %IW1007.15:

- Bit 0: This bit is valid for the standalone PLC, for the Master PLC and for the slave PLC.
  Bit 0 = 1 (TRUE), there is no class 2 error present.
  Bit 0 = 0 (TRUE), there is a class 2 error.

- Bit 1: This bit is valid only for the slave PLC.
  Bit 1 = 1 (TRUE), the slave PLC is added to the bus cycle of the master PLC.
  Bit 1 = 0 (FALSE), the slave PLC is not added to the bus cycle of the master PLC.

- Bit 2 = 1 (TRUE), time and date are valid

- Bit 3 = 1 (TRUE), battery available

- Bits 4..7 are not used

- Bits 8...15: Maximum number of remote modules which have been existing in the CS31 bus cycle of the master PLC since the last power ON or the last cold start. This number may be larger than the number of remote modules which are currently existing in the CS31 bus cycle.

1.7.10 Meaning of the contents of the error word flags

Explanation of the following table:

- Address = Memory address where the error was detected.
- Group number = Module address of the remote module.
- Channel number = Number of the faulty channel.
- Module type / Meaning
  - 000 Binary input
  - 001 Analog input
  - 002 Binary output
  - 003 Analog output
  - 004 Binary input/output
  - 005 Analog input/output
  - 255 Bus master of slave basic unit where the error has occurred and where it is stored.

FK1 - Fatal errors

Error class	**Error description**	**Error identifier in MW 254,00/ %MW1254.0** **Dec Hex**	**Detailed info 1 in MW 254,01/ %MW1254.1**	**Detailed info 2 in MW 254,02/ %MW1254.2**	**Detailed info 3 in MW 254,03/ %MW1254.3**	**Further detailed infos in MW 254,04/ %MW1254.4 : MW 254,07/ %MW1254.7**
FK1	Checksum error of the system EPROM	1_D 1_H	-	-	-	-
Fatal error	Operating system of the basic unit is defective or a defective RAM is detected when a cold start is performed (complete RAM test)	2_D 1_H	Address	-	-	-

FK2 - Serious errors

Error class	Error description	Error identifier in MW 254,08/ %MW1254.8 Dec Hex	Detailed info 1 in MW 254,09/ %MW1254.9	Detailed info 2 in MW 254,10/ %MW1254.10	Detailed info 3 in MW 254,11/ %MW1254.11	Further detailed infos in MW 254,12/ %MW1254.12 : MW 254,15/ %MW1254.15
FK2	Illegal master/slave identifier	129_D 81_H	-	-	-	-
Serious error	A serious error has occurred when the CS31 bus interfacing was initialized. The CS31 bus processor does not give any response to the PLC side within the specified time.	130_D 82_H	-	-	-	-
	The PLC is overloaded, the cycle time is too short.	131_D 83_H	-	-	-	-
	Checksum error in the Flash EPROM.	133_D 85_H	-	-	-	-
	The CS31 bus processor reports an error via EW 07,15 / %IW1007.15 bit 0. This bit is checked prior to each start of the PLC program.	136_D 88_H	-	-	-	-
	An illegal value has been configured for specifying the size of the I/O area between the master PLC and the slave PLC (KW 00,10 / %MW3000.10 or KW 00,11 / %MW3000.11).	137_D 89_H	-	-	-	-
	A serious error has occurred when the I/O processor was initialized. The I/O processor does not give any response to the PLC side within the specified time.	139_D $8B_H$	-	-	-	-
	The PLC is overloaded, the flashing process is not started.	180_D $B4_H$	-	-	-	-
	The program is too large, flashing process cannot be performed.	181_D $B5_H$	-	-	-	-
	Exception in the runtime system.	182_D $B6_H$	-	-	-	-
	Erro of internal couplers (except ARCNET).	200_D $C8_H$	For detailed information see "System technology of the internal couplers".			
	An unknown operator or block is detected in the user program during the runtime.	258_D 102_H	7 characters representing the block name.			
	The CS31 bus processor does not work correctly. The life identifier does not change.	259_D 103_H	-	-	-	-
	The I/O processor does not work correctly. The life identifier does not change.	260_D 104_H	-	-	-	-

FK3 - Light errors

Error class	Error description	Error identifier in MW 255,00/ %MW1255.0 Dec Hex	Detailed info 1 in MW 255,01/ %MW1255.1	Detailed info 2 in MW 255,02/ %MW1255.2	Detailed info 3 in MW 255,03/ %MW1255.3	Further detailed infos in MW 255,04/ %MW1255.4 : MW 255,07/ %MW1255.7
FK3	Remote module disconnected.	15_D F_H	Module type	Group number	-	-
Light error	CS31 bus error (no remote module on the system bus). **Note:** If there are only analog modules connected to the CS31 system bus, this error message may occur when the supply voltage is switched on although the analog modules have been correctly added to the CS31 bus cycle after a certain time. **Reason:** The analog modules only appear on the CS31 system bus after the quite long initialization phase is completed. During the initialization time the master PLC cannot recognize them.	16_D 10_H	-	-	-	-
	The Flash EPROM cannot be programmed.	128_D 80_H	Address of defective memory cell	-	-	-
	The Flash EPROM cannot be deleted.	129_D 81_H	Address of memory cell which is undeletable	-	-	-
	The PLC operation mode configured in the system constant KW 00,00 / %MW3000.0 has not yet been activated. Please activate it (see also system constant KW 00,00 / %MW3000.0).	130_D 82_H	Value of KW 00,00/ %MW3000.0 activated last	Value of KW 00,10/ %MW3000.10 activated last	Value of KW 00,11/ %MW3000.11 activated last	-
	Inserted SMC user program card has a wrong version identifier for the operating system.	150_D 96_H	-	-	-	-
	Inserted SMC firmware card has a wrong identifier of the module type.	151_D 97_H	-	-	-	-

Error class	Error description	Error identifier in MW 255,00/ %MW1255.0 Dec Hex	Detailed info 1 in MW 255,01/ %MW1255.1	Detailed info 2 in MW 255,02/ %MW1255.2	Detailed info 3 in MW 255,03/ %MW1255.3	Further detailed info in MW 255,04/ %MW1255.4 : MW 255,07/ %MW1255.7
FK3	Error with Flash task	180_D $B4_H$	-	-	-	-
Light error	Not enough memory with Alloc()	181_D $B5_H$	-	-	-	-
	Error during write/read of the SMC card	182_D $B6_H$	Error code:	-	-	-
			001_D 001_H	Block number and number of blocks greater than max. allowed		
			002_D 002_H	Sector number greater than max. allowed		
			004_D 004_H	Block cannot be programmed, or check sum error		
			008_D 008_H	Block already contains data		
			016_D 010_H	Sector cannot be erased		
			032_D 020_H	Card not initialized		
			064_D 040_H	Card not inserted or wrong ID Code		
			128_D 080_H	Block is empty		
	Error of internal couplers (except ARCNET)	200_D $C8_H$	Detailed information in ‚System technology of the internal couplers'			

FK4 - Warning

Error class	Error description	Error identifier in MW 255,08/ %MW1255.8 Dec Hex	Detailed info 1 in MW 255,09 %MW1255.9	Detailed info 2 in MW 255,10/ %MW1255.10	Detailed info 3 in MW 255,11 %MW1255.11	Further detailed infos in MW 255,12/ %MW1255.12 : MW 255,15/ %MW1255.15
FK4 Warning	Internal error of a remote module.	1_D 1_H	Module type	Group number	Channel number	-
	Cut wire (open circuit)	2_D 2_H	Module type	Group number	Channel number	-
	Incorrect level on an analog output.	3_D 3_H	Module type	Group number	Channel number	-
	Overload	4_D 4_H	Module type	Group number	Channel number	-
	Overload + cut wire	6_D 6_H	Module type	Group number	Channel number	-
	Short-circuit	8_D 8_H	Module type	Group number	Channel number	-
	Cut wire (at analog modules)	9_D 9_H	Module type	Group number	Channel number	-
	Short-circuit + cut wire "out of range" at analog modules	10_D A_H	Module type	Group number	Channel number	-
	Overload + short-circuit	12_D C_H	Module type	Group number	Channel number	-
	Short-circuit + overload + cut wire	14_D E_H	Module type	Group number	Channel number	-
	Internal error (non-maskable internal interrupt occurred)	136_D 88_H	-	-	-	-
	Internal error (inhibited interrupt occurred)	137_D 89_H	-	-	-	-
	The module uses default adjustment values for the direct analog inputs and ouputs instead of the factory settings.	140_D $8C_H$	-	-	-	-
	Error of internal couplers (except ARCNET)	200_D $C8_H$	Detailed information in "System technology of the internal couplers".			

1.7.11 Reaction of the bus master basic unit and the remote modules in case of errors

No.	Error	Display/reaction of the bus master basic unit	Display/reaction of the input/output rem. modules	Display/reaction of the slave basic units
1	Bus master basic unit has failed, e.g. due to a power failure.	No display, all outputs are set to FALSE (0).	LED (3) lights up. All outputs change to FALSE (0).	07 KR 91 / 07 KT 9x: - LED BA lights up LED RE flashes - Bit 1 = 0 (FALSE) in the status word EW 07,15/%IW1007.15 07 KR 31 / 07 KT 31: - Error LED flashes - Bit 1 = 0 (FALSE) in the status word EW 07,15/%IW1007.15
2	The bus master function of the basic unit (Serial unit) has failed, e.g. the bus processor is defective.	Displays: FK2 = Serious Error SE = Serial Unit Error Flags: M255,10/%MX255.10 = 1 M255,12/%MX255.12 = 1 Further flags, see 1.6.7		
3a 3b	The CS31 system bus including at least 2 remote modules is disconnected (all remote modules are disconnect.) or the CS31 system bus is short-circuited.	Displays: FK3 = Light Error BE = Bus Error RE = Remote Unit Error SE = Serial Unit Error Flags: M255,10/%MX255.10 = 1 M255,13/%MX255.13 = 1 Further flags, see 1.6.7		
4a	The CS31 system bus is disconnected (some of the remote modules are disconnected).	Displays: FK3 = Light Error BE = Bus Error RE = Remote Unit Error Flags: M255,10/%MX255.10 = 1 M255,13/%MX255.13 = 1 Further flags, see 1.6.7	Remote modules **without** connection to the bus master basic unit: reaction see no. 1	Slave basic units **without** connection to the bus master basic unit: reaction see no. 1
4b			Remote modules **with** connection to the bus master basic unit: no display/no reaction	Slave basic units **with** connection to the bus master basic unit: no display/no reaction
5a	The master basic unit does not find any remote modules on the CS31 bus after power ON or after cold start or CS31 system bus with at least 3 remote modules: 2 remote modules are disconnected.	Displays: BE = Bus Error Flags: M255,10/%MX255.10 = 1 M255,13/%MX255.13 = 1 For further flags, see 1.6.7	Remote modules **with** connection to the bus master basic unit: no display/no reaction	Slave basic units **with** connection to the bus master basic unit: no display/no reaction
5b	No connection to the CS31 system bus.		Remote modules **without** connection to the bus master basic unit: reaction see no. 1	Slave basic units **without** connection to the bus master basic unit: reaction see no. 1
5c	Defective remote modules.		Not clear	Error class FK1 / FK2, all outputs change to FALSE (0).
5d	Power supply failure.		All outputs change to 0	All outputs change to FALSE (0)
6a	An error has occurred at the inputs or outputs of a remote module, e.g. a short-circuit.	RE = Remote Unit Error Flags: M255,10/%MX255.10 = 1 M255,14/%MX255.14 = 1 (FK4)	Involved remote module: LED (3) lights up The LEDs (1) provide detailed information when using the test button (4).	Involved 07 KT 9x: LED Ovl. = Short-circuit 07 KT 31: Error LED ON Flags (07 KT 9x, KT 31): M255,10/%MX255.10=TRUE(1) M255,14/%MX255.14=TRUE(1) For further flags, see 1.6.7
6b			Not involved remote mod.: no display/no reaction	Not involved slave basic units: no display/no reaction

Reaction of the bus master basic unit and the remote modules in case of errors (continued)

No.	Error	Display/reaction of the bus master basic unit	Display/reaction of the input/output remote modules	Display/reaction of the slave basic units
7a	Two remote modules of the same type with inputs are set to the same address.	This error is detected only when the signal states of the two modules become different. In this case, the telegram is faulty and the modules are considered as disconnected. Display: RE = Remote Unit Error Flags: M255,10/%MX255.10 = 1 M255,13/%MX255.13 = 1 For further flags, see 1.6.7 / 1.6.10	Involved modules: reaction see no. 1 Other modules: no display/no reaction	Reaction see no. 1
7b	Two remote modules of the same type are set to the same address.	No reaction, unless there is a large distance between the remote modules.	Error-free operation of the two modules, unless there is a large distance between the modules.	Not applicable because inputs and outputs are always present.
7c	Two remote modules of different types, but with overlapping ranges are set to the same address, e.g. ICSI 16 D1 and ICSK 20 F1.	The error is already detected during initialization. The remote modules are not added into the bus cycle.	Involved modules: reaction see no. 1 Other modules: no display/no reaction	Involved modules: reaction see no. 1 Other modules: no display/no reaction
7d	Address 62 or 63 has been set for a binary remote module.	Is not detected.	- Signal output in parallel to the bus master. - Input signals are ignored.	-
7e	An address > 5 has been set for an analog remote module.	Is not detected.	Reaction see no. 1	-

1.8 Serial interfaces COM1 and COM2

1.8.1 Operating modes of the serial interfaces

Interface standard: EIA RS-232

The interface operating mode is set depending on the particular application.

Programming and test
or
Man-Machine Communication MMC

Active mode: The active mode is used for programming and testing the basic unit, i.e. it provides access to all programming and test functions of the basic unit.

Passive mode: The passive mode serves for the communication between the user program and a module which is connected to the serial interface (e.g. MODBUS protocol).

1.8.2 Basic features of the serial interfaces

The serial interfaces of the 07 KT 97 can be used in two different modes: active mode and passive mode. Active mode always means the programming and online access with 907 AC 1131. Passive modes are such interface modes which are set up by means of a function block.

The interfaces can be set to the modes independently of each other. The passive modes are installed by function blocks (e.g. MODINIT, COMINIT) from the user project. The selection of the interface number is performed automatically. It is explained in the following sections.

Caution:
Under no circumstances, both serial interfaces can be set to the active mode **at the same time.**

- Both interfaces are able to support the active mode, however, only one at a time.
- Both interfaces are able to support the passive mode. It is possible to use them in passive mode at the same time. The combination of the operating modes MODBUS slave, MODBUS master and ‚free mode‘ COM is no subject to any restrictions.
- In order to use an interface in the passive mode, a connection between the pins 6 and 8 is necessary at the interface cable on the PLC side.
- Per serial interface, only one passive mode initialization function block may be used (e.g. 1 x MODINIT or 1 x COMINIT)
- In the MODBUS Slave operating mode, the driver functionality is completely realized by the MODINIT function block.
- In the MODBUS Master operating mode, transmitting and receiving of messages is handled by the additional function block MODMAST. It is recommended to use only one MODMAST block for each interface within one project. If necessary, the input values can be changed during the course of the program according to the requirements.

- If several MODMAST blocks are used, it is absolutely necessary to lock the enabling inputs of the blocks against one another.
- In the free operating mode COM, only one COMREC receiving block may be used.
- In the free operating mode COM, several COMSND blocks may be used for each interface.

1.8.3 Behaviour of the serial interfaces

The serial interfaces have several automatic functions concerning setting and changing the operating mode. The resulting functionality is explained in the following chapters.

Starting-up / Booting the PLC

While starting up the PLC, at first the serial interfaces are checked for connections between the pins 6 and 8. In case of connected cables (e.g. MODBUS via COM1) the valid programming interface is detected in this way.

The following table shows the automatically set programming interface detected while starting up the PLC depending on the connected interface cables.

COM1	COM2	Programming access via
no connection (pins 6 and 8)	**no** connection (pins 6 and 8)	COM1
no connection (pins 6 and 8)	connection (pins 6 and 8)	COM1
connection (pins 6 and 8)	**no** connection (pins 6 and 8)	COM2
connection (pins 6 and 8)	connection (pins 6 and 8)	COM1

If cables with connections between pins 6 and 8 are installed at both serial interfaces, COM1 will automatically set as the programming interface at first while starting up the PLC. This makes sure that under all conditions a programming access is available. In order to establish a connection between 907 AC 1131 and the PLC, the cable containing the connection between the pins 6 and 8 has to be replaced by a suitable programming cable.

Configuration by using function blocks in the user program

If an interface initialization block (e.g. MODINIT, COMINIT) has successfully re-configured the currently valid serial programming interface, it automatically checks, if the other of the two interfaces is occupied by another initialization block. If this interface is free, it will be configured as the programming interface.

Example:

While starting up the PLC, no connection was detected between the pins 6 and 8 at the COM1 interface. As a consequence, COM1 will be configured as the programming interface.

In the user program, a COMINIT block is successfully used with COM1 (all input parameters have valid values, there is a connection between the pins 6 and 8 of COM1). As a result, COM1 will be configured in the free mode and COM2 as a programming interface.

Later on, the user program applies a MODINIT block successfully on COM2 (all input parameters have valid values, there is a connection between the pins 6 and 8 of COM2). From now on, COM1 will continue to be in the free mode and COM2 will run as a MODBUS interface. Now, both interfaces are occupied and, as a consequence, there is no longer a programming interface.

PLC RUN->STOP

If none on the two serial interfaces is available as programming interface, then, in case of STOP condition, the programming interface is automatically re-configured to that serial interface which was set with the start-up of the PLC, regardless of a connection between the pins 6 and 8 (see example above, last paragraph). In this case, the parameters (e.g. baud rate etc.) required for the access to 907 AC 1131 are also restored.

Removing an existing connection between the pins 6 and 8 of the serial interface during a running user program

If an interface was successfully configured in the user program in passive mode with MODINIT or COMINIT and, if this initialization block is still enabled (EN = TRUE, normal condition), the following applies:

The blocks automatically detect if the cable is unplugged (connection between pins 6 and 8 is missing) and signalize a corresponding error. If there is no serial interface configured as programming interface in this moment (both interfaces are used in passive mode by MODINIT and/or COMINIT) the block makes the interface the programming interface temporarily. The required parameters (baud rate etc.) are also set then.

This function is especially useful for the test of programs at PLCs without ARCNET connection and with both serial interfaces used in passive mode.

If the program was not modified (only monitored, no download, no online change), the block continues the previously configured normal operation after re-plugging the cable containing the connection between the pins 6 and 8. The pre-set parameters are still used. Otherwise the program must be re-started.

Note:
This function is supported by MODBUS blocks with run-time system versions 4.04 and later. In older run-time systems, the MODBUS blocks do not support the functions described above.

Automatic 907 AC 1131 login detection

As an option, the automatic 907 AC 1131 login detection can be activated for a serial interface in passive mode. Normally, the automatic login detection is deactivated. It can be activated by means of the COMAUTOLOGIN function from the COM_S90_V41.LIB library.

It is recommended, to activate the automatic login detection only with those projects which need it explicitly. Communication via the serial interface runs slower with the automatic login detection activated.

If the login detection is activated, the 907 AC 1131 login procedure via a serial interface in passive mode is automatically detected, regardless of a connection between the pins 6 and 8. This is very useful, if, for instance, at first a modem connected to a serial interface has to be initialized and set into operation by means of the blocks COMINIT, COMSND and COMREC, in order to establish a remote maintenance access via 907 AC 1131 during running operation.

The login procedure of 907 AC 1131 is detected via the following character string.

AA_{HEX}	AA_{HEX}	01_{HEX}	00_{HEX}	01_{HEX}	00_{HEX}	58_{HEX}	01_{HEX}	01_{HEX}
BLOCK IDENT (Low-Byte)	BLOCK IDENT (High-Byte)	BLOCK SIZE (Low-Byte)	BLOCK SIZE (High-Byte)	BLOCK NUMBER (Low-Byte)	BLOCK NUMBER (High-Byte)	CHECK SUM	LAST BLOCK	SERVICE ID (LOGIN)

Note:
Make sure that this character string does not appear within a normal communication message if the automatic login detection is activated. Otherwise, the character string will be interpreted as the 907 AC 1131 login character string. This can cause an unintended malfunction of the application.

Note:
The automatic login detection at an interface in passive mode is only possible, if the corresponding initialization block (COMINIT or MODINIT) is active for this interface (EN = TRUE).

Note:
This function is only supported by MODINIT blocks with run-time system versions 4.04 and later. In older run-time systems, the MODINIT block does not support the functions described above.

The possibility of the login detection depends also on the type of connection between the PLC and 907 AC 1131.

If the connection is established directly via RS-232, a login message can only be detected if during initialization of the interface the same parameters were set as used by 907 AC 1131 (19200 baud, 1 stop bit, no parity bit, character length 8 bits).

If the connection is established via an RS232/RS485 interface converter, a login message can also only be detected if the initialization parameters for the passive mode are equal to the parameters used by 907 AC 1131. Since in such an application normally several participants are connected to the RS-485 transmission line, also the following has to be taken into account:

The 907 AC 1131 login message does not include an address of the participant. For that reason, all participants on the RS-485 transmission line will detect the login message, provided that they are programmable with 907 AC 1131 and that their interface can interpret the login message (interface in active mode or passive mode with activated COMAUTOLOGIN). During the following acknowledgment of the login message by these participants message collisions can occur. As a consequence, the communication may be aborted.

If the connection between 907 AC 1131 and the PLC is established via modem, the communication is not influenced by the interface parameters set by COMINIT and MODINIT. The parameters, required by the modem, are to be set. After initialization, the modem converts the received data according to the settings. The assignment of the login message to a certain PLC is also guaranteed, because the connection only can be established by the modem-allocated telephone number or by MSN.

Logging-in with 907 AC 1131 causes a re-initialization of the interface at first. All blocks which access this interface are locked during the length of the online session, i.e. they do not perform any operation. The outputs of the blocks have the following values during this time

RDY	= FALSE
ERNO	= COM_AC1131_REMOTE_ACCESS

After logging-out with 907 AC 1131 the blocks are activated again.

The login monitoring for an interface only takes place, if 907 AC 1131 has not been logged-in via an other interface before (e.g. ARCNET, other COM). In addition, only such interfaces are monitored which are configured in a passive mode.

1.8.4 Interface parameters

Active mode:

The interface parameter settings are fixed and cannot be changed.

Data bits:	
Stop bits:	1
Parity bit:	none
Transfer rate:	19200 baud
Synchronization:	RTS/CTS

Passive mode:

The interface parameter settings are performed by the respective function block.

Data bits:	5, 6, 7 or 8
Stop bits:	1, 1,5 (for 5 data bits) , 2 (for 6, 7 or 8 data bits)
Parity bit:	none
Transfer rate:	300, 1200, 2400, 4800, 9600, 14400 or 19200 Baud
Synchronization:	RTS/CTS

1.9 Programming and test

1.9.1 Programming system 907 AC 1131

The programming system 907 AC 1131 can be used to program all categories of the controllers 07 KT 95, 07 KT 96, 07 KT 97 and 07 KT 98. The controllers 07 KR 91, 07 KT 92, 07 KT 93, 07 KT 94, 07 Kx 3x, 07 Kx 4x and 07 Kx 5x **cannot** be programmed using the 907 AC 1131.

The software can be executed on IBM/AT compatible Personal Computers having the operating system **Windows NT (version 4.0 or later, service pack 4 or higher), Windows 95 and Windows 98 SE (Second Edition)** installed. The drivers necessary for the serial communication and the communication via ARCNET are provided with the programming system.

The software package 907 AC 1131 is installed on the PC by a installation program which is extensively automated (see volume 4 "Installation"). The ARCNET drivers have to be installed additionally.

Apart from the usual functionality, the 907 AC 1131 presents the following outstanding features:

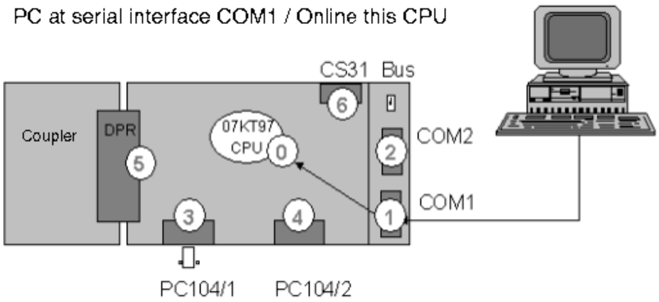
- The following PLC languages are supported:
  - Function block diagram (FBD)
  - Ladder diagram (LD)
  - Instruction list (IL)
  - Sequential function chart (SFC)
  - Structured text (ST)
- Simple handling
- Rapidity and low use of resources
- Very short compilation times
- Compact project files (one file per project)
- Complete and practical operation via keyboard, particularly for the graphic editors
- Many powerful commissioning features:
  - Monitoring of variable values
  - Writing and forcing of variable values
  - Single cycle
  - Breakpoints, single step operation and calling stack
- Sampling Trace: cycle exact recording and graphical display of variable values
- Powerflow: Display of passed lines,
  display of the accumulator content in IL or of the line status in FBD
- Integrated visualization to be used for commissioning, training, service and offline tests
- Offline simulation (not for external libraries, such as ABB-BIBx.lib)
- Online modifications (changes can be performed during operation)
- Conversion between PLC languages (IL, FBD, LD)
- Extensive support of the following IEC types:
  BOOL, 8, 16 and 32 bit integer values, 32 and 64 bit reals, strings, time, date, time of date, arrays (also multidimensional), structures, alias types, any combinations of structures, arrays and the remaining data types

1.9.2 Programming via the serial interfaces

The operation modes of the serial interfaces COM1 and COM2 are described in chapter 1.7. The serial interface COM1 is used as the default programming interface (without MMC).

The connection between PC and PLC is established using the system cable 07 SK 90.

In the programming software choose GATEWAY / Local and the driver ABB RS232 with the communication parameters 19200 baud, 8 data bits, no stop bit (refer to the 907 AC 1131 documentation, chapter 4 "The individual components / Online communication parameters for the use of gateways").

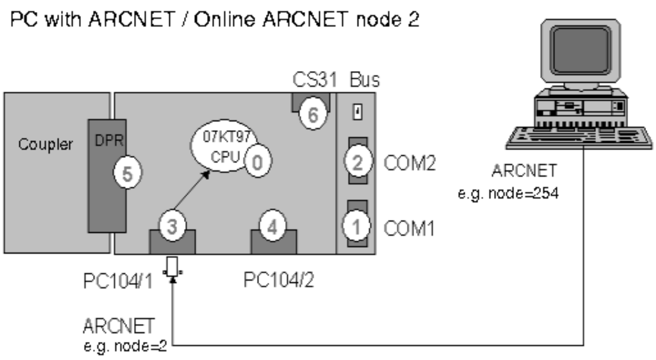


1.9.3 Programming via ARCNET

The programming via ARCNET is only possible on a PC with an included ARCNET card. The installation of the card(s) and the driver is described in volume 4 "Installation".

When programming via ARCNET, the PC is a participant in the ARCNET network. The participant number 254 is assigned to the PC as default value.

In the programming software choose GATEWAY / Local and the driver ABB ARCNET with the communication parameters Timeout 2000 ms and Node 254 (or any other free participant number). For more detailed information refer to the 907 AC 1131 documentation, chapter 4 "The individual components / Online communication parameters for the use of gateways".



Caution:
Always only one instance of the 907 AC 1131 may be opened and communicate with the PLC.

1.10 List of function blocks

All libraries and their function blocks are listed in tables in the description "Overview of the libraries" of 907 AC 1131.

1.11 Man-Machine Communication

1.11.1 MODBUS RTU for the serial interfaces

Protocol description

The MODBUS protocol is used all over the world. The **MODICON MODBUS® RTU** protocol is integrated in the basic unit 07 KT 97.

Numerous automation devices, such as PLC installations, displays, variable-frequency inverters or monitoring systems, for instance, have a MODBUS® RTU interface by default or as an option and can therefore communicate with the basic unit 07 KT 97 without any problems via the serial interfaces COM1 and COM2 (RS-232).

MODBUS® is a master-slave protocol. The master sends a request to the slave and receives its response.

Description of the MODBUS® protocol:

• Supported standard	EIA RS-232
• Number of connection points	1 master max. 1 slave with RS-232 interface max. 31 slaves with RS-232/RS-485 converter
• Protocol	MODBUS® (Master/Slave)
• Data transmission control	CRC 16
• Data transmission speed	up to 19200 baud
• Character encoding	1 start bit 8 data bits 1 parity bit, even or odd parity (optional) 1 or 2 stop bits
• Max. length of line	for RS-485: 1200 m at 19200 baud

The MODBUS® blocks transferred by the master contain the following information:

- the MODBUS® address of the interrogated slave (1 byte)
- the function code which defines the request of the master (1 byte)
- the data to be exchanged (n bytes)
- the CRC16 control code (2 bytes)

The basic unit 07 KT 97 processes only the following MODBUS® operation codes:

Function codes (HEX)	Description
01 or 02	read n bits
03 or 04	read n words
05	write one bit
06	write one word
07	rapid reading of the flags M01,00...M01,07 / %MX1.0..%MX1.7
0F	write n bits
10	write n words

The following restrictions apply to the length:

Function code (HEX)	max. length for master	max. length for slave
01 or 02	192 bits	192 bits
03 or 04	96 words / 48 double words	96 words / 48 double words
05	1 bit	1 bit
06	1 word	1 word
07	8 bits	8 bits
0F	192 bits	192 bits
10	96 words / 48 double words	96 words / 48 double words

MODBUS operation modes of the serial interfaces COM1 and COM2

Both serial interfaces of the basic unit 07 KT 97 can be operated simultaneously as MODBUS interfaces.

