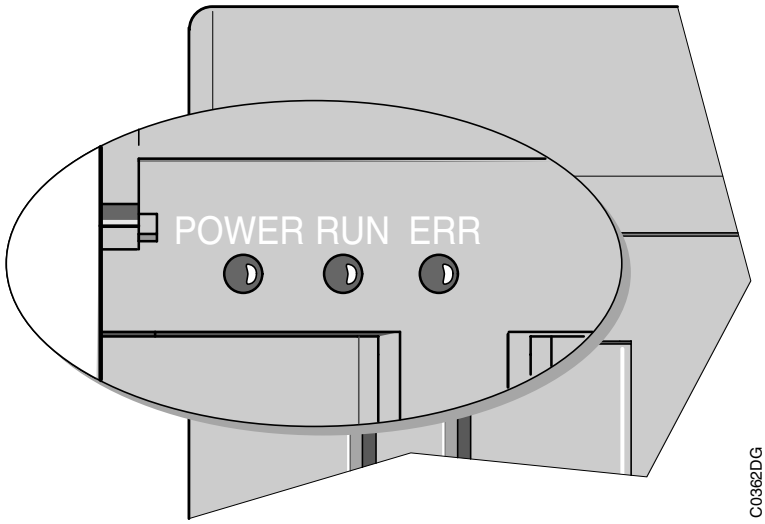
System Description

Advant Controller 31

Intelligent Decentralized Automation System

System Technology
40/50 Series



ABB

Contents

1 System Technology of basic units series 40..50 3

1.1 Introduction 3

1.2 Operands of the basic units S40..50 3
1.2.1 Inputs S40..50 / %I - area 5
1.2.2 Outputs S40..50 / %Q - area 6
1.2.3 Internals S40..50 / %M - area 7
1.2.4 The historical values of S40..50 8
1.2.5 Variables and their physical memory addresses in S40..50 9
1.2.6 System constants 12
1.2.7 System and diagnosis flags 24

1.3 Starting the controller S40..50 27
1.3.1 Terms S40..50 27
1.3.2 Initialization and backup of data areas S40..50 28
1.3.3 Initialization of the central units S40..50 30

1.4 Operation of the central units S40..50 31
1.4.1 Functional diagram 31
1.4.2 Program execution 32
1.4.3 The central unit cycle time S40..50 33
1.4.4 Calculation of a cycle time S40..50 34
1.4.5 Bus refresh times 36
1.4.6 Central unit reply time 36

1.5 Adressing and data exchange with CS31 system bus 37
1.5.1 Introduction / structure examples 37
1.5.2 Addressing of input/output variables 40
1.5.3 Addressing central units masters or stand-alones with extensions 41
1.5.4 Addressing slave central units or remote extension units on the CS31 bus 42
1.5.5 Summary of addressing 50

1.6 I/O configuration for S40..50 52
1.6.1 Configuration with function block CS31CO 52
1.6.2 Configuration of analog extensions XM 06 B5 and XE 08 B5 52

1.7 Diagnosis 58
1.7.1 Introduction in diagnosis 58
1.7.2 Types of errors detected 58
1.7.3 Detection of errors 59
1.7.4 Read the PLC status 61
1.7.5 Description of the diagnosis variables 62

1.7.6 Meaning of the contents of the error word flags 63
1.7.7 Programming examples for error handling 66

2 Communication with S40..50 68

2.1 Network communication with the incorporated MODBUS® interface 68
2.1.1 Introduction 69
2.1.2 MODBUS® protocol description 70
2.1.3 Configuration of the MODBUS® interface 72
2.1.4 Programming MODBUS® 75

2.2 Point to point communication with the incorporated ASCII interface COM1 80

2.3 Point to point communication with the programming protocol 81
2.3.1 Programming protocol via the CS31 connection 82
2.3.2 Protocol description 83

3 List of function for central units S40..50 85

3.1 List of functions S40..50 85

4 Index 89

1 System Technology of basic units series 40..50

1.1 Introduction

The programming and testing software 907 AC 1131 is a complete development environment for the ABB Advant Controller system of 90 (S90) and 40..50 (S40..50) series.

The main difference between PLCs of the 90 series and the 40..50 series is the operating system. In the 90 series, the 907 AC 1131 programming system and the multi-tasking run-time system belong inseparably together. When compiling a project, processor code is generated for the 90 series. The basic units of the 40..50 series still work with the EBS operating system. With the 40..50 series, the project **projekt.pro** is translated into the IL code of the EBS operating system. It is possible, to look at the translated code by opening the ASCII files **projekt.cod** or **projekt.adr.** The code contained inside them is sent to the PLC.

In this part of the documentation the system technology for the 40 and 50 series is described. In the file "Peculiarities_e.pdf" (Volume 8 of the documentation), the main differences and restrictions are described from the point of view of the 907 AC 1131 programming system.

1.2 Operands of the basic units S40..50

The peculiarities of the operands of the 40..50 series basic units are described in the file Peculiarities_e.pdf under "2.2 Variables of S40..50".

There are 5 types of variables in the user program:

- ⇨ The physical, binary or analog Input / Output variables
- ⇨ The internal bit, word or double word variables used in the user program for the intermediate calculations.
- ⇨ The indirect bit, word or double word constants.
- ⇨ The chain steps: a chain step enables running sequential operations. Each chain step enables the definition of a step. One single step may be active at a time.
- ⇨ The historical values: certain functions require the execution of numerous program cycles in order to execute. A historical value is an internal register used by this type of function to store the result, of the function, in the program cycle n-1 during the functions process.

  The historical values are not directly accessible, in the user program, with the 40 and 50 series central units.

Declaration of operands:

The declaration of the operands is done as following:

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

[.] – optional for constants

Note::

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

The following table shows all operands series S40..50 and the addresses in 907 AC 1131:

Variable in PLC	Typ	Address in 907 AC 1131
E00,00 .. E79,15	Inputs BOOL	%IX000.00 .. %IX079.15
A00,00 .. A79,15	Outputs BOOL	%QX000.00 .. %QX079.15
M000,00 .. M099,15 M230,00 .. M255,15	Internal bits BOOL	%MX000.00 .. %MX099.15 %MX230.00 .. %MX255.15
S000,00 .. S125,15	Step BOOL	%MX5000.00 .. %MX5125.15
K0,0 .. K0,1	Constant BOOL	FALSE and TRUE
EW00,00 .. EW68,15	Analog Inputs INT	%IW1000.00 .. %IW1068.15
AW00,00 .. AW68,15	Analog Outputs INT	%QW1000.00 .. %QW1068.15
MW000,00 .. MW099,15 MW230,00 .. MW255,15	Internal words INT	%MW1000.00 .. %MW1099.15 %MW1230.00 .. %MW1255.15
KW00,00 .. KW31,15	Word constants INT	%MW3000.00 .. %MW3031.15
MD00,00 .. MD07,15	Internal double words DINT / TIME	%MD2000.00 .. %MD2007.15
KD00,00 .. KD07,15	Double word constants DINT / TIME	%MD4000.00 .. %MD4007.15

1.2.1 Inputs S40..50 / %I - area

Type	Variables / Address		Description
	From	To	
Binary Inputs	E 00,00 %IX000.0	E 61,15 %IX061.15	CS31 bus binary inputs
	E 62,00 %IX062.0	E 62,07 %IX062.7	Binary inputs on the basic unit
	E 63,00 %IX063.0	E 68,15 %IX068.15	Binary inputs on the central unit extensions
Analog Inputs	EW 00,00 %IW1000.0	EW 61,15 %IW1061.15	CS31 bus analog inputs
	EW 62,00 %IW1062.0		1. potentiometer of the central units
	EW 62,01 %IW1062.1		2. potentiometer of the central units
	EW 62,02 %IW1062.2	EW 62,07 %IW1062.7	Reserved variables
	EW 62,08 %IW1062.8		Seconds (0...59)
	EW 62,09 %IW1062.9		Minutes (0...59)
	EW 62,10 %IW1062.10		Hours (0...23)
	EW 62,11 %IW1062.11		Days of the week (1...7)
	EW 62,12 %IW1062		Days of the month (1...31 / depending on the month)
	EW 62,13 %IW1062.13		Month (1...12)
	EW 62,14 %IW1062.14		Year (00...99)
	EW 62,15 %IW1062.15		Status word: Bit 0: for class 2 error detection Bit 1: used in the slave central units for CS31 bus recognition Bit 3: battery failure detection Bit 8 to 15: used on the master units; they indicate the number of remote units recognized on the CS31 bus.
	EW 63,00 %IW1063.0	EW 68,15 %IW1068.15	Analog Inputs on the extensions of a central unit

Examples for the declaration of inputs:

The declaration of the operands is done as following:

Symbol AT address : Type; (* comment *)

Binary Input E 62,00:
E62_00_Input_0 AT %IX62.0 : BOOL; (* This is Input 0 *)

potentiometer 1 of the central unit:
Poti1 AT %IW1062.0 : INT; (* potentiometer 1 / EW62,0 *)

1.2.2 Outputs S40..50 / %Q - area

Type	Variables / Address		Description
	From	To	
Binary Outputs	A 00,00 %QX0.0	A 61,15 %QX61.15	Binary outputs on the CS31 bus
	A 62,00 %QX62.0	A 62,05 %QX62.5	Binary outputs on the central units
	A 63,00 %QX63.0	A 68,15 %QX68.15	Binary outputs on the extensions of the central units.
Analog Outputs	AW 00,00 %QW1000.0	AW 61,15 %QW1061.15	Analog outputs on the CS31 bus
	AW 62,00 %QW1062.0	AW 62,15 %QW1062.15	Reserved variables
	AW 63,00 %QW1063.0	AW 68,15 %QW1068.15	Analog outputs on the extensions of the central units.

Examples for the declaration of outputs:

The declaration of the operands is done as follows:

Symbol AT address : Type; (* comment *)

Binary output A 62,00:
A62_00_Output_0 AT %QX62.0 : BOOL; (* This is output 0 *)

Analog output on the extension of central unit AW63,00:
A63_00_Analog_1 AT %QW1063.0 : INT; (* Analog output 1 / AW63,0 *)

1.2.3 Internals S40..50 / %M - area

Type	Variablee / Address		Description
	From	To	
Chain steps	S 000,00 %MX5000.0	S 125,15 %MX5125.15	Chain steps
Internal bits	M 000,00 %MX0.0	M 099,15 %MX99.15	Internal bits useable in the program
	M 230,00 %MX230.0	M 254,15 %MX254.15	Internal bits useable in the program
	M 255,00 %MX255.0		2 Hz oscillator variable
	M 255,01 %MX255.1		1 Hz oscillator variable
	M 255,02 %MX255.2		0.5 Hz oscillator variable
	M 255,03 %MX255.3		0.01667 Hz oscillator variable (period = 1 minute)
	M 255,04 %MX255.4	M 255,05 %MX255.5	Reserved variables
	M 255,06 %MX255.6		Switch MODBUS® / active Mode for COM2
	M 255,07 %MW255.7		Watchdog MODBUS® COM2
	M 255,08 %MX255.8		Watchdog MODBUS® COM1
	M 255,09 %MX255.9		Switch MODBUS® / active Mode for COM1
	M 255,10 %MX255.10	M 255,14 %MX255.14	Diagnosis bits
	M 255,15 %MX255.15		Variable, always set to "0" FALSE on startup, which may be used for the detection of the first program cycle
Internal words	MW 000,00 %MW1000.0	MW 099,15 %MW1099.15	Internal words useable in the program
	MW 230,00 %MW1230.0	MW 253,15 %MW1253.15	Internal words useable in the program
	MW 254,00 %MW1254.0	MW 254,07 %MW1254.7	Class 1 error information
	MW 254,08 %MW1254.8	MW 254,15 %MW1254.15	Class 2 error information
	MW 255,00 %MW1255.0	MW 255,07 %MW1255.7	Class 3 error information
	MW 255,08 %MW1255.8	MW 255,15 %MW1255.15	Class 4 error information
Internal double words	MD 00,00 %MD2000.0	MD 07,15 %MD2007.15	Double words useable in the program
Bit constants	K 00,00 FALSE	K 00,01 TRUE	Indirect bit constants
Word constants	KW 00,00 %MW3000.0	KW 00,15 %MW3000.15	System word constants reserved for the configuration
	KW 01,00 %MW3001.0	KW 31,15 %MW3031.15	Indirect word constants
Double word constants	KD 00,00 %MD4000.0		System double word constants reserved for the cycle time
	KD 00,01 %MD4000.1	KD 07,15 %MD4007.15	Indirect double word constant

1.2.4 The historical values of S40..50

Historical values are central unit specific internal variables which enable memorizing a functions intermediate results required for numerous cycles.

In fact, certain functions require the result obtained during cycle N-1 so that cycle N will execute correctly (for example: the PI and PIDT1 controllers).

The historical values are not directly accessible in the user program with the 40 and 50 series central units. The maximum number of historical values allowed in a project is 1000 inside main program and 256 inside sub-program.

The historical values of the timer functions are independent of the list of historical values of the other functions. The number of authorized timer functions is unlimeted, however the number of simultaneous timer functions is limited to 42.

The functions and the number of historical values which they use are given in the following table:

Timer functions	
ASV	2 (supplementary memory for the timers limited to 42 variables)
ESV	2 (supplementary memory for the timers limited to 42 variables)
MOA	2 (supplementary memory for the timers limited to 42 variables)
MOK	2 (supplementary memory for the timers limited to 42 variables)
TON	2 (supplementary memory for the timers limited to 42 variables)
TOF	2 (supplementary memory for the timers limited to 42 variables)
TP	2 (supplementary memory for the timers limited to 42 variables)
MOAT	2 (supplementary memory for the timers limited to 42 variables)
PDM	1 (supplementary memory for the timers limited to 42 variables)
Counter functions	
CTU	2
CTUH	2
VRZ	3
CS31 bus functions	
CONFIO1	3
CONFIO4	3
CONFIO8	3
CS31CO	1
Communication functions	
MODBUS®	2
EMAS	1
DRUCK	2
SINIT	1
Controller functions	
PI	3
PIDT1	5
High order functions	
BMELD	3 + number of inputs E
NPULSE	1
UHR	1

1.2.5 Variables and their physical memory addresses in S40..50

This paragraph describes the correspondence between the variables and their physical memory addresses. This information is necessary for certain functions such as COPY, data exchange between CS31 bus master and slave central unit.

The variable address is given by the segment and offset value.

16#0000 (0) is the segment for all variables.

The correspondence between the variable and the offset is given in the following table:
8 bits used 1 byte – 1 word used 2 bytes.

Mapping addressing method:

For the variable with group := 00 and channel := 00 we take the base address ADDD 0 from the table:
VAR 00,00 → ADDR 0 ; i.e. 6400 for E00,00

Bits:

(VAR = type E, A, S, M) (Address of the first selected variable in decimal)
VAR XX,YY = ADDR 0 + (XX * 2)

CAUTION:
For variable type M, you have two area and you have to use M 000,00 = ADDR 0, for the the first area, and M 230,00 = ADDR 0 for the second area.

Words:

(VAR = type EW, AW, KW, MW) (Address of the first selected variable in decimal)
VAR XX,YY = ADDR 0 + (XX * 32) + (YY * 2)

CAUTION:
For variable type KW the ADDR 0 is 28672. For variable type MW you have two area and you have to use MW 000,00 = ADDR 0 for the first area and
MW 230,00 = ADDR 0 for the second area.

Double words:

(VAR = type MD, KD) (Address of the first selected variable in decimal)
VAR XX,YY = ADDR 0 + (XX * 64) + (YY * 4)

Example:

Find the mapping address of variables A 62,00, AW 62,15, MW 240,15 and MD 002,07

A 62,00 = 6912 + (62 * 2) = **7036**
AW 62,15 = 18432 + (62 * 32) + (15 * 2) = **20446**
MW 240,15 = 10368 + (10 * 32) + (15 * 2) = **10718**
MD 002,07 = 15872 + (2 * 64) + (7 * 4) = **16028**

Variables	Physical addresses	
	Hexadecimal	Decimal
E00,00..E00,07 / E00,08..E00,15 E01,00..E01,07 / E01,08..E01,15 E02,00..E02,07 / E02,08..E02,15 .. E61,00..E61,07 / E61,08..E61,15 E62,00..E62,07 / E62,08..E62,15 .. E68,00..E68,07 / E68,08..E68,15	1900 / 1901 1902 / 1903 1904 / 1905 .. 197A / 197B 197C / 197D .. 1988 / 1989	6400 / 6401 6402 / 6403 6404 / 6405 .. 6522 / 6523 6524 / 6525 .. 6536 / 6537
M00,00..M00,07 / M00,08..M00,15 .. M99,00..M99,07 / M99,08 M99,15	1A00 / 1A01 .. 1AC6 / 1AC7	6656 / 6657 .. 6854 / 6855
M230,00..M230,07 / M230,08..M230,15 .. M255,00..M255,07 / M255,08..M255,15	1AC8 / 1AC9 .. 1AFA / 1AFB	6856 / 6857 .. 6906 / 6907
A00,00..A00,07 / A00,08..A00,15 A01,00..A01,07 / A01,08..A01,15 A02,00..A02,07 / A02,08..A02,15 .. A61,00..A61,07 / A61,08..A61,15 A62,00..A62,07 / A62,08..A62,15 .. A68,00..A68,07 / A68,08..A68,15	1B00 / 1B01 1B02 / 1B03 1B04 / 1B05 .. 1B7A / 1B7B 1B7C / 1B7D .. 1B88 / 1B89	6912 / 6913 6914 / 6915 6916 / 6917 .. 7034 / 7035 7036 / 7037 .. 7048 / 7049
S00,00..S00,07 / S00,08..S00,15 S01,00..S01,07 / S01,08..S01,15 .. S125,00..S125,07 / S125,08..S125,15	3D00 / 3D01 3D02 / 3D03 .. 3DFA / 3DFB	15616 / 15617 15618 / 15619 .. 15866 / 15867

Variables	Physical addresses	
	Hexadecimal	Decimal
AW00.00 AW00.01 .. AW00.15 .. AW62.00 .. AW68.15	4800 4802 .. 481E .. 4FC0 .. 509E	18432 18434 .. 18462 .. 20416 .. 20638
KW00.00 .. KW01.00 KW01.01 .. KW01.15 KW31.15	7000 .. 7020 7022 .. 703E .. 73FE	28672 .. 28704 28706 .. 28734 .. 29694
KD00.00 KD00.01 KD00.15 KD07.15	7400 7404 .. 743C .. 75FC	29696 29700 .. 29756 .. 30204
EW00.00 EW00.01 .. EW00.15 .. EW62.00 .. EW68.15	5200 5202 .. 521E .. 59C0 .. 5A9E	20992 20994 .. 21022 .. 22976 .. 23198
MW00.00 MW00.01 .. MW99.00 .. MW99.15	1C00 1C02 .. 2860 .. 287E	7168 7170 .. 10336 .. 10366
MW230.00 .. MW230.15 .. MW255.15	2880 .. 289E .. 2BBE	10368 .. 10398 .. 11198
MD00.00 MD00.01 .. MD00.15 .. MD07.15	3E00 3E04 .. 3E3C .. 3FFC	15872 15876 .. 15932 .. 16380

1.2.6 System constants

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

The system constants are declared in the export file S40_Syconst.exp. When creating a new project using "File/New", this file is automatically stored in the „Resources“ menu and there under „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, MODBUS on CS31 connection)
KW 00,01 : %MW3000.1	Initialization: Binary flag area
KW 00,02 : %MW3000.2	Initialization: Word flag area
KW 00,03 : %MW3000.3	Initialization: Double word area
KW 00,04 : %MW3000.4	Initialization: Step chain flags area
KW 00,05 : %MW3000.5	Initialization: Historical values
KW 00,06 : %MW3000.6	Application mode of serial interface COM1
KW 00,07 : %MW3000.7	PLC reaction to class 3 errors
KW 00,08 : %MW3000.8	**without function**
KW 00,09 : %MW3000.9	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	**without function**
KW 00,13 : %MW3000.13	**without function**
KW 00,14 : %MW3000.14	**without function**
KW 00,15 : %MW3000.15	**without function**

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:	AC31 module addresses	0-61
- MODBUS® on CS31 connection:	Addresses (0..255)	1100-1355
– Range of values:	-2, -1, 0-61, 1100-1355	
– 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 a slave PLC no. 5:
KW00_00_MAST_SLV AT %MW3000.0 : INT := 5; (* Slave PLC *)

Declaration for a MODBUS Master on the CS31 connection:
KW00_00_MAST_SLV AT %MW3000.0 : INT := 1100; (* MODBUS Master *)

Declaration for a MODBUS Slave no. 5:
KW00_00_MAST_SLV AT %MW3000.0 : INT := 1105; (* MODBUS Slave no. 5 *)

CAUTION:
The central units serie 40 can be used only in standalone mode. They have no CS31 connection.