The MODBUS operating mode of an interface is set by means of the function block MODINIT. The interface is selected via the block input COM. The input MAST_SLAV determines the operating mode of the selected interface:

- MODBUS master: MAST_SLV = 100
- MODBUS slave: MAST_SLV = 100 + slave address

The settings are applied together with the interface parameters when a FALSE/TRUE edge occurs at the input FREI.

MODBUS master (MAST_SLV = 100)

In the operating mode MODBUS master, the telegram interchange with the slave(s) is performed via the MODMAST function block. The function block MODMAST sends the MODBUS request telegrams to the slave via the set interface and receives the MODBUS response telegrams from the slave via this interface.

All the telegrams are to be processed via a MODMAST function block.

MODBUS slave (MAST_SLV = 100 + slave address)

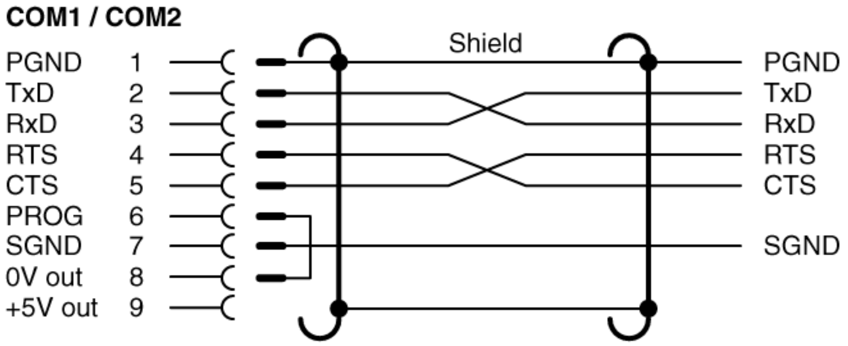
In the operating mode MODBUS slave, no function block is required for the MODBUS communication. Sending and receiving of MODBUS telegrams is performed automatically.

Initialization of the interfaces COM1 and COM2

The initialization of the interface is performed via the MODINIT block. The interface parameters are applied together with the selection of the interface (COM) and the presetting for the operating mode (MAST_SLV) when a FALSE/TRUE edge occurs at the input FREI.

Cable:

The cable used as the data transmission line must have the pins 6 and 8 bridged in the connector on the controller side, like the interface cable 07 SK 91.



PGND	Protective Ground	RTS	Request To Send
TxD	Transmit Data	CTS	Clear To Send
RxD	Receive Data	PROG	Switch-over active/passive mode
		SGND	Signal Ground

Connection diagram COM1 / COM2 for MODBUS

MODBUS telegrams

The sending and receiving telegrams shown are not visible in the PLC. In both operating modes the function block MODINIT only indicates whether a telegram was received (REC = TRUE) or sent (SND = TRUE) and which function code the telegram contains (FCT). However, the complete telegrams can be displayed using a serial data analyzer which is inserted into the connection line between master and slave, if required.

The amount of user data depends on the properties of the master and the slave.

For the following examples it is assumed that a SST MODBUS module is used as slave. There may be different properties if modules of other manufacturers are used.

FCT 1 or 2: Read n bits

n = 1...96

Master request

Slave	Func.	Op. addr. slv.		No. of bits		CRC	
address	code	High	Low	High	Low	High	Low

Slave response

Slave	Func.	No. of	...data...	CRC	
address	code	bytes		High	Low

Example: MODBUS interface of the master: COM1
Master reads from: Slave 1
Data: M01,04 = FALSE; M01,05 = TRUE; M01,06 = FALSE
Source address on the slave: M01,04: 2014_{HEX} = 8212_{DEC}
Target address on the master: M10,01
The values of the flags M01,04...M01,06 on the slave are stored on the master starting with M10,01.

MODBUS request of the master

Slave	Func.	Op. addr. slv.		No. of bits		CRC	
address	code	High	Low	High	Low	High	Low
01_{HEX}	01_{HEX}	20_{HEX}	14_{HEX}	00_{HEX}	03_{HEX}	37_{HEX}	CF_{HEX}

MODBUS response of the slave

Slave	Func.	No. of	Data	CRC	
address	code	bytes		High	Low
01_{HEX}	01_{HEX}	01_{HEX}	02_{HEX}	$D0_{HEX}$	49_{HEX}

Parameter setting for the block inputs MODMAST

NB = number of bits

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	1	Application specific	8212	3	ADR (M10,01)

FCT 3 or 4: Read n words

n = 1...96

Master request

Slave	Func.	Op. addr. slv.		No. of words		CRC	
address	code	High	Low	High	Low	High	Low

Slave response

Slave	Func.	No. of	...data...	CRC	
address	code	bytes		High	Low

Example: MODBUS interface of the master: COM1
Master reads from: Slave 1
Data: MW00,04 = 4; MW00,05 = 5; MW00,06 = 6
Source address on the slave: MW00,04: 2004_{HEX} = 8196_{DEC}
Target address on the master: MW10,01
The values of the flag words MW00,04...MW00,06 on the slave are stored on the master starting with MW10,01.

MODBUS request of the master

Slave	Func.	Op. addr. slv.		No. of words		CRC	
address	code	High	Low	High	Low	High	Low
01_{HEX}	03_{HEX}	20_{HEX}	04_{HEX}	00_{HEX}	03_{HEX}	$4F_{HEX}$	CA_{HEX}

MODBUS response of the slave

Slave	Func.	No. of	Data		Data		Data		CRC	
address	code	bytes	High	Low	High	Low	High	Low	High	Low
01_{HEX}	03_{HEX}	06_{HEX}	00_{HEX}	04_{HEX}	00_{HEX}	05_{HEX}	00_{HEX}	06_{HEX}	40_{HEX}	$B6_{HEX}$

Parameter setting for the block inputs MODMAST

NB = number of words

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	3	Application specific	8196	3	ADR (MW0,01)

FCT 3 or 4: Read n double words

The function code "read double word" is not defined in the MODBUS RTV standard. This is why the double word is composed of a low word and a high word (manufacturer specific).

n = 1...48

Master request

Slave	Func.	Op. addr. slv.		No. of words		CRC	
address	code	High	Low	High	Low	High	Low

Slave response

Slave	Func.	No. of	...data...	CRC	
address	code	bytes		High	Low

Example: MODBUS interface of the master: COM1
Master reads from: Slave 1
Data: MD00,02 = 32; MD00,03 = 80000
Source address on the slave: MD00,02: 4002_{HEX} = 16386_{DEC}
Target address on the master: MD00,00
The values of the flag double words MD00,02...MD00,03 on the slave are stored on the master starting with MD00,00.

MODBUS request of the master

Slave	Func.	Op. addr. slv.		No. of words		CRC	
address	code	High	Low	High	Low	High	Low
01_{HEX}	03_{HEX}	40_{HEX}	02_{HEX}	00_{HEX}	04_{HEX}	$F0_{HEX}$	09_{HEX}

MODBUS response of the slave

Slave	Func.	No. of	Data		Data		Data		Data		CRC	
address	code	bytes	High	Low	High	Low	High	Low	High	Low	High	Low
01_{HEX}	03_{HEX}	08_{HEX}	00_{HEX}	00_{HEX}	00_{HEX}	20_{HEX}	00_{HEX}	01_{HEX}	38_{HEX}	80_{HEX}	57_{HEX}	$B0_{HEX}$

Parameter setting for the block inputs MODMAST

NB = number of bits

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	31	Application specific	16386	4	ADR (MD00,00)

FCT 5: Write 1 bit

For the function code "write 1 bit", the value of the bit to be written is encoded in one word.

BIT = TRUE → Data word = FF 00 $_{HEX}$

BIT = FALSE→ Data word = 00 00 $_{HEX}$

n = 1

Master request

Slave	Func.	Op. addr. slv.		Data		CRC	
address	code	High	Low	High	Low	High	Low

Slave response

Slave	Func.	Op. addr. slv.		Data		CRC	
address	code	High	Low	High	Low	High	Low

Example: MODBUS interface of the master: COM1
Master writes to: Slave 1
Data: M10,01 = TRUE
Source address on the master: M10,01
Target address on the slave: M01,04: 2017_{HEX} = 8215_{DEC}
The value of the flag M10,01 on the master is stored on the slave in M01,04.

MODBUS request of the master

Slave	Func.	Op. addr. slv.		Data		CRC	
address	code	High	Low	High	Low	High	Low
01_{HEX}	05_{HEX}	20_{HEX}	17_{HEX}	FF_{HEX}	00_{HEX}	37_{HEX}	FE_{HEX}

MODBUS response of the slave (mirrored)

Slave	Func.	Op. addr. slv.		Data		CRC	
address	code	High	Low	High	Low	High	Low
01_{HEX}	05_{HEX}	20_{HEX}	17_{HEX}	FF_{HEX}	00_{HEX}	37_{HEX}	FE_{HEX}

Parameter setting for the block inputs MODMAST

NB = number of bits

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	5	Application specific	8215	1	ADR (M10,01)

FCT 6: Write 1 word

n = 1

Master request

Slave	Func.	Op. addr. slv.		Data		CRC	
address	code	High	Low	High	Low	High	Low

Slave response

Slave	Func.	Op. addr. slv.		Data		CRC	
address	code	High	Low	High	Low	High	Low

Example: MODBUS interface of the master: COM1
Master writes to: Slave 1
Data: MW10,01 = 7
Source address on the master: MW10,01
Target address on the slave: MW00,07: 2007_{HEX} = 8199_{DEC}
The value of the flag word MW10,01 on the master is stored on the slave in MW00,07.

MODBUS request of the master

Slave	Func.	Op. addr. slv.		Data		CRC	
address	code	High	Low	High	Low	High	Low
01_{HEX}	06_{HEX}	20_{HEX}	07_{HEX}	00_{HEX}	07_{HEX}	72_{HEX}	09_{HEX}

MODBUS response of the slave (mirrored)

Slave	Func.	Op. addr. slv.		Data		CRC	
address	code	High	Low	High	Low	High	Low
01_{HEX}	06_{HEX}	20_{HEX}	07_{HEX}	00_{HEX}	07_{HEX}	72_{HEX}	09_{HEX}

Parameter setting for the block inputs MODMAST

NB = number of words

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	6	Application specific	8215	1	ADR (MW0,01)

FCT 7: Fast reading M01,00...M01,07

Master request

Slave address	Func. code	CRC	
		High	Low

Slave response

Slave address	Func. code	Data byte	CRC	
			High	Low

Example: MODBUS interface of the master: COM1
Master writes to: Slave 1
Data: M01,00 = TRUE; M01,01 = FALSE;
M01,02 = TRUE; M01,03 = FALSE;
M01,04 = TRUE; M01,05 = FALSE;
M01,06 = TRUE; M01,07 = FALSE
Source address on the slave: M01,00: 2010_{HEX} = 8208_{DEC}
Target address on the master: M01,08
The values of the flags M01,00...M01,07 on the slave are stored on the master in M01,08...M01,15.

MODBUS request of the master

Slave address	Func. code	CRC	
		High	Low
01_{HEX}	07_{HEX}	41_{HEX}	$E2_{HEX}$

MODBUS response of the slave

Slave address	Func. code	Data byte	CRC	
			High	Low
01_{HEX}	07_{HEX}	55_{HEX}	$E2_{HEX}$	$0F_{HEX}$

Parameter setting for the block inputs MODMAST

NB = number of bits

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	7	Application specific	8208	8	ADR (M10,01)

FCT 15: Write n bits

n = 1...192 (if 07 KT 9X is the MODBUS slave)

Master request

Slave address	Func. code	Op. addr. slv.		No. of bits		No. of bytes	...data...	CRC	
		High	Low	High	Low			High	Low

Slave response

Slave address	Func. code	Op. addr. slv.		No. of bits		CRC	
		High	Low	High	Low	High	Low

Example: MODBUS interface of the master: COM1
Master writes to: Slave 1
Data: M01,01 = TRUE; M01,02 = FALSE; M01,03 = TRUE
Source address on the master: M01,01
Target address on the slave: M01,01: 2011_{HEX} = 8209_{DEC}
The values of the flags M01,01...M01,03 on the master are stored on the slave in M01,01...M01,03.

MODBUS request of the master

Slave address	Func. code	Op. addr. slv.		No. of bits		No. of bytes	Data	CRC	
		High	Low	High	Low			High	Low
01_{HEX}	$0F_{HEX}$	20_{HEX}	11_{HEX}	00_{HEX}	03_{HEX}	01_{HEX}	05_{HEX}	$B4_{HEX}$	37_{HEX}

MODBUS response of the slave

Slave address	Func. code	Op. addr. slv.		No. of bits		CRC	
		High	Low	High	Low	High	Low
01_{HEX}	$0F_{HEX}$	20_{HEX}	11_{HEX}	00_{HEX}	03_{HEX}	$4E_{HEX}$	$0F_{HEX}$

Parameter setting for the block inputs MODMAST

NB = number of bits

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	15	Application specific	8209	3	ADR (M01,01)

FCT 16: Write n words

n = 1...96 (if 07 KT 9X is the MODBUS slave)

Master request

Slave address	Func. code	Op. addr. slv.		No. of words		No. of bytes	...data...	CRC	
		High	Low	High	Low			High	Low

Slave response

Slave address	Func. code	Op. addr. slv.		No. of words		CRC	
		High	Low	High	Low	High	Low

Example: MODBUS interface of the master: COM1
Master writes to: Slave 1
Data: MW01,01 = 1; MW01,02 = 2; MW01,03 = 3
Source address on the master: MW01,01
Target address on the slave: MW00,01: 2001_{HEX} = 8193_{DEC}
The values of the flag words MW01,01...MW01,03 on the master are stored on the slave in MW00,01...MW00,03.

MODBUS request of the master

Slave address	Func. code	Op. addr. slv.		No. of words		No. of bytes	Data		Data		Data		CRC	
		High	Low	High	Low		High	Low	High	Low	High	Low	High	Low
01_{HEX}	10_{HEX}	20_{HEX}	01_{HEX}	00_{HEX}	03_{HEX}	06_{HEX}	00_{HEX}	01_{HEX}	00_{HEX}	02_{HEX}	00_{HEX}	03_{HEX}	$C0_{HEX}$	84_{HEX}

MODBUS response of the slave

Slave address	Func. Code	Op. addr. slv.		No. of words		CRC	
		High	Low	High	Low	High	Low
01_{HEX}	10_{HEX}	20_{HEX}	01_{HEX}	00_{HEX}	03_{HEX}	DA_{HEX}	08_{HEX}

Parameter setting for the block inputs MODMAST

NB = number of words

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	16	Applicaton specific	8193	3	ADR (MW01,01)

FCT 16: Write n double words

The function code "write double word" is not defined in the MODBUS RTV standard. This is why the double word is composed of a low word and a high word (manufacturer specific).

n = 1...48

Master request

Slave address	Func. code	Op. addr. slv.		No. of words		No. of bytes	...Data...	CRC	
		High	Low	High	Low			High	Low

Slave response

Slave address	Func. code	Op.-Adr. Slave		No. of words		CRC	
		High	Low	High	Low	High	Low

Example: MODBUS interface of the master: COM1
Master writes to: Slave 1
Data: MD00,00 = 18; MD00,01 = 65561
Source address on the master: MD00,00
Target address on the slave: MD00,00: 4000_{HEX} = 16384_{DEC}
The values of the flag double words MD00,00...MD00,01 on the master are stored on the slave in MD00,00...MD00,01.

MODBUS request of the master

Slave address	Func. Code	Op. addr. slv.		No. of words		No. of bytes	Data		Data		Data		Data		CRC	
		High	Low	High	Low		High	Low	High	Low	High	Low	High	Low	High	Low
01_{HEX}	10_{HEX}	40_{HEX}	00_{HEX}	00_{HEX}	04_{HEX}	08_{HEX}	00_{HEX}	00_{HEX}	00_{HEX}	12_{HEX}	00_{HEX}	01_{HEX}	00_{HEX}	19_{HEX}	60_{HEX}	$B3_{HEX}$

MODBUS response of the slave

Slave address	Func. code	Op.-Adr. Slv		No. of words		CRC	
		High	Low	High	Low	High	Low
01_{HEX}	10_{HEX}	40_{HEX}	00_{HEX}	00_{HEX}	04_{HEX}	DA_{HEX}	$0A_{HEX}$

Parameter setting for the block inputs MODMAST

NB = number of words = 2 x number of double words

EN	COM	SLAVE	FCT	TIMEOUT	ADDR	NB	DATA
FALSE ->TRUE	1	1	16	Application specific	16384	4	ADR (MD00,00)

Error telegram

In the operating mode MODBUS master, the 07 KT 97 sends a telegram only if the parameters at the MODMAST inputs are logically correct. Nevertheless it can happen that a slave cannot process the request of the master or that the slave cannot interpret the request due to transmission errors. In those cases the slave sends an error telegram to the master. As an identification for the error telegram the slave sends back a function code resulting of a logical OR interconnection of the function code received from the master and the value 80_{HEX}.

Slave response

Slave address	Func. code OR 80_{HEX}	Error code	CRC	
			High	Low

Possible error codes of the slave

Code	**Meaning**
01_{DEC}	The slave does not support the function requested by the master
02_{DEC}	Invalid operand address in the slave
02_{DEC}	Operand area exceeded
03_{DEC}	At least one value is outside the permitted value range
12_{DEC}	The amount of data is higher than the slave can process
13_{DEC}	The telegram contains an odd number of words in case of double word access
10_{DEC}	The length specifications in the telegram do not match
11_{DEC}	The type of operand area and the function do not match

Example:

MODBUS request of the master:

Function code: 01 (read n bits)

Operand address slave: 4000_{HEX} = 16384 $_{DEC}$ (flag area double word)

MODBUS response of the slave

Function code: 81_{HEX}

Error code: 02

Function block MODINIT

This function block initializes and configures a serial interface to be a MODBUS interface. It is required for the operating mode MODBUS master as well as for the operating mode MODBUS slave. This function block can be applied to the local controller interfaces (COM1/COM2) and to the interfaces of the MODBUS coupler 07 KP 93 (COM3/COM4). A separate instance of the function block must be used for each interface.

MODINT is a part of the libraries ABB.BIB4.LIB and COM_S90_V41.LIB.

Function block MODMAST

This function block is only required in the operating mode MODBUS master. It deals with the communication (sending telegrams to the slaves and receiving telegrams from the slaves). This function block can be applied to the local controller interfaces (COM1/COM2) and to the interfaces of the MODBUS coupler 07 KP 93 (COM3/COM4). A separate instance of the function block must be used for each interface.

MODMAST is a part of the libraries ABB.BIB4.LIB and COM_S90_V41.LIB.

Cross-reference list

Caution:
An access to reserved operands (e.g. E065_00 = %IX065.00) or undefined operands (e.g. E64_08 = %IX064.08) is not permissible and leads to unforeseeable results. This applies to the operand area in the master (MODMAST input DATA) as well as to the operand area in the slave (MODMAST input ADDR).

Please note that the forbidden operands can also be reached by a valid start address and a length NB (e.g. DATA = ADR(E064_00) and NB = 8 or ADDR = 400_{HEX} and NB = 8). Such combinations of addresses and lengths must be absolutely avoided.

Caution:
Writing accesses to inputs must be avoided.

When writing via MODBUS to inputs of a 07 KT 97 which acts as a MODBUS slave, the written values are immediately overwritten by the "true" input values. When the 07 KT 97 acting as MODBUS master reads via MODBUS and the data are stored to inputs, the values which were read are also immediately overwritten again by the "true" input values.

Caution:
When writing via MODBUS to constants of the 07 KT 97 which acts as a MODBUS slave, the written values are first only applied to the running operation. When the 07 KT 97 acting as MODBUS master reads via MODBUS and the data are stored to constants, the values which were read are also first only applied to the running operation.

In order to overwrite constants permanently, they must be additionally saved in the Flash memory. Otherwise the original values of the project are loaded after the supply voltage is switched on once again.

Operands (symbolic)	**Operands (IEC)**	**MODBUS address (HEX)**	**Operand description**
E000_00 : E061_15	%IX000.00 : %IX061.15	0000_{HEX} : $03DF_{HEX}$	Binary inputs, CS31 bus
E062_00 : E063_15	%IX062.00 : %IX063.15	$03E0_{HEX}$: $03FF_{HEX}$	Binary inputs, local
E064_00 : E064_07	%IX064.00 : %IX064.07	0400_{HEX} : 0407_{HEX}	Binary inputs, local (binary access to EW006_00..EW006_07)
E065_00 : E099_15	%IX065.00 : %IX099.15	0410_{HEX} : $063F_{HEX}$	Binary inputs, central expansion (reserved)
E100_00 : E163_15	%IX100.00 : %IX163.15	0640_{HEX} : $0A3F_{HEX}$	Binary inputs, 2nd non-central expansion (reserved)
E200_00 : E255_15	%IX200.00 : %IX255.15	$0C80_{HEX}$: $0FFF_{HEX}$	Binary inputs, 3rd non-central expansion (reserved)
E256_00 : E263_15	%IX256.00 : %IX263.15	no direct access	Binary inputs, 3rd non-central expansion (reserved)
	%IX1.1.0000.0 : %IX1.1792.15	no direct access	Binary inputs, PROFIBUS line 1
	%IX2.1.0000.0 : %IX2.1792.15	no direct access	Binary inputs, PROFIBUS line 2

Operands (symbolic)	Operands (IEC)	MODBUS address (HEX)	Operand description
A000_00 : A061_15	%QX000.00 : %QX061.15	1000_{HEX} : $13DF_{HEX}$	Binary outputs, CS31 bus
A062_00 : A063_15	%QX062.00 : %QX063.15	$13E0_{HEX}$: $13FF_{HEX}$	Binary outputs, local
A064_00 : A099_15	%QX064.00 : %QX099.15	1400_{HEX} : $163F_{HEX}$	Binary outputs, central expansion (reserved)
A100_00 : A163_15	%QX100.00 : %QX163.15	1640_{HEX} : $1A3F_{HEX}$	Binary outputs, 2nd non-central expansion (reserved)
A200_00 : A255_15	%QX200.00 : %QX255.15	$1C80_{HEX}$: $1FFF_{HEX}$	Binary outputs, 3rd non-central expansion (reserved)
A256_00 : A263_15	%QX256.00 : %QX263.15	no direct access	Binary outputs, 3rd non-central expansion (reserved)
	%QX1.1.0000.0 : %QX1.1792.15	no direct access	Binary outputs, PROFIBUS line 1
	%QX2.1.0000.0 : %QX2.1792.15	no direct access	Binary outputs, PROFIBUS line 2

Operands (symbolic)	Operands (IEC)	MODBUS address (HEX)	Operand description
M000_00 : M254_15	%MX0000.00 : %MX0254.15	2000_{HEX} : $2FEF_{HEX}$	Binary flags
M255_00 : M255_15	%MX0255.00 : %MX0255.15	$2FF0_{HEX}$: $2FFF_{HEX}$	Binary flags (system)
M256_00 : M279_15	%MX0256.00 : %MX0279.15	no direct access	Binary flags (system) (reserved)
M280_00 : M511_15	%MX0256.00 : %MX0511.15	no direct access	Binary flags (system)
S000_00 : S255_15	%MX5000.00 : %MX5255.15	3000_{HEX} : $3FFF_{HEX}$	Steps

Operands (symbolic)	Operands (IEC)	MODBUS address (HEX)	Operand description
EW000_00 : EW005_15	%IW1000.00 : %IW1005.15	0000_{HEX} : $005F_{HEX}$	Analog inputs, CS31 bus
EW006_00 : EW006_07	%IW1006.00 : %IW1006.07	0060_{HEX} : 0067_{HEX}	Analog inputs, local
EW007_00 : EW007_14	%IW1007.00 : %IW1007.14	0070_{HEX} : $007E_{HEX}$	Analog inputs (reserved)
EW007_15	%IW1007.15	$007F_{HEX}$	Analog input, status word CS31 bus
EW008_00 : EW015_15	%IW1008.00 : %IW1015.15	0080_{HEX} : $00FF_{HEX}$	Analog inputs, CS31 bus
EW016_00 : EW034_15	%IW1016.00 : %IW1034.15	0100_{HEX} : $022F_{HEX}$	Analog inputs, central expansion (reserved)
EW100_00 : EW107_15	%IW1100.00 : %IW1107.15	0640_{HEX} : $06BF_{HEX}$	Analog inputs, 1st non-central expansion (reserved)
EW200_00 : EW207_15	%IW1200.00 : %IW1207.15	$0C80_{HEX}$: $0CFF_{HEX}$	Analog inputs, 2nd non-central expansion (reserved)
	%IW1.0000 : %IW1.1792	no direct access	Analog inputs, PROFIBUS line 1
	%IW2.0000 : %IW2.1792	no direct access	Analog inputs, PROFIBUS line 2

Operands (symbolic)	Operands (IEC)	MODBUS address (HEX)	Operand description
AW000_00 : AW005_15	%QW1000.00 : %QW1005.15	1000_{HEX} : $105F_{HEX}$	Analog outputs, CS31 bus
AW006_00 : AW006_03	%QW1006.00 : %QW1006.03	1060_{HEX} : 1063_{HEX}	Analog outputs, local
AW007_00 : AW007_15	%QW1007.00 : %QW1007.15	1070_{HEX} : $107F_{HEX}$	Analog outputs (reserved)
AW008_00 : AW015_15	%QW1008.00 : %QW1015.15	1080_{HEX} : $10FF_{HEX}$	Analog outputs, CS31 bus
AW016_00 : AW034_15	%QW1016.00 : %QW1034.15	1100_{HEX} : $122F_{HEX}$	Analog outputs, central expansion (reserved)
AW100_00 : AW107_15	%QW1100.00 : %QW1107.15	1640_{HEX} : $16BF_{HEX}$	Analog outputs, 1st non-central expansion (reserved)
AW200_00 : AW207_15	%QW1200.00 : %QW1207.15	$1C80_{HEX}$: $1CFF_{HEX}$	Analog outputs, 2nd non-central expansion (reserved)
	%QW1.0000 : %QW1.1792	no direct access	Analog outputs, PROFIBUS line 1
	%QW2.0000 : %QW2.1792	no direct access	Analog outputs, PROFIBUS line 2

Operands (symbolic)	Operands (IEC)	MODBUS address (HEX)	Operand description
MW000_00 : MW253_15	%MW1000.00 : %MW1253.15	2000_{HEX} : $2FDF_{HEX}$	Word flags
MW254_00 : MW255_15	%MW1254.00 : %MW1255.15	$2FE0_{HEX}$: $2FFF_{HEX}$	Word flags (error message)
MW256_00 : MW259_15	%MX1256.00 : %MX1259.15	no direct access	Word flags (system) (reserved)
MW260_00 : MW511_15	%MW1260.00 : %MW1511.15	no direct access	Word flags (user area)
KW000_00 : KW000_15	%MW3000.00 : %MW3000.15	3000_{HEX} : $300F_{HEX}$	Word constants (system)
KW001_00 : KW079_15	%MW3001.00 : %MW3079.15	3010_{HEX} : $34FF_{HEX}$	Word constants
KW080_00 : KW089_15	%MW3080.00 : %MW3089.15	3500_{HEX} : $359F_{HEX}$	Word constants (system)
MD000_00 : MD063_15	%MD2000.00 : %MD2063.15	4000_{HEX} : $43FF_{HEX}$	Double word flags
KD000_00	%MD4000.00	5000_{HEX}	Double word constant (system)
KD000_01 : KD023_15	%MD4000.01 : %MD4023.15	5001_{HEX} : $517F_{HEX}$	Double word constants

MODBUS example program:

07 KT 97 configured with COM1 as MODBUS master and COM2 as MODBUS slave

Task: The bits %M0.1...%MX0.8 at the slave (07 KT 97 COM2) are to be read with a maximum frequency and stored in %MX0.9...%MX0.15 at the master (07 KT 97 COM1).

Note: No serial programming interface is available during the running operation because both serial interfaces of the basic unit are simultaneously operated as MODBUS interfaces. The online operation of the 907 AC 1131 is only possible via ARCNET. Alternatively the program can be loaded into two separate controllers, each having only one serial interface operated as a MODBUS interface. In this case the online access of the 907 AC 1131 can also be performed via the other serial interface which is not used as MODBUS interface.

In the program header various variables are first declared and initialized. This way of implementation increases the clarity of the program and additionally simplifies the incorporation of changes possibly required at a later time. Alternatively the values can also be declared directly at the block inputs.

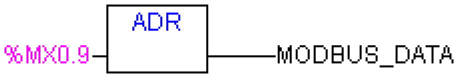
```
PROGRAM MODBUSEXAMPLE
VAR
    MODINIT_EN:       ARRAY[1..2] OF BOOL:= TRUE, TRUE;    (*Release MODINT*)
    MODMAST_EN:       ARRAY[1..2] OF BOOL:= FALSE, FALSE;  (*Release MODMAST*)
    MODBUS_IDENT:     ARRAY[1..2] OF INT  := 100, 101;     (*COM1 master, COM2 slave with Addr. 1*)
    MODBUS_BAUD:      INT                 := 9600;         (*9600 baud*)
    MODBUS_PTY:       INT                 := 0;            (*no parity*)
    MODBUS_STOP:      INT                 := 1;            (*1 stop bit*)
    MODBUS_RTSCTRL:   BOOL                := FALSE;        (*no RTS control*)
    MODBUS_TLS:       INT                 := 0;            (*0 ms carrier lead time*)
    MODBUS_CDLY:      INT                 := 0;            (*0 ms carrier delay time*)
    MODBUS_CHTO:      INT                 := 3;            (*3 ms character timeout*)
    MODBUS_TIMEOUT:   INT                 := 1000;         (*1 s telegram timeout*)
    MODBUS_FCT:       INT                 := 1;            (*function "read n bits"*)
    MODBUS_ADDR:      INT                 := 16#2001;      (*read starting from %MX0.1*)
    MODBUS_NB:        INT                 := 7;            (*7 bits*)
    MODBUS_DATA:      DWORD;                               (*memory address for bits read*)

    MODINIT_ERROR:    ARRAY[1..2] OF INT;                  (*storage of errors*)
    MODMAST_ERROR:    ARRAY[1..2] OF INT;                  (*storage of errors*)


    MODINIT_COM1:     MODINIT;                             (*instance*)
    MODINIT_COM2:     MODINIT;                             (*instance*)
    MODMAST_COM1:     MODMAST;                             (*instance*)
END_VAR
```

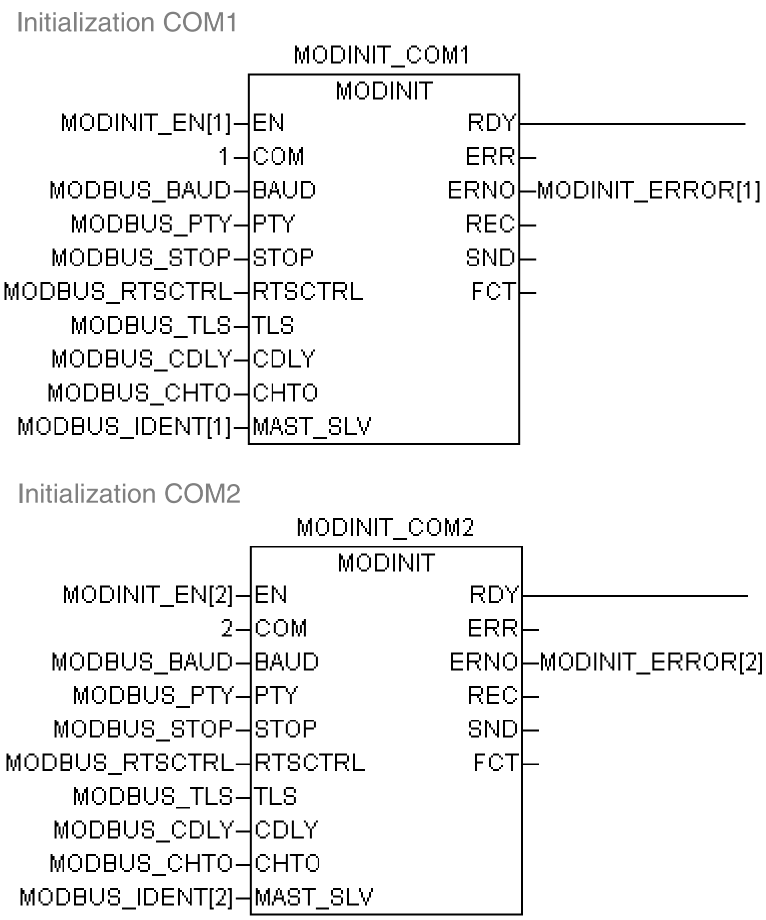
Example for a variable declaration and initialization

The initialization of the operand address in the master (MODBUS_DATA) is performed in the instruction part of the program by means of the ADR operator.



Pre-assignment of the operand address for the operating mode MODBUS master

The initialization of the serial interface is performed via instances of the MODINIT block with the help of the variables defined in the program header. A communication via the serial interfaces is only possible, if both interfaces are set to the same transmission parameters (BAUD – CHTO). The EN inputs of the MODINIT type instances are internally pre-initialized with FALSE. Consequently an intialization of the MODINIT_EN variables with TRUE causes a FALSE→TRUE edge at the EN input of the instances during the first program cycle and a permanent value of TRUE during the following cycles. This causes an initialization of the interfaces during the first cycle and an operation as MODBUS interfaces during the following cycles.



Initialization of the MODBUS interfaces

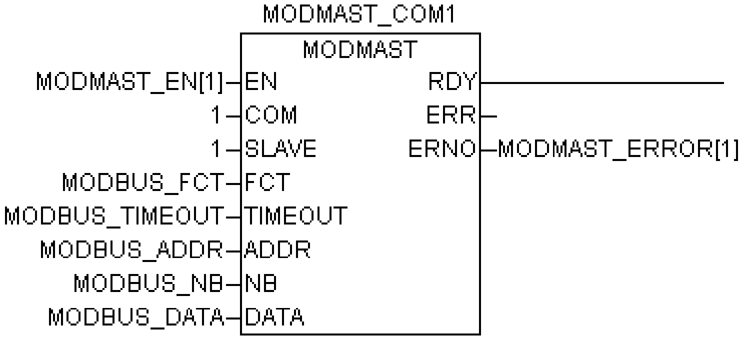
In the next step the MODBUS master functionality for COM1 must be realized with the help of the MODMAST block. In case of a FALSE → TRUE edge at the input EN, the master sends a request telegram to a slave which corresponds to the remaining block inputs. The RDY output of the MODMAST block is pre-initialized with TRUE. During processing of the telegram RDY is FALSE. RDY becomes TRUE after the processing was successful or when an error occurs. Therefore the RDY signal can be directly used to activate the block via the input EN. Because sending of telegrams is only possible after the corresponding interface was successfully initialized to be a MODBUS master interface, it is sensible to additionally interconnect the release with the RDY output of the associated MIDINIT block.

Release of master function COM1

Release signal for the MODMAST block

The MODMAST block is now activated as follows:

Master function COM1



Realization of the MODBUS master functionality

This example program does not consider the processing of possible errors. The current error messages are only stored to the ARRAYs MODINIT_ERROR and MODMAST_ERROR. For real projects, an additional application specific error reaction should be programmed.

Caution:

If the block is activated as described in the example program mentioned above, no jobs are executed anymore if at least one input of the MODMAST block is invalid. In order to initiate a new job, the MODMAST block requires a FALSE→TRUE edge at the release input EN. If the values at the inputs are correct, the block sets RDY = FALSE and executes the job. After the job is processed, the block sets RDY = TRUE and in case of an error it sets ERR = TRUE. Using the activation shown above, this generates a rising edge at the input EN. But if one of the inputs experiences an error, the RDY output stays TRUE and ERR is set. Thus, no new edge is generated at the input EN. See also the description of the block MODMAST.

2 System technology of the internal couplers

2.1 The ARCNET coupler

2.1.1 Brief overview

Short description and field of application

(**A**ttached **R**esource **C**omputer **Net**work)

ARCNET is a system used for data transfer in local networks. The ARCNET protocol is based on the token passing principle. An identifier (token) which is forwarded from one station to the next guarantees that at every point in time only one station can start a data transfer (collision-free data transfer). The order in which the individual stations are called is automatically adapted to the present situation on the network. This means that the network is automatically reconfigured every time a station is switched on or off.

Realizing the idea of decentralized intelligence, the ARCNET SST is used for a quick and easy networking of AC31 basic units of the series 90. It is possible to connect AC31 input and output modules or further pre-processing devices (AC31 basic units, converters etc.) to these basic units. The coupler is integrated in the basic unit 07 KT 97 (optional). So, the communication interface stays free and can be used to connect a RCOM coupler 07 KP 90 for data remote transfer, for example.

Data in short

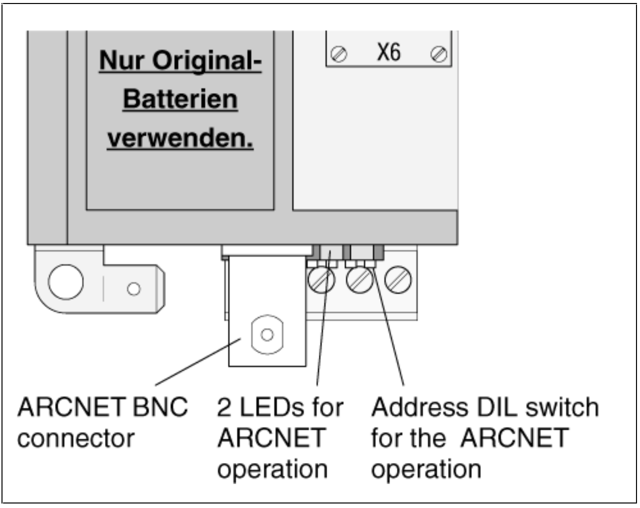
- Very short access times obtained by a high data transfer rate of 2.5 Mbit/s.
- Up to 430 m bus length (line: 300 m) and up to 6.5 km when repeaters / star couplers are used.
- Token master, up to 9 participants without repeaters and up to 255 participants when repeaters/ star couplers are used.
- Programming via ARCNET bus, see 907 AC 1131.
- Attachment plug X4 built as BNC socket for the connection of coaxial cables.
- Signaling green LED (BS) Operating condition "controller active", i.e. the PLC is sending via the ARCNET.
- green LED (TX) Operating condition "transmit active", i.e. the PLC is sending via the ARCNET.
- Electrical isolation against supply voltage, inputs and outputs and against interfaces COM1/COM2.
- The ARCNET coupler is supplied from the internal 24 V DC power supply.
- The DIL switch for ARCNET address setting is located on the outside of the module.

2.1.2 Connection and transfer media

Attachment plug:

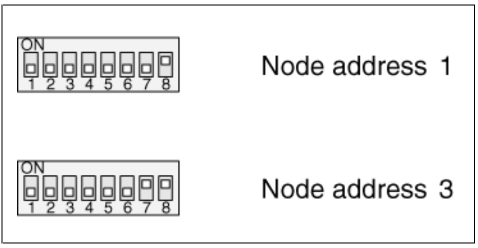
The ARCNET coupler is designed as a bus with BNC connectors for coaxial cables. The ARCNET bus is earthed inside the module by a capacitor. As an EMC measure and for protection against dangerous contact voltages, the bus has to be earthed directly at a central place.

The 07 KT 97 is connected to the ARCNET by means of a BNC T-connector on plug X4.



Setting the participant address:

The DIL switch for setting the participant address is located below the plug X4. The address setting is performed binary. The following allocation applies:
DIL 8 = 2^0 = 1
DIL 7 = 2^1 = 2
DIL 6 = 2^2 = 4
:
DIL 1 = 2^7 = 128



Transfer media:

Bus cable:
Cable RG 62 A/U:
e.g. Lapp Kabel, Stuttgart
Phone: +49 711 7838-0

Bus cable plugs:
BNC connector 75 Ω : Order number B-9005
BNC T-connector 75 Ω : Order number B-9083
BNC terminator 93 Ω : Order number B-9093

Rufenach Vertriebs-GmbH
Phone: +49 6221 8443-0
Fax: +49 6221 8443-99

Repeaters, active star couplers:

Supplier:
APEX Automatisierungstechnik GmbH, Braunschweig
Phone: +49 531 37040

Coaxial repeater (2 port amplifier):
010214005/HKXKX

Active star coupler, consisting of:
Modular 8 port amplifier (basic unit with controller and power supply):
010214001/modH8P

Coaxial port plug-in unit for star connection (LAND):
010214002/MHKXP

Coaxial port plug-in unit for bus connection (HIT):
010214003/MHKXP

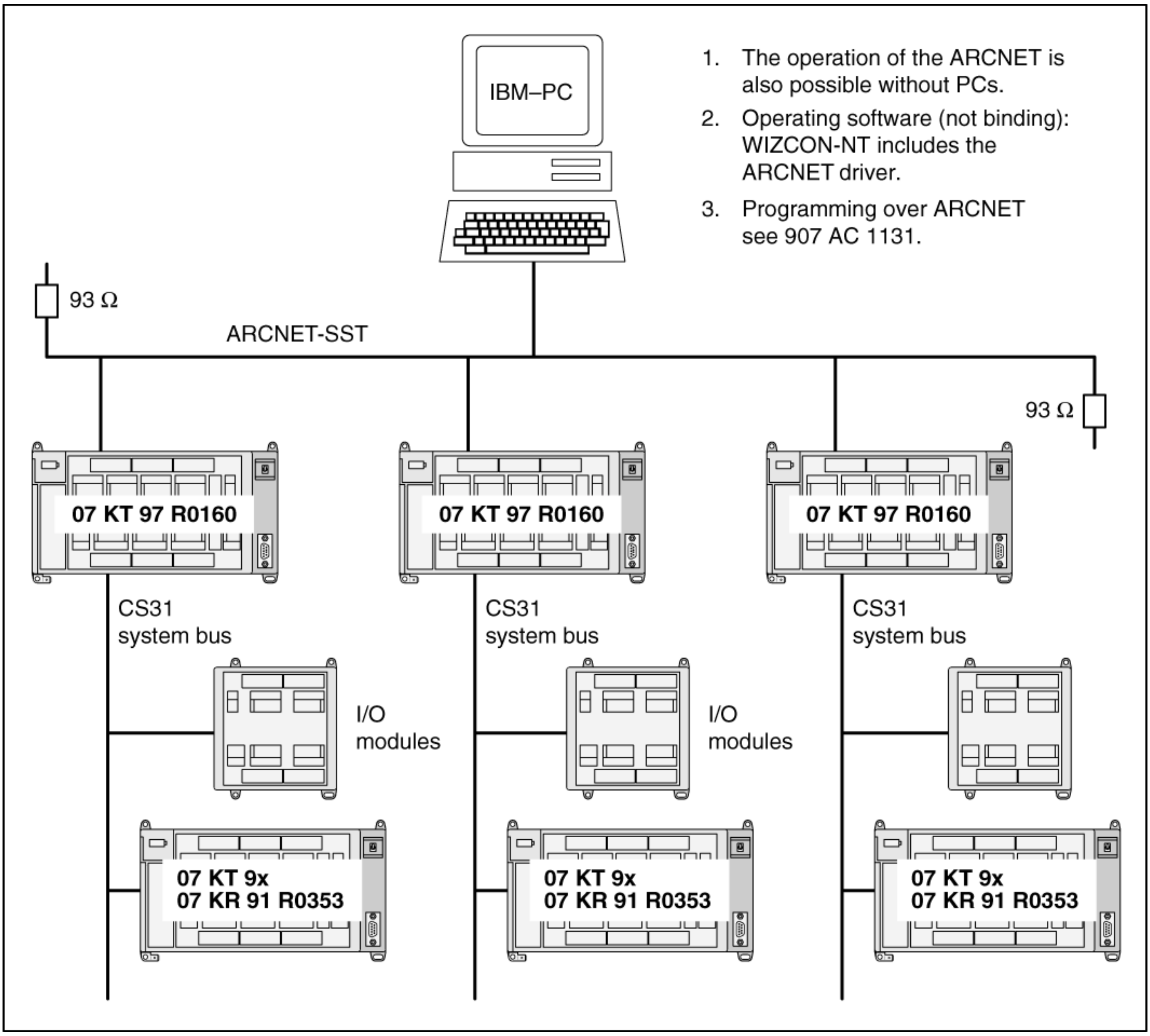
Optical fibre converter, plug:
Refer to "Conversion from coaxial cable to optical fibre" in section 2.1.3.

2.1.3 Possibilities for networking

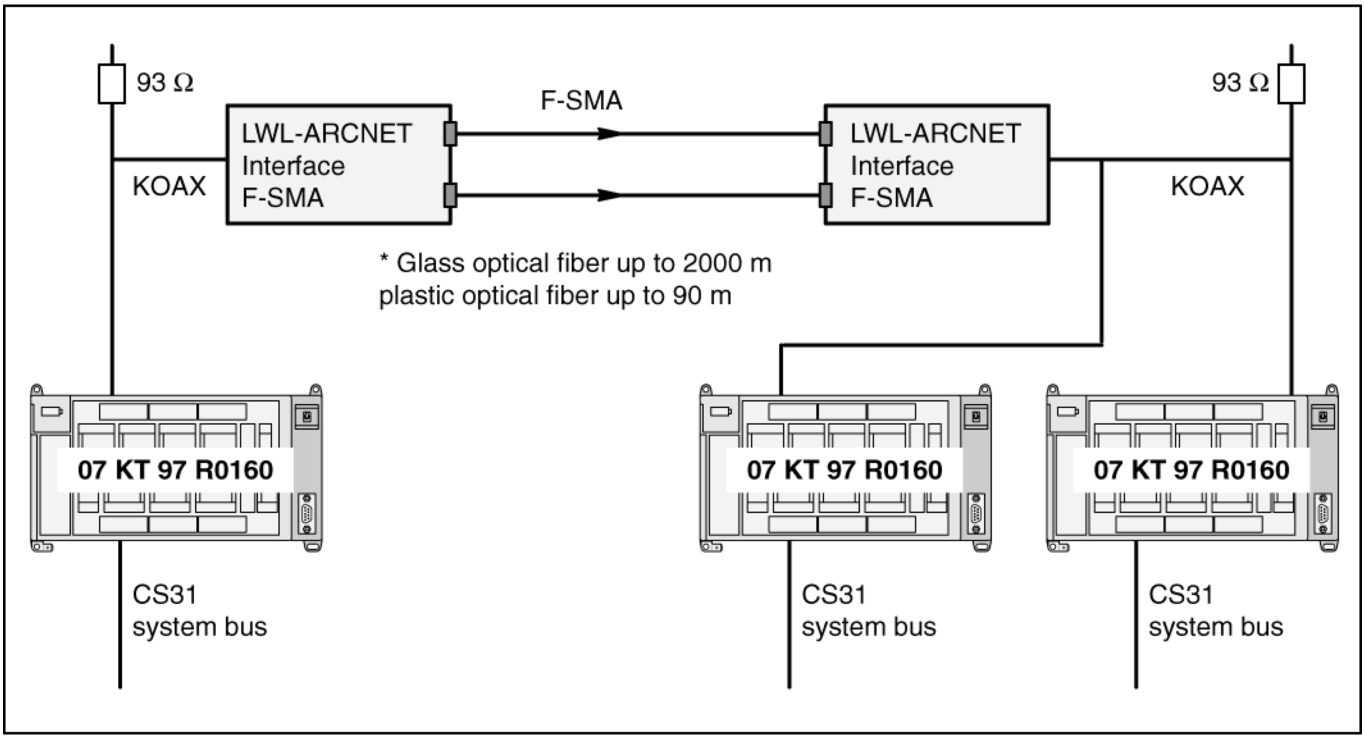
Linear ARCNET is the simplest configuration. Here, the coaxial cable (RG-62 A/U, 93 Ω) is laid from one station to the next. The connection to the individual stations is done using a BNC T-connector. The ends of the cabling must be terminated by terminating resistors of 93 Ω.

Linear ARCNET

Linear ARCNET connects the individual stations directly to each other, without inserted distributors. The individual stations are connected to the network via T-connectors. Terminating resistors must be connected to both ends of the cabling. A maximum of 8 stations can be connected to the network. The maximum length is 300 m. By means of an active hub (active distributor), an additional segment can be connected to the end of a linearly cabled segment.



Conversion from coaxial cable to optical fibre



Supplier of optical fibre parts:

Harting Elektronik GmbH
D-32339 Espelkamp
Phone: +49 5772 47-263
Fax: + 49 5772 47-461

LWL-ARCNET interface for glass fibre:
Order number 20 40 002 3711

F-SMA connector 50/125:
Order number 20 10 125 1212

F-SMA connector 50/125 glass:
Order number 20 20.050 1022

Other cables, for example for underground wiring up to approx. 5.32/m (on inquiry).

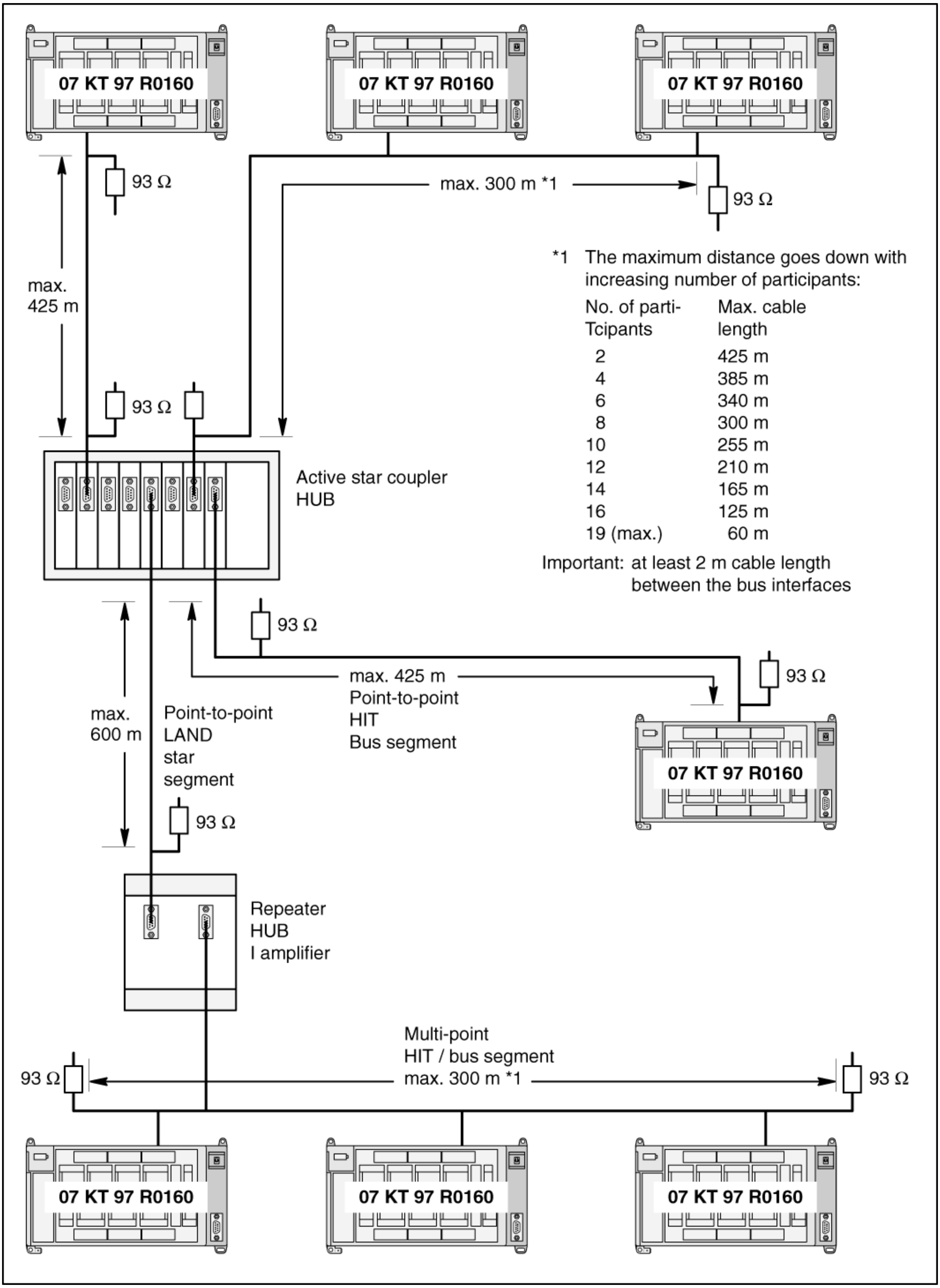
For technical questions concerning optical fibre equipment, please consult Harting (phone: +49 5772 47-225).

Connecting optical fibres requires practice as a craftsman. Company for connecting optical fibres:

Magronic in Munich
Phone: +49 89 3838-650

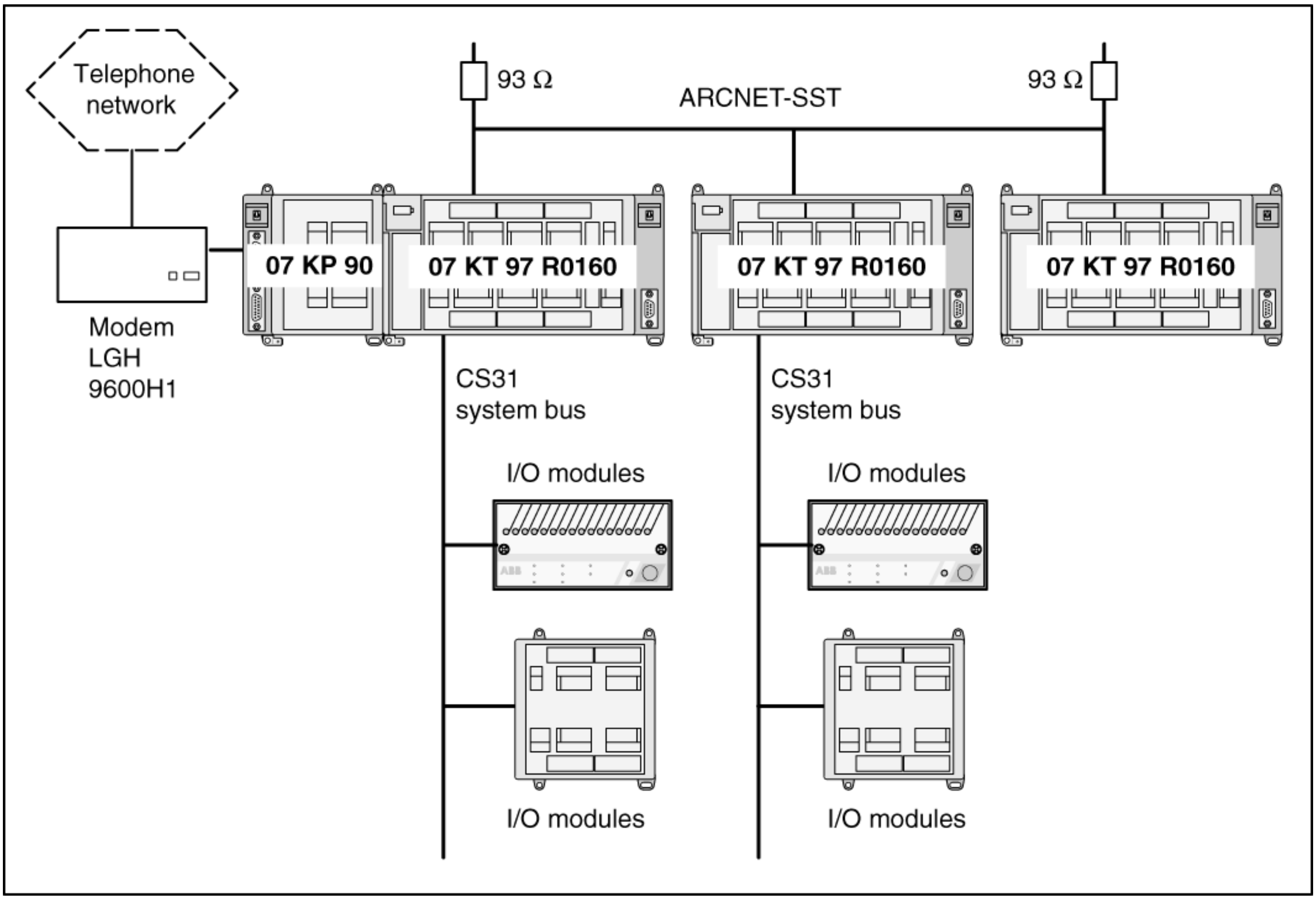
Linear ARCNET, expanded by active distributors (active hubs)

Active hubs amplify the incoming signals and therefore stabilize the network and enable longer distances in the network. The active hub uncouples the individual connections. As a result, the failure of one connection does not lead to a complete failure of the entire network. The maximum length of the network is 6.5 km. A maximum of 255 stations can be connected to the network.


07 KT 97 R0160
07 KT 97 R0160
07 KT 97 R0160
93 Ω
max. 300 m *1
93 Ω
max.
425 m
*1 The maximum distance goes down with increasing number of participants:
No. of parti-Tcipants | Max. cable length
2 | 425 m
4 | 385 m
6 | 340 m
8 | 300 m
10 | 255 m
12 | 210 m
14 | 165 m
16 | 125 m
19 (max.) | 60 m
Important: at least 2 m cable length between the bus interfaces
93 Ω
Active star coupler
HUB
93 Ω
max. 425 m
Point-to-point
HIT
Bus segment
93 Ω
07 KT 97 R0160
max.
600 m
Point-to-point
LAND
star
segment
93 Ω
Repeater
HUB
I amplifier
Multi-point
HIT / bus segment
max. 300 m *1
93 Ω
93 Ω
07 KT 97 R0160
07 KT 97 R0160
07 KT 97 R0160

Stations with an additional coupler:

Telephone network connection for 3 substations



2.1.4 ARCNET implementation

The operating system has three protocols implemented which distinguish by a DIN identifier in the 1st byte of the user data:

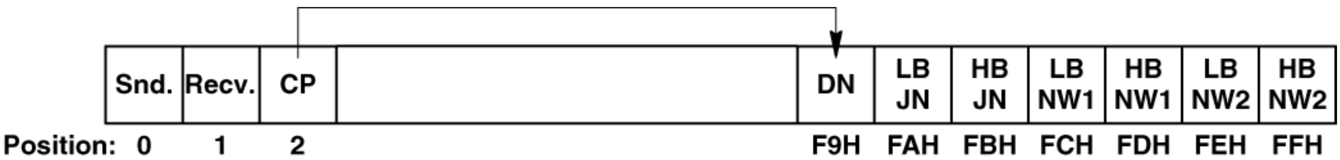
- Programming and test protocol, DIN identifier 4F $_{hex}$ and 3F $_{hex}$
- MMC protocol, DIN identifier 5F $_{hex}$
- Data transfer from station to station, DIN identifier 7F $_{hex}$

The protocols for MMC and programming and test are directly handled by the operating system. Both protocols initiate a response telegram to the sender. The address of the sender is retrieved from the received telegram.