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
- 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 1.5.4. – Addressing slave central units..):

- The value range for the addresses is 0..61. 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 operation in word area, the sending and receiving data are on the channels 0-7.

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

Meaning of the initialization value **n** of the constant:	
→Initialized binary flag area (set to FALSE / 0)	
n = 0 (Default)	M 000,00-M 099,15 / %MX0.0-%MX99.15 und M 230,00-M 255,15 / %MX230.0-%MX255.15
n = 1-99	M n,00-M 099,15 / %MXn.0-%MX99.15 und M 230,00-M 255,15 / %MX230.0-%MX255.15
n = 100-229	M 230,00-M 255,15 / %MX230.0-%MX255.15
n = 230-254	M n,00-M255,15 / %MXn.0-%MX255.15
n < 0, n > 255	M 255,00-M 255,15 / %MX255.0-%MX255.15

Example:

KW 00,01 := 52 and battery is charged:

KW00_01_INIT_M AT %MW3000.1 : INT := 52; (* Initialization of binary flag *)

Flags initialized with 0:	M 52,00-M 99,15 / %MX52.0-%MX99.15 and M 230,00-M 255,15 / %MX230.0-%MX255.15
Flags buffered:	M 00,00-M 51,15 / %MX0.0-%MX51.15

Comment:

The bits M 255,00-M 255,15 / %MX255.0-%MX255.15 are system flags
(see chapter 1.2.7 -system and dignosis flags)

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

Meaning of the initialization value **n** of the constant:	
→Initialized word flag area (set to 0)	
n = 0 (Default)	MW 000,00-MW 099,15 / %MW1000.0-%MW1099.15 and MW 230,00-MW 255,15 / %MW1230.0-%MW1255.15
n = 1-99	MW n,00-MW 099,15 / %MW1000+n.0-%MW1099.15 and MW 230,00-MW 255,15 / %MW1230.0-%MW1255.15
n = 100-229	MW 230,00-MW 255,15 / %MW1230.0-%MW1255.15
n = 230-255	MW n,00-MW 255,15 / %MW1000+n.0-%MW1255.15
n < 0, n > 255	no initialization

Example:

KW 00,02 := 52 and battery is charged:

KW00_02_INIT_MW AT %MW3000.2 : INT := 52; (* Initialization of word flags *)

Flags Initialized with 0:	MW 52,00-MW 099,15 / %MW1052.0-%MW1099.15 and MW 230,00-MW 255,15 / %MW1230.0-%MW1255.15
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 constant:	
→Initialized double word flag area (set to 0)	
n = 0 (Default)	MD 00,00-MD 07,15 / %MD2000.0-%MD2007.15
n = 1-7	MD n,00-MD 07,15 / %MD2000+n.0-%MD2007.15
n < 0, n > 7	no initialization

Example:

KW 00,03 := 3 and battery is charged:

KW00_03_INIT_MD AT %MW3000.3 : INT := 3; (* Initialization of double word flags *)

Flags initialized with 0:	MD 03,00-MD 07,15 / %MD2003.0-%MD2007.15
Flags buffered:	MD 00,00-MD 02,15 / %MD2000.0-%MD2002.15

KW 00,04 / %MW3000.4:
Initialization of chain steps

Meaning of the initialization value **n** of the constant:	
→Initialized chain steps (set to step 0)	
n = 0 (Default)	S 00,00-S 125,15 / %MX5000.0-%MX5125.15
n = 1-125	S n,00-S 125,15 / %MX5000+n.0-%MX5125.15
n < 0, n > 125	no initialization

Example:

KW 00,04 := 52 and battery is charged:

KW00_04_INIT_S AT %MW3000.4 : INT := 52; (* Initialization of chain steps *)

Chains set to step 0:	S 52,00-S 125,15 / %MX5052.0-%MX5125.15
Chain steps buffered:	S 00,00-S 51,15 / %MX5000.0-%MX5051.15

KW 00,05 / %MW3000.5:
Initialization of historical values

Meaning of the initialization value **n** of the constant:	
→Initialized historical values (set to 0)	
n = 0 (Default)	Initializasion of all historical values
n < 0, n > 0	no initialization

Example:

KW 00,05 := 1 and battery is charged:

KW00_05_INIT_VW AT %MW3000.5 : INT := 1; (* no initialization of historical values *)

All historical values are buffered.

Note:

A change of the system constants KW 00,01 - KW 00,15 / %MW3000.1 - %MW3000.5 comes into effect immediately, i.e. without cold start or warm start.

The constant is a 5-digit decimal number:

Tenthousands	Thousands	Hundreds	Tens	Ones
RTS signal level	RTS delay time	Communication mode (0..355)		

Setting the communication mode:

The serial port of the basic unit can be used in various modes:

- Programming mode to program and test the central unit (active mode)
- ASCII mode for communicating between the central unit and another ASCII device (passive mode)
- MODBUS® mode for communicating between the central unit and another MODBUS® device.

To use one of these communication modes you must: (see the following table):

- Configure the serial interface with the system constant KW 00,06 / %MW3000.6
- Know the position of the RUN/STOP buttom
- Use the right cable:
  - For programming: 07 SK 50 or 07 SK 52 (Active mode)
  - For ASCII/MODBUS®: 07 SK 51 or 07 SK 53 (Passive mode)

Parameter-selection	**KW 06,00 / %MW3000.6**	**RUN/STOP position**	**Cable**	**Communication mode**
Standard	0, <0, >2	STOP	X	Programming
	<100	RUN	Programming	Programming
	>355		ASCII/MODBUS®	ASCII
Programming	1	X	X	Programming
ASCII	2	RUN	X	ASCII
		STOP	X	Programming
MODBUS®	>99	X	ASCII/MODBUS®	MODBUS®
	<356		Programming	Programming

X: choice with no effect

The communications parameters are defined in accordance with the operating mode:

Operating modet	**Default parameters**	**Modification of parameters**
Programming	9600 bauds no parity 8 data bits 1 stop bit	Parameters non modifiable
ASCII	No default parameters defined	Parameters are defined by using the SINIT function in the user program
MODBUS®	9600 bauds no parity 8 data bits 1 stop bit	Modification of parameters by using the SINIT function in the user program

Communication mode n:

The communication mode of the serial interface COM1 is set by the digits Ones, Tens and Hundreds of system constant KW00,06 / %MW3000.6. Following settings are possible:

n = 0 (Default)	+ 07 SK 50/52 cable	Programming mode (Active mode)
n = 0 (Default)	+ 07 SK 51/53 cable	ASCII mode (Passive mode)
n = 1	+ 07 SK 50/52 cable	Programming mode (Active mode)
n = 2	+ 07 SK 50/52 cable	ASCII mode (Passive mode)
n = 100	+ 07 SK 51/53 cable	MODBUS® master
n = 100 + slave no. 1-255	+ 07 SK 51/53 cable	MODBUS® slave 1-255

Setting the delay time of the RTS signal:

The delay time of the RTS signal is set in the "thousands" digit of the system constant KW00,06 / %MW3000.6. The following settings are possible:

Default value: 0 = the RTS signal delay depends on the number of characters in the buffer

Delay time for MODBUS®:

1000 = 1 character
2000 = 32 ms
3000 = 64 ms
4000 = 96 ms
5000 = 128 ms

Delay time in programming and ASCII mode:

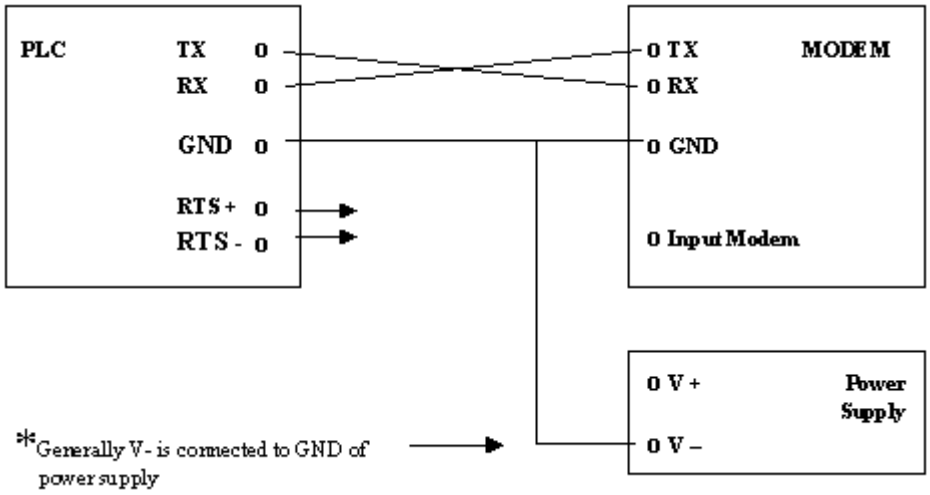
>1000 = 3 characters

Setting the RTS signal level:

On S40..50 the RTS signal can be inverted by setting the "tenthousands" digit in the system constant KW00,06 / %MW3000.6.

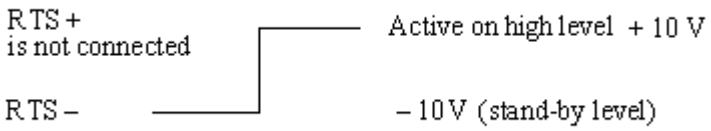
RTS signal level = 0 no invertion (active on high level +10 V)
10000 invertion of the RTS signal level (active on low level –10 V)

The following figure shows the general cabling between PLC and modem:

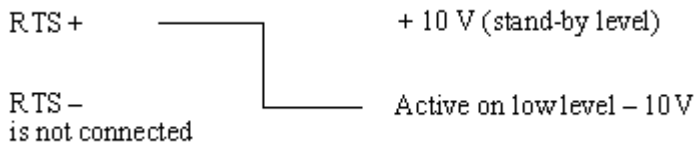


RTS signal with Serie 40:

By setting **"no invertion / active on high level"**:



By setting **"Invertion / active on low level"**:



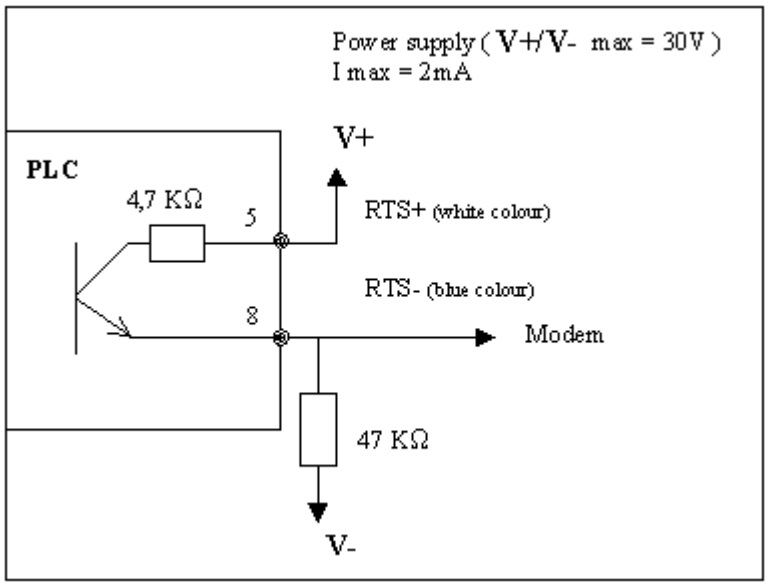
RTS signal with serie 50:

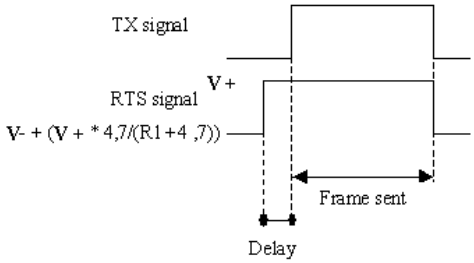
The settings for the RTS signals of the 50 series on version inferior with:

Designation of product	Index of version
07 KR 51 – 24 V DC	Q22 and Q30
07 KR 51 – 120/230 V AC	Q22 and Q30
07 KT 51 – 24 V DC	P22 and P30

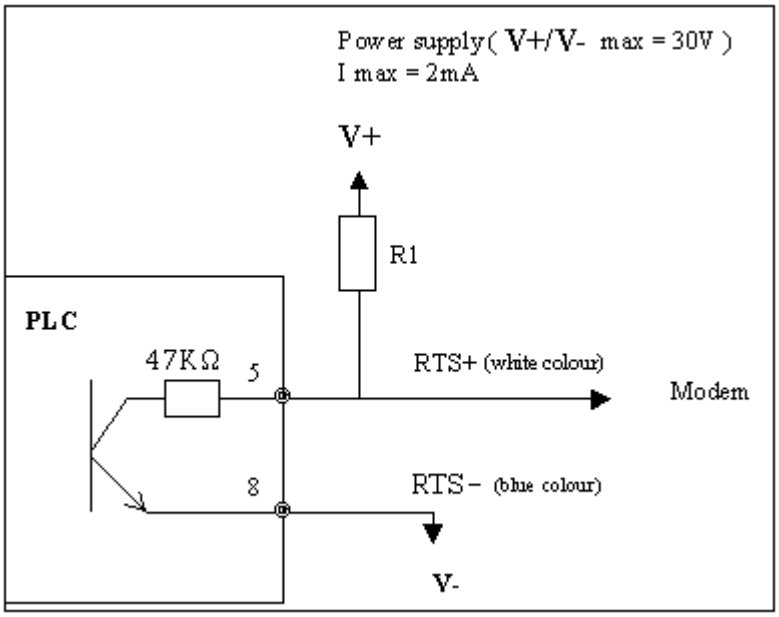
You can find the index of version on the label on the left side of the 50 series central unit after the designation of product, for example 07 KR 51-P30.

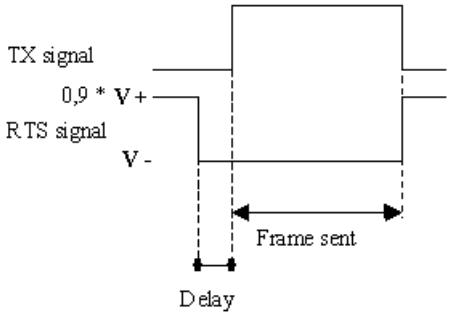
By setting **"No invertion / active on high level"** :





By setting **"Invertion / active on low level"** :



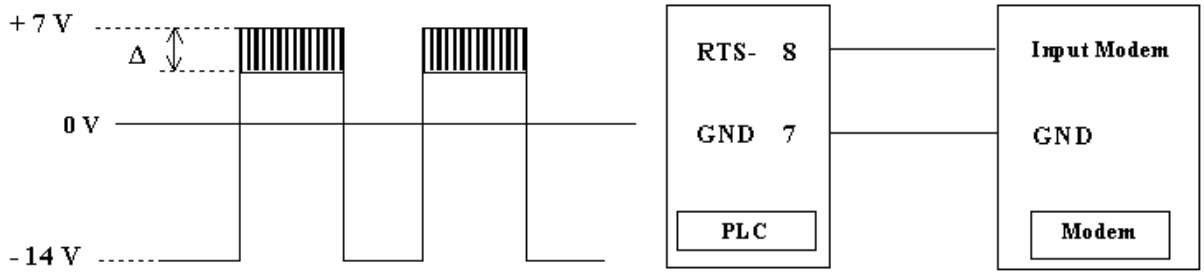


The settings for the RTS signals of the 50 series starting from version:

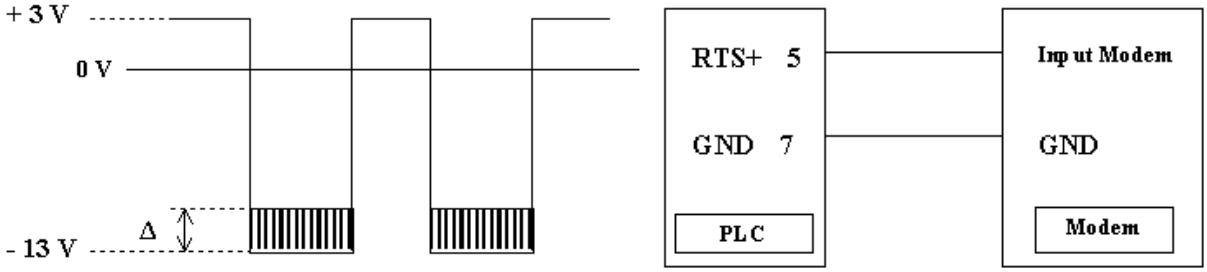
Designation of product	Index of version
07 KR 51 – 24 V DC	Q22 and Q30
07 KR 51 – 120/230 V AC	Q22 and Q30
07 KT 51 – 24 V DC	P22 and P30

You can find the index of version on the label on the left side of the 50 series central unit after the designation of product, for example 07 KR 51-P30.

By setting **"No invertion / active on high level"** :



By setting **"Invertion / active on low level"** :



Calculate initialization value of the system constant:

The initialization value of the system constant KW 00,06 / %MW3000.6 is the summery of the values for communication mode, RTS delay time and RTS signal level:

Init value := communication mode + RTS delay time + RTS signal level

Example:

Set COM1 to MODBUS slave no. 217 with a RTS delay time of 64 ms and RTS signal active on low level:

Communication mode: 100 + 217 = 317
RTS delay time: 3000
RTS signal level: 10000

Initialization value := 317 + 3000 + 10000 := 13317

KW06_00_MODE_SST AT %MW3000.6 : INT := 13317; (* MODBUS slave 217 *)

Changing the system constants takes effect immediately.

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

The AC31 basic units feature a dignosis system designed to localize errors quickly and efficiently. This diagnosis system is divided into 4 error classes:

- Class 1: fatal errors
- Class 2: serious errors
- Class 3: light errors
- Class 4: warning

If class 1 or 2 error occurs, the program is aborted or not started. If a class 4 error occurs, the program is not interrupted.

If a class 3 error occurs, the program may be aborted or continued:

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 KW 00,07 / %MW3000.7 takes immediately effect, i.e. without cold start or warm start.

Example:

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

KW 00,09 / %MW3000.9:
Starting up the AC31 system serie 50

This system constant takes effect if the basic unit is configured as the bus master by stating up the AC31 system after power ON, warmstart or coldstart. On serie 40 the system constant has no effect.

Meaning of the initialization value **n** of the constant:	
n = 0 (Default)	The user program is started. The basic unit not take any notice of whether the remote units are initialized and adopted into the CS31 bus cycle.
n = 1..31	The user program is not started until at least n remote modules have been initialized and adopted into the CS31 bus cycle.

Changes to this system constant KW 00,09 / %MW3000.9 take effect on the next warm- or coldstart.

Example:

```
KW 00,09 := 0
KW00_09_HOCHFAHR AT %MW3000.9 : INT := 0; (* start user program *)
```

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

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

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 areat:	Sending 2..15 bytes	0, 2..15
– For use in word area:	Sending 1..8 words	100, 101..108
– Range of values:	0, 2..15 and 100, 101..108	
– Default value:	0	

Changes to this system constant take effect on the next warm- or coldstart.

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 constant:

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 2..15 bytes	0, 2..15
– For use in word area:	Sending 1..8 words	100, 101..108
– Range of values:	0, 2..15 and 100, 101..108	
– Default value:	0	

Changes to this system constant take effect on the next warm or coldstart.

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.

KD 00,00 / %MD4000.0:
Cycle time of the central unit

The central unit program executes in cycles for the duration which is defined in the system constant "Cycle time". The value is given in millisecondss.

- For a master cental unit:
  the cycle time is between 0 and 100 ms, where only multiples of 5 ms are authorized.

- For a slave or stand alone central unit:
  the cycle time is between 0 and 250 ms.

When the 0 value is selected, the cycle time is the minimum time which the central unit requires for each cycle. In this case the cycle time is not constant.

The default value is 10 ms.

For more informations refer to chapter 1.4 Operation of the central units S40..50

Example:

Cycle time KD 00,00 := 30 ms

```
KD00_00_Zykluszeit AT %MD4000.0 : DINT := 30; (* Cycle time 30 ms *)
```

1.2.7 System and diagnosis flags

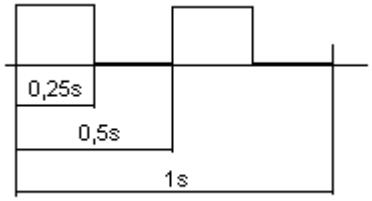
The following areas are reserved for system flags and system flag words. It is also not allowed to use flags or flag words from this areas which are not yet seized for other purposes.

Flags:	M255,00 – M255,15	%MX255.0 - %MX255.15
Flag words:	MW254,00 – MW255,15	%MW1254.0 - %MW1255.15

The system and diagnosis flags are declared in the export file S40_Fehlermerker.exp. When a new project is being created with „File / New“, this file is automatically stored in the „Resources“ menu under the „Global Variables“ object. Otherwise it 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 0,01667 Hz (Period approx. 1 Minute (1/64 Hz))



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

M255,6 / %MX255.6: Switch MODBUS® / Active Mode for COM2

This system flag can be used to change the protocol on the CS31 bus connection of central units serie 50 (COM2) between MODBUS® and Programming mode (Active mode). It is necessary to be already in MODBUS® slave configuration on CS31 connection (COM2).

M255,06 / %MX255.6 := FALSE	MODBUS®
M255,06 / %MX255.6 := TRUE	Programming mode (Active mode)

M255,7 / %MX255.7: Watchdog MODBUS® COM2

The binary flag M255,7 / %MX255.7 can be used as a watchdog for the MODBUS® Slave on the CS31 bus connection (COM2). This binary flag is set to TRUE when the slave sends answer to the master.

M255,8 / %MX255.8: Watchdog MODBUS® COM1

The binary flag M255,8 / %MX255.8 can be used as a watchdog for the MODBUS® Slave on the serial interface COM1. This binary flag is set to TRUE when the slave sends answer to the master.

M255,09 / %MX255.9: Switch MODBUS® / Acive mode for COM1

This system flag can be used to change the protocol on the serial interface COM1 between MODBUS® and programming mode (Active mode).

M255,09 / %MX255.9 := FALSE	MODBUS®
M255,09 / %MX255.9 := TRUE	Aktive Mode (Programmiermodus)

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 1.7 "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 FALSE ("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 TRUE ("1"), allows to determine whether the user program was restarted or not.

MW254,00-MW255,15 /%MW1254.00-%MW1255.15: Fehlermeldungen der SPS:

The detailed messages of the error flags are contained in the area MW254,00-MW255,15 / %MW1254.0-%MW1255.15. Refer to chapter 1.7 "Diagnosis" for a detailed description of the error messages.

EW 62,08-EW 62,14 / %IW1062.8 – IW1062.14::
Real-time clock:

The clock parameters are accessible in the following variables:

EW 62,08 / %IW1062.8	seconds (0..59)
EW 62,09 / %IW1062.9	minutes (0..59)
EW 62,10 / %IW1062.10	hours (0..23)
EW 62,11 / %IW1062.11	day of the week (1..7)
EW 62,12 / %IW1062.12	day (1..31/depending on the month)
EW 62,13 / %IW1062.13	month (1..12)
EW 62,14 / %IW1062.14	year (0..99)

EW 62,15 /%IW1062.15:
CS31 status word

Bit 0 = 1 :	no CS31 error of class 2 present
Bit 1 = 1 :	PLC is added to the CS31 bus cycle (only relevant for use as slave)
Bit 2 :	not used
Bit 3 = 0 :	battery failure
Bit 4...7 :	not used
Bit 8..15 :	maximum number of modules which were detrected on the CS31 system bus up to the current point in time(only relevant for use as master)

1.3 Starting the controller S40..50

1.3.1 Terms S40..50

Cold start:

- All RAM memory modules are tested and erased.
- If there is no user program present in the EEPROM, all system constants are set to the default values (corresponds to the settings on delivery).
- The operating modes predetermined by the system constants are set.
- The CS31 system bus is reinitialized (only for CS31 system bus master).

Initiating a cold start:

- command KALT <CR> with a terminal or
- menu item ”Online/Cold start” in the programming system

Warm start:

- All RAM memory excluding the program area and the operands area are tested and erased.
- If there is a user program present in the EEPROM, 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 (only for CS31 system bus master).

Initiating a awrm start:

- Voltage ON/OFF or
- Command WARM <CR> with a terminal or
- Menu itwm ”Online/Reset” in the programming system

Buffering of data areas:

The buffering of data is the maintenance of data (for example internal makers) after voltage OFF/ON. Buffering is possible if the battery is charged.

1.3.2 Initialization and backup of data areas S40..50

Backup of data areas:

The variables are all initialized by default at each program start.

Backup of data, i.e. saving data with power OFF/ON, is only possible if the battery is charged. Data can be stored up to 20 days long at an ambient temperature of 25°C. The battery (Vanadium-Lithium) recharges from 0 to 100% in 12 hours when the power supply is on.

A battery failure is detected with the bit 3 of the status word EW 62.15 / %IW1062.15. If the value of bit 3 is 0 (xxxx xxxx xxxx 0xxx), then the battery is exhausted. The life time depends on the use of the CPU.

Cases of use:

- the power supply is OFF every night — life time = 15 years
- the power supply is OFF every week-end — life time = 12 years
- the power is ON only 4 days a week — life time = 6 years

The following can be buffered completely or in parts:

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

In order to backup particular data, these data must be excluded from the initialization process. The settings are performed with system constants (see chapter 1.2.6).

Initialization of data areas:

Initialization to 0 takes place on program start-up, with the (sub) areas to be initialized being defined by system constants. Initialization functions as shown in the following table:

Conditions **Actions**	**Flags, step chains and historical values which are initialized (set to FALSE/0)**
Menu item -> STOP -> Cold start	All
RUN/STOP switch set to RUN -> power ON	In accordance with the values of system constants (see chapter 1.2.6)
RUN/STOP switch -> RUN	
Menu item -> STOP -> START	

If the basic unit is configured as master on the CS31 system bus, the bus system is also initialized with the program start.

1.3.3 Initialization of the central units S40..50

The basic unit executes a complete series of auto tests at each startup. The program can only be started if no errors have been detected.

The auto tests verify the following:

- the program syntax,
- the data transmission,
- the status of extensions,
- the status of remote units, if the central unit is configured as master,
- the startup conditions (internal memories are reset or not – depending on the configuration by the user).

The initialization of the central unit depends on the startup type:

Startup type	**Initialization**
Power ON or software RESET (warm start)	- Clear the program in RAM - Copy the Flash EPROM contents to RAM - Clear the data in RAM according to the configuration
Central unit STOP / RUN button	- Copy the Flash EPROM contents to RAM The RAM remain unchanged if there isn't a program in the Flash EPROM. - Clear the data in RAM according to the configuration
Software cold start	- Clear the program and data in RAM - Copy the Flash EPROM contents to RAM

The bus system is also initialized after startup if the central unit is the master on the CS31 system bus.

Initialization with the factory settings:

When a new program is transmitted to the PLC, the old program is automatically overwritten. It is, however, advisable to reconfigure the central unit with the factory parameters before sending a completely **new** program:

- Stand-alone central unit
- The default system configuration of the central unit (see chapter 1.2.6)
- No program

A complete initialization of the central unit with the factory parameters is obtained by selecting the following in the programming software 907 AC 1131:

"Online / Delete flash"

followed immediately by

"Online / Coldstart"

1.4 Operation of the central units S40..50

1.4.1 Functional diagram

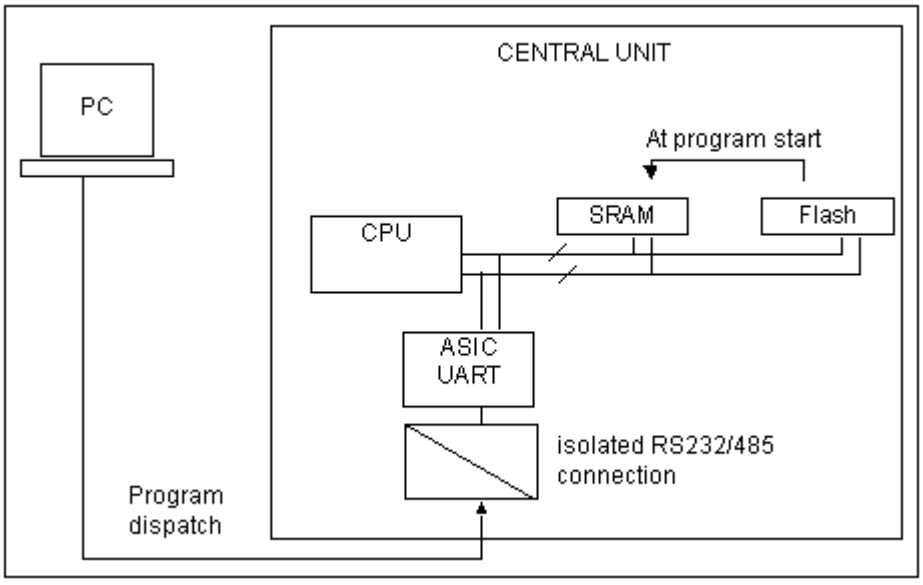
The memory of the 40 and 50 series is composed of two distint areas:

- A RAM memory where the user program and data is loaded.

- A Flash EPROM memory which contains:
- a backup of the user program with the program constants,
- the configuration data
- and the system program, which is protected against access by the user program.

An incorporated battery (50 series) also enables the backup of internal variables.

The system program is a set of universal functions designed to cover all applications and ensure all the basic PLC functions (see chapter 3.1). The user program is developed with the 907 AC 1131 programming and test software stating with version V4.2. After being translated into instructions which can be understood by the central unit, the program is loaded in RUN*) or STOP mode into the SRAM and then saved to the Flash EPROM. Thereby, at each program launch the user program, saved in the Flash EPROM, is copied to the SRAM for processing by the microprocessor (see following figure).

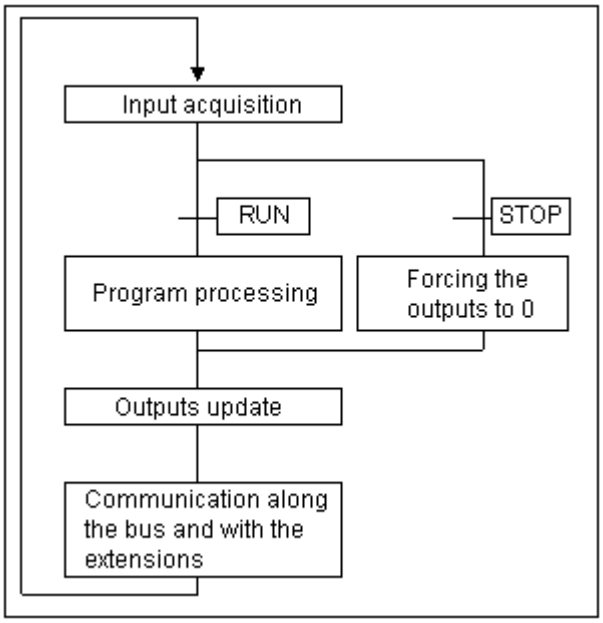


Functional diagram of the central unit

*) **CAUTION:**
This is not possible for central units type R2002 (or firmware version 22).

1.4.2 Program execution

The central unit microprocessor ensures the cyclic execution of the system as shown in the following figure.



The program execution cycle

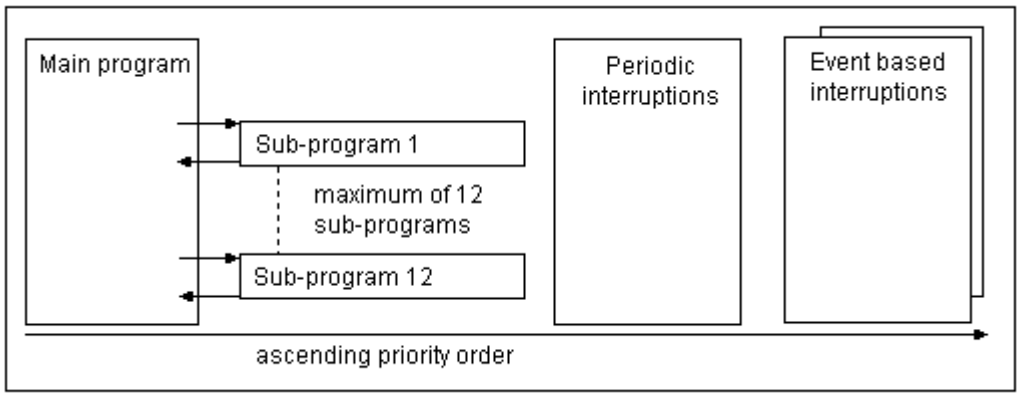
The internal processing:
- PLC monitoring and control
– and processing requests from the terminal operator
is executed in parallel with the previously described cycle.

The main program is processed sequentially. It may call to maximum 12 sub-programs. Each sub-program may be called numerous times in the main program.

Three types of interruptions may be executed parallel to the main program:

- A cyclic interruption
- An interruption triggered by an event on the input E62,02 / %IX62.2
- An interruption triggered by an event on the input E62,03 / %IX62.3

The interruption have priority over the main program execution. If all three interruptions are triggered simultaneously then the interruption triggered by input E62,03 / %IX62.3 has priority over the E62,02 / %IX62.2 input interruption which in turn has priority over the cyclic interruption. Once an interruption has been launched it cannot be interrupted by another (see also figure).



Task priorities

The execution duration of a cycle (bus cycle + program cycles) is controlled by the central unit. Any excess of the cycle time defined by the user in 907 AC 1131 is signaled by the ERR LED, at the front of the central unit, as of the first program cycle.

1.4.3 The central unit cycle time S40..50

The central unit program executes in cycles for the duration which is defined in the system constant "Cycle time". The value is given in milliseconds.

- For a master cental unit:
  the cycle time is between 0 and ms, where only multiples of 5 ms are authorized.

- For a slave or stand alone central unit:
  the cycle time is between 0 and 250 ms.

When the 0 value is selected, the cycle time is the minimum time which the central unit requires for each cycle. In this case the cycle time is not constant.

The cycle time is set with the system constant KD 00,00 / %MD4000.0.

It must be greater than the program processing time of the user program, the data handling time and the waiting times belonging to them. The cycle time is also a time base for some time-related functions, such as the PI controller.

1.4.4 Calculation of a cycle time S40..50

The cycle time Tc can be calculated by the user with the following equation:

$Tc \geq Tb + Tp$

with Tb = the CS31 bus transmission time

and Tp = the program execution time

The program execution time corresponds to the addition of all the times of the functions present in the user program (see table in chapter 3.1)

In general, the program execution time for 1 000 bytes is:

- 0.4 ms for 100 % binary instructions
- 1.2 ms for 65 % of binary instructions and 35 % of word instructions.

The bus transmission time is calculated from the installation configuration. It requires adding together the times of all the units on the bus. The total time for the extensible remote units is given by adding the unit time and the times of the connected extensions (see following table).

CS31 bus communication times:

CS31 bus communication times	
Basic time of the master central unit	2000 µs
Slave central units without extensions	
07 KR 51*	750 µs
07 KT 51*	750 µs
07 KR 91*	750 µs
07 KT 92*	750 µs
07 KT 93*	750 µs
07 KT 94*	
* with default configuration	
Times according to configuration:	
2 bytes on transmission and 2 bytes at reception	516 µs
4 bytes on transmission and 4 bytes at reception	750 µs
8 bytes on transmission and 8 bytes at reception	1300 µs
12 bytes on transmission and 12 bytes at reception	1850 µs
8 words on transmission and 8 words at reception	2500 µs
Remote binary input units	
ICSI 08 D1	323 µs
ICSI 08 E1	323 µs
ICSI 08 E3/E4	323 µs
ICSI 16 D1	387 µs
ICSI 16 E1	387 µs

CS31 bus communication times			
Remote output units			
ICSO 08 R1	260 µs		
ICSO 08 Y1	260 µs		
ICSO 16 N1	340 µs		
07 DI 92	516 µs		
Remote Input / Output units			
ICSC 08 L1	387 µs		
ICFC 16 L1	516 µs		
ICSK 20 F1	452 µs		
ICSK 20 N1	452 µs		
07 DC 91	516 to 590 µs depending on configuration		
07 DC 92	516 to 590 µs depending on configuration		
Remote extensible Input / Output units			
ICMK 14 F1	340 µs without extension		
ICMK 14 N1	340 µs without extension		
Extension Input / Output units	**On master central unit**	**On extensible remote units**	**On save central unitst**
XI 16 E1	1000 µs	1000 µs	2500 µs
XO 08 R1	1000 µs	1000 µs	2500 µs
XC 08 L1	1000 µs	1000 µs	2500 µs
XO 16 N1	1000 µs	1000 µs	2500 µs
XK 08 F1	1000 µs	1000 µs	2500 µs
XM 06 B5	1000 µs	1000 µs	2500 µs
XE 08 B5	1000 µs	1000 µs	2500 µs
XTC 08	1000 µs	1000 µs	2500 µs
IP65 binary Input / Output units			
07 DI 93-I	387 µs		
07 DO 93-I	260 µs		
07 DK 93-I	340 µs		
Remote analog units			
ICSM 06 A6	1162 µs		
ICSE 08 A6	1355 µs		
ICSE 08 B5	1355 µs		
ICST 08 A7	1355 µs		
ICST 08 A8	1355 µs		
ICST 08 A9	1355 µs		
07 AI 91	1355 µs		
ICSA 04 B5	700 µs		
High-speed counter			
ICSF 08 D1	1300 µs		

1.4.5 Bus refresh times

The bus refresh time corresponds to the transmission time over the bus. The calculation corresponds to the addition of all the communication times of the remote units on the bus and the base time of the master central unit.