The protocol for the data transfer from station to station must be planned by the PLC user. For this purpose, the function blocks AINIT, ASEND1, ASEND4, ASEND16 and AREC are available.

The coupler is initialized exclusively for sending and receiving of short data packages. Following this, a data package for the data transfer from station to station has the following format:

- 3 bytes of check data: sender, receiver, CP
- max. 253 bytes of user data, consisting of:
  1 byte DIN identifier + 2 bytes job no. + max. 125 words user data



Explanation:

Snd.	Station address of the sender
Recv.	Station address of the receiver
CP	Continuous Pointer; contains the position ($F9_H$) of the 1. byte of user data
DN	DIN identifier ($7F_H$)
LB JN	Low byte of the job number
HB JN	High byte of the job number
LB NW1	Low byte of user data word 1
HB NW1	High byte of user data word 1
LB NW2	Low byte of user data word 2
HB NW2	High byte of user data word 2

Reading the data package from the ARCNET coupler (data reception) is performed interrupt controlled. The interrupt routine stores the data package in the receiving buffer. The AREC block reads the receiving buffer and supplies the data in the user memory. The blocks ASEND1, ASEND4 or ASEND16 serve for the sending direction. All the blocks store the data packages in the sending buffer. From here, the data are transferred interrupt controlled. The block ASEND1 sends a data package to another station. The block ASEND4 sends a data package to up to 4 other stations simultaneously. The block ASEND16 sends a data package to up to 16 other stations simultaneously.

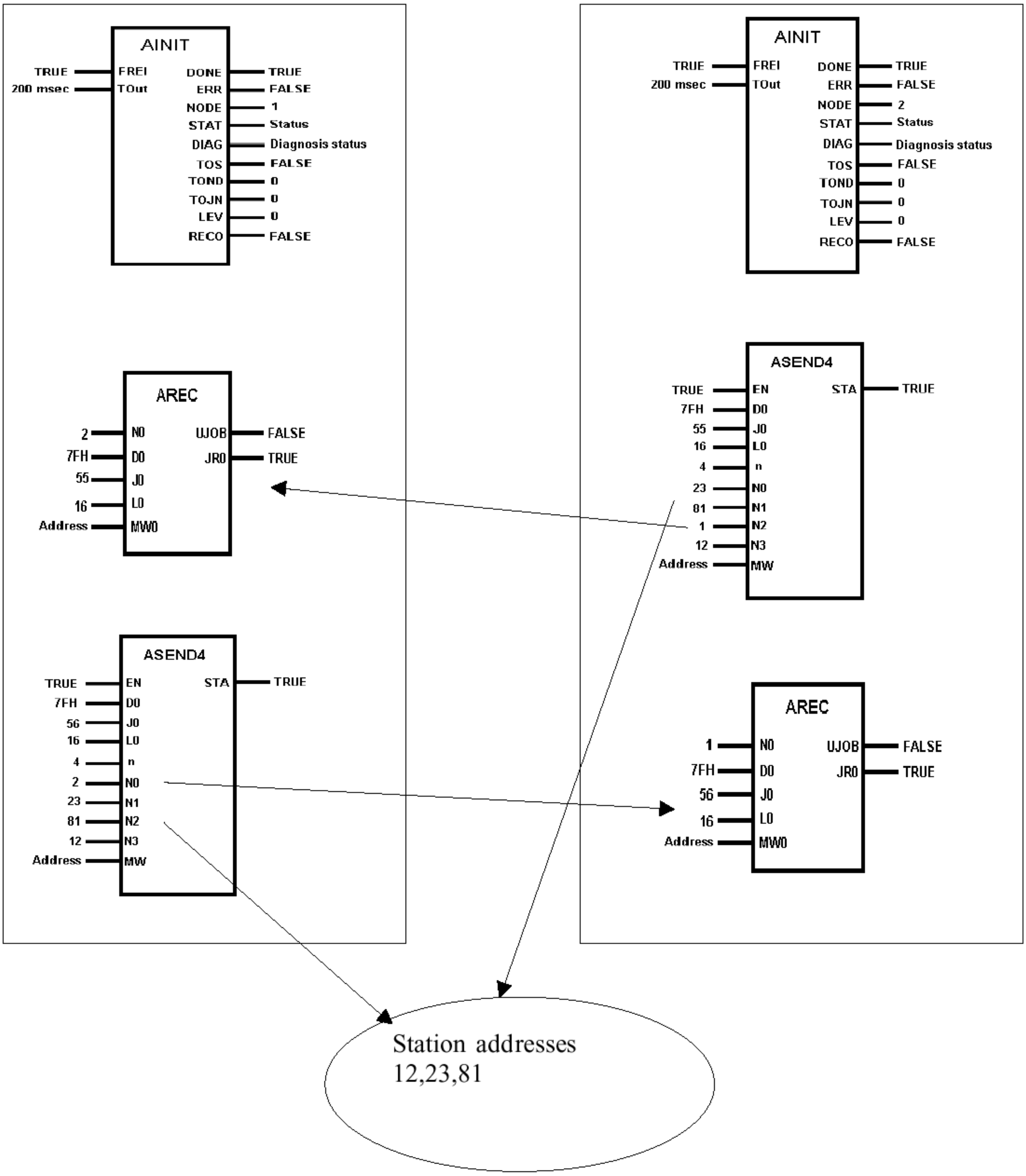
2.1.5 Example of a program structure for the data transfer from station to station

As mentioned above, this data transfer protocol occurs exclusively with the DIN identifier $7F_H$.

Caution:

In each station just one AINIT block is required. All planned blocks AINIT, AREC, ASEND1, ASEND4 and ASEND16 are to be planned in one task. It must be ensured that all implemented AREC blocks are processed during every cycle and always in the same order. The sending blocks ASEND1, ASEND4 and ASEND16 require a correctly planned receiving block AREC in an other station for the transfer (see also the descriptions of the blocks AINIT, AREC, ASEND1, ASEND4, ASEND16). The station address for the ARCNET operation is set on a DIL switch.

Below a planning example for station addresses 1 and 2 is shown. Station 1 is sending to station 2 and to the addresses 12, 23 and 81. Station 2 is sending to station 1 and also to the addresses 12, 23 and 81. This example makes clear that the receiving block AREC must have the same DIN identifier and job number as it is entered at the corresponding ASEND4 block. The data length is not checked on the reception.


AINIT
TRUE FREI DONE TRUE
200 msec TOut ERR FALSE
NODE 1
STAT Status
DIAG Diagnosis status
TOS FALSE
TOND 0
TOJN 0
LEV 0
RECO FALSE
AREC
2 N0 UJOB FALSE
7FH D0 JR0 TRUE
55 J0
16 L0
Address MW0
ASEND4
TRUE EN STA TRUE
7FH D0
56 J0
16 L0
4 n
2 N0
23 N1
81 N2
12 N3
Address MW
AINIT
TRUE FREI DONE TRUE
200 msec TOut ERR FALSE
NODE 2
STAT Status
DIAG Diagnosis status
TOS FALSE
TOND 0
TOJN 0
LEV 0
RECO FALSE
ASEND4
TRUE EN STA TRUE
7FH D0
55 J0
16 L0
4 n
23 N0
81 N1
1 N2
12 N3
Address MW
AREC
1 N0 UJOB FALSE
7FH D0 JR0 TRUE
56 J0
16 L0
Address MW0
Station addresses
12,23,81

2.2 The PROFIBUS-DP coupler

2.2.1 Brief overview

Fundamental properties and fields of application

PROFIBUS-DP is designed for the rapid transfer of process data between central controller modules (such as PLC or PC) and decentralized field modules (such as I/O modules, drives and valves) in the field level. The communication occurs mainly cyclic. For intelligent field modules, additionally acyclic communication functions are required for parameterizing, diagnosis and alarm handling during the running cyclic data transfer.

During normal operation a central controller (DP master of class 1) cyclically reads the input data of the connected decentralized I/O modules (DP slaves) and sends output data to them. Per slave a maximum of 244 bytes of input and output data can be transferred in one cycle.

Apart from the user data traffic the PROFIBUS-DP provides extensive commissioning and diagnosis functions. The present diagnosis messages of all slave modules are summarized in the master. This enables a quick localization of errors.

Using PROFIBUS-DP both mono master and multi master systems can be realized. Multi master systems are built of functional independent subsystems which each consist of one master and a portion of the slaves which are integrated in the entire system. Normal bus masters cannot exchange information with each other.

PROFIBUS-DP distinguishes two types of masters. The class 1 master carries out the cyclic transfer of user data with the slaves and supplies the data. The class 1 master can be called by a class 2 master using specific functions. These functions are restricted to support and diagnosis information, for example the interrogation of diagnosis information of the slaves or the master itself. Thus a class 2 master is also considered as a programming and diagnosis device.

PROFIBUS-DP uses the hybrid bus access method. This guarantees on the one hand that complex automation devices used as DP masters obtain the opportunity to handle their communication tasks in defined time intervals. On the other hand it enables the cyclic and real-time related data exchange between the master and peripheral devices (DP slaves). The assigned slave modules on the bus are handled by the master one after the other using the polling operation mode. So, each slave becomes active only after it was requested by the master. This avoids simultaneous access to the bus.

The hybrid access method of the PROFIBUS allows a combined operation of multiple bus masters and even the mixed operation of PROFIBUS-DP and PROFIBUS-FMS in one bus section. This, however, assumes the correct configuration of the bus system and the unique assignment of the slave modules to the masters.

The characteristic properties of a PROFIBUS-DP module are documented in the form of an electronic data sheet (modules master data file, GSD file). The modules master data describe the characteristics of a module type completely and clearly in a manufacturer independent format. Using this defined file format strongly simplifies the planning of a PROFIBUS-DP system. Usually the GSD files are provided by the module's manufacturer. In addition the PROFIBUS user organization makes the GSD files of numerous PROFIBUS-DP modules available for a free of charge download via internet in their GSD library. The address of the PROFIBUS user organization (PNO) is: http://www.profibus.com.

Features:

Modes of operation:

- Operation as DP master of class 1 or as DP slave, as desired.
- Automatic setting of the operation mode during configuration.

Transmission technique:

- RS485, potential separated, insulation voltage up to 850 V.
- Twisted pair cable or optical fibre as a medium for the bus.
- Transfer rate from 9.6 kbit/s up to 12 Mbit/s.
- Bus length up to 1200 m at 9.6 kbit/s and up to 100 m at 12 Mbit/s.
- Up to 32 participants (master and slave modules) without repeater and up to 126 participants on one bus with repeater.
- 9-pole SUB-D socket for bus connection; assignment according to standard.
- Integrated repeater controller.
- Automatic baud rate detection when in operating mode DP slave.

Communication:

- Cyclic user data transfer between DP master and DP slave.
- Incyclic data transfer from master to master.
- Check of the slave configuration.
- Efficient diagnosis functions, 3 graduated diagnosis messaging levels.
- Synchronization of inputs and/or outputs via control commands.
- Per slave a maximum of 244 bytes of input and output data possible.

Protection functions:

- Message transfer with hamming distance HD = 4.
- Errors in the data transfer are detected by the CRC check and cause the repetition of the telegram.
- Access protection for inputs and outputs of the slaves.
- Incorrect parameter settings are avoided because participants with faulty parameters are not included in the user data operation.
- A failure of a participant is registered in the master and indicated via a common diagnosis.
- Response monitoring for DP slaves. Failure of a transmission line is detected and causes the outputs to be switched off.

Status indication via 4 LEDs

- READY (yellow): The coupler is ready for operation.
- RUN (green): Status of configuration and communication.
- ERROR (red): Error on PROFIBUS.
- STATUS (yellow): Data exchange

2.2.2 Connection and transfer media

Attachment plug for the bus cable:

9-pole SUB-D connector

Assignment:

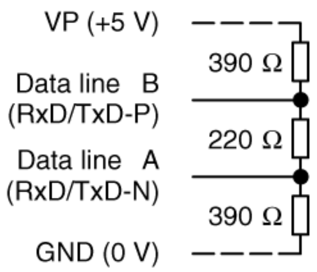
Pin No.		Signal Meaning
1	Shield	Shielding, protective earth
2	not used	
3	RxD/TxD-P	Receiving / sending line, positive
4	CBTR-P	Control signal for repeater (optional)
5	DGND	Reference potential for data lines and +5V
6	VP	+5V, supply voltage for bus termination resistors
7	not used	
8	RxD/TxD-N	Receiving / sending line, negative
9	CNTR-N	Control signal for repeater, negative (optional)

Supplier:

e.g. Erbic® BUS Interface Connector
ERNI Elektroapparate GmbH
Seestraße 9
D-72099 Adelberg
Phone: +49 7166 50 176
Fax : +49 7166 50 103
Internet: http://www.erni.com

Bus terminating resistors:

The line ends (of the bus segments) have to be terminated using bus terminating resistors according to the drawing below. The bus terminating resistors are usually placed inside the bus connector.



Configuration of the resistors

Symbol

Bus terminating resistors connected to the line ends

Bus cable:

Type:	Twisted pair cable (shielded)
Characteristic impedance:	135...165 Ω
Cable capacity (distributed capacitance):	< 30 pF/m
Diameter of line cores (copper):	≥ 0,64 mm
Cross-section of line cores:	≥ 0,34 mm^2
Line resistance per core:	≤ 55 Ω/km
Loop resistance (resistance of 2 cores):	≤ 110 Ω/km

Supplier:

e.g. UNITRONIC® BUS
U.I. LAPP GmbH
Schulze-Delitzsch-Straße 25
D-70565 Stuttgart
Phone: +49 711 7838 01
Fax: +49 711 7838 264
Internet: http://www.lappkabel.de

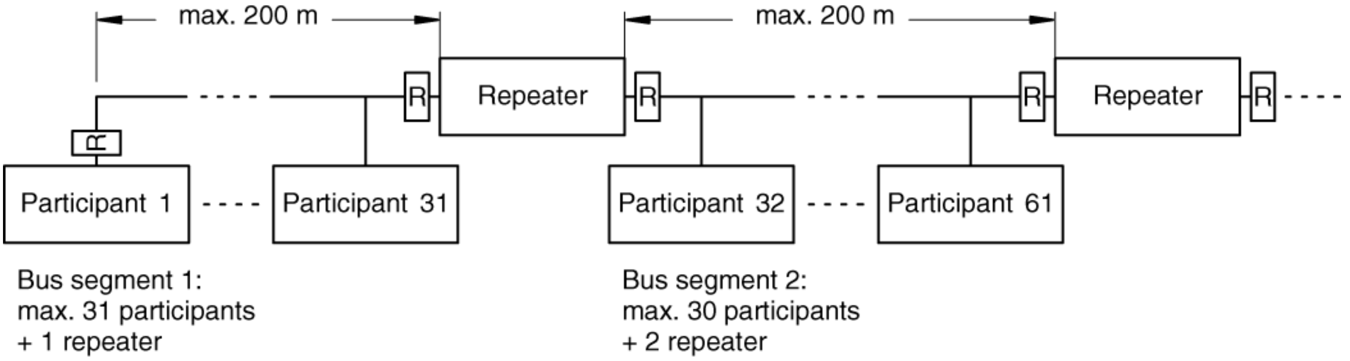
Maximum line lengths (bus segment):

1200 m	at a transfer rate of 9.6 / 19.2 / 93.75 kbit/s
1000 m	at a transfer rate of 187.5 kbit/s
400 m	at a transfer rate of 500 kbit/s
200 m	at a transfer rate of 1500 kbit/s
100 m	at a transfer rate of 3000 / 6000 / 12000 kbit/s

Branch lines are generally permissible for baud rates of up to 1500 kbit/s. But in fact they should be avoided for transmission rates higher than 500 kbit/s.

Repeaters:

One bus segment can have up to 32 participants. Using repeaters a system can be expanded to up to 127 participants. Repeaters are also required for longer transfer lines. Please note that a repeater's load to the bus segment is the same as the load of a normal participant. The sum of normal participants and repeaters inside of one bus segment must not exceed 32.



Principle example for a PROFIBUS-DP system with repeaters (1500 kbit baud rate)

2.2.3 Possibilities for networking

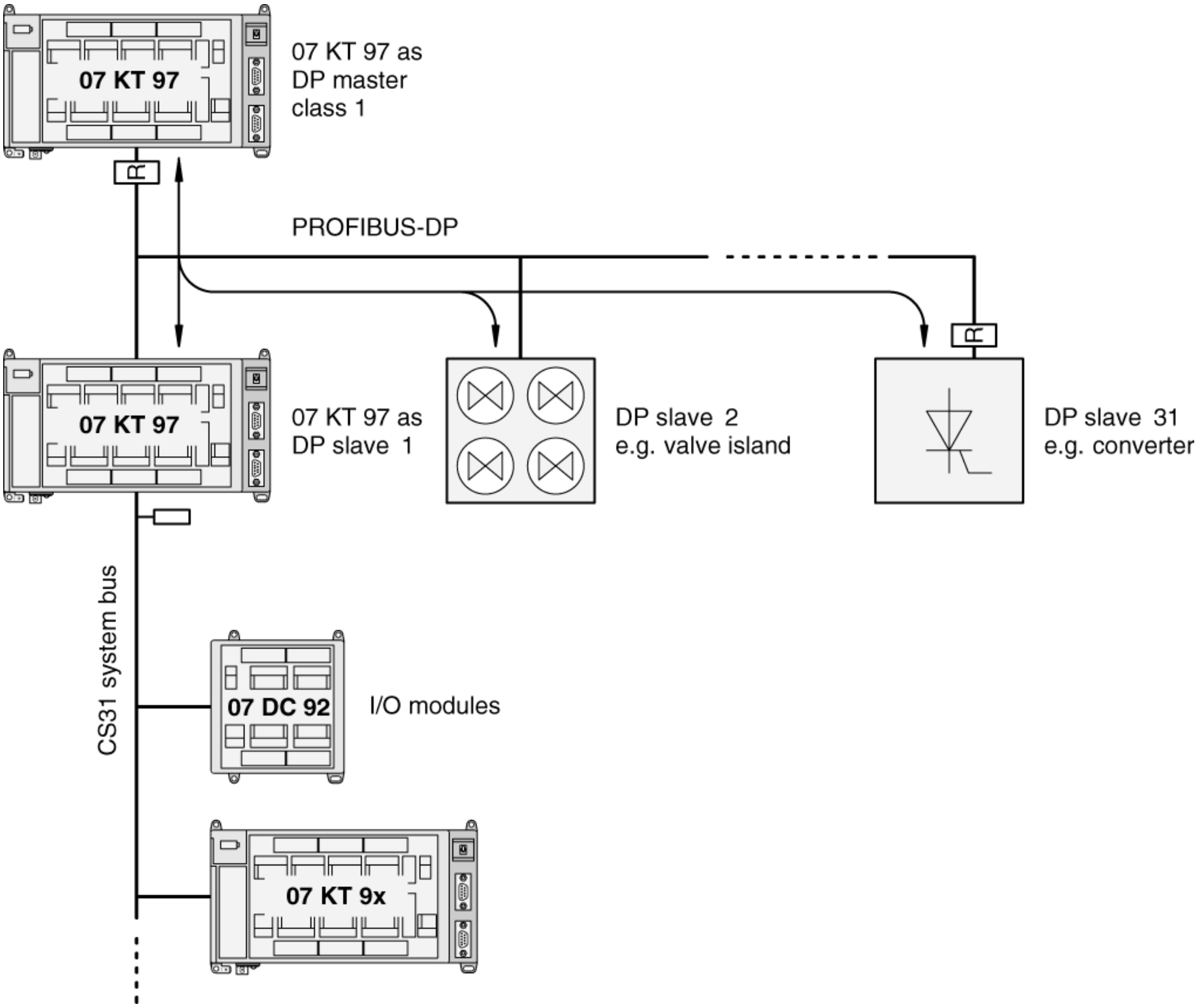
The PROFIBUS coupler is connected to the bus via the bus connector (9-pole SUB-D socket) on the upper left-hand side of the controller housing. As an EMC measure and for protection against dangerous contact voltages, the shield of the bus line must be connected to protective earth outside the housing.

Single master system

The single master system is the simplest version of a PROFIBUS network. It consists of a class 1 DP master and one or more DP slaves. Up to 31 DP slaves can be connected to the bus without using a repeater. If the number of bus segments is increased by means of repeaters, up to 126 DP slaves can be handled. The line ends of the bus segments have to be terminated using bus terminating resistors.

The DP master of class 1 is able to:

1. Parameterize DP slaves (e.g. time supervision, bus interchange).
2. Configure DP slaves (e.g. type / number of channels).
3. Read input and output data of the DP slaves.
4. Write output data of the DP slaves.
5. Read diagnosis data of the DP slaves.
6. Send control commands to the DP slaves (e.g. freezing input signals).



Single master system example

Multi master system:

A PROFIBUS network containing several DP masters is called a multi master system. Up to 32 participants (DP masters and DP slaves) can be operated on one bus segment. Using repeaters the system can be expanded to up to 127 participants. In a multi master system no data exchange between the DP masters is performed. The entire system is divided into logical subsystems inside of which one DP master communicates with the assigned DP slaves. Each DP slave can be assigned to only one DP master. The master has unlimited access to its assigned slaves while all other masters on the bus can only read the input and output data of these slaves.

All the DP masters of class 1 (normal bus master, here 07 KT 97) and the DP masters of class 2 (commissioning device, typically a PC) can read the input and output data of all slaves.

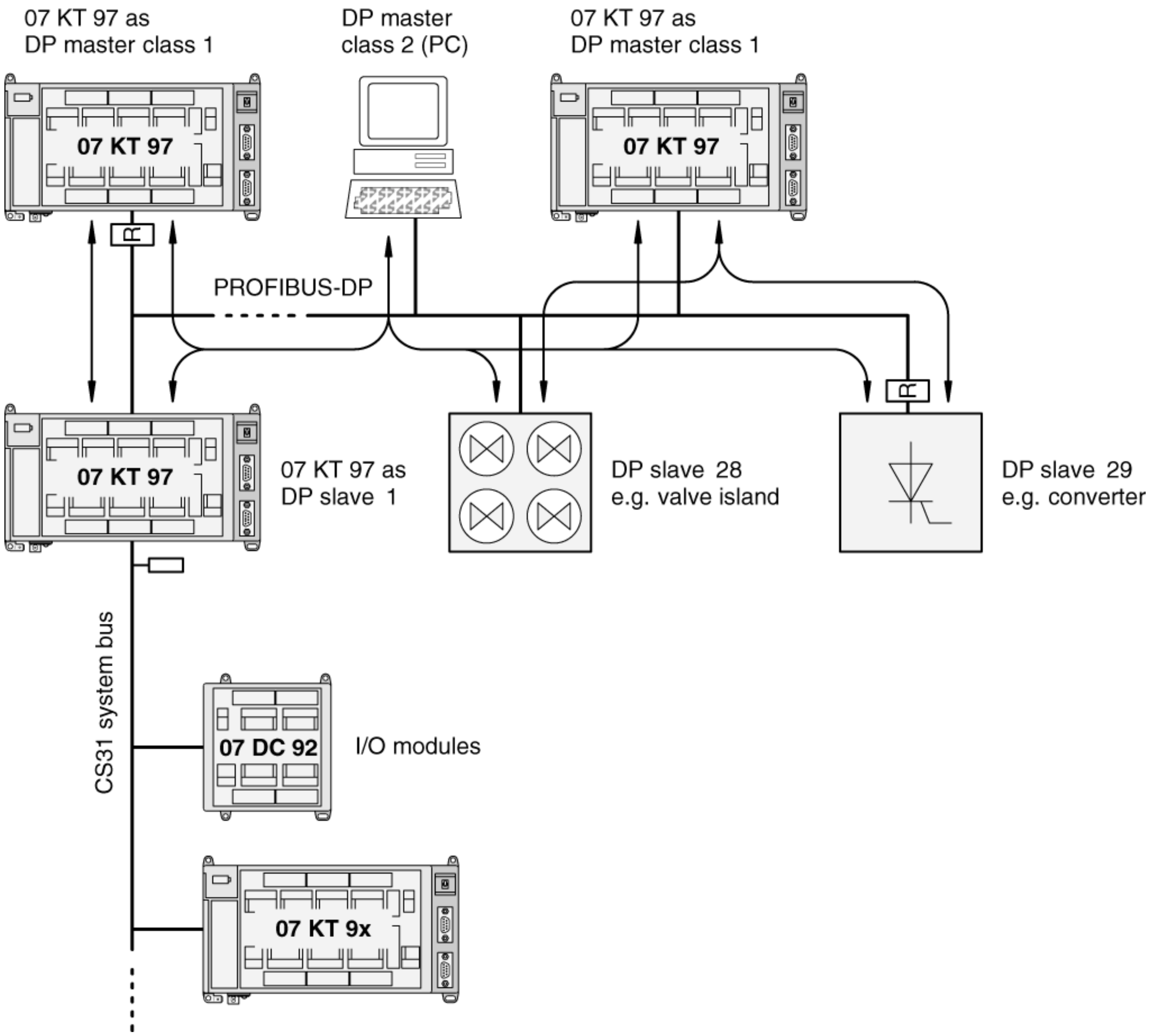
Additionally the DP masters of class 1 and class 2 have the following access possibilities to their assigned DP slaves. They are able to:

- Parameterize DP slaves (e.g. timing supervision, bus interchange).
- Configure DP slaves (e.g. type / number of channels).

- Write output data of the DP slaves.
- Read diagnosis data of the DP slaves.
- Send control commands to the DP slaves (e.g. freezing input signals).

A DP master of class 2 is additionally able to:

- Read and write configuration data of the class 1 DP masters.
- Read configuration data of the DP slaves.
- Read diagnosis data of the class 1 DP masters.
- Read out the diagnosis data of the DP slaves assigned to the respective DP master.



Multi-master system example

2.2.4 PROFIBUS implementation

Configuration:

Correct configuration data are an assumption for the correct function of the PROFIBUS coupler. The configuration data are a definite element of a project. In 907 AC 1131 they are created selecting Resources | Control configuration (refer to the 907 AC 1131 documentation). The coupler configuration data are transferred to the controller together with the actual program. The operating system of the PLC first carries out a validity check and then transfers the configuration data to the coupler. If an error occurs here, it is displayed via the FKx error flags and LEDs. In addition, the configuration data can be saved together with the user program in the Flash memory of the control. Configuration data which are saved in the Flash memory are automatically loaded the next time the power is switched on.

The PROFIBUS coupler selects its operating mode according to the configuration data and signalizes that it is ready for operation. The coupler’s readiness for operation is indicated by cyclic flashing of the LED RDY which is located beside the bus connection.

Running operation:

The PROFIBUS-DP protocol is automatically handled by the coupler and the operating system of the control. The coupler is only active on the bus if it was correctly initialized before and if the user program runs. No function blocks are required for the cyclic exchange of process data via PROFIBUS-DP. Special PROFIBUS-DP functions can be realized using the function blocks of the PROFIBUS library.

Operating mode DP-Slave:

In the DP-Slave operating mode the coupler only responds to diagnosis requests of the DP master if the user program is not active. By setting the Station_not_ready bit in the response telegram, the coupler signalizes to the requesting master that it is not ready for data exchange.

After the user program is started the coupler signalizes that it is ready on the bus and consequently it is added to the cyclic telegram traffic by the DP master. When the master could take it successfully into operation the LED RDY indicates this by lighting continuously. The coupler stores the data received from the master in the operand areas defined in the configuration and sends the configured process data to the master. Some special acyclic PROFIBUS-DP functions are handled by the coupler automatically while others require a corresponding function block (in preparation).

If the user program is stopped, the coupler logs off from the bus.

Operating mode DP-Master:

In the DP-Master operating mode the coupler starts the communication via PROFIBUS-DP after the user program is started and attempts to initialize the planned slaves. After a slave is initialized successfully it is added to the cyclic process data exchange. The LED RDY lights up continuously after at least one slave was successfully taken into operation. The cyclic I/O data exchange with the slaves is performed automatically. Special acyclic PROFIBUS-DP functions require a corresponding function block (in preparation).

If the user program is stopped, the coupler shuts down the PROFIBUS system in a controlled manner.

Error diagnosis:

PROFIBUS-DP communication errors are generally indicated by the red LED ERR beside the bus connector. A malfunction of the PROFIBUS driver or the coupler itself is indicated via the FK error flags and the corresponding LEDs. Furthermore the PROFIBUS library provides different function blocks which allow a detailed error diagnosis. In both operating modes DP master and DP slave, the state of the coupler itself can be read out via a corresponding block. Additionally in the DP master operating mode the detailed PROFIBUS diagnosis of an individual slave and a system diagnosis overview can be retrieved.