1.4.6 Central unit reply time

The central unit reply time corresponds to the time taken to activate an output after the activation of an input.

The maximum reply time is obtained by adding the filtering time of the input, the bus refresh time, the delay time for the output and twice the cycle time.

The filtering time for an input on the 40 and 50 series is 5 ms.

The delay time for a transistor output is considered as 0 milliseconds and the delay time for a relay output is 6 ms.

1.5 Adressing and data exchange with CS31 system bus

1.5.1 Introduction / structure examples

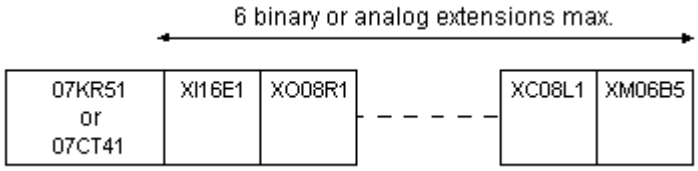
An AC31 system S40..50 always needs an AC31 central unit. There are the following types of central units:

- The 40 series central unit with a local inputs/outputs extensions interface
- The 50 series central unit with a local inputs/outputs extensions interface and a CS31 bus interface

Each central unit incorporates a specific number of binary inputs/outputs. It is possible, depending on the central unit, to increase the number of central od decrentralized inputs/outputs.

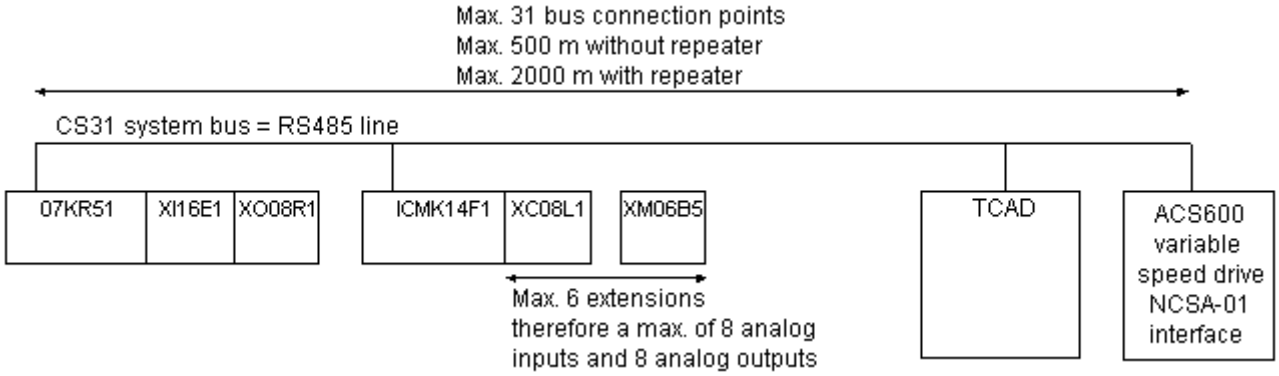
Central units with central extensions:

With the 40 and 50 series, it is possible to increase the number of inputs/outputs of the basic unit by adding up to 6 local extension units of either type, binary or analog



Central units with CS31 bus:

With the 50 and 90 series, it is possible to increase the number of inputs/outputs of the basic unit by adding remote units. The central unit controlling the system is called the MASTER central unit. The maximum bus length is **500 m** without an amplifier and **2000 m** with 3 amplifiers (1 NCB or NCBR unit enables bus amplification for 500 m).



The master central unit can manage up to **31 connection points** called SLAVES which are identified by an address between 0 and 61, such as:

- A remote unit with extension possibilities (a maximum of 6 extension units comprising of a maximum of 8 analog input channels and 8 analog output channels)
- A remote unit series 30 and 90 (without extension possibilities) with analog and binary inputs and outputs
- High-speed counters
- An ABB NCSA-01 frequency converter
- Valve blocks
- Central units series 50
- Central units serie 90 as slaves with own user program
- Or any other devices which support the CS31 communication

Note:
A slave fitted solely with binary channels occupies **1 connection point**.
A slave with binary and analog channels (i. e. ICMK14xx with analog extensions) occupies **2 connection points** of the available **31** ones.

The maximum number of analog remote units depends on the MASTER central unit:

- 50 series: - a maximum of **31** remote analog input units

or - a maximum of **31** remote analog output units

or - a maximum of **15** extensible remote units (ICMK14F1) with analog input/output extensions + **1** remote analog input/output unit (15 x 2 + 1 = 31)

or - a mixed binary/analog configuration within the previous limits

- 90 series - a maximum of **28** analog remote analog input units

or - a maximum of **28** analog remote analog output units

or - a maximum of **28** slaves with analog extensions

or - a mixed binary/analog configuration within the previous limits

Bus transmission:

The master central unit, as well as the remote units, may be positioned in any order on the bus. The order of the extensions on the central units and remote extensible units is free.

The addressing is not according to the order of the units on the bus. The addresses from 1 to 61 may be attributed in any order to the slaves: the first remote unit on the CS 31 bus can have address 5 followed by a remote unit with address 3 followed by another remote unit with address 12, and so on.

The master central unit manages the transmission of messages to the various slaves over a RS-485 serial connection.

The messages are transmitted under the following format:

- Request from the master central unit:

Address	Data	CRC8

- Request from the remote units:

Start	Data	CRC8

The messages always end with an end of frame control: checksum CRC8.

The length of the exchanged frames depends on the type of unit. Frames exchanged with an analog unit are the longest.

The central unit interrogates the remote units in order to establish an initial image of the system configuration during initialization.

The remote units are then interrogated with each bus cycle. This enables the recognition of newly added or removed remote units and the updating of the diagnosis information.

If the central unit receives a message indicating a CRC8 error it will not be signaled immediately and the frame will not be taken into account. After nine consecutive transmission errors a “bus error” is signaled by the central unit. The bus error is also signaled to the remote units after a 250 ms time-out.

1.5.2 Addressing of input/output variables

Addressing the CS31 bus enables programming the utilization of the inputs/outputs in a manner completely transparent to the user.

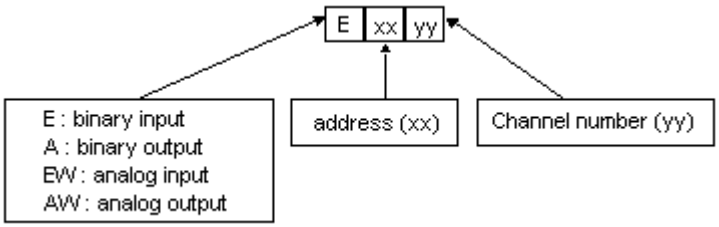
The inputs/outputs of an AC31 remote unit are defined by:

- their type (input or output, binary or analog),

- the unit address (max. 68),

- their channel number on the unit (max. 15).

They are recognized in the following manner by the program: Exx.yy



In **907 AC 1131** programming software, the inputs and outputs have to be declared as follows: (see chapter 1.2 - Operands):

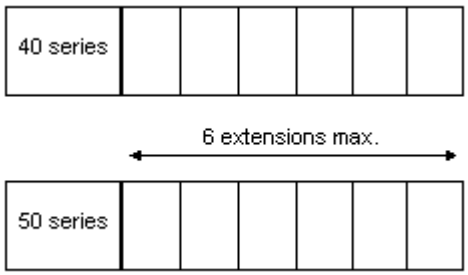
Input/Output	Declaration in 907 AC 1131
Exx,yy	**Exx_yy_Symbol AT %IXxx.yy : BOOL; (* comment *)**
Axx,yy	**Axx_yy_Symbol AT %QXxx.yy : BOOL; (* comment *)**
EWxx,yy	**EWxx_yy_Symbol AT %IW1000+xx.yy : INT; (* comment *)**
AWxx,yy	**AWxx_yy_Symbol AT %QW1000+xx.yy : INT; (* comment *)**

The symbolic name (p. e. Exx_yy_Symbol) and the comment are free.

1.5.3 Addressing central units masters or stand-alones with extensions

Utilization of the standalone, master or slave mode of the central unit is software defined with the system constant KW 00,00 / %MW3000.0 with programming software 907 AC 1131 (see chapter 1.2.6 – System constants).

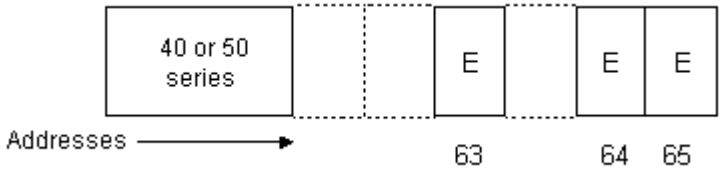
A maximum of 6 binary or analog extensions may be connected to a central unit in any order.



Master- or Stand-alone central unit

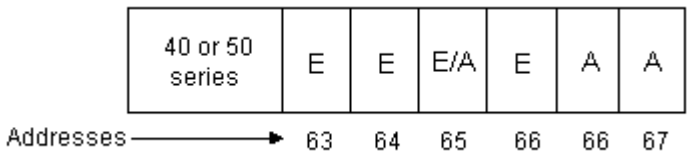
- The address 62 is assigned to the inputs / outputs of a central unit.
- The addresses of the extensions are assigned automatically according to the order of the extension:

  - the first binary inputs extension takes the address 63, and those that follow are incremented by 1, up to 68.



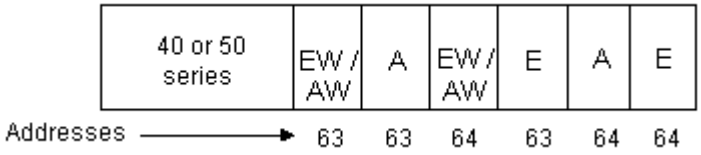
  - the first binary output extension also takes the 63 address and those that follow are incremented by 1, up to 68.

  - an extension with mixed or configurable inputs / outputs is considered by the addressing as an input extension **and** an output extension => The address of the following extension, whether it is input or output, is incremented by 1.



In order to optimize the addressing, the mixed or configurable inputs/outputs extension, or extensions, will be situated at the extremities.

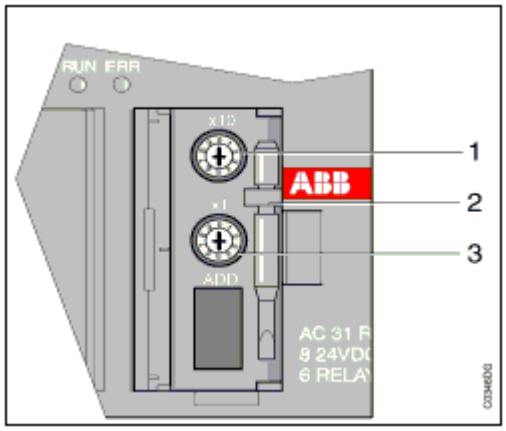
- the address of the first analog extension starts at 63 and is incremented for the following by 1, up to 68.



1.5.4 Addressing slave central units or remote extension units on the CS31 bus

Addressing extensible remote units:

The slave number of the extensible remote units is given by the rotative switches situated beneath the first cover on the front face. Each switch can be set from 0 to 9. The switch 1 is for tens and the switch 3 is for units (see figure).
Values greater than 61 are not allowed: 0 <= X <= 61.



Rotative switches for addressing the remote extension unit ICMK 14xx

Limitation for analog extension:

The maximum number of analog channel per extensible remote unit is:

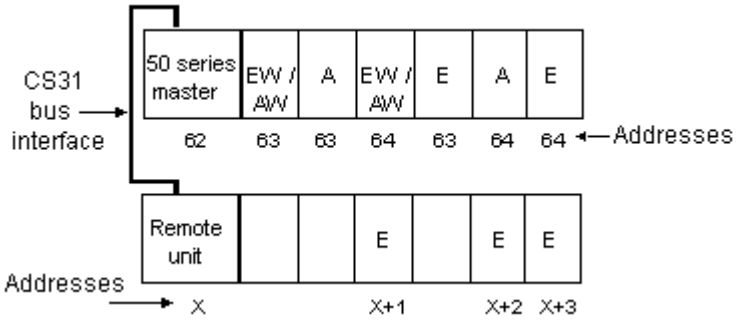
- 8 analog inputs and 8 analog outputs

The maximum configuration per extensible remote unit is:

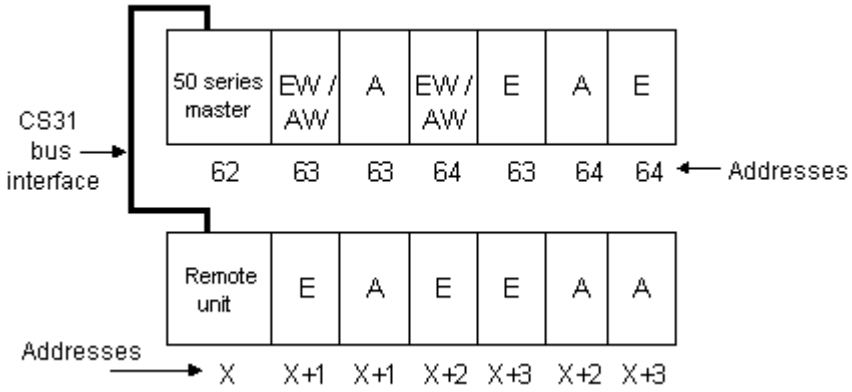
- a maximum of 1 XE 08 B5 extension unit + 5 binary extension units may be used
- a maximum of 1 XTC 08 B5 extension unit + 5 binary extension units may be used
- a maximum of 1 XE 08 B5 and 1 XTC 08 B5 extension unit + 4 binary extension units may be used
- a maximum of 2 XM 06 B5 extension units + 4 binary extension units may be used

Addressing rule:

- The first binary input extension takes the address X+1. The following is incremented by 1 and so on until 61.

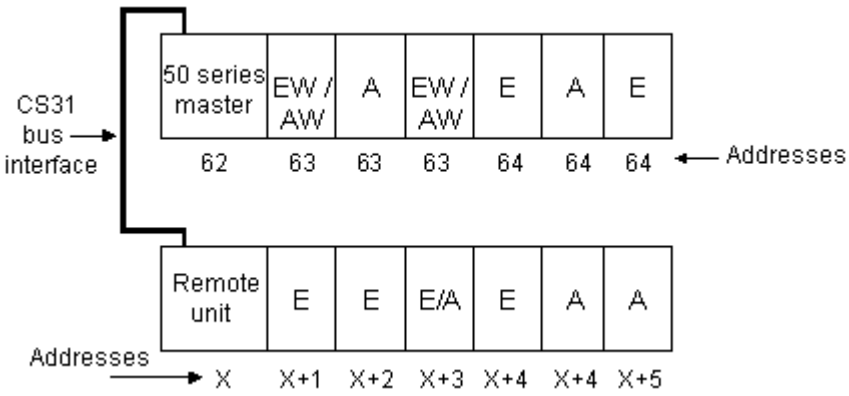


- The first binary output extension also takes the X+1 address and those that follow are incremented by 1, up to 61.



- A mixed or configurable input/output extension is considered, for addressing, as an input extension **and** an output extension => the address of the following binary extension, whether input or output, is incremented by 1.

None of the addresses defined by the configuration may be used by another remote unit of the 30 or 90 series. In the following example an output unit of the 30 or 90 series can not have the address X+1 or X+2. It's address has to be greater than X+5.

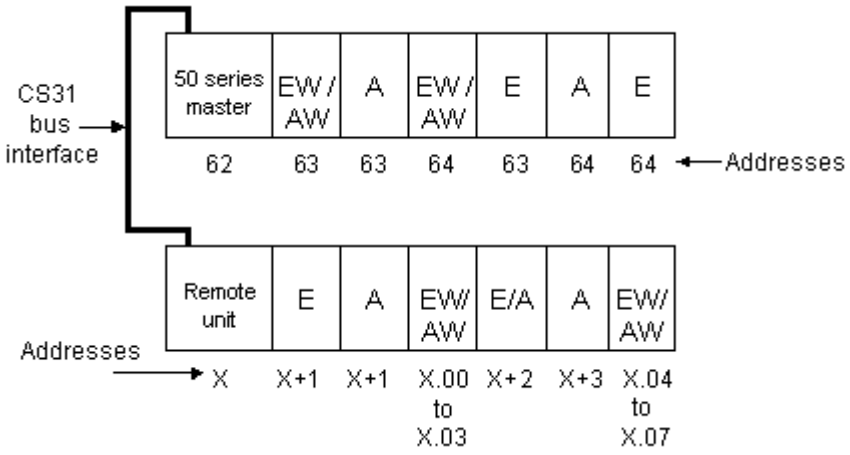


When an analog extension unit is present amongst the extension, the unit takes the same address value as that of the extensible remote unit to which it is associated.

Address values = (0 <= X <= 61)

When an XM 06 B5 type analog extension unit is used, it is possible to associate a maximum of two units to the extensible remote unit, because the number of inputs and outputs of this unit is less than eight.

The first XM 06 B5 has the channels X,00 to X,03, the second XM 06 B5 X,04 to X,07.



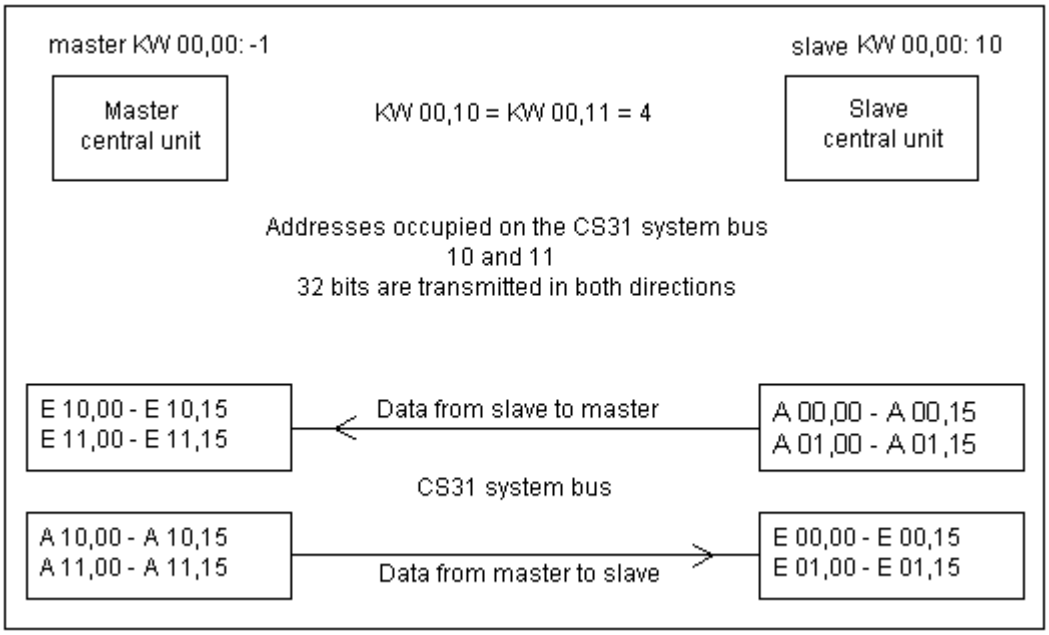
Adressing slave central units:

Information exchanges between a master central unit and a slave central unit isn't limited to the exchange of physical input/output variables. It is possible to exchange a data table, whose size may be defined in bits or words, as follows:

- bit exchange in 8 bit packets (1 byte) = from 2 to 15 bytes
  Address = (0 <= Y <= 61)
  Exchange max. 120 inputs and 120 outputs
- word exchange from 1 to 8 words
  Address = (0 <= Y <= 61)

The addressing of the slave central unit is done by the system constants KW 00,00 / %MW3000.0 with the programming software 907 AC 1131. The number of exchanged data is set in the slave unit with the system constants KW 00,10 / %MW3000.10 and KW 00,11 / %MW3000.11. (see chapter 1.2.6 - Sytem constants).

Example: Master - slave operation: data exchange at digital level:



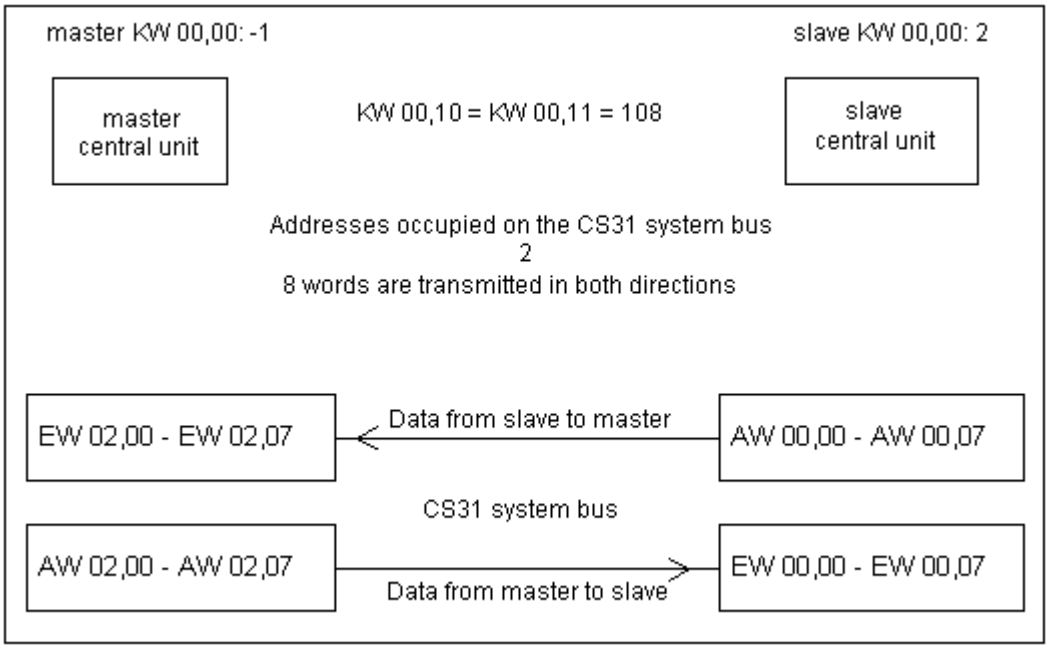
In the programming software 907 AC 1131 must be declared in the project for **the master central unit** (i. e. master.pro) the following operand:

```
KW00_00_Mast_Slav AT %MW3000.0 : INT := -1; (* master on CS31 bus *)

E10_00 AT %IX10.0 : BOOL; (* Output A00,00 in slave 10 *)
E10_01 AT %IX10.1 : BOOL; (* Output A00,01 in slave 10 *)
..
E10_15 AT %IX10.15 : BOOL; (* Output A00,15 in slave 10 *)

E11_00 AT %IX11.0 : BOOL; (* Output A01,00 in slave 10 *)
E11_01 AT %IX11.1 : BOOL; (* Output A01,01 in slave 10 *)
..
E11_00 AT %IX11.15 : BOOL; (* Output A01,15 in slave 10 *)

A10_00 AT %QX10.0 : BOOL; (* Input E00,00 in slave 10 *)
A10_01 AT %QX10.1 : BOOL; (* Input E00,01 in slave 10 *)
..
A10_15 AT %QX10.15 : BOOL; (* Input E00,15 in slave 10 *)

A11_00 AT %QX10.0 : BOOL; (* Input E01,00 in slave 10 *)
A11_01 AT %QX10.1 : BOOL; (* Input E01,01 in slave 10 *)
..
A11_15 AT %QX11.15 : BOOL; (* Input E01,15 in slave 10 *)
```

In the project of the **slave central unit** (i. e. slave.pro) must be declared the following operands:

```
KW00_00_Mast_Slav AT %MW3000.0 : INT := 10; (* slave 10 on CS31 bus *)
KW00_10_SLV_SEND AT %MW3000.10 : INT := 0; (* Default, 4 bytes send to master *)
KW00_11_SLV_REC AT %MW3000.11 : INT := 0; (* Default, 4 bytes receive from master *)

E00_00 AT %IX0.0 : BOOL; (* Output A10,00 in master *)
E00_01 AT %IX0.1 : BOOL; (* Output A10,01 in master *)
..
E00_15 AT %IX0.15 : BOOL; (* Output A10,15 in master *)

E01_00 AT %IX1.0 : BOOL; (* Output A11,00 in master *)
E01_01 AT %IX1.1 : BOOL; (* Output A11,01 in master *)
..
E01_15 AT %IX1.15 : BOOL; (* Output A11,15 in master *)

A00_00 AT %QX0.0 : BOOL; (* Input E10,00 in master *)
A00_01 AT %QX0.0 : BOOL; (* Input E10,01 in master *)
..
A00_15 AT %QX0.0 : BOOL; (* Input E10,15 in master *)

A01_00 AT %QX1.0 : BOOL; (* Input E11,00 in master *)
A01_01 AT %QX1.1 : BOOL; (* Input E11,01 in master *)
..
A01_15 AT %QX1.15 : BOOL; (* Input E11,15 in master *)
```

Example: Master – slave operation: data exchange at word level:



In the programming software 907 AC 1131 must be declared in the project for **the master central unit** (i. e. master.pro) the following operand:

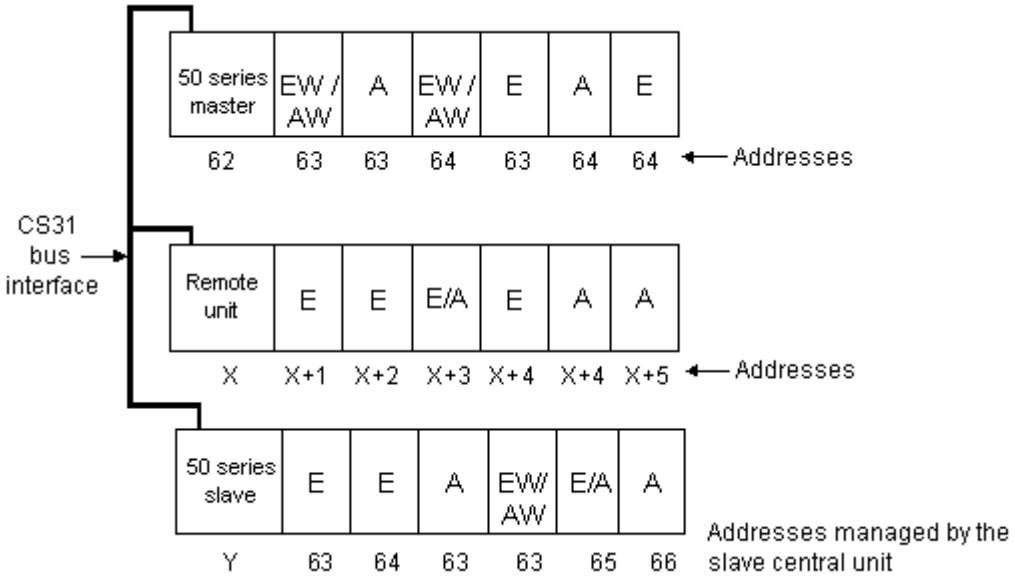
```
KW00_00_Mast_Slav AT %MW3000.0 : INT := -1; (* Master on CS31 bus *)

EW02_00 AT %IW1002.0 : INT; (* Output AW00,00 in slave 2 *)
EW02_01 AT %IW1002.1 : INT; (* Output AW00,01 in slave 2 *)
..
EW02_07 AT %IW1002.7 : INT; (* Output AW00,07 in slave 2 *)

AW02_00 AT %QW1002.0 : INT; (* Input EW00,00 in slave 2 *)
AW02_01 AT %QW1002.1 : INT; (* Input EW00,01 in slave 2 *)
..
AW02_07 AT %QW1002.7 : INT; (* Input EW00,07 in slave 2 *)
```

In the project of the **slave central unit** (i. e. slave.pro) must be declared the following operands:

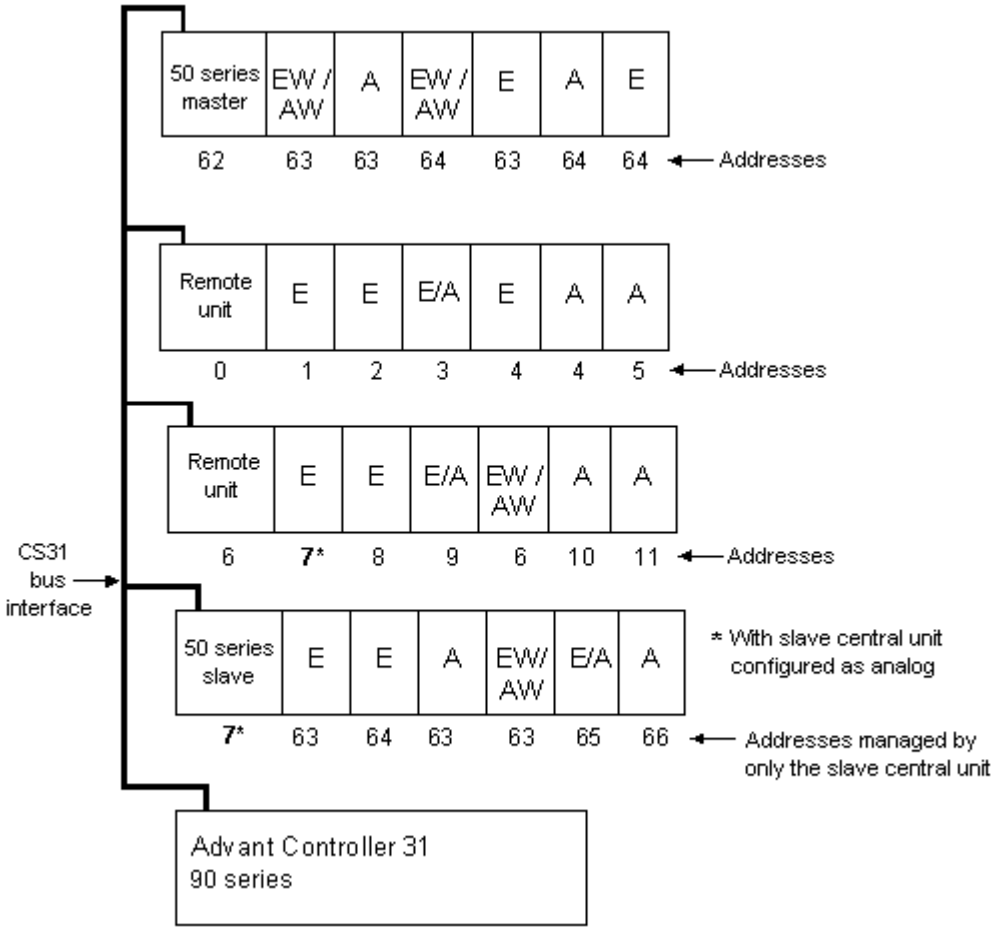
```
KW00_00_Mast_Slav AT %MW3000.0 : INT := 2; (* Slave 2 on CS31 bus *)
KW00_10_SLV_SEND AT %MW3000.10 : INT := 108; (* 8 words send to master *)
KW00_11_SLV_REC AT %MW3000.11 : INT := 108; (* 8 words receive from master *)

EW00_00 AT %IW1000.0 : INT; (* Output AW02,00 in master *)
EW00_01 AT %IW1000.1 : INT; (* Output AW02,01 in master *)
..
EW00_07 AT %IW1000.7 : INT; (* Output AW02,07 in master *)

AW00_00 AT %QW1000.0 : INT; (* Input EW02,00 in master *)
AW00_01 AT %QW1000.1 : INT; (* Input EW02,01 in master *)
..
AW00_07 AT %QW1000.7 : INT; (* Input EW02,07 in master *)
```

Addressing of extensions for a slave remote central unit:

The addressing of extensions of the central unit is undependent of the CS31 system bus addressing. She is defined and managed true the slave unit.



Addressing example:



This example takes 5 connection points (CS31 bus addresses)

1.5.5 Summary of addressing

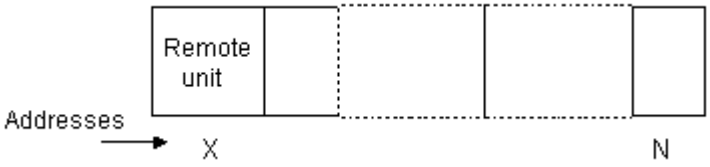
Addresses of the central unit and the central extensions:

Master central unit 40 and 50 series	Addresses	Address configuration
- Incorporated Inputs/Outputs	62	Standard
- Binary extensions	63 to 68	Automatically
- Analog extensions	63 to 68	Automatically

Analog channel addressing:

	On central units	On extensible remote units (address Y)	
Address	63 <= X <= 68	0 <= Y <= 61	
XM 06 B5	**Max 6 extensions**	**Max 2 extensions**	
		First extension	**Second extension**
- inputs	EW X,00..EW X,03 %IW100X.0..%IW100X.3	EW Y,00..EW Y,03 %IW100Y.0..%IW100Y.3	EW Y,04..EW Y,07 %IW100Y.4..%IW100Y.7
- outputs	AW X,00..AW X,01 %QW100X.0..QW100X.1	AW Y,00..AW Y,01 %QW100Y.0..%QW100Y.1	AW Y,04..AW Y,05 %QW100Y.4..%QW100Y.5
- internal values	AW X,02..AW X,03 %QW100X.2..%QW100X.3	AW Y,02..AW Y,03 %QW100Y.2..%QW100Y.3	AW Y,06..AW Y,07 %QW100Y.6..%QW100Y.7
XE 08 B5	**Max 6 extensions**	**Max only one**	
- inputs	EW X,00..EW X,07 %IW100X.0..%IW100X.7	EW Y,00..EW Y,07 %IW100Y.0..%IW100Y.7	
XTC 08	**Max 6 extensions**	**Max only one**	
- internal values	AW X,00..AW X,07 %QW100X.0..%QW100X.7	AW Y,00..AW Y,07 %QW100Y.0..%QW100Y.7	

Addresses of slaves on the central unit:



With X = address of the extensible remote unit,
and N = maximum number of extensions of one type on an extensible remote unit.

	Master central unit		Address configuration
	50 series	**90 series**	
- extensible remote unit			By rotative switches
If solely binary extension	0<=X+N<=61	0<=X+N<=61	Automatically
If at least one analog extension	0<=X<=61	0<=X<=5 und 8<=X<=15	Automatically
- 50 series slave central unit			
Binary configuration	0 to 61	0 to 61	By programming software KW00,00 / %MW3000.0
Analog configuration	0 to 61	0 to 5 and 8 to 15	By programming software KW 00,00 / %MW3000.0
- 90 series slave central unit			
Binary configuration	0 to 61	0 to 61	By programming software KW 00,00 / %MW3000.0
Analog configuration	0 to 5	0 to 5 and 8 to 15	By programming software KW 00,00 / %MW3000.0

1.6 I/O configuration for S40..50

1.6.1 Configuration with function block CS31CO

The CS31CO function block enables the configuration of certain remote units and obtains their current configuration.

The description of each unit indicates whether a remote unit or extension requires configuration with the CS31CO block.

The possible configurations, depending on the units, are:

- Configuration for detecting a cut wire on an input or digital output
- Configuration for configurable I/O units, of a channel as an input or an output
- Configuration of analog input/outputs as power or voltage
- Configuration of the filtering time of a digital input

A detailed description of the function block can be found in the online help of the programming software 907 AC 1131 and in the documentation of the library Base_S40_Vxx.LIB.

1.6.2 Configuration of analog extensions XM 06 B5 and XE 08 B5

The channel and the display value can be configured for the analog extensions XM 06 B5 and XE 08 B5.

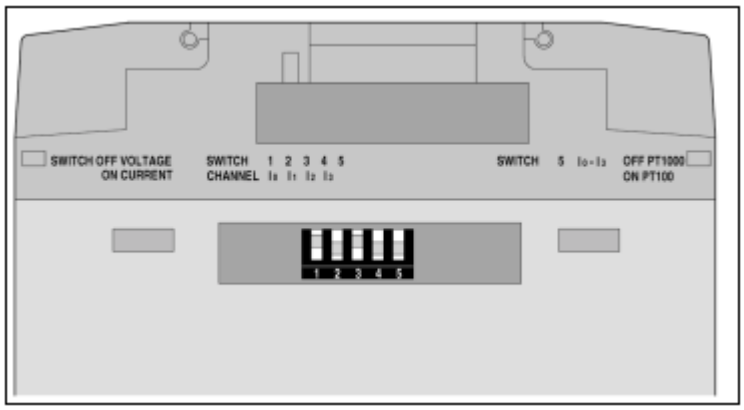
The channel type has to be chosen between voltage (-/+ 10 V), current (0-20 mA or 4-20 mA) or temperature measurement(Pt100 or Pt1000 2, 3 or 4 wires, NI1000, Balco500). This configuration is selected channel by channel with the position of a dip switch, the push bottom on the front plate or by the user program.