Function blocks:

Libraries

- PROFI_40.LIB and PROFIBUS_S90_V41.LIB

General

- PROFI_INFO: Reading of coupler information

DP master

- DPM_STAT: Reading the coupler status
- DPM_SLVDIAG: Reading the detailed PROFIBUS diagnosis of a DP slave
- DPM_SYSDIAG: Reading of the system diagnosis

DP slave

- DPS_STAT: Reading the coupler status

2.2.5 Planning examples

DP slave:

Below a planning example for the control 07 KT 97 R120 used as DP slave is showed. The PLC receives cyclically one word from the master and shall output the value at the analog output AW06,00. The values read at the binary inputs E62,00...E62,07 are to be reported to the master. The bus address no. 2 of the slave is predetermined.

Before the actual user program can be built, the behaviour of the coupler on the bus must be determined. For this purpose the hardware configuration part of 907 AC 1131 must be called by selecting Resources|Control configuration.

Selecting Attach subelement | DP slave opens a window that displays all DP slaves whose GSD files are available. Select 07 KT 97-DPS for the module name here. The module number to be selected depends on the slot where the PROFIBUS coupler is installed. This is slot 1 for the 07 KT 97 R0120 or slot 2 for the 07 KT 97 R162. Thus, in this example the module number 1 must be entered. After you have confirmed the selection with OK the coupler properties for this project must be determined.

As basic parameters first the start addresses of the input and output data in the operand area must be preset. The entry field for the diagnosis address is without function. Because it is a word that shall be received from the master, the input address %IW1.0 is selected in this example. The values of the binary inputs E62,00...E62,07 are packed in one byte for the transfer to the master. Hence, %QB1.0 is selected for the output address.

According to the default setting the value 2 must now be entered for the station address. Now a station name can be assigned optionally. It is not necessary to enter the transfer rate because this is preset by the DP master and automatically detected by the coupler.

Now the I/O configuration is to be determined. Please note that the PLC is considered as a simple I/O device for the configuration. The designation of the I/O modules is always performed from the view of the process. "1 word output" and "1 byte input" must be selected for the I/O configuration because the control receives via PROFIBUS-DP one word that it has to put out to the process and it transfers one byte to the master that it reads from the process. The coupler configuration is now completed. Furthermore symbolic variable designators for the I/O data can be assigned optionally. Thus, the following hardware configuration is obtained:

```
⊟-----Hardware-Konfiguration
      ⊟-----07 KT 97-DPS [DP-Slave]
            ⊟-----1  word output consistent (IEC-Input)
            :     └-------Analogwert AT %IW1.0: WORD ;
            ⊟-----1  byte input consistent (IEC-Output)
                  └-------Binaere_Eingaenge_gepackt AT %QB1.0: BYTE ;
```

Example: Hardware configuration 07 KT 97 as DP slave

Now the user program must be made which realizes the required behaviour of the control. First the inputs and outputs of the control can not be called using their symbolic names but only as IEC operators. So the IEC operators %IX62.0 - %IX62.7 correspond to the binary inputs E62,00...E62,07 and %QW1006.0 corresponds to the analog output AW6,0. Alternatively the binary inputs E62,00...E62,07 can also be called via %IB62.0 packed in a byte (E62,08...E62,15 = %IB125). If you are not familiar with this representation, symbolic variables can be assigned to the inputs and outputs by declaring global variables using Resources | Global variables. These global variables can also be accessed anywhere in the program. The declaration could be as follows:

```
VAR_GLOBAL
(*Declaration of the analog output AW6,0*)
    AW6_0       AT %QW1006.0:   INT:

(*Declaration of binary inputs E62,00...E62,07 as single inputs*)
    E62_00      AT %IX62.0:     BOOL;
    E62_01      AT %IX62.1:     BOOL;
    E62_02      AT %IX62.2:     BOOL;
    E62_03      AT %IX62.3:     BOOL;
    E62_04      AT %IX62.4:     BOOL;
    E62_05      AT %IX62.5:     BOOL;
    E62_06      AT %IX62.6:     BOOL;
    E62_07      AT %IX62.7:     BOOL;

(*Declaration of binary inputs E62,00...E62,07 as group input*)
    E62_00_07   AT %IB62.0:     BYTE;
END_VAR
```

Example of a global variable declaration

In order to output the analog value received via PROFIBUS-DP to AW6,0 the following assignment is required in the program:

Analog value----AW6_0

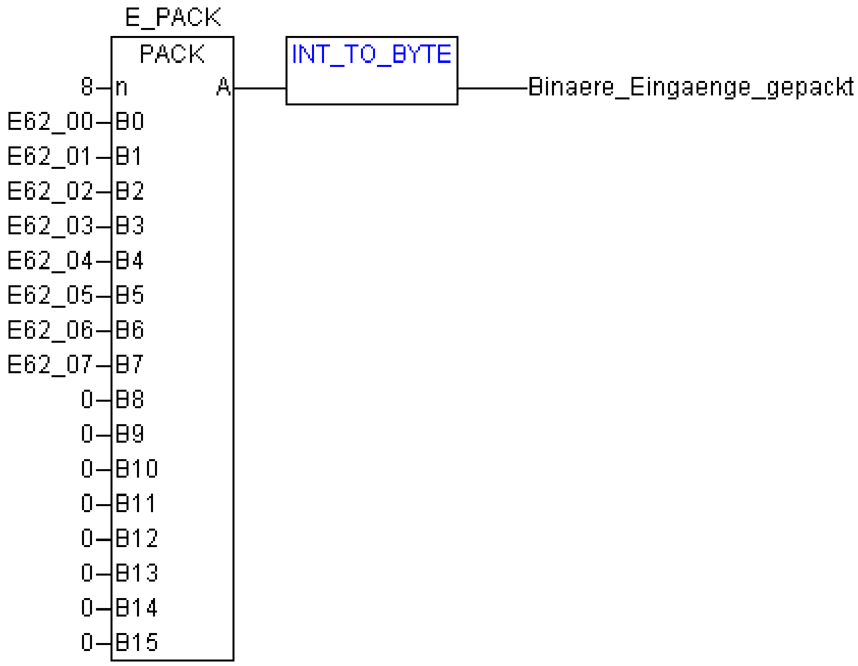
Alternatively and without a previous variable declaration also the following assignment can be used:

%IW1.0----%QW1006.0

For transferring the binary inputs E62,00...E62,07 to the master also different possibilities exist. Below only those two possibilities using the symbolic designators are represented. If the inputs are called as a group via E62_00_07 only the following assignment is required to transfer the values to the master:

E62_00_07----Binaere_Eingaenge_gepackt

If the binary inputs are called individually they must be packed in one byte prior to the assignment or transfer:

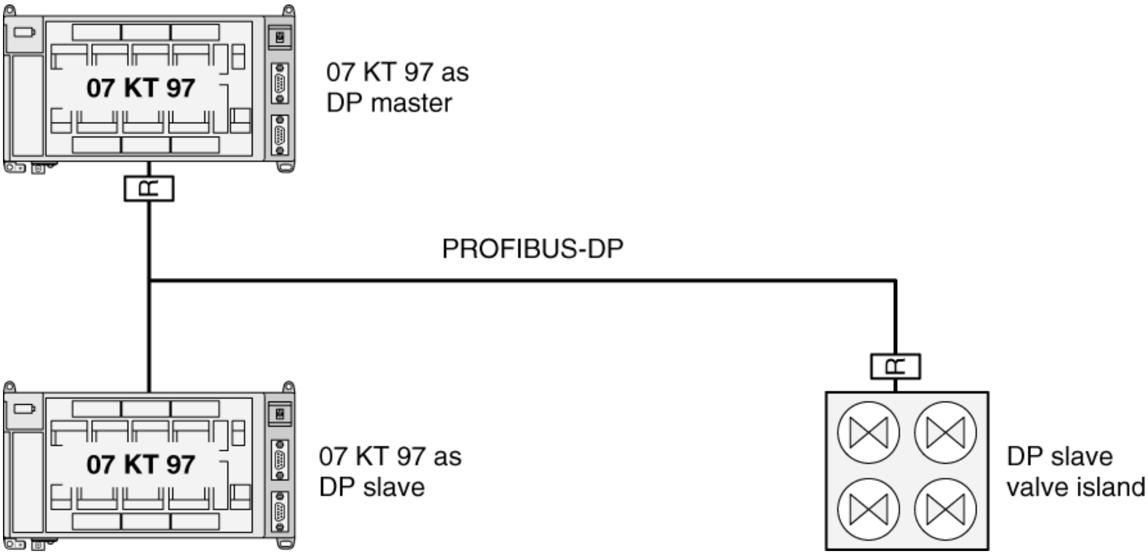


Example: Packing of binary inputs and transfer via PROFIBUS-DP

In addition, the state of the coupler can be read out with the help of the function block DPS_STAT.

DP master:

Below, a planning example for the 07 KT 97 used as DP master is showed. Beside the 07 KT 97 which is used as the DP master, one valve island and the control mentioned in the example above are present on the bus. So the resulting configuration is as follows:



PROFIBUS-DP example configuration

The DP master control sends the difference resulting from the values which are measured at the local analog inputs EW6,00 and EW6,01 to the 07 KT 97 used as DP slave and receives eight binary values packed in one byte from the DP slave. The DP master PLC sends the received binary values as control signals to the valve island (e.g. FESTO type 02 FB 13, 4 valves). The following bus addresses are agreed: 1 for the 07 KT 97 used as DP master, 2 for the 07 KT 97 used as DP slave, 3 for the valve island.

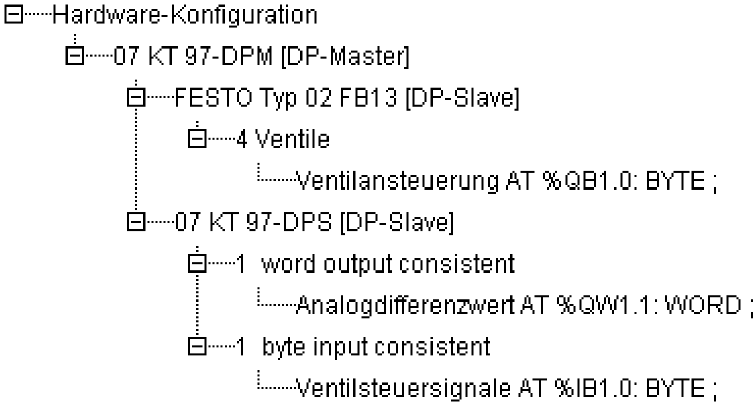
Prior to taking the master PLC into operation it makes sense to configure the slaves first. Proceed as shown in the example above to configure the 07 KT 97 as DP slave. Refer to the manufacturer's documentation for the configuration procedure of the valve island. Such devices normally have DIP switches where the bus address can be set. The I/O configuration is usually fixed and the transfer rate is automatically detected.

After the DP slaves have been configured the configuration of the master can be started. By selecting Resources | Control configuration the hardware configuration part of the 907 AC 1131 is accessed.

Selecting Attach subelement | DP master opens a window that displays all DP masters whose GSD files are available. Select 07 KT 97-DPM for the module name here. The module number to be selected depends on the slot where the PROFIBUS coupler is installed. This is slot 1 for the 07 KT 97 R0120 or slot 2 for the 07 KT 97 R162. In this example, the 07 KT 97 R0120 shall be used, so the module number 1 must be entered. After you have confirmed the selection with OK the coupler properties for this project must be determined.

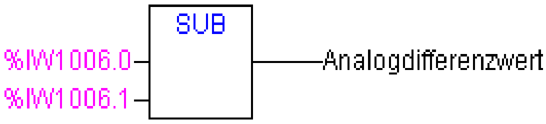
As basic parameters an individual name and the bus address can be assigned. The highest station address is automatically calculated. The entry field for the diagnosis address and the selection "Automatic start" are without function. The behaviour of the master in case of an error (e.g. failure of a slave) is determined using the parameter "Auto Clear Mode". If the Auto Clear mode is activated the outputs of all slaves are switched into the safe state in case of an error. The master itself performs a controlled shutdown of the PROFIBUS-DP system and attempts once again to take all slaves into operation. If the Auto Clear mode is deactivated the master continues the cyclic data exchange when an error occurs. The occurred error is only reported. Using "Automatic addresses" you can choose whether the IEC operand address shall be generated automatically or entered by hand. It is recommended that the address administration is performed automatically by the configurator. In this application example it is not required to set the group properties. Refer to the 907 AC 1131 documentation for information about this. The baud rate should be set manually to the highest possible value supported by all participants on the bus. The selected baud rate must also be set for all modules in the system which do not have an automatic baud rate detection. The calculation of the bus parameters should be left to the configurator. Incorrect manual inputs can lead to an undefined system behaviour. The setting of the coupler properties is now complete.

Using Attach DP slave the coupler must now be told which slaves to handle. First 07 KT 97-DPS is selected from the appearing list and confirmed with OK. This procedure is repeated for the valve island (here FESTO type 02 FB 13). In the next step the properties of the slaves must be announced to the coupler. For explanations about the individual parameters, refer to the configurator description (see 907 AC 1131 documentation). Here, only the bus address assignment and the setting of the I/O configuration shall be described. According to the defaults, the station address 2 is to be set for the 07 KT 97 used as slave. "1 word output" and "1 byte input" must be selected for the I/O configuration because the DP master sends one word to and receives one byte from the slave. The valve island must be set to station address 3 and "4 valves". If now symbolic variable designators are also entered, the following hardware configuration results:



Example: Hardware configuration 07 KT 97 as DP master

Now, the actual user program can be built. The calculation of the difference resulting from the analog signals at the inputs EW6,00 and EW6,01 is performed via the corresponding IEC operators.



Example: Transferring the difference resulting from the analog signals

The signals for the control of the valve island received by the 07 KT 97 used as DP slave can be immediately forwarded to the valve island using the assignment

Valve control signals----Valve activation

In addition, the condition of the coupler can be read out with the help of the function block DPM_STAT. If necessary, the detailed diagnosis of the DP slaves can be polled using function blocks of the type DPM_SLVDIAG and a survey over the conditions of all slaves in the system can be polled using the function block DPM_SYSDIAG.

2.2.6 PROFIBUS error messages

The PROFIBUS error messages are listed in section "Error messages of the internal couplers".

2.2.7 Further information

Standardization

EN 50170

DIN 19245 part 1

DIN 19245 part 3

Important address

PROFIBUS Nutzerorganisation e.V. (PNO)

Haid-und-Neu-Straße 7

D-76131 Karlsruhe

Phone:	+49 721 9658 590
Fax:	+49 721 9658 589
Internet:	http://www.profibus.com

Terms, definitions and abbreviations used

PROFIBUS-DP	**PRO**CESS **FI**ELD**BUS** - DECENTRALIZED PERIPHERY
DPM1	DP master (class 1), normal bus master
DPM2	DP master (class 2), commissioning device
DPS	DP slave
GSD	Modules master data
DPV1	Guideline for functional expansions of PROFIBUS-DP
PNO	PROFIBUS user organization (**P**ROFIBUS **N**utzer **O**rganisation)

2.3 Error messages of the internal couplers

Range of validity

The described error messages are valid for all internal couplers, except ARCNET.

Structure of the error messages of the internal couplers

Error class	Error identifier Dec Hex	Detailed info 1	Detailed info 2	Detailed info 3	Detailed info 4	Detailed info 5
FK2 / FK3 / FK4	200_D $C8_H$ (Error identifier of internal couplers)	ID (Card number with error)	Error class of internal couplers	Cause of error	Additional information	Additional information

Error identifier

All error messages, concerning the internal couplers, have the error identifier 200_D or $C8_H$.

Detailed info 1 – Card number

With all errors, the detailed info 1 contains the card number (ID) of the involved coupler. The only exception is the appearance of an invalid ID (only 1 and 2 are valid) with the configuration, for instance. In this case, the invalid ID is displayed.

Detailed info 2 – Error class

Besides the classification into the global error classes FK1 to FK4, the errors of the internal couplers are classified deeper in the detailed info 2. The following error classes are defined:

Detailed info 2 Dez	 Hex	Error class field bus couplers	
256_D	100_H	Initialization error	(Error when initializing the coupler or the driver in the PLC)
512_D	200_H	Operating system error	(operating system of the coupler reports an error)
768_D	300_H	Task1 status	(Task1 of the coupler has an improper status)
1024_D	400_H	Task2 status	(Task2 of the coupler has an improper status)
1280_D	500_H	Task3 status	(Task3 of the coupler has an improper status)
1536_D	600_H	Task4 status	(Task4 of the coupler has an improper status)
1792_D	700_H	Task5 status	(Task5 of the coupler has an improper status)
2048_D	800_H	Task6 status	(Task6 of the coupler has an improper status)
2304_D	900_H	Task7 status	(Task7 of the coupler has an improper status)
2560_D	$A00_H$	Run-time error	(error of the coupler or driver in the PLC during running operation)
2816_D	$B00_H$	Planning error	(the project contains none or faulty configuration data)
3072_D	$C00_H$	Message error	(Error in the message interface of the coupler)
65535_D	$FFFF_H$	Fatal error	

Detailed info 3 – Cause of errors

In the detailed information 3 the cause of the field bus coupler error is more precisely specified. The following table shows the generally applicable error causes. Operating system errors or task errors are excluded here. The codes of these errors are listed further down.

General

Detailed info 3 Dec	Hex	Cause of error
1_D	1_H	DP RAM of the coupler not accessible
2_D	2_H	Error when testing the coupler watchdog
3_D	3_H	Invalid size of the coupler DP RAM memory
4_D	4_H	Coupler not found
5_D	5_H	Incorrect coupler type (different coupler than on delivery)
6_D	6_H	Incorrect coupler model (different participant type and/or different protocol than installed on delivery)
7_D	7_H	Unknown coupler
8_D	8_H	Heap error (control has not enough memory to apply the required resources)
9_D	9_H	Coupler has no configuration data
10_D	A_H	Life identifier error of the coupler (the coupler did not handle the life identifier)
11_D	B_H	Reset error (coupler performed a reset independently)
12_D	C_H	Reset error (coupler did not carry out the reset request within the predetermined time)
13_D	D_H	Ready error (coupler did not signalize readiness for operation within the predetermined time)
16_D	10_H	Sending mailbox error (coupler did not fetch the request message within the predetermined time)
17_D	11_H	Receiving mailbox error (coupler did not respond to the request message within the predetermined time)
18_D	12_H	The called coupler is not a DP master (the project contains wrong configuration data)
19_D	13_H	The called coupler is not a DP slave (the project contains wrong configuration data)
25_D	19_H	Invalid configuration data
26_D	$1A_H$	Invalid card number (the project contains wrong configuration data)
27_D	$1B_H$	Transfer of the configuration data to the coupler not yet finished
28_D	$1C_H$	Error when writing/erasing of the coupler's Flash memory
67_D	43_H	Error or faulty coupler response when downloading the configuration data (Start_Seq)
68_D	44_H	Error or faulty coupler response when downloading the configuration data (Download)
69_D	45_H	Error or faulty coupler response when downloading the configuration data (End_Seq)

Operating system errors and task errors of the coupler

Detailed info 3 Dec	Hex	Cause of error
1_D	1_H	Task without communication
2_D	2_H	Task in idle state
50_D	32_H	Faulty initialization base
100_D	64_H	UART parity error
101_D	65_H	UART framing error
102_D	66_H	UART overrun error
103_D	67_H	Incorrect number of data
104_D	68_H	Check sum error
105_D	69_H	Timeout
106_D	$6A_H$	Protocol error
107_D	$6B_H$	Data error
108_D	$6C_H$	No acknowledgment

110_D	$6E_H$	Incorrect protocol base
150_D	94_H	Error in message header
151_D	95_H	Incorrect message length
152_D	96_H	Incorrect message command
153_D	97_H	Incorrect message structure
154_D	98_H	Message error
155_D	99_H	Message timeout
160_D	$A0_H$	Incorrect telegram header
161_D	$A1_H$	Incorrect module address in message
162_D	$A2_H$	Incorrect data area identifier in message
163_D	$A3_H$	Incorrect data address in message
164_D	$A4_H$	Incorrect data index in message
165_D	$A5_H$	Incorrect number of data in message
166_D	$A6_H$	Incorrect data type in message
167_D	$A7_H$	Incorrect function in message
170_D	AA_H	Error in message base
200_D	$C8_H$	Task not initialized
201_D	$C9_H$	Task blocked
202_D	CA_H	Segment error
203_D	CB_H	User error
210_D	$D2_H$	Data base error
211_D	$D3_H$	Data base write error
212_D	$D4_H$	Database read error
213_D	$D5_H$	Faulty structure
214_D	$D6_H$	Faulty parameter
215_D	$D7_H$	Faulty configuration
216_D	$D8_H$	Faulty function list
217_D	$D9_H$	System error
220_D	DC_H	System data base error
250_D	FA_H	Task in initialization state
251_D	FB_H	Task is waiting for initialization
252_D	FC_H	Task is activated
253_D	FD_H	Task implemented
254_D	FE_H	Task in configuration state
255_D	FF_H	Task not found

FK2 - Serious errors
Error identifier in MW 254,08/%MW1254.8: 200D   C8H

Error class	Error description	Detailed info 1 in MW 254,09/ %MW1254.9	Detailed info 2 in MW 254,10/ %MW1254.10 Dec Hex	Detailed info 3 in MW 254,11/ %MW1254.11 Dec Hex	Detailed info 4 in MW 254,12/ %MW1254.12	Detailed info 5 in MW 254,13/ %MW1254.13	Further detailed infos in MW 254,14/ %MW1254.14 : MW 254,15/ %MW1254.15
FK2	Error during write/read test on coupler DP RAM memory	ID	256_D 100_H	1_D 1_H	-	-	-
Serious error	Error during dynamic memory allocation	ID	256_D 100_H	8_D 8_H	-	-	-
	Operating system error of the coupler	ID	512_D 200_H	Error no. [1]	-	-	-

[1] See "Operating system errors and task errors of the coupler"

FK3 - Light errors
Error identifier in MW 255,00/%MW1255.0: 200D   C8H

Error class	Error description	Detailed info 1 in MW 255,01/ %MW1255.1	Detailed info 2 in MW 255,02/ %MW1255.2 Dec Hex	Detailed info 3 in MW 255,03/ %MW1255.3 Dec Hex	Detailed info 4 in MW 255,04/ %MW1255.4 Dec Hex	Detailed info 5 in MW 255,05/ %MW1255.5 Dec Hex	Further detailed infos in MW 255,06/ %MW1255.6 : MW 255,07/ %MW1255.7
FK3	Error when testing the coupler watchdog	ID	256_D 100_H	2_D 2_H	-	-	-
Light error	Coupler not found	ID	256_D 100_H	4_D 4_H	Coupler type[1]	Coupler model[2]	-
	Incorrect coupler type	ID	256_D 100_H	5_D 5_H	Original coupler type[1]	Coupler type found[1]	-
	Incorrect coupler model	ID	256_D 100_H	6_D 6_H	Original coupl. model[2]	Coupler model found[2]	-
	Unknown coupler model	ID	256_D 100_H	7_D 7_H	-	-	-
	No coupler reaction to reset	ID	256_D 100_H	12_D C_H	-	-	-
	Coupler does not signalize readiness for operation	ID	256_D 100_H	13_D D_H	-	-	-
	Coupler reports error when the I/O structure for the slave is requested	ID	256_D 100_H	15_D F_H	Slave address	Error no. [3]	-
	No buffer for sending message	ID	256_D 100_H	16_D 10_H	-	-	-
	Coupler does not react to the request to erase its Flash memory	ID	256_D 100_H	17_D 11_H	28_D $1C_H$	4_D 4_H	-
	Coupler does not respond to the transfer of the 907 AC 1131 configuration data for the slave	ID	256_D 100_H	17_D 11_H	Slave address	-	-
	Coupler does not respond to the transfer of the 907 AC 1131I bus parameters	ID	256_D 100_H	17_D 11_H	127_D $7F_H$	-	-
	Coupler externally configured by configuration tool; no response when the I/O structure for the slave is requested or response is invalid	ID	256_D 100_H	17_D 11_H	15_D F_H	Slave address	-

Error class	Error description	Detailed info 1 in MW 255,01/ %MW1255.1	Detailed info 2 in MW 255,02/ %MW1255.2 Dec Hex	Detailed info 3 in MW 255,03/ %MW1255.3 Dec Hex	Detailed info 4 in MW 255,04/ %MW1255.4 Dec Hex	Detailed info 5 in MW 255,05/ %MW1255.5 Dec Hex	Further detailed infos in MW 255,06/ %MW1255.6 : MW 255,07/ %MW1255.7
FK3 Light error	Coupler reports invalid configuration data (no RUN of the coupler after the transfer of the configuration data)	ID	256_D 100_H	25_D 19_H	-	-	-
	Coupler reports error when erasing the Flash memory	ID	256_D 100_H	28_D $1C_H$	4_D 4_H	Error No.[3]	-
	Coupler reports error on Start_Seq request when downloading the configuration data for the slave	ID	256_D 100_H	67_D 43_H	Slave address	Error No.[3]	-
	Coupler reports error on download request when downloading the configuration data for the slave	ID	256_D 100_H	68_D 44_H	Slave address	-	-
	Coupler reports error on download request for the slave	ID	256_D 100_H	68_D 44_H	Slave address	Error No.[3]	-
	Coupler reports error on End_Seq request when downloading the configuration data for the slave	ID	256_D 100_H	69_D 45_H	Slave address	-	-
	Coupler reports error on End_Seq request for slave	ID	256_D 100_H	69_D 44_H	Slave address	Error No.[3]	-
	Error in task 1 of the coupler	ID	768_D 300_H	Error No.[4]	-	-	-
	Error in task 2 of the coupler	ID	1024_D 400_H	Error No.[4]	-	-	-
	Error in task 3 of the coupler	ID	1280_D 500_H	Error No.[4]	-	-	-
	Error in Task 4 of the coupler	ID	1536_D 600_H	Error No.[4]	-	-	-
	Error in Task 5 of the coupler	ID	1792_D 700_H	Error No.[4]	-	-	-
	Error in Task 6 of the coupler	ID	2048_D 800_H	Error No.[4]	-	-	-
	Error in Task 7 of the coupler	ID	2304_D 900_H	Error No.[4]	-	-	-
	Life identifier error of the coupler	ID	2560_D $A00_H$	A_D A_H	-	-	-
	RUN/STOP switch is switched from STOP to RUN, but the configuration phase is not yet finished	ID	2560_D $A00_H$	27_D $1B_H$	-	-	-
	RUN/STOP switch is switched from STOP to RUN, but coupler has no configuration data	ID	2816_D $B00_H$	9_D 9_H	-	-	-
	Coupler has 907 AC 1131 configuration data as DP master, but cannot be operated as DP master	ID	2816_D $B00_H$	18_D 12_H	-	-	-
	Coupler has 907 AC 1131 configuration data as DP slave, but cannot be operated as DP slave	ID	2816_D $B00_H$	19_D 13_H	-	-	-

	907 AC 1113 configuration data for the coupler are of unknown type	ID	2816_D $B00_H$	27_D 19_H	7_D 7_H	-	-

[1] See ‚Coupler types‘
[2] See ‚Coupler models‘
[3] See ‚Message errors‘
[4] See ‚Operating system errors and task errors of the coupler‘

Coupler types

Detailed info 4 / 5		Type
Dec	**Hex**	
55_D	37_H	Coupler with fixed operating mode
57_D	39_H	Coupler with changeable operating mode

Coupler models

Detailed info 4 / 5		Model
Dec	**Hex**	
48_D	30_H	Reserved
49_D	31_H	PROFIBUS master / PROFIBUS slave
49_D	31_H	Reserved
51_D	33_H	PROFIBUS-DP master
52_D	34_H	PROFIBUS-DP slave
53_D	35_H	Reserved
54_D	36_H	Reserved
76_D	$4C_H$	DeviceNet master

Message errors

Detailed info 5		Error
Dec	**Hex**	
19_D	13_H	Coupler not in the offline configuration mode
21_D	15_H	Incorrect parameter in request
48_D	30_H	Timeout
52_D	34_H	Unknown area code
53_D	35_H	Permitted buffer length exceeded
55_D	37_H	Incorrect parameter in data
57_D	39_H	Sequence error
58_D	$3A_H$	Requested data not available
59_D	$3B_H$	Data incomplete or incorrect

FK4 Warnings
Error identifier in MW 255,08/%MW1255.8: 200D C8H

Error class	Error description	Detailed info 1 in MW 255,09/ %MW1255.9	Detailed info 2 in MW 255,10/ %MW1255.10 Dec Hex	Detailed info 3 in MW 255,11/ %MW1255.11 Dec Hex	Detailed info 4 in MW 255,12/ %MW1255.12	Detailed info 5 in MW 255,13/ %MW1255.13	Further detailed info in MW 255,14/ %MW1255.14 : MW 255,15/ %MW1255.15
FK4	907 AC 1131 incorrect configuration data set	ID	2816_D $B00_H$	27_D 19_H	-	-	-
Warning	907 AC 1131internal configuration data set contains an invalid ID (card number)	invalid ID	2816_D $B00_H$	28_D $1A_H$	-	-	-
	907 AC 1131 configuration data contain PROFIBUS master data for card number ID, but there is no PROFIBUS master coupler inserted in the slot ID	ID	2816_D $B00_H$	18_D 12_H	-	-	-
	907 AC 1131 configuration data contain PROFIBUS master data for card number ID, but there is no PROFIBUS master coupler inserted in the slot ID	ID	2816_D $B00_H$	19_D 13_H	-	-	-

3 System technology of the external couplers

3.1 07 KP 90 - Coupler for RCOM/RCOM+

3.1.1 Features of the module 07 KP 90 R303

- The module can be planned as RCOM master or as RCOM slave.
- Up to 254 RCOM slaves are possible in a network (max. 8 slaves when using MasterPiece 200 and max. 30 slaves when using dial-up operation).
- The RCOM protocol is compatible to MP200/1 with DSCA 180A. All RCOM services are available (cold start, warm start, normalization, clock synchronization, writing and reading data, event polling).
- The RCOM interface for the connection of the modem corresponds to EIA RS-232. In addition an operation according to EIA RS-485 is possible.
- An additional user interface (CONSOLE) according to EIA RS-232 is available as a commissioning utility (display of the course of communication, planning of phone numbers, etc.).
- Software clock. This time can be used by the PLC program.