The display value can also be configured. The scale and the position of the dot can be modified. Therefore, it is possible to display a process data, i.e. Pressure, speed, etc.

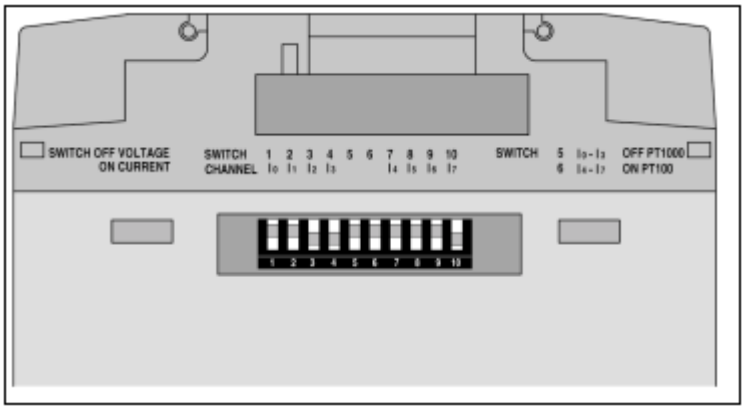
It is also possible to change the filtering: the choise is Fast, 60 Hz, 50 Hz, integration or standard.

Hardware configuration:

Dip switches on the left side of the extension have to be set on the right position.



Dip switches on XM 06 B5 (below: ON)



Dip switches on XE 08 B5 (below: ON)

Configuration of the channels 0 to 3:

Dip switch N°1 for Input 0	OFF	voltage or Pt100/Pt1000
	ON	current
Dip switch N°2 for Input 1	OFF	voltage or Pt100/Pt1000
	ON	current
Dip switch N°3 for Input 2	OFF	voltage or Pt100/Pt1000
	ON	current
Dip switch N°4 for Input 3	OFF	voltage or Pt100/Pt1000
	ON	current
Dip switch N°5	OFF	current source is set to 0.25mA for Pt1000/NI1000/Balco500
	ON	current source is set to 2.5mA for Pt100 (Used for channels 0 to 3)

CAUTION:

The XM 06 B5 product can be deliver with a dip switch component with 6 dip switches. In this case, the dip switch N°6 will be without funcionality.

Configuration of the channels 4 to 7:

Dip switch N°7 for Input 4	OFF	voltage or Pt100/Pt1000
	ON	current
Dip switch N°8 for Input 5	OFF	voltage or Pt100/Pt1000
	ON	current
Dip switch N°9 for Input 6	OFF	voltage or Pt100/Pt1000
	ON	current
Dip switch N°10 for Input 7	OFF	voltage or Pt100/Pt1000
	ON	current
Dip switch N°6	OFF	current source is set to 0.25mA for Pt1000/NI1000/Balco500
	ON	current source is set to 2.5mA for Pt100 (Used for channels 4 to 7)

Choice of channel to display:

The push button "SELECT CHANNEL" on the front plate is used to chose the channel to display value. Channel numbers are scrolled every time the push button is pressed and is displayed (green colour) in the right segment "CHANNEL".

The value is always displayed according to the selected format in the 4 segments "VALUE".

The two analog outputs of analog extension XM 06 B5 have the number 4 and 5.

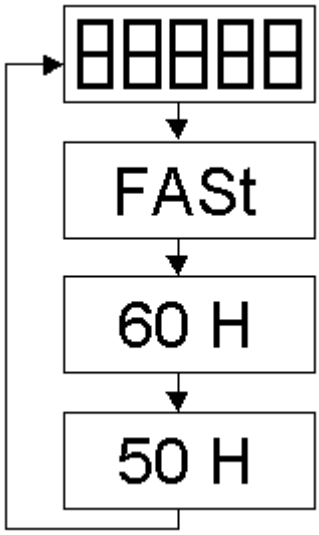
After the last channel the display is tested (all segments ON) without 50 / 60 Hz filtering.

When the push button is pressed for 5 seconds during the display test the configuration mode will be selected. The display starts to blink. The filtering type and fast refresh time can be selected by short pressing the push buttom.

The choice 50 Hz, 60 Hz or Fast mode is selected for all channels of module. When you chose a filtering configuration you increase the acquisition time (5 seconds) of all channels.

With the mode Fast, the refresh time is 50 ms instead of 120 ms, but the value is less stable that means the value is moving a little bit (1 LSB).

When the choice is selected, release the push button for 5 seconds and the new filtering is stored in the EEPROM. The configuration mode is closed and the display value mode is again active. (see following figure).



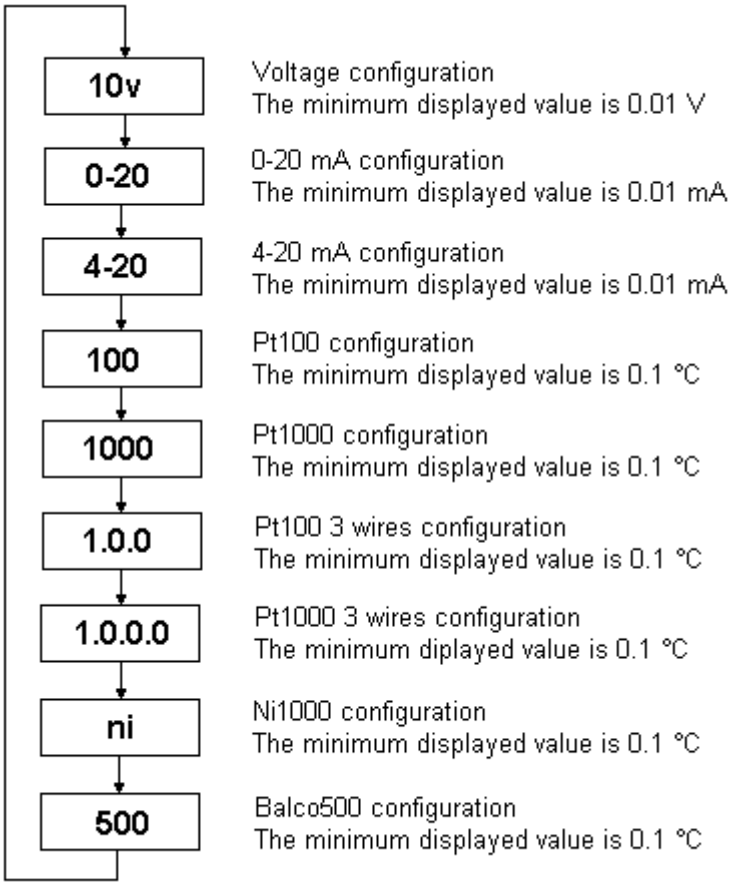
In the FAST mode, the refresh time is 50 ms instead of 120 ms. The value, however, is less stable, i.e. the least significant bit toggles.

Configuration of the channel type:

The channel type can also be set with the push button "SELECT CHANNEL" on the front plate of the extension.

The channel, which will be configured, is chosen with the push button (see choice of channel to display). The configuration mode is set when the push button is pressed for 5 seconds. The display starts to blink and the the current format is displayed. The factory setting is +/- 10 V.

Channel types are scrolled every time the push button is pressed (see following figure).



When the push button is not pressed within 5 seconds, the configuration mode is closed and the display value mode is again active.

The configuration is stored in an internal EEPROM and it is saved in case of power fail.

Configuration by the user program:

It is possible to configure the type of each channel by the user program with the function blocks CONFIO1, CONFIO4 or CONFIO8. If the analog extension is removed, then ist configuration can be load again to the new one.

CAUTION:

The second extension XM 06 B5 connected to a remote unit can not be configured by the user program. Only **one** extension can be configured on a remote unit.

The function blocks CONFIO1, CONFIO4 and CONFIO8 are detailled described in the online help of programming software 907 AC 1131 and in the documentation of library Base_S40_Vxx.LIB.

With the function blocks CONFIO1, CONFIO4 and CONFIO8 is it possible to lock the configuration. The lock function means that the configuration can not be modified with the push button. The locked function is never stored and has to be sent every time that the extension is supplied.

The latest configured channel on one extension is always the channel number displayed. It is possible to select a channel number to the display from the user program.

1.7 Diagnosis

1.7.1 Introduction in diagnosis

The aim of the diagnosis for the 40 and 50 series central units is to ensure a rapid and efficient localization of breakdowns.

The detected errors are transmitted to the central unit which signals their presence on the red ERR LED situated on the front of the central unit.

1.7.2 Types of errors detected

The detected errors are grouped by type into four error classes.

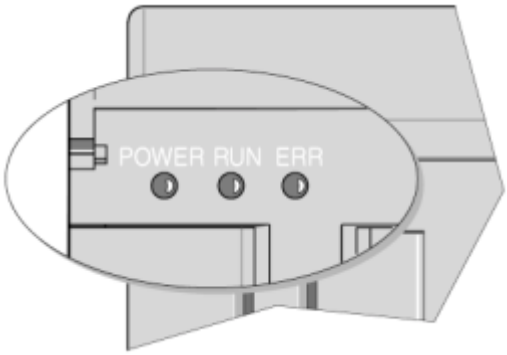
Class 1 errors: fatal errorsr	**Class 2 errors: serious errors**	**Class 3 errors: light errors**	**Class 4 errors: Warnings**
Access to the Flash EPROM is no longer assured. - Flash EPROM checksum error	The operating system functions correctly but the execution of the user program is not guaranteed. Detected errors: - Defective RAM - Too many timers active simultaneously	Communication errors. Stopping the program depends on the users choice during configuration according to the application. Detected errors: - Disconnected unit - Bus error - NCB / NCBR error - Cycle time too short - Addressing fault	Errors occurring on the units or syntax errors whose effects will only become apparent later. The user decides which actions to initialize according to the application. - Internal unit error - Cut wire*1, overload, short circuit - Analog output level error - Defective10 V Output - Program size, program syntax, sub-program or interruption program error - too many historical values - All of the units on the bus are not initialized *2

*1 error detected if prior configuration by programming with the CS31CO block.

*2 error detected if prior software configuration for KW 00,09 / %MW3000.9.

1.7.3 Detection of errors

The detected errors are transmitted to the central unit which signals their presence on the red ERR LED situated on the front of the central unit.



An error on a remote unit is also signaled by the ERR LED of the unit concerned.

The SUPPLY LED flashes on the extension when an error occurs on the extension.

As soon as the error has been detected and corrected by the user it can be acknowledged:

- By restarting the central unit
- Through the software
- Or by program

Only one error per class is memorized. If more than one error of the same class occurs at the same time:

- Only the first is memorized.
- The first error should be acknowledged to allow the following to be read. And so on through to the last error.
- Those following errors which disappear, before the acknowledgment of the first error, are never signaled.

Table showing a summary of all errors:

	Class 1 errors: fatal errors	**Class 2 errors: serious errors**	**Class 3 errors: Light errors**	**Class 4 errors: Warnings**
Detection:	Immediate	Immediate	- Bus error: if the central unit detects a frame control error (CRC) during 9 successive cycles or a timing error or the absence of a reply from a unit - Cycle time error: if the system detects the overriding of the predefined cycle time after 16 consecutive cycles	- Unit error: the central unit queries one slave per cycle. An error is detected between 1 and 31 cycles (depending on the bus structure). - Program syntax error: The central unit detects this type of error when passing from STOP to RUN via the switch or via the software or by online validation of a program modification.
LED status				
- on the central unit:	ERR LED on RUN LED off even if the RUN/STOP switch is on RUN	ERR LED on RUN LED off even if the RUN/STOP switch is set to RUN	ERR LED on According to the configuration, the RUN LED is off even if the RUN/STOP switch is on RUN	ERR LED on Following a program syntax type error, the RUN LED is off
- on the extensible remote units:			ERR LED on or flashing on the error in question	ERR LED on
- on the extensions:			Flashing SUPPLY LED	Flashing SUPPLY LED
- on non-extensible remote unit:			ERR LED on or flashing depending on the error in question	ERR LED on
Reaction while powering up or while the central unit is in use:	All outputs stay at 0 or are set to 0. The programming software no longer has access to the central unit. The central unit remains OFF as long as the error is still present.	All outputs stay at 0 or are set to 0. The programming software retains access to the central unit. The user program is not started or is stopped.	The program is not stopped, by default. With prior configuration you can have the system stopped in the event of an error (see chapter 1.2.6 – System constants)	The program is not stopped
Acknowledgement after error recovery:	- Power on - Cold start	- Power on - Software cold start - Software warm start or RESET - Software acknowledgement	- Switch RUN/STOP from STOP to RUN - Program start via Software - Software warm start - Software cold start - Power on - Software acknowledgement (command: MAIL) - Acknowledgement via programming (CS31QU) - Test button on the 30 and 90 series remote units	
				- Automatic acknowledgement if the error number <= 15

1.7.4 Read the PLC status

Detailed information concerning the status of the central unit and the detected erroors can be obtained in 907 AC 1131 by clicking the command "Online/CPU Display Status". For more information refer to document "Peculiarities of the 907 AC 1131 for S40..50" or use the online help.

You get the following information:

Capacity utilization of the CPU („CPU-Last“) in %
PLC mode: (RUN or STOP)
Memory allocation factor in %
Cycle time in ms
Error list – display of a detected error in text form
Miscellaneous – Answer of the PLC on command „ST“

1.7.5 Description of the diagnosis variables

Each error is identified by a class number, an error code within this class and the arguments. Only one error per class may memorized and detailed by the central unit within a predefined group of internal variables.

The variables values can be accessed by the user for program error management.

If you include the importfile "S40_Fehlermerker.exp" into your project on 907 AC 1131 programming software, so you have access with the PC to the diagnosis variables.

	Class 1 errors: fatal errors	**Class 2 errors: serious errors**	**Class 3 errors: light errors**	**Class 4 errors: Warnings**
Error signaled by the error bit:	M 255,10 / MX255.10 = TRUE			
Type of error signaled by:	M 255,11 %MX255.11	M 255,12 %MX255.12	M 255,13 %MX255.13	M 255,14 %MX255.14
Error code in:	MW 254,00/%MW1254.0	MW 254,08/%MW1254.8	MW 255,00/%MW1255.0	MW 255,08/%MW1255.8
Detailed information in:				
Information 1	MW 254,01/%MW1254.1	MW 254,09/%MW1254.9	MW 255,01/%MW1255.1	MW 255,09/%MW1255.9
Information 2	MW 254,02/%MW1254.2	MW 254,10/%MW1254.10	MW 255,02/%MW1255.2	MW 255,10/%MW1255.10
Information 3	MW 254,03/%MW1254.3	MW 254,11/%MW1254.11	MW 255,03/%MW1255.3	MW 255,11/%MW1255.11
Information 4	MW 254,04/%MW1254.4	MW 254,12/%MW1254.12	MW 255,04/%MW1255.4	
Hard version of unit*	MW 254,05/%MW1254.5	MW 254,13/%MW1254.13	MW 255,05/%MW1255.5	}
Soft version of unit*	MW 254,06/%MW1254.6	MW 254,14/%MW1254.14	MW 255,06/%MW1255.6	} MW 255,12/%MW1255.12
Unit serial number*	MW 254,07/%MW1254.7	MW 254,15/%MW1254.15	MW 255,07/%MW1255.7	MW255,13/%MW1255.13 and MW255,14 / %MW1255.14
Acknowledgement through programming in the central unit This acknowledgement only acknowledges the errors on the central unit not those signaled by a remote uni			By setting M 255,13/%MX255.13 to FALSE after the disappearance of the error	By setting M 255,14/%MX255.14 to FALSE after the disappearance of the error
Acknowledgement through programming in the central unit and remote units			With the CS31QU function block	

* Information unavailable on certain AC31 units

The error bit M 255,10/%MX255.10 is TRUE, if any of the error bits M 255,11-M 255,14 / %MX255.11-%MX255.14 are at TRUE.
The central unit has not found an error if M 255,10/%MX255.10 = FALSE.
The error bit M 255,10/%MX255.10 is automatically reset to FALSE when the error type bits are acknowledged.

A class 4 error (M 255,14/%MX255.14=TRUE) when MW 255,08/%MW1255.8 ≤ 15 acknowledges automatically. Class 3 and 4 type errors may be acknowledged by setting M 255,13/%MX255.13 or M 255,14/%MX255.14 to FALSE.

The information word values are updated with each new error. These words are not automatically reset to 0. Reset is possible by writing the value 0 in these words, either online or via a program.

MW 255,12 / %MW1255.12 gives information about versions (hard & soft). Value is in decimal and must be interpreted in hexadecimal.

1.7.6 Meaning of the contents of the error word flags

Explanation of the tables:

Information 1:	Memory address	= Program menory address, where the erros was detected.
Information 2:	Address	= Address of unit or defective extension
Information 3:	Channel number	= Number of the defective channel

Information 4: Unit type

000	Digital inputs
001	Analog inputs
002	Digital outputs
003	Analog outputs
004	Digital inputs/outputs
005	Analog inputs/outputs
016	XO 08 R1
017	XI 16 E1
018	XC 08 L1
019	ICMK 14 F1
020	ICMK 14 N1
023	XK 08 F1
024	XO 16 N1
025	XO 08 Y1
026	XO 08 R2
080	XM 06 B5
081	XE 08 B5
082	XTC 08
192	ASI-GATEWAY
224	07 CR 41
225	07 KR 51
226	07 CT 41
227	07 KT 51
255	Master or slave central unit where the error was detected and stored.

FK1 -Fatal errors

Class 1 error description	Error number 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
Flash EPROM checksum error	- -	-	-	-

FK2 - Serious errors

Class 2 error description	Error code 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
Defective RAM (user program or data memory)	128_D 80_H	Memory address	-	-
Too many simultaneous timers active on the central unit during program execution (maximum 42)	255_D FF_H	-	-	-

FK3 - Light errors

Class 3 error description	Error code in MW255,00 / %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
Remote unit disconnected	15_D $0F_H$	Unit type	Address	-
CS31 system bus error (no remote-units connected to the bus) Note: If there are only analog units connected to the CS31 system bus, then this error could occur when powering up while the analog units are not initialized. Reason: Analog units have a fairly long initialization period. They are therefore not recognized by the central unit during this initialization phase.	16_D 10_H	-	-	-
NCB or NCBR error Note: Error on one of the CS31 lines. Check the red status LED of NCB or NCBR to locate the defective line.	17_D 11_H	-	-	-
Superposed address	18_D 12_H			
Cycle time too short	200_D $C8_H$	-	-	-

FK4 - warnings

Class 4 error description	Error code 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
Internal unit errors	1_D	01_H	Unit type	Address	Channel number
Cut wire (detection of open circuit)*	2_D	02_H	Unit type	Address	Channel number
Analog output lever error	3_D	03_H	Unit type	Address	Channel number
Overload	4_D	04_H	Unit type	Address	Channel number
Defective 10 V output	5_D	05_H	Unit type	Address	Channel number
Overload and cut wire*	6_D	06_H	Unit type	Address	Channel number
Short circuit	8_D	08_H	Unit type	Address	Channel number
Short circuit + cut wire*	10_D	$0A_H$	Unit type	Address	Channel number
Overload + short circuit	12_D	$0C_H$	Unit type	Address	Channel number
Short circuit + Overload + cut wire*	14_D	$0E_H$	Unit type	Address	Channel number
The end of program is not detected by the system on startup	129_D	81_H	-	-	-
A program syntax error is detected on startup	131_D	83_H	Program address	-	-
On startup the system detects thats the historical values memory is too small	132_D	84_H	-	-	-
The cycle time is missing and detected by the system on startup	133_D	85_H	-	-	-
A missing label for a conditional step is detected by the system on startup.	135_D	87_H	Program address	-	-
The program is not started because the number of units initialized on the CS31 system bus is less than the number specified during configuration with system constant KW 00,09 / %MW3000.9	138_D	$8A_H$	Number of units selected during configuration (KW 00,09 / %MW3000.9)	Number of actual units on the CS31 system bus	-
The program is too large for the memory size	140_D	$8C_H$	-	-	-
The system detects a missing end of sub-program or missing program during startup	142_D	$8E_H$	-	-	-
The system detects a missing program interrupt or an interrupt validation	143_D	$8F_H$	-	-	-
The system detects too many sub-programs (max. 12) during startup	144_D	90_H	-	-	-
The system detects too many historical values in the sub-programs during startup. (max. 128 = Sum of nb of calls x nb of historical values in the sub-programs)	145_D	91_H	-	-	-
The system does not detect a sub-program, corresponding to a call, during startup	146_D	92_H	-	-	-

* An error is detected if previously configured by programming the CS31CO block

1.7.7 Programming examples for error handling

Example of a reaction / command following a specific error:

The following example enables commanding an alarm following an overload or short circuit on the number 5 output of a ICMK 14 N1 remote unit with address 3 on the CS31 system bus.

A second bit could be set at the same time as the alarm bit. This bit should be used, for example, to shut down the power supply.

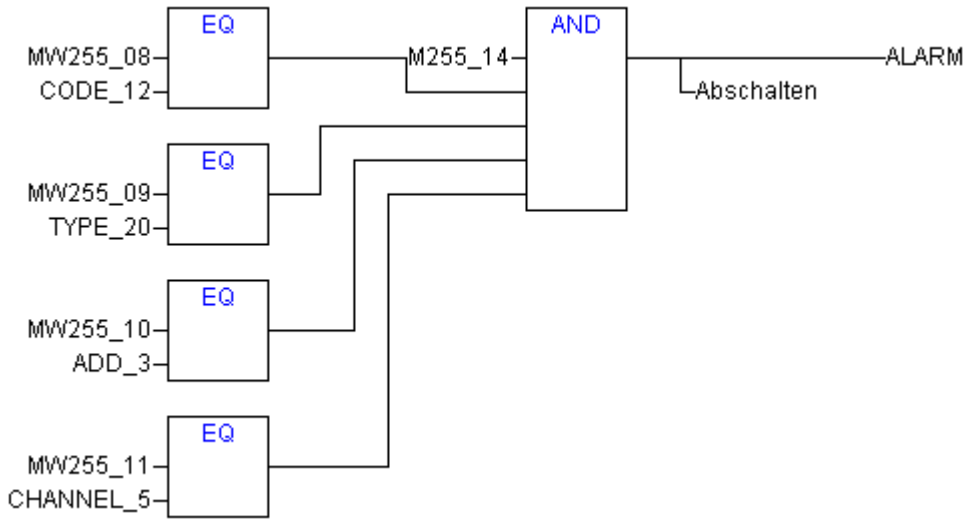
This is a class 4 type error.
The class 4 error uses the following variables:

- M 255.14 : class 4 presence bit
- MW 255.08 : the error code (12 comply with overload and short circuit, see chapter 1.7.6)
- MW 255.09 : the unit type (20 comply to type ICMK14N1, see chapter 1.7.6)
- MW 255.10 : the unit address (address of the reporting modul)
- MW 255.11 : the defective channel number

These variables are predefined in the export file "S40_FEHLERMERKER.EXP" and are read in by command "File/New" or "Project/Import" into "Ressources/Global Variables/S40_Fehlermerker". The following variables has to be declared:

```
CODE_12 AT %MW3001.0 : INT := 12; (* Overload or short circuit *)
TYPE_20 AT %MW3001.1 : INT := 20; (* ICMK 14 F1 *)
ADD_3 AT %MW3001.2 : INT := 3; (* Modul address 3 *)
CHANNEL_5 AT %MW3001.3 : INT := 5; (* Channel 5 *)
ALARM AT %QX1.0 : BOOL; (* ALARM lamp *)
Abschalten AT %MX1.0 : BOOL; (* Shut down the power supply *)
```

Program in FBD looks as follows:

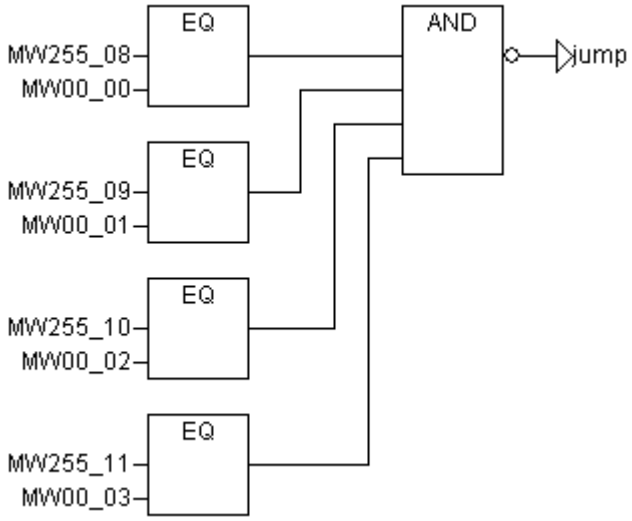


Example for saving numerous errors of the same class:

Only one error can be memorized and detailed in the internal diagnosis on the 40 and 50 series central units. An automatic programmed acknowledgment enables reading other eventual errors. An adapted program enables saving them when necessary.The program in this case is applied to class 4 errors but may also be applied to class 3 errors.

In the present case the last three errors are only saved if the occurred error is different to the previous. The M 255,14 / %MX254.14 bit signals the error presence. At that moment, if the M 000,00 / %MX0.0 status is TRUE than the error is acknowledged due the CS31QU block.

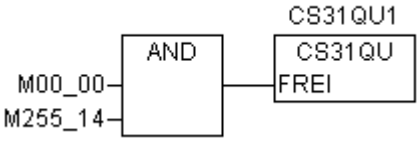
The variables are read in with the import files "M00_00_bis_M00_15.EXP" and "MW00_00_bis_MW00_15.EXP" by command "Project/Import".



MW00_04 ——— MW00_08
MW00_05 ——— MW00_09
MW00_06 ——— MW00_10
MW00_07 ——— MW00_11
MW00_00 ——— MW00_04
MW00_01 ——— MW00_05
MW00_02 ——— MW00_06
MW00_03 ——— MW00_07
MW255_08 ——— MW00_00
MW255_09 ——— MW00_01
MW255_10 ——— MW00_02
MW255_11 ——— MW00_03

jump:

M00_00○——M00_00

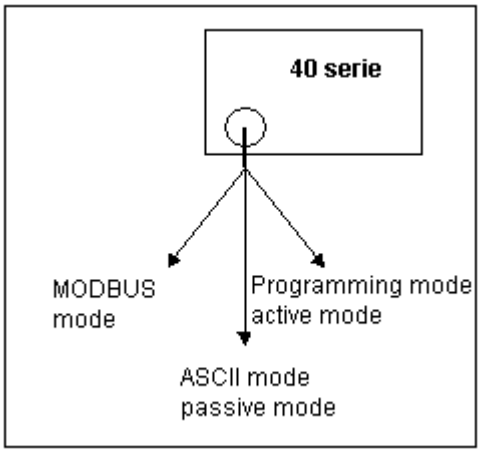


2 Communication with S40..50

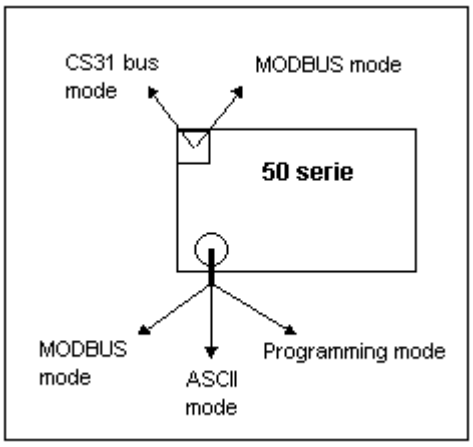
2.1 Network communication with the incorporated MODBUS® interface

Communication is of greater importance in wide area installations. The AC31 system possesses other interfaces, apart from the CS31 system bus, adapted to different communication modes.

This chapter presents, in particular, the interface available with the 40 serie central units which possess three communication protocols on the same serial interface and the interfaces available with the 50 series central units which posses three communication protocols on two different serial interfaces (see figures). The MODBUS® mode is the easiest and most powerful of the three for communicating with other devices.



Communication protocols with the 40 serie



Communication protocols with the 50 serie

2.1.1 Introduction

The MODBUS® protocol is used worldwide. The **MODICON MODBUS® RTU** protocol is integreated in the 40 and 50 series central units.

Numerous automation devices such as PLCs, displays, frequency converters or monitoring systems feature by default, or as an option, a MODBUS® RTU interface and can therefore easily communicate with the 40 or 50 basic units via the serial interface (RS-232 or RS-485) or via the CS31 connection configured in Modbus mode.

On series 50:
MODBUS® slave via the CS31 connection is available starting from software **version 1.7**

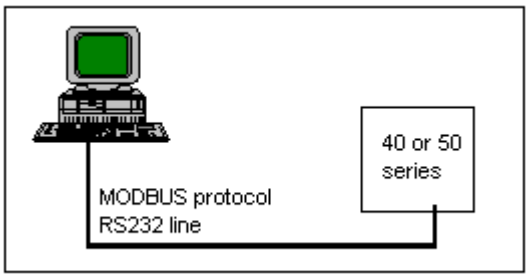
Designation of products	Index of version
07 KR 51 - 24VDC	From K17
07 KR 51 – 120/230 VAC	From K17
07 KT 51 – 24VDC	From J17

MODBUS® master via the CS31 connection is available starting from software **version 2.0**

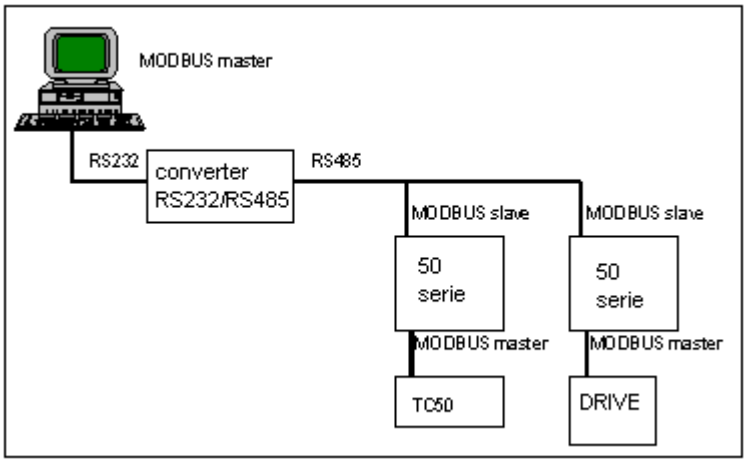
Designation of products	Index of version
07 KR 51 - 24VDC	From N20
07 KR 51 – 120/230 VAC	From N20
07 KT 51 – 24VDC	From M20

You can find the version on the label on the left sideof the 50 series central unit.

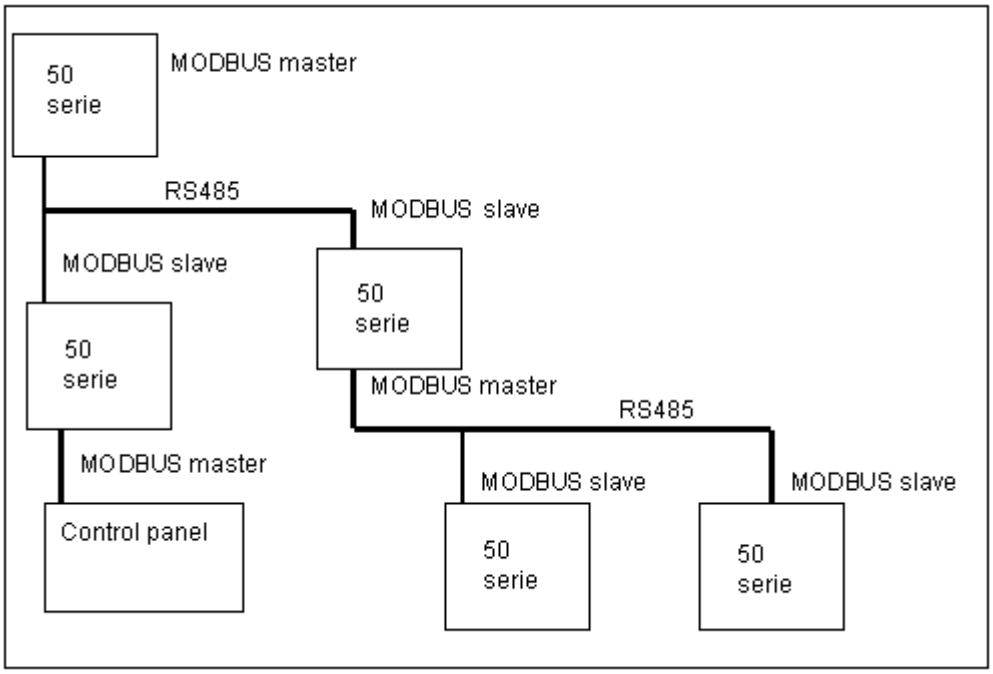
The following figures shows three configuration examples:



Point-to-point connection with a PC



Network connection with a PC



Network connection between central units

MODBUS® is a question/answer protocol that is still occasionally referred to as master/slave: the master sends a request to the slave and waits for the slave's reply.

The master devices on a MODBUS® network are generally central units, displays or supervisory systems. The slaves on the MODBUS® network are generally PLCs, frequency converter, etc.

2.1.2 MODBUS® protocol description

Support	Serial line COM1	CS31 connection (COM2)
Mode	half-duplex	
Number of connection points	**1** single master max. **1** slave with RS232 interface max. **12** slaves with build-in RS485 interface max. **255** slaves with repeaters	**1** single master **1** slave with RS485 interface **31** slaves without repeater max. **255** slaves with repeaters
Protocol	MODBUS® (master / slave)	
Transmission control	CRC 16	
Speedt	Up to 19 200 Bauds	Up to 187 500 Bauds
Maximum length	On RS-485: 1200 m at 19200 Bauds 1500 m at 300 Bauds	On RS-485: 600 m at 76800 Bauds 1200 m at 19200 Bauds 1500 m at 300 Bauds

It is recommended, to install a terminating resistor of 120 Ω if the bus length exceeds 100 m. The CS31 bus connection for the MODBUS® communication with the 50 series basic units is a RS-485 interface. It is connected with a shielded twisted-pair cable.

The MODBUS® frames transmitted by the master contain the following information:

- The MODBUS® address of the interrogated slave (1 byte)

- The function code defining the master request (1 byte)

- The data to be exchanged (N bytes)

- The CRC16 checksum (2 bytes)

The frame has a maximum length of 240 bytes, thereby enabling the exchange of a maximum of 100 data words or 255 binary data.

The slave's reply contains the request confirmation, the data to be returned and also a frame control code (checksum). The slave returns an error code in the even of an error.

Only the following MODBUS® operation codes may be processed by the 40 and 50 series central units:

Function codes		**Description**
In hexadecimal	In decimal	
01 or 02	01 or 02	Read n bits
03 or 04	03 or 04	Read n words
05	05	Write a bit
06	06	Write a word
07	07	Fast reading of 8 bits
08	08	Diagnosis / Initialization
0F	15	Write n bits
10	16	Write n words

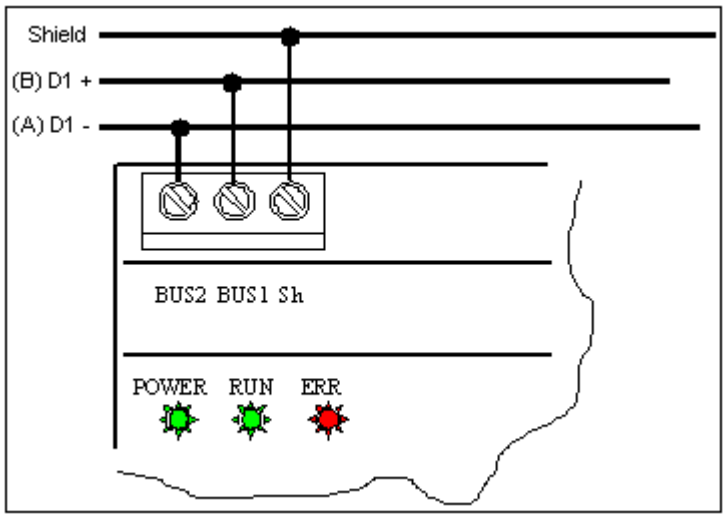
The codes generated on error are:

Error codes	**Description**
01	Unknown function code
02	Address error
03	Data error
09	Time out
10	Checksum error

2.1.3 Configuration of the MODBUS® interface

With serial line COM1:
MODBUS® communication with the 40 and 50 series central units is only allowed if the **black 07 SK 51 or 07 SK 53 cables** are used.

With CS31 connection:
MODBUS® communication on the CS31 connection with the 50 series central units is a RS485 interface and consists of a shielded twisted pair:



Cabling MODBUS® network on CS31 connection (RS485)

The interfaces of the 40 and 50 series central units have a default **MODBUS® configuration** with the following parameters:

	Serial line COM1	**CS31 connection (COM2)**
Mode	MODBUS® slave 1	Stand alone
Transmission speed	9600 Bauds	19200 Bauds
Number of stop bits	1	1
Number of data bits	8	8
Parity	None	None

With the serial line it is not necessary to configure the communication if your application uses these default parameters. The mode can be changed with the system constant KW00,06 / %MW3000.6 (see chapter 1.2.6.).

The MODBUS® communication on the CS31 connection must be configured with the system constant KW00,00 / %MW3000.0.

KW00,00 / %MW3000.0 = MODBUS® slaveaddress + 1100

Example: MODBUS® slave address 5

KW00,00 = 5 + 1100 = 1105

Declaration in 907 AC 1131:
KW00_00_Mast_Slv AT %MW3000.0 : INT := 1105; (* MODBUS slave 5)

The MODBUS® mode remains available, even if the central unit program is stopped (the front switch on OFF), when the black 07 SK 51 or 07 SK 53 cable is connected and the serial interface is configured for MODBUS® communication. The programming mode for tests and modifications becomes available when the grey 07 SK 50 or 07 SK 52 cable is used.

A change in the speed, stop bits, data bits or parity parameters is programmed with the **SINIT** block (refer to documentation of library Base_S40_Vxx.LIB or onlinehelp of the programming software 907 AC 1131). The usage of the SINIT block is not necessary when communication uses the default parameters.