3.1.2 Communication via RCOM/RCOM+

The communication between the basic unit 07 KT 9x and the coupler 07 KP 90 R303 is performed via the networking interface using function blocks of the programming software 907 AC 1131 (see section 3.1.2).

The programming and test software 907 KP 90 R202 is not required to perform the programming of the basic units 07 KT 95, 07 KT 96 and 07 KT 97. The required function blocks are part of the ABB libraries in the programming software 907 AC 1131.

Annex A contains all chapters of the programming and test software 907 KP 90 R202 which are relevant to the programming using 907 AC 1131. These are:

- Volume 3 / Chapter 2 "Introduction to RCOM"
- Volume 3 / Chapter 3 "Planning"
- Volume 3 / Chapter 4 "Using dial-up modems"
- Volume 8 / Chapter 1 "Operator"
- Volume 10 / Appendix "Error codes"

3.1.3 List of function blocks for RCOM/RCOM+

The function blocks are part of the programming software 907 AC 1131.

Library: **RCOM_V40.LIB or RCOM_S90_V41.LIB**

The communication between the basic units 07 KT 9x and 07 KP 90 R303 is performed using the following function blocks:

- CLOCK Set clock
- COLDST Cold start
- DIAL Dial communication partner
- HANGUP Hang up phone
- NORMAL Normalization
- POLL Perform event polling
- RCOM Initialize 07 KP 90 R303 for RCOM protocol
- RCOM_PL Initialize 07 KP 90 R303 for RCOM+ protocol
- READ Read data from RCOM slave
- READ_S Provide data for READ job
- RECV Receive data from RCOM partner
- SYS_S RCOM system service
- TRANSM Send data to RCOM partner
- WARMST Warm start

3.2 07 KP 93 - Coupler for MODBUS RTU

3.2.1 Features of the module 07 KP 93 R1163

The communication module 07 KP 93 R1163 is an interface module with 2 serial MODBUS RTU interfaces.

Devices which use the MODBUS RTU protocol for the communication can be coupled to the Advant Controller 31 system using this communication module.

The most essential features of the communication module are:

- 2 serial interfaces,
  can be used as EIA RS-232 or EIA RS-485 (COM3, COM4) as desired.

  Possible operating modes:

COM3	COM4	
Master	Slave	(Master-Master is not possible)
Slave	Master	
Slave	Slave	

- The communication to AC31 basic units is performed via function blocks which are part of the programming software 907 AC 1131.

3.2.2 Communication via the MODBUS interfaces COM3 and COM4

The communication between the basic unit 07 KT 9x and the coupler 07 KP 93 is performed via the networking interface using function blocks of the programming software 907 AC 1131 (see section 3.2.2).

The software 907 KP 93 is not required to perform the programming of the basic units 07 KT 95, 07 KT 96 and 07 KT 97. The required function blocks are part of the ABB libraries in the programming software 907 AC 1131.

Annex B contains all chapters of the programming and test software 907 KP 93 which are relevant to the programming using 907 AC 1131. These are:

- Volume 7 / Chapter 1 "Communication via the MODBUS interfaces COM3 and COM4"

3.2.3 List of function blocks for 07 KP 93

The function blocks are part of the programming software 907 AC 1131.

Library: **ABB-BIB4.LIB** or **COM_S90_V41.LIB**

The communication between the basic units 07 KT 9x and 07 KP 93 is performed using the following function blocks:

- MODINIT MODBUS initialization
- MODMAST MODBUS master

3.3 Communication module 07 MK 92 R0161

3.3.1 Features of the communication module 07 MK 92 R1161

The communication module 07 MK 92 R1161 is a freely programmable interface module with 4 serial interfaces.

Using this communication module, devices of other manufacturers can be coupled to the Advant Controller 31 system via the serial interface.

The communication protocols and the transfer rates can be freely defined by the user.

Programming is done using the programming and test software 907 MK 92 on a PC.

The communication module is connected to the Advant Controller 31 basic units 07 KR 91 R151 and 07 KT 9x via the networking interface.

The most essential features of the communication module are:

- 4 serial interfaces
  where
  - 2 serial interfaces can be configured as desired according to EIA RS-232 or EIA RS-422 or EIA RS-485 (COM3, COM4) and

  - 2 serial interfaces can be configured according to EIA RS-232 (COM5, COM6).
- Freely programmable by means of an extensive function library.
- Communication to AC31 basic unit via connection elements.
- LEDs for diagnosis can be planned.
- Programming and testing on a PC using COM3.
- Data can be saved in a Flash EPROM.

The processing of the serial interfaces and the networking interface are planned in an application program.

The programming is performed using the high-level language "C".

The data exchange between the serial interface module and the Advant Controller 31 basic unit is realized in the basic unit using function blocks.

3.3.2 Notes concerning the programming software 907 MK 92

The programming software 907 MK 92 can be executed on normal IBM compatible personal computers having the following technical features:

- 640 kB RAM
- Hard disk drive
- Floppy disk drive, 3 ½", 1.44 Mbyte
- EIA RS-232 (V24) serial interface for the control system
- Parallel or second serial interface for printer
- MS-DOS operating system V5.0 or higher

Order notes:

907 MK 92
Programming and test software for the communication module 07 MK 92 R1161

Order number: GJP5 2073 00 R0102

3.3.3 List of function blocks for the communication with the basic unit 07 MK 92

The function blocks are part of the programming software 907 AC 1131.

Library: **ABB-BIB4.LIB** or **COM_S90_V41.LIB**

The communication between the basic units 07 KT 9x and 07 MK 92 R1161 is performed using the following function blocks:

- SISEND Sending data to the networking interface
- SIREC Receiving data from the networking interface

4 Index

%

A

B

C

D

E

A Appendix

A.1 Error codes

Various errors identified with numbers may occur when using the RCOM coupler. The error number is indicated at output FeNR of the corresponding connection elements.

The errors can be subdivided into two groups: recoverable errors and fatal errors

Recoverable errors

The recoverable errors include all errors occurring during transmission of telegrams (character losses, parity errors etc.). In some cases, the coupler attempts retransmission in the case of transmission errors; the number of attempts can be specified on the RCOM connection element (parameter Retr).

In the case of recoverable errors, LED “ERR” lights briefly. The LED goes out again as soon as the error has been remedied.

Fatal errors

If fatal errors occur, the coupler terminates both communication with the CS31 CPU and communication with the RCOM partners.

It is still possible to make entries via the operator interface. A Reset is required in order to reactivate the coupler, e.g. by switching off and switching back on or by using the Reset key. If you use the Reset key, you must abort the PLC program, and restart it in order to re-initialize the 07 KP 90 R202.

LED “ERR” lights steadily in the case of fatal errors. In addition, LED “RUN” goes out, thus indicating that no further communication is occurring.

Table of error codes

The table overleaf contains all possible error codes (hexadecimal and decimal), their cause and whether the error is fatal or not.

Error code (hex)	(dec.)	Cause	Rmks.
0000	0	No error	
0001	1	PLC does not respond (possible cause: missing CE or PLC stopped)	
1001	4097	Event queue full	
2001	8193	Initialization error: Incorrect address (parameter NOD)	
2002	8194	Initialization error: Incorrect baud rate (parameter Baud)	
2005	8197	Initialization error: Incorrect parity (parameter Pari)	
2006	8198	Initialization error: Incorrect duplex mode (parameter Dupl)	
2007	8199	Initialization error: Incorrect line stab. time (parameter TLS)	
2008	8200	Initialization error: Incorrect carrier delay (parameter CDly)	
2009	8201	Initialization error: Incorrect char. timeout (parameter CTO)	
2010	8208	Initialization error: Incorrect turnaround time (parameter TAT)	
2011	8209	Initialization error: Incorrect retransmissions (parameter Retr)	
2012	8210	Initialization error: Incorrect number of preambles (parameter NoPr)	
2015	8213	Initialization error: Incorrect modem type (parameter Mdm)	
2018	8216	Initialization error: Incorrect debug level (parameter Dbg)	
3001	12289	No CTS during transmission	
3002	12290	Timeout, no response telegram	
3003	12291	Telegram error (incorrect length code)	
3004	12292	Telegram error (incorrect checksum)	
3005	12293	Incorrect slave responding	
3006	12294	Telegram error (incorrect response code)	
4001	16385	Service not known	
4002	16386	Incorrect length entry (parameter LEN)	
4003	16387	Incorrect slave number (parameter NOD)	
4004	16388	Incorrect network number (parameter NET))	
4005	16389	Incorrect data set number (parameter IDT)	
4006	16390	Response telegram does not match service	
4007	16391	Slave not normalized	
4008	16392	Event queue blocked (not normalized)	
4010	16400	Invalid time (connection element CLOCK)	
4020	16416	Slave responds: “application part not ready” (e.g. FREI = 0)	
4030	16432	Slave responds ”illegal command”	
5000	20480	Coupler not initialized (after Reset)	1)
6001	24577	Telephone directory or setup data not found	
6002	24578	Incorrect entry in telephone directory or setup data	
6003	24579	No telephone modem planned	
6004	24580	Modem not yet online (service started without DIAL)	
6005	24581	Modem already online (repeated DIAL without HANGUP)	
7000	28672	EEPROM cannot be programmed	
7001	28673	EEPROM cannot be erased	
FFFF	–1	Internal error	fatal

Remarks:
1) Requires restart of the PLC (abort program and restart of the program)
fatal Fatal error

Table: Error codes

A.2 Operator messages

The tables on the next few pages contain all possible messages which may be output on the operator interface and their significance.

Certain messages contain designations for services which are specified in the following table.

Abbreviated	Name of the service	Triggered in the master by CE	Handled in the slave by CE
norm comm part	normalize communication part	1)	3)
quer comm part	status query communication part 1)	3)	
cold start	cold start	COLDST	3)
warm start 1	block all blocks	WARMST	3)
warm start 2	block unique blocks	1)	3)
set clock	set clock	CLOCK	3)
norm user part	normalize user part	1)	3)
norm all blocks	normalize all blocks	NORMAL	3)
norm sep blocks	normalize sepereate blocks	1)	3)
write dataset	write dataset	TRANSM	RECV
write cntrl	write data to control register	1)	3)
read dataset	read dataset	READ	READ_S
event request	event request	POLL	3)
repeat read	repeat read command	2)	3)
repeat write	repeat write command	2)	3)
dial	dial up slave	DIAL	3)
hangup	hangup phone	HANGUP	3)

Remarks:
1) not used as master on 07 KP 90 R202
2) is performed automatically in the case of transmission errors
3) is handled automatically in the 07 KP 90 R202

Table: Identification of services

In the table showing all messages, the first column specifies the actual message of the communication module. The next column specifies the significance of this message and the third column shows remarks which may indicate the cause of the error.

–EPL–E–PLCTO	service timeout, no PLC reaction	An event which has arrived has not been fetched by the PLC. RECV CE missing	Rmks 1
–EPL–I–EVENT	event – ds: .., len: .., ..	Event arrived	
–EPL–I–PLCACC	service accepted by PLC	Service accepted by PLC	
–EPL–I–SYSMES	system message	System event has arrived (is ignored)	
–EPL–W–PLCREJ	service rejected by PLC,	Event rejected by PLC (FREI in the case of RECV = 0)	
–ERR–F–FATAL	fatal error, communication canceled	Fatal error has occurred, communication has been terminated	
–EVT–I–BLOCK	blocking event queue	Blocking event queue	
–EVT–I–CLEAR	clearing event queue	Clearing event queue	
–EVT–I–DEBLK	deblocking event queue	Enable event queue Event queue	
–EVT–I–GET	event queue empty	Event queue is empty	
–EVT–I–GET	getting event	Fetching event from queue	
–EVT–I–PUT	putting event	Inserting event in queue	
–EVT–W–PUT	event–queue full	Event queue is full	
–INI–E–COMGRP	error reading com group, code	Error reading special flags for communication processors	Rmks 2
–INI–E–EXTINI	external init error, ...	Error during initialization	2
–INI–E–GRESI	error resetting PLC communication, ...	It has not been possible to reset system bus communication	2
–INI–E–MOD	error initializing telephone modem, ...	It has not been possible to initialize telephone modem	3
–INI–E–OCCUP	error occupying PLC, ...	It has not been possible to assign PLC	2
–INI–E–OPER	error initializing operator, ...	It has not been possible to initialize commissioning interface	3
–INI–E–PARAM	error in parameter, ...	Error in parameter	1
–INI–E–RCOM	error during RCOM–init, ...	It has not been possible to initialize RCOM mechanism	
–INI–E–RCOM	error initializing RCOM–channel, ...	It has not been possible to initialize RCOM interface	3
–INI–E–READ	error reading parameters, ...	It has not been possible to read RCOM parameters	2
–INI–E–RS	error reading RUN/STOP, ...	It has not been possible to read RUN/STOP switch	2
–INI–E–RSINI	error initializing PLC communication, ...	It has not been possible to initialize system bus communication	2
–INI–I–CHECK	checking RCOM–parameters	Checking RCOM parameters	
–INI–I–COMGRP	waiting for valid com group	Waiting for valid communication area (in special flag for communication processors)	
–INI–I–EXTINI	external init done, start flag: MW nn,0	External initialization completed, communication flag range = MW ...	
–INI–I–EXTINI	waiting for external init	Waiting for external initialization	
–INI–I–GRES	resetting PLC communication	Resetting system bus communication	
–INI–I–RUN	waiting for RUN–switch	Waiting for RUN/STOP switch = RUN	
–KPM–I–EXIT	exit main loop, reason code ...	Quitting RCOM communication; cause ...	Rmks 4
–KPM–I–GOODM	good morning!!	It is midnight	
–KPM–I–LCNT	lifecount ...	Cycle counter set to ...	
–MOD–E–DIAL	cannot connect	It has not been possible to establish dial–up connection	5
–MOD–E–DIAL	modem already connected	Modem is already on line	5
–MOD–E–DIAL	no modem available (modem type = 0)	No modem planned	1
–MOD–E–ENTRY	bad entry in phone file	Entry error in telephone directory	5
–MOD–E–HANGUP	cannot hangup	It has not been possible to hang up	3
–MOD–E–INIT	error during modem init	Error during modem initialization	3
–MOD–E–NOFILE	no valid files on EEPROM	No valid setup/telephone files on EEPROM	5
–MOD–E–RING	no modem available (modem type = 0)	No modem planned	1
–MOD–I–ANS	answer ...	Modem response: ...	
–MOD–I–ANSCMP	compare ... – ...	Comparing modem response with ...	
–MOD–I–ANSCMP	strings are equal	Strings are identical	
–MOD–I–DIAL	connected	It has been possible to set up connection (dial)	
–MOD–I–DIAL	dialing node ...	Dialing station ...	
–MOD–I–DIAL	ring (...)	Dialing station ...	
–MOD–I–HANGUP	hangup phone	Hanging up	

–MOD–I–INIT	answer ...	Modem respone:...	
–MOD–I–INIT	init modem (...)	Initializing modem	
–MOD–I–RING	connected	It has been possible to set up connection (going off hook)	
–MOD–I–RING	ring received	Telephone ringing	
–MOD–W–DIAL	retry ...	Retry dialing	
–MOD–W–RING	cannot connect	It has not been possible to set up connection (going off hook)	Rmks 5
–MST–E–ADDR	error reading reply (address), ...	It has not been possible to read address from response	6
–MST–E–DATA	error reading reply (data), ...	It has not been possible to read data from response	6
–MST–E–HEADR	error reading reply (header), ...	It has not been possible to read header from response	6
–MST–E–LCODE	illegal length–code in reply	Incorrect length code in response	6
–MST–E–POSTA	error reading reply (checksum/postambles),...	It has not been possible to read checksum/trailer from response	6
–MST–E–PREA	error reading reply (preambles), ...	It has not been possible to read leader from response	6
–MST–E–RES	command not reset by PLC	Command not acknowledged by PLC	7
–MST–E–SEND	error sending telegram, ...	It has not been possible to transmit job	3
–MST–E–SUM	checksum error in reply	Checksum error in response	6
–MST–I–POLL	checking slave ...	Checking whether slave ... has dialed	
–MST–I–POLL	polled slave ..., result ...	It has been possible to poll slave ...; result ...	
–MST–I–RESULT	..., result ...	Service ... terminated; result ...	
–MST–I–SERV	...	Service ... started	
–MST–W–NOSRV	no services within hangup time	No service within the hangup time, now hanging up	Rmks 1
–MST–W–RETRY	retry ...	Retrying job telegram	6
–OPR–E–CMD	unknown command ’...’	Unknown operator command	
–OPR–E–OCCUP	error occupying PLC, ...	It has not been possible to assign PLC	2
–OPR–E–RELEA	error releasing PLC, ...	It has not been possible to release PLC	2
–OPR–I–INIT	operator init done	Initialization commissioning interface terminated	
–PLC–E–GETEND	error reading line area from plc, ...	Error in system bus communication	2
–PLC–E–GETEND	error reading rx area from plc, ...	Ditto	
–PLC–E–GETEND	error reading tx area from plc, ...	Ditto	
–PLC–E–GETKP	error reading kp number from plc	Ditto	
–PLC–E–GETREQ	error reading com group from plc, ...	Ditto	
–PLC–E–GETREQ	error reading line area from plc, ...	Ditto	
–PLC–E–GETREQ	error reading rx area from plc, ...	Ditto	
–PLC–E–GETREQ	error reading tx area from plc, ...	Ditto	
–PLC–E–SETEND	error writing control block to plc, ...	Ditto	
–PLC–E–SETEND	error writing rx area to plc, ...	Ditto	
–PLC–E–SETEND	error writing tx area to plc, ...	Ditto	
–PLC–E–SETREQ	error writing control block to plc, ...	Ditto	
–PLC–E–SETREQ	error writing rx area to plc, ...	Ditto	
–PLC–E–SETREQ	error writing tx area to plc, ...	Ditto	
–PLC–W–GETEND	timeout during read com group from plc, abort: ...	Timeout during system bus communication, error during abort: ...	Rmks 2
–PLC–W–GETEND	timeout during read line area from plc, abort: ...	Ditto	
–PLC–W–GETEND	timeout during read rx area from plc, abort: ...	Ditto	
–PLC–W–GETEND	timeout during read tx area from plc, abort: ...	Ditto	
–PLC–W–SETEND	timeout during write control block to plc, abort: ...	Ditto	
–PLC–W–SETEND	timeout during write rx area to plc, abort: ...	Ditto	
–PLC–W–SETEND	timeout during write tx area to plc, abort: ...	Ditto	
–RCS–I–SET	RCSW set to ...	RCOM status word set to ...	
–RDS–E–PLCTO	service timeout, no PLC reaction	PLC not responding to read job (READ_S connection element missing)	Rmks 1

–RDS–I–PLCACC	service accepted by PLC	Service “read data set” accepted by PLC	
–RDS–W–PLCREJ	service rejected by PLC, ...	Service “read data set” rejected by PLC (FREI with READ_S=0)	
–RPL–E–REPLY	internal error: ...	Internal error .... when setting up response	Rmks 7
–RPL–E–REPLY	reply error: ...	Error in response telegram	5, 6
–RPL–I–LEN	... data bytes in reply	... data bytes in response	
–RPL–I–REPLY	...	Response: ...	
–RPL–I–REPLY	..., result ...	Error ... in response	Rmks 6
–SCL–I–TIME	date: %ld, time %ld	New RCOM time ... arrived	
–SLV–E–ADDR	error reading telegram (address), ...	It has not been possible to read address from job	Rmks 6
–SLV–E–BREAK	error checking for break	It has not been possible to check BREAK	3
–SLV–E–DATA	error reading telegram (data), ...	It has not been possible to read data from job	6
–SLV–E–HEADR	error reading telegram (header), ...	It has not been possible to read header from job	6
–SLV–E–LCODE	illegal length–code in telegram	Incorrect length code in job	6
–SLV–E–POSTA	error reading telegram (checksum/postambles), ...	Incorrect checksum/trailer in job	6
–SLV–E–PREA	error reading telegram (preambles), ...	It has not been possible to read leader from job	6
–SLV–E–RES	command not reset by PLC	Command has not been cancelled by PLC	7
–SLV–E–SEND	error sending reply, ...	It has not been possible to send response	3
–SLV–E–SUM	checksum error	Checksum error	6
–SLV–I–ADR	not my address (...)	Job not for me but for slave ...	
–SLV–I–NOREP	no reply sent (broadcast request)	No response transmitted (broadcast job)	
–SLV–I–RESULT	..., result ...	Service terminated; result ...	
–SLV–I–SERV	...	Service detected	
–SLV–W–MODE	event queue blocked	Event queue barred (normalization missing)	Rmks 1
–SLV–W–MODE	slave mode program	Data transmission barred (normalization missing)	1
–SLV–W–NOSRV	no services within hangup time	No jobs have arrived within the hangup time, now hanging up	
–TEL–E–SERV	internal error: ...	Internal error ...	8
–TEL–I–LEN	... data bytes in request	... data bytes in job	
–TEL–I–SERV	...	Service ... detected	
–TEL–I–SERV	... ds: ..., len: ...	Service ... detected; data set ...; length ...	
–WDS–E–PLCTO	service timeout, no PLC reaction	PLC not responding to write job (RECV connection element missing)	Rmks 1
–WDS–I–PLCACC	service accepted by PLC	Service “Write data set” accepted by PLC	
–WDS–W–PLCREJ	service rejected by PLC, ...	Service “Write data set” rejected by PLC (FREI with RECV = 0)	

Remarks:

1 A planning error has probably occurred. Check whether all required CEs are present and whether the correct parameters have been assigned to them.

2 Error during system bus communication. It is either remedied automatically or it leads to fatal errors. May occur in the case of configuration errors of the T200 system. May also be triggered by 907 PC 32, e.g. when transmitting programs

3 There is probably a fault in the cable. Check the wiring of RTS and CTS

4 The following causes are possible:
- 2: Reinitialization has been performed by the PLC (RCOM connection element started)
- 3: RUN/STOP switch has been set to STOP
- 4: A fatal error has occurred

5 There is probably an error in modem control. Check: Modem settings, SETUP and PHONE data.

6 A transmission error has occurred. Check all RCOM parameters and timeout times.

7 This occurs if the CPU is set to STOP

8 Internal error. This may not occur. Attempt to reset the unit and reinitialize it.

2 Introduction to RCOM

This chapter explains the fundamental terms and functions of the RCOM protocol. It also briefly presents the functions of the 07 KP 90 R202.

2.1 What is RCOM?

RCOM is a transmission protocol which is particularly suitable for data transmission over large distances (RCOM = Remote COMmunication).

The protocol is based upon a simple V.24 interface so as to permit the use of standard data teletransmission devices (e.g. modems)

The main fields of application for RCOM couplings are as follows:

- Coupling from ABB Procontic CS31 to ABB MasterPiece control systems
- Networking ABB Procontic CS31 to ABB Procontic T200 controllers with 07 KP 64
- Inter–networking CS31 stations

You can use either dedicated lines (e.g. existing cable paths or leased lines) or telephone lines with dial–up modems for communication.

RCOM networks

An RCOM network consists of two or more users, e.g. control computers or substations etc. One user is always planned as the RCOM master. All other users are RCOM slaves

The users are interconnected by means of a transmission medium. In the case of RCOM, this may comprise direct lines for instance (point–to–point connections), dedicated lines with multidrop modems (these permit coupling of several users to one line) or dial–up connections via the public telephone network.

Each user has an address via which it can be addressed. This address is a number between 1 and 254 for slaves, and the master has the address 0. .

The illustration below shows an RCOM network with multidrop modems:

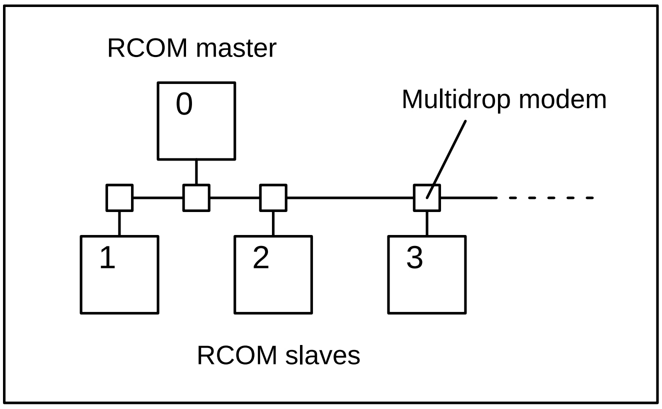


Figure 2.1: RCOM network with multidrop modems

With each job, the master transfers the address of the addressed slave, and only this slave will respond to the job.

Master–slave structure

There is a simple convention for controlling data communication on the line: One user in the network is the **master**. Only the master can send jobs to other users, the **slaves**. The slaves respond to a job telegram with a telegram which indicates whether the job has been understood and whether it has been possible to execute it.

This always results in the sequence job telegram –> response telegram on the data line.

2.2 Data transmission

Data sets

All user data in the RCOM network are transmitted in the form of so–called data sets (DSs for short). They consist of maximum sixteen 16–bit data words (eight 32–bit double words in the case of MasterPiece).

A maximum of one data set can be transferred in each job telegram. If it is intended to transmit large quantities of data, it is necessary to start several jobs.

For identification, each data set has a number which is also transferred in the telegram. This data set number is abbreviated to 'IDT'.

These data sets are stored in word flags in the CS31. Thus, 16 word flags in which the user data are to be stored must be reserved for one data set.

The user must specify the data set number ('IDT'), the first of 16 word flags (start word flag 'AMW') and the number of data words to be transferred ('LEN') at the corresponding connection element for transfer of the data set.

There are three options for transmission of data sets:

- Write data sets to slave
- Read data sets from slave
- Event–driven transmission

Write data set

The master can write a data set to the slave by reading the user data from the planned flags and sending them via telegram to the slave. The data are stored in the planned flags in the slave. The slave confirms reception of the data in the response telegram.

Read data set

The master reads a data set from the slave by first sending a job telegram to the slave. The slave receives this telegram, reads the data from its flags and returns the data in the response telegram. The data from the response are stored in the planned flags in the master.

Event polling

In many applications, it is necessary for the slave to transfer data to the master of its own accord, e.g. if it has recognized an important event in the process.

Normally, the slave cannot start communication of its own accord (master–slave structure). So the slave would have to wait until the master reads the required data set.

The RCOM protocol has a simple mechanism for overcoming this problem: Event polling.

If a slave wishes to send a data set to the master of its own accord, it can do this as follows:

- The slave transfers the data set to a queue (event queue) on the 07 KP 90 R202.
- The master cyclically polls all slaves consecutively in order to establish whether they have events in their queues. If so, the addressed slave sends the data set in the response. If not, it sends the response 'Event queue empty'.

This means that the slave can signal events to the master very easily. If no events have occurred, no user data are exchanged either. So the transmission will be completed very quickly in this case.

Since there is no temporal relationship between triggering of the event (insertion in the queue) and event polling (read–out of the queue by the master), a time stamp which provides information on when the event occurred is stored in the data set. For this reason, only maximum 14 data words can be entered in the data set, and the last two words contain the time stamp.

The 07 KP 90 R202 can store a maximum of 20 events in the queue. Other events are rejected with an error message.

The illustration below is intended to clearly show event–driven transmission.

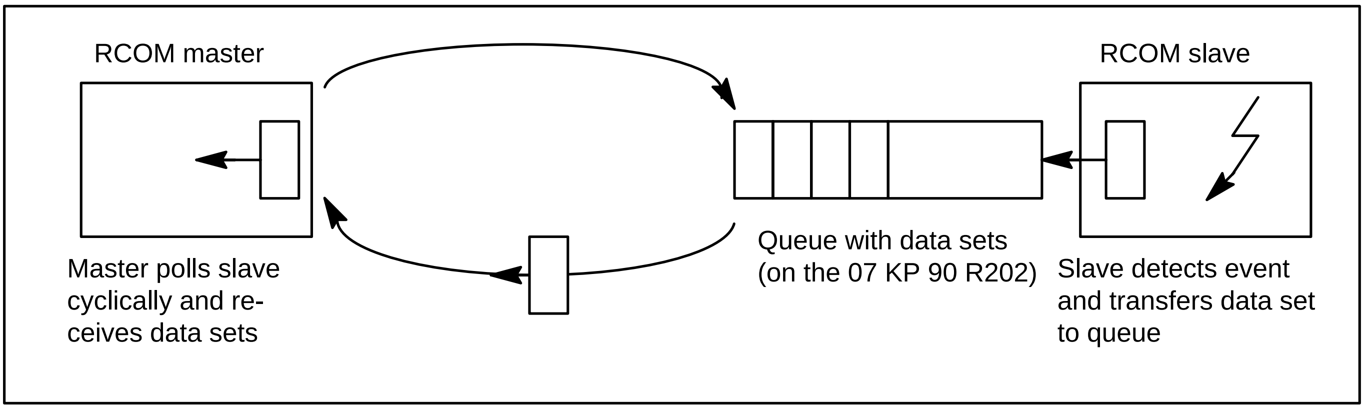


Figure 2.2: Event polling

2.3 Job types

There are several job types which can be subdivided into two groups for data transmission and for management of the RCOM network:

- System services, e.g. for connection set–up and cleardown and reinitializing the network etc.
- Services for data transmission. After the RCOM network has been started successfully, the master triggers these jobs in order to transmit user data.

System services

The following services are provided for management of the RCOM network:

- Cold start: The addressed slave is reset, i.e. all protocol control characters are set to initial state. In addition, the contents of the event queue are deleted. After a cold start, data sets cannot be transmitted again until normalization has occurred.
- Warm start: This also deletes the contents of the event queue. A warm start can be started after a transmission error, for instance, in order to resynchronize master and slave. After a warm start, data sets cannot be transmitted again until after normalization.
- Normalization (normalize user part): This enables transmission of data sets after a cold start or warm start. This job **must** be used in order to permit communication to commence.
- Set clock (clock synchronization): The 07 KP 90 R202 has a clock which generates the time stamps for events. This clock can be set by the master.

All system services can be started in the master with corresponding connection elements. In the slave, the system services are handled automatically by the 07 KP 90 R202, i.e. nothing needs to planned for the system services in the RCOM slaves.

Please do not confuse the RCOM services Cold start and Warm start with the commands of the same name on the CS31 PLC. The terms Cold start and Warm start always refer to the RCOM services in this manual.

Data transmission services

The following services are provided for data transmission:

- Write data set
- Read data set
- Poll event queue (event polling)

2.4 Addressing in the RCOM network

Addressing data sets

A complete address must be specified in order to address a data set on a specific slave. This address consists of the following parts:

- Number of the RCOM network (NET). Since only one RCOM network can be connected on the 07 KP 90 R202, NET must always be zero.
- Number of the slave (NODE). Up to 254 slaves may be present in an RCOM network. The number zero is used by the master. NODE is abbreviated to NOD in the connection elements used for planning the coupler.
- Number of the data set (IDT). IDENT is abbreviated to IDT in the connection elements.
- Number of data words of the data set to be transmitted (LEN). A data set does not always need to be transmitted completely but at least two data words must be transmitted and the number of data words to be transmitted must be an even number. Transmission always commences with the first data word.

The illustration below shows an example of an RCOM system and the addressing path:

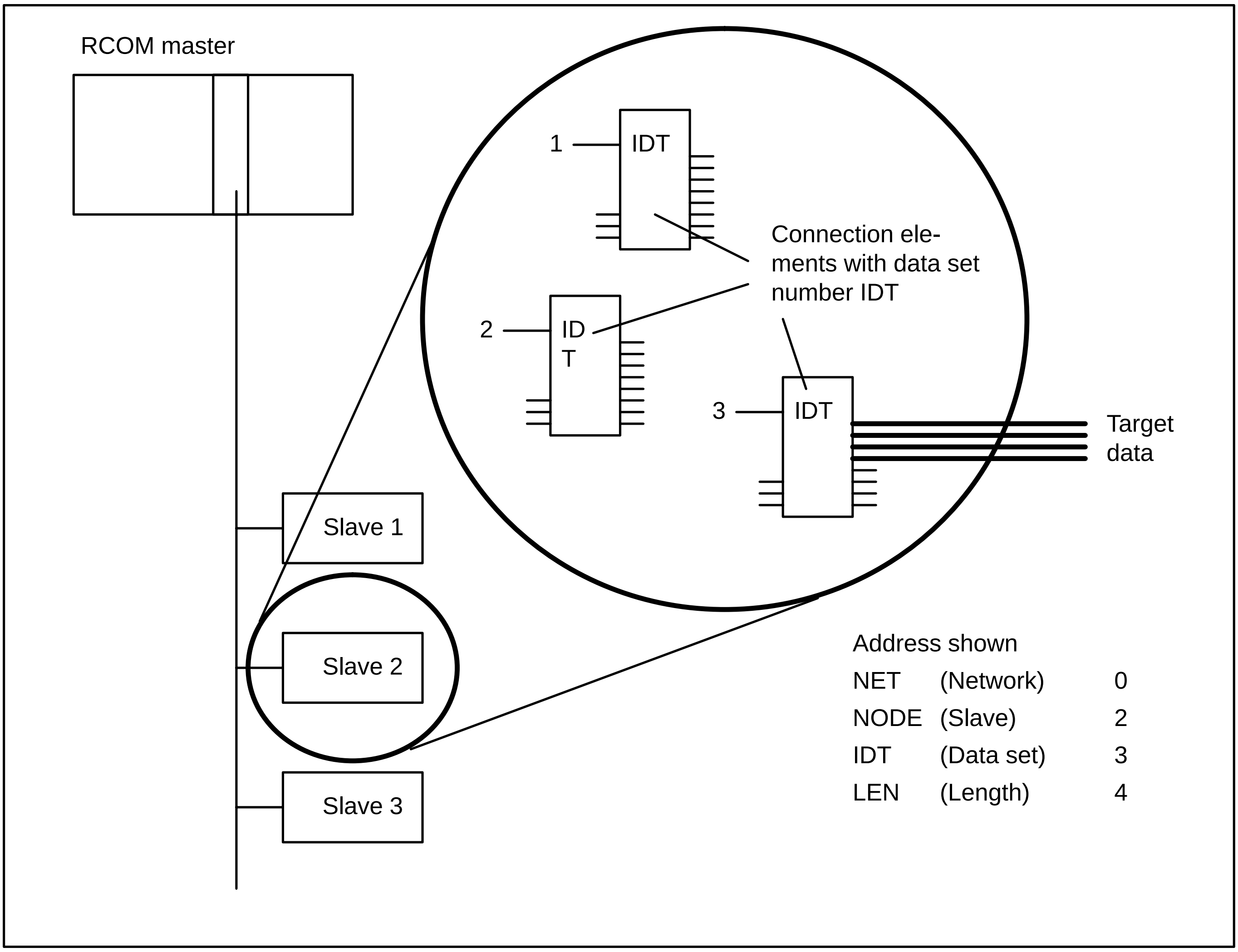


Figure 2.3: Addressing in the RCOM system

Broadcast

Slave number FF_H (255) can be used to transmit specific services to all slaves simultaneously. Such a service is **not** answered by **any of the slaves** and is thus repeated several times by the master as a safety measure.

Broadcast is permitted only for job types in which no user data are transmitted (system services).

2.5 Overview of the 07 KP 90 R202

Control of the coupler

The coupler is controlled by the CS31 master station with the aid of commands. Commands from the master station and responses of the coupler are exchanged via the networking interface of the CS31 master station.

The user controls the coupler only with the aid of the supplied connection elements (abbreviated to “CE” or “block” below). There are CEs for all system and data transmission services. This permits the user himself to define the required communication sequence very simply.

Parameters

All important parameters (baud rate, timeout times etc.) for the coupler are defined in an initialization connection element (RCOM). They are read once by the coupler during initialization and are then used unchanged until the next initialization.

Parameters for the individual services (e.g. slave address, data set number etc.) are preset directly on the connection element for this service.

Comissioning

A terminal can be connected to the 07 KP 90 R202 in order to simplify commissioning. The coupler then issues messages concerning incoming commands from the PLC, received and transmitted RCOM telegrams and any errors which have occurred.

This function can be deactivated after commissioning. The communication processor then continues operation without a terminal.

The Annex provides a list of all messages of the coupler and their significance.

2.6 Differences between RCOM and RCOM+

There are the following differences concerning the data transmission with RCOM and RCOM+:

– the “BREAK” character of RCOM is replaced by a transmission break of configurable length for RCOM+

– the 8 bit “Exclusive–or–check sum” of RCOM is replaced by a 16 bit CRC16 check sum for RCOM+

– in addition, the variable DIGI_time was added for delaying the transmission when using the DIGILINE modems

2.7 Switching between RCOM and RCOM+

Switching over between RCOM and RCOM+ is carried out by configuring the CEs RCOM or RCOM+.

The CEs have the same parameters. Internally, the variable RCOM_typ is set to “1” in case of RCOM+. In case of RCOM, RCOM_typ is “0”.

The description of the CEs will be found in volume 7.

3 Planning

3.1 Overview

Connection elements permitting simple use of the coupler are provided for all required services:

- The coupler is initialized with the “RCOM” connection element. It defines all parameters required for RCOM operation.
- For the system services, there are blocks for:
  – the Cold start service
  – the Warm start service
  – normalization (normalize user part)
  – event polling
  – setting the time
  – telephone dialing and
  – telephone hangup
- There are the CEs TRANSM (Transmit data set) and RECV (Receive data set) for writing data sets. These connection elements are also used for event–driven transmission. Block “POLL” which triggers polling of a slave must be planned in the master for event polling.
- The RCOM master uses the READ connection element for reading data sets. The addressed slave makes available the data to be read in a READ_S connection element.

Initialization, cold start and normalization

The illustration on the next page shows the application of the connection elements RCOM, COLDST and NORMAL. These CEs must be used for initializing the communication processors and for starting the RCOM protocol.

Each RCOM user is initialized with the RCOM connection element, i.e. the transmission parameters, the network address and the timeout times etc. are defined.

The RCOM master must then perform an RCOM Cold start service (COLDST). Data transmission must now be enabled with normalization (NORMAL). Cold start and normalization are implemented in the example with Broadcast telegrams (NOD = 255) so that all slaves are addressed simultaneously.

The RCOM network is ready for data transfer after the above–described procedures.

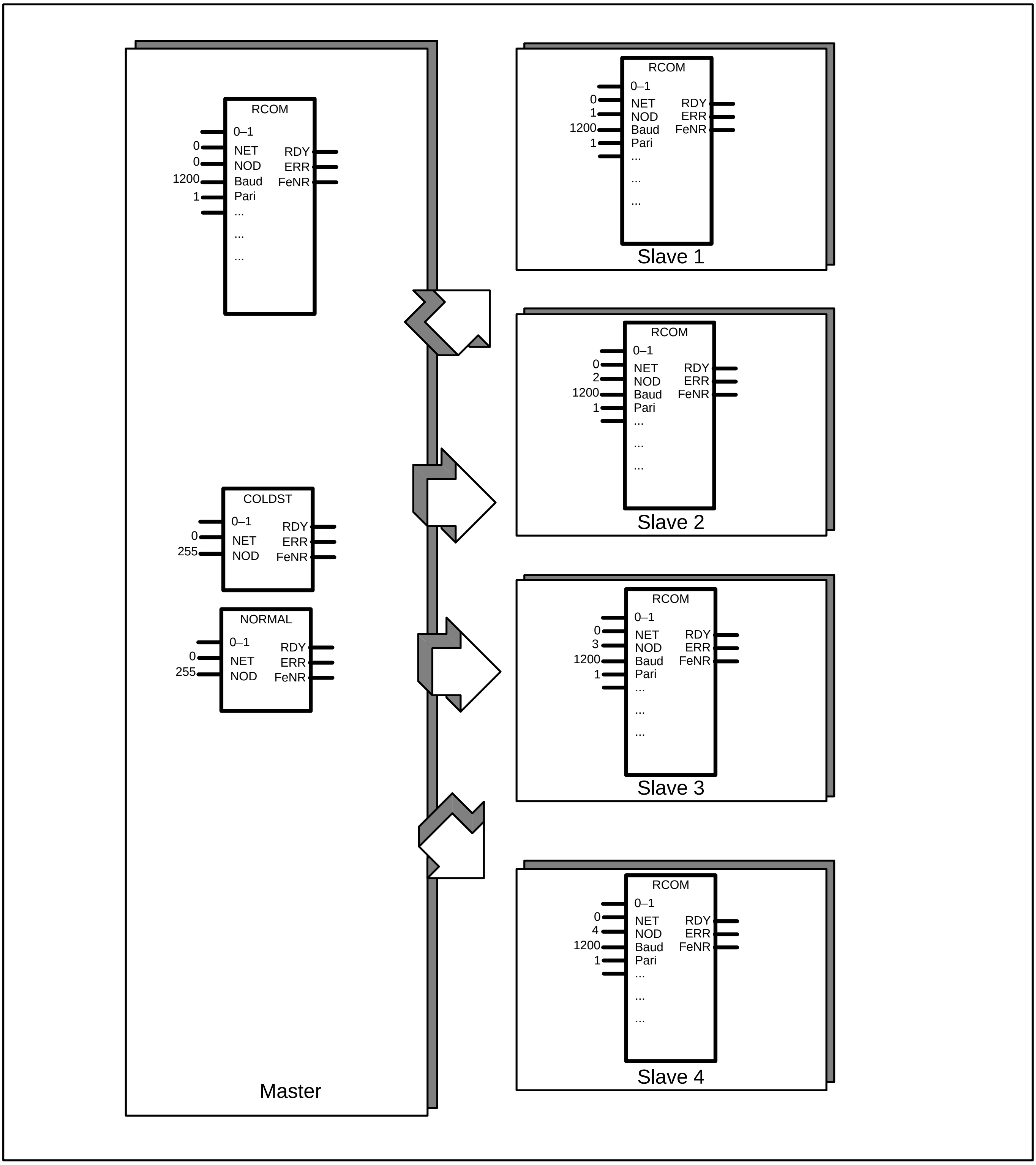


Figure 3.1: Application of the connection elements for initialization and the RCOM services Cold start and Normalization. The CEs have been shown in simplified form.

Data transmission

The illustration below shows an example of transmission of data sets. It is intended to clearly illustrate the relationship between blocks and the significance of address and data set number IDT.

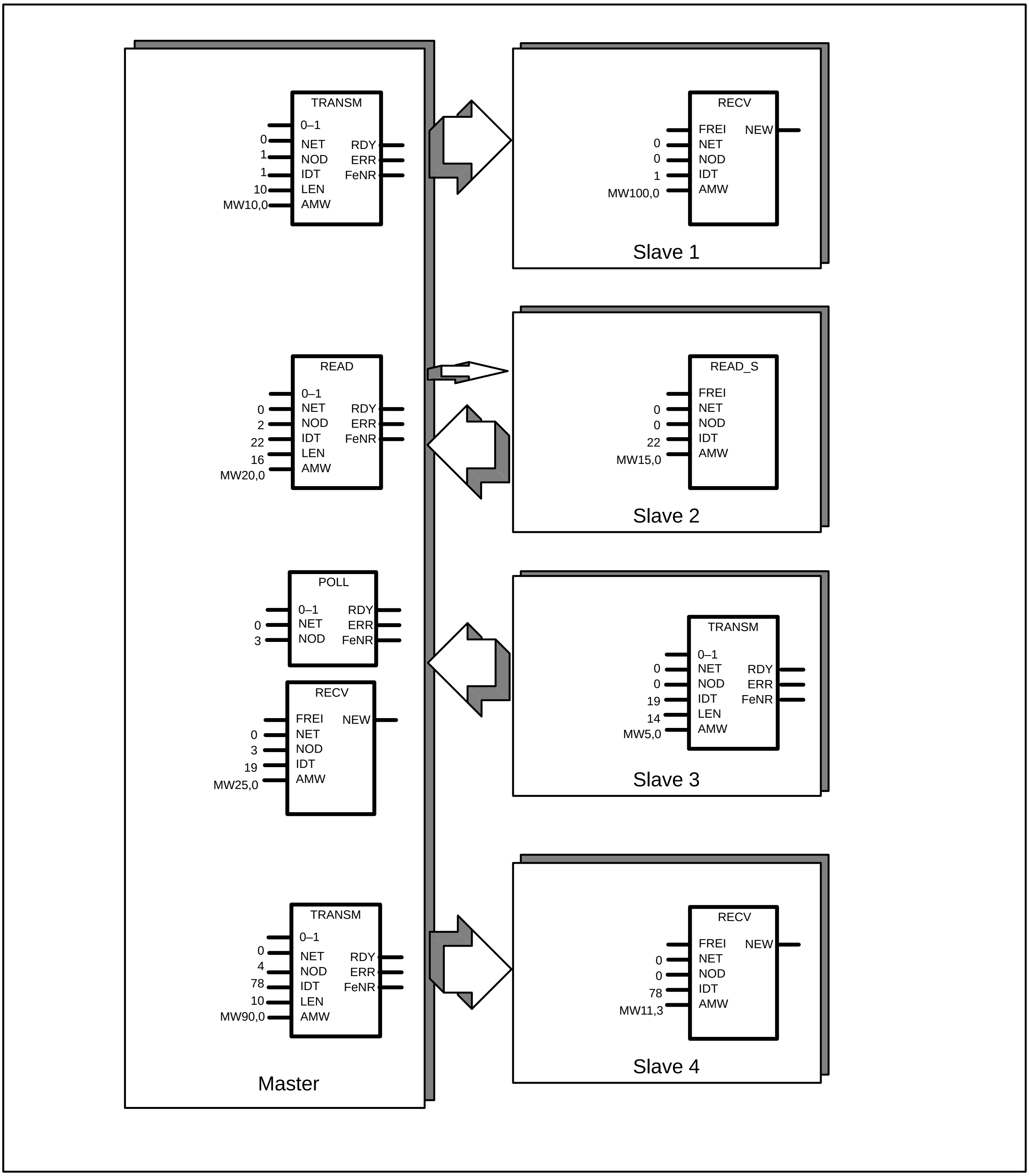


Figure 3.2: Application of the connection elements for a data transmission. The CEs are shown in simplified form.

The following transmissions can be performed in this example:

- Data set 1 is transmitted from the master to slave 1. The data are taken from MW10,0 to MW10,9 in the master and stored in flags MW100,0 to MW100,9 in the slave.
- Data set 22 is read from slave 2. The data are taken from flags MW15,0 to MW15,15 in the slave and stored in MW20,0 to MW20,15 in the master.
- Data set 19 is sent from slave 3 to the master (event–driven). The master must first poll slave 3 (block POLL). The data are fetched from flags MW5,0 to MW5,13 in the slave and are stored in flags MW25,0 to MW25,13 in the master.
- Data set 78 is sent from the master to slave 4. The data are read from MW90,0 to MW90,9 in the master and stored in MW11,3 to MW11,12 in slave 4.

This illustration does not show the logic for enabling etc.

Please note:

- For each connection element in the master, there is a partner in the slave. These two CEs have the same data set number IDT.
- The actual user data are not applied as inputs or outputs on the connection element, but parameters are assigned to them only as a reference to where the data are stored.
- All data sets in the slave have zero as address NOD since jobs can come only from the master.

Flag M255,15

Flag M255,15 requires special handling so that the RCOM connection elements can be initialized correctly when starting the user program.

This flag is always initialized to zero each time the user program is started, i.e. independently of the initialization presets by the system constants.

M255,15 must be set to 1 **at the end of the PLC program**, i.e. directly before connection element PE so that the RCOM connection elements are able to determine whether the user program has been restarted or not. The flag must not be changed after this.

The RCOM connection elements do not work correctly if this rule is not observed.

3.2 Structure of the data sets

The data sets must always lie in the normal word flag range MW xx,yy at a fixed address. This address must be specified at input AMW on the connection elements.

Word flags

Word flags can be transmitted directly. You merely have to transfer the data from the process to the data set, i.e. by means of an assignment or a COPY block.

Binary flags

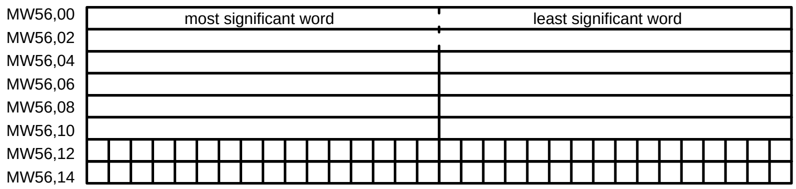
Binary flags must be packed to form words with the PACK block for transmission. They can unpacked again accordingly with the UNPACK block after transmission.

Double words

Double words must also be separated to form words before transmission. In this case, the high–order word must lie first and at an even address in the data set (example: data set starts at MW10,0; possible positions for a double word: MW10,0, MW10,2, MW10,4 etc.). This ensures compatibility with MasterPiece control systems.

Example of a data set

The illustration below shows a data set in which various data types are packed:



Example of data set: user data: 2 double words, 8 words, 64 binary flags

User data are located between MW56,00 and MW56,15

Figure 3.3: Data set in which various data types are packed

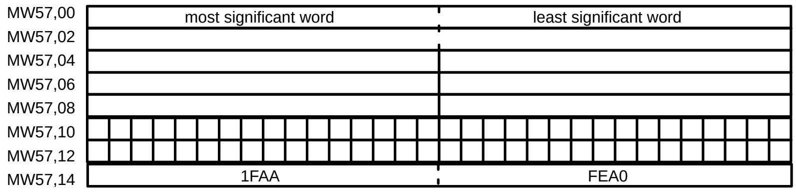
Data sets in case of event–driven transmission

In data sets transmitted as an event from the slave to the master, the time stamp is a double word **after** the actual user data (max. 14 words user data + 2 wordstime stamps = 16 words). If the data set contains 4 user data words for instance, the time stamp will lie in words 4 and 5.

The time stamp indicates when the event occurred. The time is calculated as a double word in 0.1 ms since midnight. Example: the event was triggered at l4:45.30: the time stamp then has the value:

```
114*60*60*1000*10 +
    45*60*1000*10 +
       30*1000*10 =  531300000 dec.
                  =  1FAAFEA0  hex.
```

This value is split into one high–order word and one low–order word, each with 16 bits, and stored directly **after** the user data in the data set, with the high–order word first:



Example of event data set: user data: 2 double words, 6 words, 64 binary flags
Time stamp: 14:45.30

The user data lie between MW57,00 and MW57,13.Time stamp at MW57,14..MW57,15

Figure 3.4: Storing event data sets in the double word

3.3 Use of the system services

A specific sequence of system services must be observed for starting an RCOM network. This sequence ensures correct initialization of the RCOM protocol.

All system services are triggered by the master with the corresponding CEs. Nothing needs to be planned in the slaves for the system services. The 07 KP 90 R202, as slave, handles all system services independently.
Important: Please do not confuse the RCOM services Cold start and Warm start with the PLC commands of the same names. The terms “Cold start” and “Warm start” always refer to the RCOM system services in this manual.

Cold start

A cold start service must be performed after the RCOM master has been initialized. The cold start can be transmitted either by broadcast to all slaves simultaneously or to each slave individually.

Cold start involves the entire protocol mechanism being reinitialized and the contents of the event queue being cleared. A special cold start event is then triggered in the addressed RCOM slaves. This event is required when operating with ABB MasterPiece systems. It is merely indicated when polling in the case of pure ABB Procontic networks.

Normalization must always be carried out after a cold start since, otherwise, it is not possible to transmit data sets.

Warm start

The event queue of one slave (or all slaves) can be cleared with a warm start service.

A warm start can be used for resuming communication after transmission errors for instance. This permits master and slave to resynchronize.

Normalization must always be carried out after a warm start since, otherwise, data sets cannot be transmitted.

Normalization

A slave must be normalized after a cold start or warm start. Normalization enables transmission of data sets and events. If a slave is not normalized, it cannot trigger events. The TRANSM connection element then issues a corresponding error message.

If a master polls a non–normalized slave, the POLL connection element signals a corresponding error.

Set clock

When a 07 KP 90 R202 is switched on, its software clock is set to 0:00 hours. You can use the CLOCK connection element to set the clock of the RCOM master and clocks of all slaves to the same time. This is important for evaluating time stamps in the case of event–driven data transmission.

You should set the clock after the cold start, and clock setting should be repeated if necessary cyclically (e.g. every 24 h).

3.4 The RCOM system time

The 07 KP 90 R202 contains a software clock for generating time stamps and for other purposes, and you can also use this software clock in your PLC program. The time information is made available at connection element RCOM at out puts NT, Std, Min and Sek and is updated there approx. every 5 seconds.

The RCOM time starts at 00:00.00 hours when the coupler is switched on.

Connection element CLOCK sets the RCOM clock of the master and sends a Set clock telegram to all slaves (NOD must be set to 255 for this purpose).

Proceed as follows when setting the RCOM time: Read the actual time from the real–time clock and start the CLOCK block in the RCOM master with this time (NOD=255 in order to address all slaves). The new time is now transferred to the RCOM clock in the master and in all s laves, and output NT is set to “1” for approx. 5 seconds. If real–time clocks are also to be used in individual slaves, you can use the edge of NT to set these clocks. The master and all slaves now use the same time.

You should use the CLOCK block even if the RCOM master does not have a real–time clock (e.g. with time 00:00.00 hours). All time stamps in events are then calculated relative to this arbitrary RCOM time.

3.5 Planning procedure

When planning, you should first precisely analyze the required communication relationships in order to avoid subsequent modifications.

You should consider the following questions:

- How many slaves are necessary? Issue the slave numbers (NOD).
- How many data words have to be transmitted for each slave? Define the subdivision of the data into data sets. Issue flag ranges for each data set.
- How must the individual data sets be transmitted? Cyclically? At the request of the PLC program? Event–driven? Define the communication sequences. Do not forget the required system services (e.g. Cold Start and Normalization). Chapter “Examples” shows how such sequences can be implemented.
- How must the PLC program respond to transmission errors? The example program shows one possible solution.

Important planning rules

The following rules must be observed when planning on the 07 KP 90 R202:

- IDT (data set ident) may have values between 1 and 255.
- Only an even number of data words may be transmitted in a data set
- The connection elements may not be skipped once they have been started. Otherwise, this disturbs the logic sequence between the CE and the 07 KP 90 R202; the connection element may block.
- The connection elements may not be started before the RCOM CE has run successfully (initializing the coupler). Otherwise, the connection elements may block.
- Data transmission with READ jobs takes approximately twice as long as an event–driven transmission. Consequently, you should give preference to event–driven transmission in the case of time–critical transmissions.
- Max. 14 words are permitted per data set with event–driven transmission. The time stamp lies directly after the user data in two words. The master must know the number of words transmitted if the master wishes to evaluate the time stamp (plan a fixed length).

3.6 Hardware handshake

The coupler can handle two types of hardware handshake for communication with a modem: half duplex and full duplex.

The two possible duplex modes are selected on the connection element RCOM with parameter “Dupl”.

Hardware handshake relates only to transmission of telegrams or jobs. All characters are accepted and interpreted as valid characters at all times in the receive direction.

Please note that a valid CTS signal must be available before transmission of a telegram. The 07 KP 90 R202 otherwise signals an error. If the modem used does not provide a CTS signal, RTS must be connected to CTS in the cable.

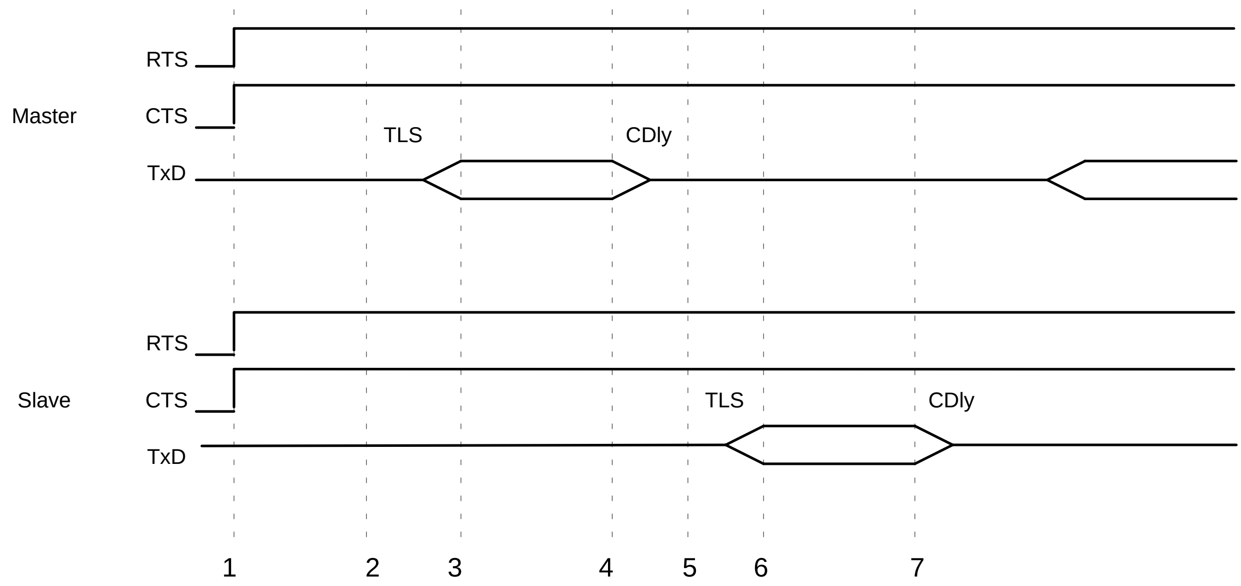
Full duplex

In the case of full duplex mode, the coupler sets its RTS line to “1” after initialization.

Before transmission of characters, it awaits a valid CTS line. The modem may set CTS to zero during transmission in order to stop data flow.

Full duplex should be used for transmission links which provide each transmission direction with its own channel, e.g. modem–zero cables, telephone connections or modem LS–01 of Messrs. Hedin–Tex.

The illustration below shows full–duplex data communication on an 07 KP 90 R202 as RCOM slave.



1 The 07 KP 90 R202s set RTS to 1 during initialization. The modem responds with CTS=1.

2 PLC starts job on 07 KP 90 R202. Coupler waits for TLS.

3 Coupler checks CTS and starts transmission of the job telegram.

4 Telegram is terminated. Coupler waits for CDly.

5 Slave recognizes job and processes telegram. TLS is awaited before transmission of the response.

6 Slave starts transmission of the response telegram.

7 After the telegram has been transmitted, the slave awaits CDly. This terminates transmission.

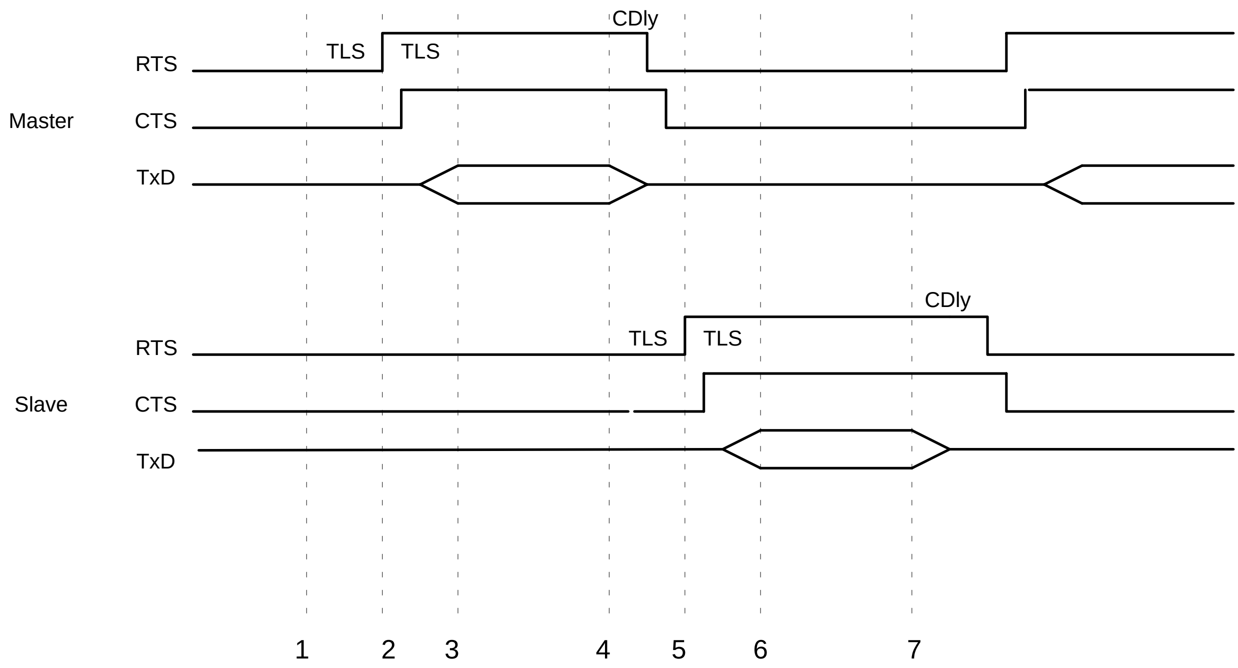
Please note: TLS is always awaited before transmission of a telegram. CDly is awaited after transmission of a telegram. TLS must be >=CDly of the remote station in order to achieve error–free communication.

Half duplex

In half duplex mode, the coupler activates RTS before each telegram and then deactivates it again

Certain modems can thus be switched over from receive mode to transmit mode. This permits a common transmission channel to be used for transmission and reception.

The illustration below is intended to clearly indicate this relationship.



1 PLC starts job on 07 KP 90 R202 (RCOM master). 07 KP 90 R202 awaits TLS.

2 Master coupler sets RTS in order to switch modem to transmit direction. Modem responds within TLS.

3 After TLS expires, the master checks CTS and starts transmission of the job telegram if CTS=1. Otherwise, the job is terminated with an error message

4 Telegram is terminated. Master awaits CDly and then sets RTS=0. Slave processes telegram and awaits the TLS.

5 Slave sets RTS=1. Modem responds within TLS.

6 After TLS has expired, the slave checks CTS and starts transmission of the response telegram.

7 After the telegram has been transmitted, the slave awaits CDly and sets RTS=0. This terminates transmission.

Please note: TLS is always awaited before RTS is set to 1. This permits the communication partner to deactivate the carrier.

After RTS has been set to 1, TLS is awaited again. This permits the modem to activate the carrier (transient recovery).

The following condition must be fulfilled so that both communication partners do not have an activated carrier at any time:

Own TLS > CDly of the remote station.

Delay times

Delays which may contribute to increasing transmission reliability can be planned before and after the telegram both in the case of Full duplex mode and in the case of Half duplex mode.

Parameter TLS (“line stab. time”) indicates the time which is awaited before transmission of a telegram and after activation of the modem carrier with CTS=1 (in the case of Half duplex only).

Parameter CDLY (“carrier delay”) indicates the delay after the telegram.

The following condition must be observed in the case of Full duplex and Half duplex mode in order to guarantee reliable transmission:

Own TLS > CDLY of the remote station.

The two delay times are entered as a number of characters (duration of transmission of a character at the given baud rate) so that longer delays result in the case of lower baud rates.

Since the internal clock runs only with the clock of 10 ms, only multiples of 10 ms are practical. Example:
Baud rate = 9600 baud –> 1 character = approx. 1 ms, practical time values: 10, 20, 30 etc.
Baud rate = 4800 baud –> 1 character = approx. 2 ms, practical time values: 5, 10, 15 etc.

Recommended values for TLS and CDLY are as follows at 1200 baud:

- with half duplex: TLS = 3 characters, CDLY = 2 characters
- with telephone connections: TLS = 2 characters, CDLY = 2 characters
- with full duplex (multidrop or point–to–point): TLS = 2 characters, CDLY = 0 characters.

Please note that these values depend very greatly upon the modem used and should be determined experimentally, in particular in the case of half–duplex links.

3.7 Special modems

Integrated RS–485 interface

The RS–485 interface integrated into the 07 KP 90 R202 can be used for connecting several communication modules for max. 500 meters around.

Therefore the following parameters are to be set (example for 1200 and 9600 Baud):

– Baud=9600
  Pari=1
  Dupl=0
  TLS=20
  CDly=12
  CTO=40
  TAT=2000
  Mdm=2
  NoPr=1

– Baud=1200
  Pari=1
  Dupl=0
  TLS=2
  CDly=1
  CTO=40
  TAT=2000
  Mdm=2
  NoPr=1

23 WT 90

The 23 WT 90 modem can be operated at the 07 KP 90 R202 with the following parameters:

– Baud=1200
  Pari=1
  Dupl=0
  TLS=20
  CDly=0
  CTO=40
  TAT=2000
  Mdm=3
  NoPr=1

Set the switches at the 23 WT 90 modem as follows:

– S1–1: ON
  S1–2: ON
  S1–3 ON

  S2–1: ON
  S2–2: OFF
  S2–3: ON
  S2–4: OFF
  S2–5: OFF
  S2–6: OFF
  S2–7: OFF
  S2–8: OFF

Note:
If operating more than one slave with the 23 WT 90, error messages appear on the slaves (error 3002). These error messages have to do with technical features of the 23 WT 90. They don‘t result in a handicap of communication as long as no error messages appear on the master.

Logem LGH 9600H1

Using this modem for dial–up operation is described in the Chapter ‘Dial–up modem’.

3.8 Behavior in the case of RUN/STOP and Single Step/Single Cycle

RUN/STOP behavior

After initialization, the 07 KP 90 R202 runs until it is reinitialized. This means that it also accepts jobs if the PLC is stopped.

If you abort a running PLC program, it may happen that you interrupt an RCOM connection element. The 07 KP 90 R202 warns you with a message “command not reset by PLC” or “no PLC reaction” if a connection element does not respond within 2 seconds.

If the coupler is planned as an RCOM slave, it issues the corresponding error message (error message ’application part not ready”, error number 4020 hex.) to the RCOM master.

If the coupler is planned as the RCOM master, the interrupted job is repeated until the PLC resets the job. This will be the case at the latest the next time initialization is performed by the RCOM connection element.

Single Step and Single Cycle

Since the communication processor monitors the reaction of the PLC in the case of incoming and triggered services, it is not possible to run the RCOM connection elements in Single–step or Single–cycle mode. You should thus always run the communication part of the PLC program in “real time”.

4 Using dial–up modems

4.1 General

The 07 KP 90 R202 communication processor is capable of handling communication via the telephone network. For this purpose, it can control Hayes–compatible modems (controlled by AT commands).

If you use dial–up modems for data transmission, you must know the following:

- Use only Hayes–compatible modems.
- Deactivate any MNP options which may be available on the modem. In the case of MNP transmission, the temporal relationship between telegrams is lost, and this impedes transmission.

  For transmission, a physical connection is necessary, which allows the transmission of breaks and binary characters, without loosing the coherence in time (duration of the break signal, intervals of the characters). For most modems these operating mode is called ‘direct mode’.
- In the PLC program, the connection must be set up (DIAL connection element) before transmission of RCOM services and it must then be cleared down again (HANGUP).

When commissioning, you should first attempt to address the modem with the operator command “MOD” (refer to Chapter “Operator”). If you enter the following command

```
OPERATOR> MOD ATI4 <CR>
```

you should see a table of the most important modem parameters. If this is not the case, the modem will probably have been configured incorrectly (baud rate or parity etc.)

Communication sequence on the RCM master

You must observe the following sequence in the PLC program of the RCOM master for data transmission:

- Initialize the 07 KP 90 R202 with CE RCOM. Parameter Mdm=1 (dial–up modems)
- Call the distant station: DIAL. If DIAL is completed and no error is signalled:
- Perform a cold start or warm start at the distant station if required. Cold start and warm start delete the event queue of the called slave. You should **not** use broadcast telegrams with these services but, rather, address the slave explicitly. You must then
- perform normalization. This service **must** always be normalized in order to initialize the protocol mechanism for data transmission. Here as well, you should address the slave explicitly with its number and not use broadcast telegrams. Only if normalization does not signal an error, you may:
- write and read data sets (TRANSM and READ) and
- poll the slave (POLL and RECV).
- You must then place the telephone back on hook (HANGUP).

In the case of any transmission errors which occur, in particular during normalization, you should place the telephone back on hook with HANGUP and start a new dialing attempt.

Communication sequence on RCOM slaves

No special CEs need to be planned in the PLC program for regular data transmission (master calls and starts services) in the case of RCOM slaves.

The slave “picks up” the telephone when it rings and then awaits telegrams from the master. If no further telegrams arrive after a waiting time has expired (HANGUP time), the slave “hangs up” automatically.

Event transmission: DIAL in slave

The slave can call the master if it wishes to transmit events to the master.

Only a DIAL CE needs to be started in the slave for this purpose. Communication runs as follows:

- Slave calls master with DIAL CE.
- Master answers the telephone.
- After a brief waiting time, the master starts to normalize all slaves planned in the telephone directory. Since only one slave can have called, only the caller will answer correctly.
- The master now **automatically** polls the recognized slave until it signals that the event queue is empty or until the number of polls defined by parameter MaxP is reached. Data sets arriving are transferred to the RECV CEs in the master.
- The master then “hangs up”.
- The slave also “hangs up” after a waiting time (“Hangup time”).

Please note:

- The master automatically attempts to poll all slaves when it is called. No POLL connection element is required for this purpose. You use the POLL CE only if the master calls the slave.
- No HANGUP CE needs to be planned in the slave since the slave does not know when transmission is completed. The slave “hangs up” automatically after expiry of the “hangup time” (refer to Section “Timeouts”).

4.2 Timeout

Whether RCOM telegrams are also actually transmitted with the telephone off hook is strictly monitored both in the master and in the slave.

If no telegrams have been received by the slave or if no services have been started in the master after expiry of the “hangup time, the telephone is placed on hook again.

This prevents “wrong callers” who have dialed the wrong number for instance from blocking the telephone continuously.

Since the RCOM slave is not capable of completely monitoring the status of the modem (control **only** with DIAL), you should select a short “hangup time” for the slave, e.g. 10 seconds.

On the master, the timeout should never respond since the modem can be monitored completely by the PLC program (DIAL and HANGUP). Consequently, you can select a long “hangup time”, e.g. 30 seconds.

4.3 Modem parameters

Correct setting of the modem is very important in order to achieve error–free communication.

Certain parameters can be stored in a non–volatile memory in the modem. All other parameters can be stored on the 07 KP 90 R202 in the Init string of the modem setup. They are then transferred to the modem when the 07 KP 90 R202 is initialized.

The following parameters **must always** be configured so that the 07 KP 90 R202 can control the modem correctly:

– Commands are echoed by the modem
– Acknowledgements from the modem on
– Acknowledgements in plain text
– Break does not clear down connection
– MNP options off
– Data compression off (direct mode)
– Dialing with DTR off
– RTS/CTS handshake between modem and 07 KP 90 R202
– Automatic call acceptance off
– Escape character: ’+’

4.4 Setup

After initialization, the 07 KP 90 R202 checks whether a dial–up modem is used on the coupler (parameter Mdm=1 on the RCOM connection element). If so, the SETUP file and the telephone directory are read. The 07 KP 90 R202 then expects a Hayes–compatible telephone modem on the RCOM interface.

```
MODEM_INIT          ATZ^M~~ATI4^M~
DIAL_PREFIX_1       ATDT
DIAL_PREFIX_2       ATDP:
DIAL_PREFIX_3       not used
DIAL_PREFIX_4       not used
DIAL_PREFIX_5       not used
DIAL_PREFIX_6       not used
DIAL_SUFFIX         ^M
CONNECT_ANS         CONNECT
MODEM_RING          RING
NO_CARRIER          NO CARRIER
COMMAND             ~~+++~~
MODEM_ANSWER        ATA^M
MODEM_HANGUP        ATH^M~~~~~~
MAX_RING_TIME       70
MAX_NO_OF_CALLS     3
CALL_DELAY          2
HANGUP_TIME         30
```

Figure 4.1: Example of Setup file

The Setup file contains all commands required for controlling the modem. It is pre–configured for the Logem LGH 9600H1 modem and should not be changed if using these modems.

The Setup file is a text file. You can enter it line–by–line via the “CONSOLE” commissioning interface of the 07 KP 90 R202.

The file has a line–by–line structure, i.e. one line defines one parameter.

Caution: Do not forget any of the lines in the file! The keywords at the start of the line may not be changed or omitted

Two characters in the file have a special significance:

~ If you see this character (tilde) in a string, this means that communication to the modem is stopped for one second (waiting time). Certain modems need a pause after certain commands. You should use this character in such cases.

^ This character (circumflex) is used to precede a CONTROL character. Thus, for instance: ^M = carriage return.

The parameters are as follows:

1. **MODEM_INIT** is a character string transmitted to the modem upon initialization. Generally, this command resets the modem (e.g. with ’ATZ’) and then changes default settings.
2. **DIAL_PREFIX_1..6** are character strings which precede the telephone number. Various prefixes are possible, e.g. in order to permit switchover between dial pulsing and DTMF. The number of the prefix is specified in the telephone directory for the relevant telephone number.
3. **DIAL_SUFFIX** is a character string sent after the telephone number to the modem.
4. **MODEM_RING** is the character string which the modem sends to the 07 KP 90 R202 when the telephone rings.
5. **NO_CARRIER** is the character string which the modem sends if the partner hangs up or if the line has been interrupted.
6. **COMMAND** is the character string which sets the modem to command mode. The character string is transmitted to the modem before the string for hanging up. Certain modems require a brief pause after this. Use “~” in the character string for this.
   If COMMAND contains no string (entry ’COMMAND ’), the 07 KP 90 R202 uses the RTS line as DTR signal for hanging up. This operating mode is described in the Chapter ’Operation with the LGH 9600H1’.
7. **MODEM_ANSWER** is the character string with which the modem accepts an incoming call (off hook).
8. **MODEM_HANGUP** is the character string which causes the modem to hang up. The modem is set to command mode beforehand with the COMMAND character string.
9. **MAX_RING_TIME** is the time in seconds for which the 07 KP 90 R202 waits for a connection to the partner. The time is calculated as of transmission of the dial string.
10. **MAX_NO_OF_CALLS** is the maximum number of attempts made to call the partner. The partner is then recognized as non–reachable (error message in the DIAL block).
11. **CALL_DELAY** is the waiting time in seconds between two call attempts.
12. **HANGUP_TIME** is the maximum time which may elapse without services arriving from the PLC or via the RCOM interface. If this time is exceeded, the 07 KP 90 R202 hangs up (refer to section Timeout).

4.5 Telephone numbers

The telephone numbers of the individual slaves are also stored in a file. The numbers are read when connection element DIAL is being processed.

The file is a normal text file.

The file has a line–by–line structure, i.e. the telephone number for slave 1 can be found in line 1 etc. Please enter a blank for slave numbers not used. The entry is then deleted.

A line consists of three parts:

- Number of the prefix
- Telephone number
- Any comment

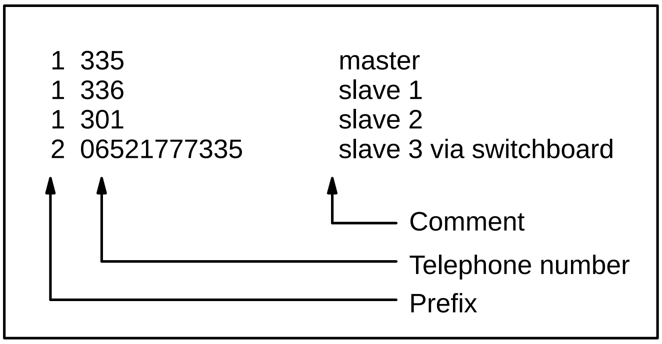


Figure: Example telephone directory

If a slave is called with block DIAL, parameter NOD must be converted to an effective telephone number. The telephone directory is used for this.

The prefix number indicates the number of the DIAL_PREFIX in the Setup file which must be used for this slave. Three parts are transmitted to the modem overall:

- the selected prefix
- the actual telephone number
- the DIAL_SUFFIX from the Setup file.

For example, let us assume that DIAL_PREFIX_1 is "ATDT" and that DIAL_SUFFIX is "^M". When slave 1 is dialed (NOD=1 on DIAL CE),

"ATDT335<return>"

is sent to the modem.

The following rules apply to the structure of the telephone directory:

- The file is line–oriented. Each slave corresponds to one line, i.e. parameter NOD on the DIAL connection element corresponds to the line number.
- A line consists of three entries:
  - Prefix number
  - Telephone number
  - Comment
- The entries are separated by blanks (at least one blank).
- The prefix number must be between 1 and 6.
- The telephone number can be any number.
  Important: Do not separate the digits. Use only digits in the telephone number.
- The comment may consist of any characters.

4.6 Entering the files

The files are entered as follows via the commissioning interface:

- Enter "PHONE" or "SETUP" on the console, dependent upon the actual file which you wish to change. If you want to change a specific entry, you can also enter "PHONE 3" or "SETUP 5" etc. Editing then starts at the specified line number.
- On the console you will see a line of the file with prefixed line number and a colon. In the next line, you will see ">" beneath the colon. Example (line 4 of the display):

```
phone 1: 2 06221777335  Slave 1
phone 1> _
```

- If you want to change the line, simply enter the line again and terminate the entry with <ENTER>. If you do not want to change the line, simply press <ENTER>.
- If you want to delete your entry and enter a line again, press CONTROL–C.
- You can copy parts from the old entry if you have not yet changed the line: in order to do this, press CONTROL–B. A word is copied from the old entry (this is helpful with the SETUP table)
- If you wish to delete an entry in the telephone directory, enter only a blank and then press <ENTER>.
- If you want to quit editing before the last line, enter CONTROL–Z.

The files are stored unchecked in the RAM and remain stored until the next time the coupler is switched off. If you want to copy the files from the RAM to the EEPROM, enter "SAVE". Programming may take up to 40 seconds.

If you want to create the files illustrated in this chapter, enter "DEFAULT". The 07 KP 90 R202 then generates these example files which you can change and then store with SAVE.

Important: Changed Setup data and telephone numbers are lost when you switch off the 07 KP 90 R202 if you have not saved them to the EEPROM beforehand with "SAVE"!

4.7 Operation with LGH 9600H1

Special features

In ’direct mode’ the LGH 9600H1 (in the following designated as LGH) cannot react on the sequence ’+++’. So switching from online to command mode for hang–up is not possible.

For remedy, there is an operating mode of the 07 KP 90 R202, which enables the hang–up by means of the DTR signal (S1/108) of the modem. If the DTR is switched from active to passive, the modem hangs–up immediately and switches to the command mode.

Call–acceptance and making requested connections is only possible with active DTR signal.

Because the 07 KP 90 R202 is not able to make an independent DTR signal available, the RTS signal is used therefore (operating mode ’RTS as DTR’). So the cable for connecting the LGH to the 07 KP 90 R202 is as follows:

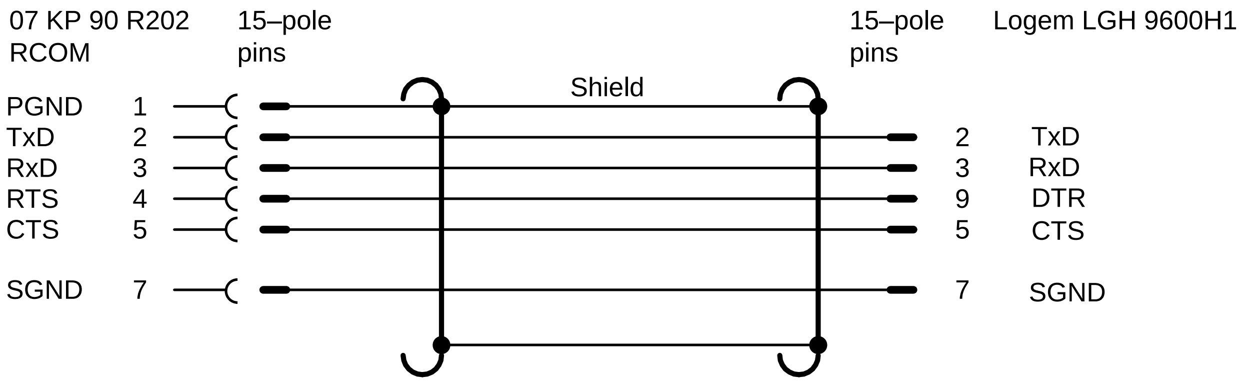


Figure: Connecting Logem LGH 9600H1 to 07 KP 90 R202

Settings at the 07 KP 90 R202

If no character is configured in the setup string ’COMMAND’, the operating mode ’RTS as DTR’ is used. In that case please pay attention leaving the string MODEM_HANGUP also empty, because for hang–up the DTR signal is used.

This setup is also stored as default setup in the 07 KP 90 R202 and can be called at any time with the command ’DEFAULT’.

```
Operator> default
Operator> show setup
 0: MODEM_INIT       ATZ^M~~
 1: DIAL_PREFIX_1    ATDTNT
 2: DIAL_PREFIX_2    ATDPNT0:
 3: DIAL_PREFIX_3    not used
 4: DIAL_PREFIX_4    not used
 5: DIAL_PREFIX_5    not used
 6: DIAL_PREFIX_6    not used
 7: DIAL_SUFFIX      ^M
 8: CONNECT_ANS      CONNECT
 9: MODEM_RING       RING
10: NO_CARRIER       NO CARRIER
11: COMMAND
12: MODEM_ANSWER     ATA^M
13: MODEM_HANGUP
14: MAX_RING_TIME    70
15: MAX_NO_CALLS     3
16: CALL_DELAY       7
17: HANGUP_TIME      30

OPERATOR>
```

Figure: Default Setup

Settings at the LGH 9600H1

When starting, the LGH is to be set to the following parameters permanently, e.g. with a terminal connected directly to the LGH:

```
at&v
Version 2.02  D
F2 E1 L1 M1 Q0 X4 V1 P \Q2 \G0 \A1 \C0 \L0 \N1 \X1 \K0 \B3 %C1 %E1 %M0 &L0 &I0
&Y0 &X0 &G0 &M0 &C2 &D2 &R1 &S1
S00=000 S01=000 S02=043 S03=013 S04=010 S05=008 S07=100 S08=002 S10=002 S12=050
S20=255 S26=004 S28=000 S37=001 S39=017 S40=019 S45=000 S46=060 S50=002 S51=004
S60=000 S61=000 S80=000 S81=000 S100=042 S101=000 S102=000

OK
```

Figure: Settings at the LGH 9600H1

During the input of settings make sure, that after the input of the parameter 'ATF2' the LGH is set permanently to 1200 Baud and that the terminal also is to be set to 1200 Baud.

Store these parameters by means of 'AT&W' in the non–volatile memory of the LGH.

The DIL switches at the LGH are to be set that way, that the basic setting 0 is set in the software mode (all switches S5.1 to S5.5 at the front panel of the unit are set to 'OFF').

1 Operator

1.1 What is ”Operator”?

The second serial interface “CONSOLE” of the 07 KP 90 R202 is also referred to as the operator interface below. The user can enter commands for the communication module at this interface, and the coupler can issue signals and messages via the interface.

There are two main applications for this interface: generation of the telephone directory and tracing the communication sequence during commissioning.

In order to use the CONSOLE interface, you will need a simple terminal or a PC with terminal emulation. The terminal must be set to 9600 baud, 8 data bits, no parity and one stop bit.

1.2 Fault–finding: Debug

The user can display all telegrams and all services processed by a 07 KP 90 R202 on the operator console.

There are two levels: In the first level, a message containing the most important parameters is output for each telegram received or transmitted by a slave and for each service started by the PLC.

In the second level, the coupler issues a signal or message with each important action (including internal operations). Thus, received data words and transmitted telegrams are displayed for instance, as are all status change s in the RCSW (RCOM status word) and all status changes in the event queue. The second debug level will probably be of interest in only a few cases. The first level suffices for testing (debugging) the PLC program.

The two levels can be set with input “Dbg” on connection element “RCOM” (0: no debug messages, 1: debug level 1, 2: level 2). After initialization, the debug level can be set to “1” by entering command

```
OPERATOR> debug 1
```

(corresponding to level 2). Deactivate the messages with

```
OPERATOR> debug 0
```

After commissioning, you should deactivate the messages for normal operation of the coupler (parameter Dbg = 0).

All operator messages have the following appearance:

```
–typ–I–identification text
–typ–W–identification text
–typ–E–identification text
–typ–F–identification text
```

Where:

- typ: Three letters indicating the origin of the message, i.e. “MST” for services performed by a master, “TEL” for telegrams received by a slave, “RPL” for responses transmitted by a slave, etc.
- I/W/E/F provides information on the type of message:

’I’ “I” is information serving to trace the sequence (debug levels 1 and 2)

’W’ is a warning which occurs if telegrams for which no CE is planned arrive for instance. Warnings do not disturb the sequence in the coupler but do indicate a planning error.

’E’ is an error message of the coupler, i.e. if an addressed slave does not respond. Errors disturb the coupler when processing the current service but can frequently be remedied by repeating the service. LED ERR lights in the event of an error. If it has been possible to remedy the error, the LED goes out again.

’F’ is a fatal error which cannot be remedied. Communication via RCOM and to the PLC is aborted by the coupler and only operator entries are then possible. LED ERR lights steadily and the RUN LED goes off. The coupler can be reactivated only by Reset (key or power off/on).

- “Identification” is an abbreviation of the error message.
- “text” contains the actual message in plain text.

1.3 Operator commands

The following operator commands can be entered on the console:

HELP
Outputs a help text on the available commands.

TIME
Outputs the current RCOM system time.

EVENT
Outputs a table with the events present in the event queue. The table contains an event number, the event type (40: event with data set, 41: cold start event), data set number and length and the time when the event occured.

RCOM
Outputs the table with the current RCOM parameters. **IMPORTANT**: All times are specified in milliseconds. Some of the times are increased slightly by the coupler.

RCSW
Outputs the RCOM status word (RCSW).

SHOW SETUP
Outputs the setup data. Refer to section "Setups for dial–up modems".

SHOW PHONE
Outputs the telephone directory. Refer to section "Setups for dial–up modems".

PHONE <n>
Editing the telephone directory as of entry "n". If you omit "n", editing is carried out as of the first entry. Refer to section "Setups for dial–up modems".

SETUP <n>
Editing the setup data as of entry "n". If you omit "n", editing is carried out as of the first entry. Refer to section "Setups for dial–up modems".

SAVE
Saving the telephone setup and data on EEPROM. Refer to section "Setups for dial–up modems".

HANGUP
Hangs up the telephone (only if a dial–up modem is connected).

DIAL n
Dials the telephone number stored under entry "n" in the telephone directory (only if a dial–up modem is connected).

MOD string
Sends "string" to the modem (only if a dial–up modem is connected). The responses of the modem are displayed. Example: "MOD AT14 <ENTER>" –> the modem responds with a table of the most important parameters.

DEBUG <n>
Sets the debug level to "n". This remains valid until the next initialization by the PLC. If you omit "n", the current debug level is displayed.

ABB STOTZ-KONTAKT GmbH
Eppelheimer Straße 82 Postfach 101680
D-69123 Heidelberg D-69006 Heidelberg

Telephone +49 6221 701-0
Telefax +49 6221 701-1111
E-Mail desst.helpline@de.abb.com
Internet http://www.abb.de/sst