Declaration of variables:

```
SINIT_1: SINIT;
SSK AT %MW3002.0: INT := 2;          (* CS31 connection = COM2 *)
BAUDRATE AT %MW3002.1: INT := 38;    (* Code 38 = 38400 Baud *)
STOPBITS AT %MW3002.2: INT := 1;     (* 1 stop bit *)
DATENBITS AT %MW3002.3: INT := 8;    (* 8 data bits *)
ENDS AT %MW3002.4: INT := 13;        (* 13 = <CR> *)
ENDE AT %MW3002.5: INT := 13;        (* 13 = <CR> *)
```

Program code:

Example of SINIT block in 907 AC 1131

Description of parameters with SINIT block:

SSK parameter:
1 = COM1 initialization of serial line
2 = COM2 initialization of CS31 port

At BAUD - parameter the value of speed 300 up to 19200 is written directly. For higher speed the value of the speed is written through code:

33600 Baud -> Code 44
38400 Baud -> Code 38
57600 Baud -> Code 25
75000 Baud -> Code 19
76800 Baud -> Code 18
125000 Baud -> Code 11
187500 Baud -> Code 7

Detection of MODBUS® communication defaults:

Two different system binary flags can be used as watchdog:

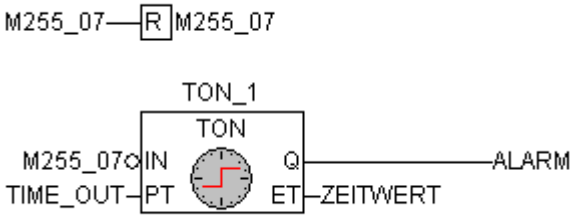
M255,07 / %MX255.7 - watchdog MODBUS® on CS31 connection (COM2)
M255,08 / %MX255.8 - watchdog MODBUS® on serial line COM1

The system binary flag is set to TRUE (1 signal) when the slave sends answers to the master. The example in the following figure shows, how detect MODBUS® communication defaults on CS31 connection.

Declaration of variables:

```
TIME_OUT AT %MD4000.0 : TIME := t#2s;  (* Timeout after 2 sec *)
ALARM AT %MX0.0 : BOOL;                (* MODBUS communication default *)
ZEITWERT AT %MD2000.1 : TIME;          (* Time passed of timeout MODBUS *)
TON_1: TON;
```

Programm code:



Example for programming M255,07 / %MX255.7 in 907 AC 1131

2.1.4 Programming MODBUS®

MODBUS® slave unit:

All data may be read or written in MODBUS® . A program isn't required for MODBUS® communication management. The MODBUS® slave central unit automatically converts the request sent by the master MODBUS® and returns a message once it has recognized the address.

The SINIT block is only required in the program when changing communication parameters.

MODBUS® master unit:

The MODBUS® blocks MODBUSB, MODBUSW, MODMASTB, MODMASTW enables sending data read / write requests to MODBUS® slaves. The blocks are described in the documentation of library Base_S40_V41.LIB and in the onlinehelp of the programming software 907 AC 1131.

A change of communication parameters also requires the usage of the SINIT block in the program.

List of cross references

An exchange of MODBUS® data is realized in a table defined by:

- The MODBUS® address of the first exchanged variable

- The list size = the total number of variables in the list.

All the variables of the 40 and 50 series central units, as described in the following table, can be read or written by the MODBUS® master.

MODBUS® addressing method :

VAR 00,00 ———————————————— ADDR 0

(VAR = type E,A,S,M,EW,AW,MW,KW)	(Address of the first selected variable in decimal) VAR XX.YY = ADDR 0 + (16 * XX) + YY
(VAR = type MD,KD)	(Address of the first selected variable in decimal) VAR XX.YY = ADDR 0 + (32 * XX) + (2 * YY)

Example : Find the MODBUS® address of variables A 62,15, M 232,01 and MD 02,07

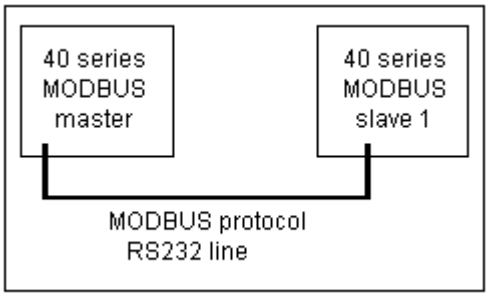
A 62,15 = 4096 + (16 * 62) + 15 = **5103**
M 232,01 = 8192 + (16 * 232) + 1 = **11905**
MD 02,07 = 4000 + (32 * 2) + (2 * 7) = **16462**

Variable types	Variables	IEC address	MODBUS® addresses in hexadecimal	MODBUS® addresses in decimal
Binary Inputs	E 00,00 E 00,01 ... E 00,15	%X00.00 %X00.01 ... %X00.15	0000 0001 ... 000F	0000 0001 ... 0015
	E 01,00 ... E 61,15	%X01.00 ... %X61.15	0010 ... 03DF	0016 ... 0991
	E 62,00 ... E 62,15	%X62.00 ... %X62.15	03E0 ... 03EF	0992 ... 1007
	E 63,00 ... E 68,15	%X63.00 ... %IX68.15	03F0 ... 044F	1008 ... 1103
Binary Outputs	A 00,00 A 00,01 ... A 00,15	%QX00.00 %QX00.01 ... %QX00.15	1000 1001 ... 100F	4096 4097 ... 4111
	A 01,00 ... A 61,15	%QX01.00 ... %QX61.15	1010 ... 13DF	4112 5087
	A 62,00 ... A 62,15	%QX62.00 ... %QX62.15	13E0 ... 13EF	5088 ... 5103
	A 63,00 ... A 68,15	%QX63.00 ... %QX68.15	13F0 ... 144F	5104 ... 5199
Internal bits	M 000,00 M 000,01 ... M 000,15	%MX000.00 %MX000.01 ... %MX000.15	2000 2001 ... 200F	8192 8193 ... 8207
	M 001,00 ... M 099,15	%MX001.00 ... %MX099.15	2010 ... 263F	8208 ... 9791
	M 230,00 ... M 254,15	%MX230.00 ... %MX254.15	2E60 ... 2FEF	11872 ... 12271
	M 255,00 ... M 255,15	%MX255.00 ... %MX255.15	2FF0 ... 2FFF	12272 ... 12287
Steps	S 00,00 S 00,01 ... S 00,15	%MX5000.00 %MX5000.01 ... %MX5000.15	3000 3001 ... 300F	12288 12289 ... 12303
	S 01,00 ... S125,15	%MX5001.00 ... %MX5125.15	3010 ... 37DF	12304 ... 14303

Variable types	Variables	IEC addresses	MODBUS® addresses in hexadecimal	MODBUS® addresses in decimal
Analog inputs	EW 00,00 EW 00,01 ... EW 00,15	%IW1000.00 %IW1000.01 ... %IW1000.15	0000 0001 ... 000F	0000 0001 ... 0015
	EW 01,00 ... EW 62,15	%IW1001.00 ... %IW1062.15	0010 ... 03EF	0016 ... 1007
	EW 63,00 ... EW 68,15	%IW1063.00 ... %IW1068.15	03F0 ... 044F	1008 ... 1103
Analog outputs	AW 00,00 AW 00,01 ... AW 00,15	%QW1000.00 %QW1000.01 ... %QW1000.15	1000 1001 ... 100F	4096 4097 ... 4111
	AW 01,00 ... AW 62,15	%QW1001.00 ... %QW1062.15	1010 ... 13EF	4112 ... 5103
	AW 63,00 ... AW 68,15	%QW1063.00 ... %QW1068.15	13F0 ... 144F	5104 ... 5199
Internal words	MW 000,00 MW 000,01 ... MW 000,15	%MW1000.00 %MW1000.01 ... %MW1000.15	2000 2001 ... 200F	8192 8193 ... 8207
	MW 001,00 ... MW 099,15	%MW1001.00 ... %MW1099.15	2010 ... 263F	8208 ... 9791
	MW 230,00 ... MW 254,15	%MW1230.00 ... %MW1254.15	2E60 ... 2FEF	11872 ... 12271
	MW 255,00 ... MW 255,15	%MW1255.00 ... %MW1255.15	2FF0 ... 2FFF	12272 ... 12287
Internal double words	MD 00,00 MD 00,01 ... MD 00,15	%MD2000.00 %MD2000.01 ... %MD2000.15	4000 4002 ... 401E	16384 16386 ... 16414
	MD 01,00 ... MD 07,15	%MD2001.00 ... %MD2007.15	4020 ... 40FE	16416 ... 16638
Indirect word constants	KW 00,00 KW 00,01 ... KW 00,15	%MW3000.00 %MW3000.01 ... %MW3000.15	3000 3001 ... 300F	12288 12289 ... 12303
	KW 01,00 ... KW 31,15	%MW3001.00 ... %MW3031.15	3010 ... 31FF	12304 ... 12799
Indirect double word constants	KD 00,00 KD 00,01 ... KD 00,15	%MD4000.00 %MD4000.01 ... %MD4000.15	5000 5002 ... 501E	20480 20482 ... 20510
	KD 01,00 ... KD 07,15	%MD4001.00 ... %MD4007.15	5020 ... 50FE	20512 ... 20734

Example using the MODBUS® blocks

An example of MODBUS® communication between two 40 series central units.



In the **MODBUS® slave central unit** a programm isn't required for MODBUS® communication. The system constant KW 00,06 / %MW3000.6 must be initialized with 101 for MODBUS® slave 1.

Example for a program in the MODBUS® master central unit:

Read the 6 bits E 62,00 / %IX62.0 to E 62,05 / %IX62.5 in the MODBUS® slave central unit 1.

The information is placed in A 62,00 / %QX62.0 to A 62,05 / %QX62.5 of the MODBUS® master central unit.

The MODBUS® address of variable E 62,00 / %IX62.0 in the slave 1 is 992 in decimal.

A delay should be defined to enable re-sending the MODBUS® message in case of error. This delay is called TIME_OUT and is generally 2 seconds.

In case of MODBUS® communication between one master 50 serie central unit and different slaves 50 series central units, different MODBUS® functions can be used. It is possible to link them together to optimize the refresh time.

Example:
Read 6 bits continuosly (E 62,00 / %IX62.0 to E 62,05 / %IX62.5) in the slave 1 central unit. The information is placed in A 62,00 / %QX62.0 to A 62,05 / %QX62.5 of the master central unit.
Write 20 words continously AW 00,00 / %QW1000.0 to AW 01,03 / %QW1001.3 in the slave 2 central unit. The information is received from the master central unit in MW 00,00 / %MW1000.0 to MW 01,03 / %MW1001.3.

In case of MODBUS® communication is used on the CS31 connection with 50 serie, you have to use the function block MODMASTB/MODMASTW, this funtion is the same as in function block MODBUSB/MODBUSW with the possibility to select the right port (COM1 for serial line, COM2 for CS31 connection).

Reply times for MODBUS® communication

The MODBUS® processing time depends on:

- The transmission speed

- The number of frame bytes

- The central units cycle time

- The central units load factor

The following times are for indication purposes only.

Reply times with a 40 or 50 series central unit:

Cycle time = 10 ms

Load factor = 80%

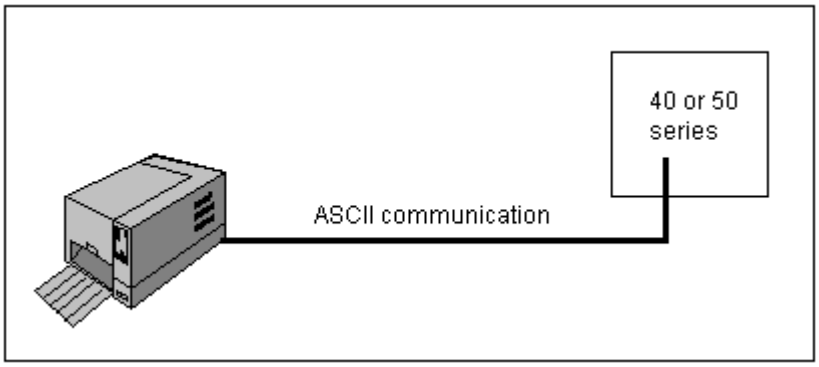
Speed = 9600 Bauds

Nr. of variables	Bits		Words	
	Read (ms)	Write (ms)	Read (ms)	Write (ms)
1	10 – 60	50	10 – 60	60
10	10 – 60	60	10 – 60	110
50	10 – 60	110	110 – 170	220
100	50 – 60	110	220 – 280	390
150	50 – 110		-	-
255	50 – 110		-	-

2.2 Point to point communication with the incorporated ASCII interface COM1

ASCII communication is a protocol which enables exchanging text coded in ASCII (American Standard Code of Information Interchange) between two devices for example.

It may be used by the 40 and 50 series central units to communicate with a printer or a terminal fitted with an ASCII interface(see figure).



Communication parameters:

There isn't a default configuration.
It is therefore **necessary** to configure the communication parameters (speed, stop bits, data bits, parity, end of frame) with the **SINIT** block (refer to description of the block) in the user program.

System cable:

For the ASCII mode the **black cables** type **07 SK 51** or **07 SK 53** are used.

Changing the programming mode to ASCII mode requires a serial configuration through the software by setting the system constant KW00,06 / %MW3000.6 (see chapter 1.2.6).

Programming ASCII mode:

Two other functions for ASCII communication exist, apart from the obligatory SINIT block, for parameter configuration.

Sending messages:

Sending an ASCII message from one 40 or 50 series central unit to another device is done with the **DRUCK block** in the user program.

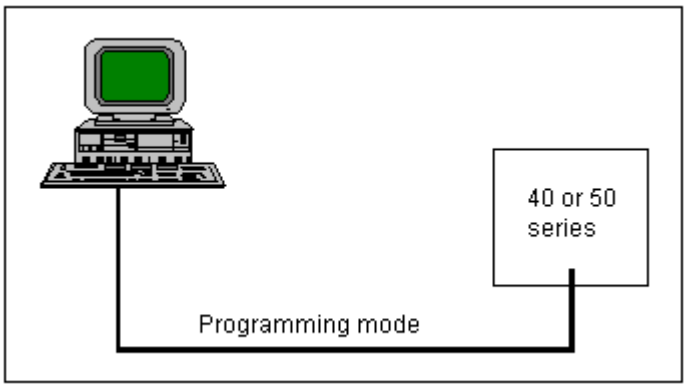
Receiving messages:

Receiving an ASCII message on a 40 or 50 series central unit is done with the **EMAS block** in the user program.

A detailed description of the SINIT, DRUCK and EMAS blocks is available in the onlinehelp of the 907 AC 1131 software or in the documentation of library "Base_S40_Vxx.LIB".

2.3 Point to point communication with the programming protocol

In programming mode the programming protocol of 40 and 50 series is used. All variable values can be read or written by this protocol without any configuration or any particular program in the central units.



Communication parameters:

The following communications parameters are permitted:

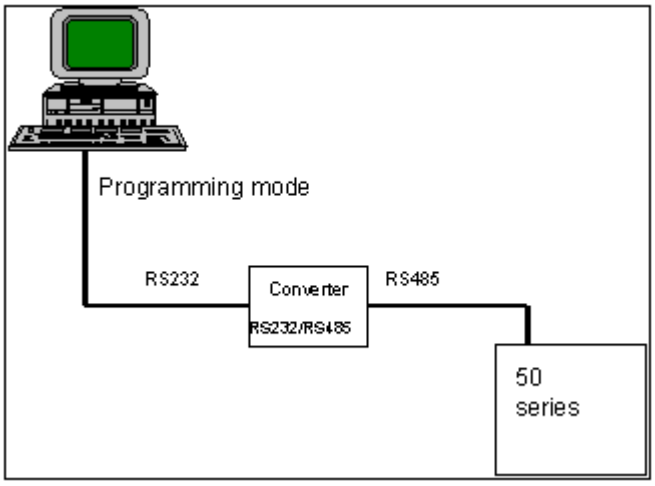
- 9600 Bauds
- 8 data bits
- 1 stop bit
- No parity

System cable:

Use for the programming mode the **grey 07 SK 50 or 07 SK 52 programming cable**.

2.3.1 Programming protocol via the CS31 connection

The programming protocol can be used on the CS31 connection (COM2) (see following figure):



To activate this mode it is necessary to be already in MODBUS® slave configuration on CS31 connection (COM2) and to change the status of the system flag M255,06 / %MX255.6.

Programming protocol via the CS31 connection is available starting from software **version 2.0**:

Designation of products	Index of version
07 KR 51 - 24VDC	From N20
07 KR 51 - 120/230 VAC	From N20
07 KT 51 – 24VDC	From M20

You can find the version on the label on the left side of the 50 series central unit after the designation of product, for example 07 KR 51-P30.

Limitations for using programming mode on the CS31 connection (COM2):

Mode on Serial Line (COM1)	Mode on CS31 connection (COM2)	Status
Programming Mode	MODBUS Master	Available
Programming Mode	MODBUS Slave	Available
Programming Mode	Programming Mode	Only COM2 active
ASCII Mode	MODBUS Master	Available
ASCII Mode	MODBUS Slave	Available
ASCII Mode	Programming Mode	**NOT** available
MODBUS Master	MODBUS Master	Available
MODBUS Master	MODBUS Slave	Available
MODBUS Master	Programming Mode	Available
MODBUS Slave	MODBUS Master	Available
MODBUS Slave	MODBUS Slave	Available
MODBUS Slave	Programming Mode	Available

Special internal system flags available:

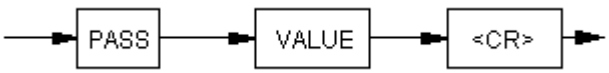
These system flags can be used to change the protocol on the serial ports COM1 and COM2 (CS31 connection). The changing of the status of these system flags can be managed through the user program, programming interface or MODBUS® interface:

M255,06 / %MX255.6	Switch MODBUS® / programming (active) mode COM2
M255,09 / %MX255.9	Switch MODBUS® / programming (active) mode COM1

2.3.2 Protocol description

The programming protocol 40 and 50 series is a simple ASCII protocol.

The following table shows the avaiable commands.

Command	Description
Commands for creating the user program	
AEND	Prepare a programming change on a running PLC program
ALT	Reactivate the user program stored in EEPROM
AL	Display capacity utilization of basic unit, usually less than 80 %
D	Display program
DEEP	Erase PLC program in EEPROM
FREI	Enable a program change to a running PLC program
IDA	Display program identification
IDR	Delete program identification
IDS	Enter program identification
K	Enter/edit values of indirect constants such as Kx,x, KWx,x, KDx,x
NOP	Delete program part, i.e. overwrite program part with NOPs
O	Optimize the program, all NOPs are deleted, syntax: O<CR>
P	Display free program memory area
S	Enter/edit PLC program (substitute)
SP	Save PLC program in EEPROM
V	Move user program (only towards the end)
Commands for testing the user program	
A	Abort user program
FEHLER	Display contents of the error register
FORC	Enter force values (max. 7 word and 31 bits)
FORCA	Display force value
FORCR	Delete forcing
G	Start user program
KALT	Perform cold start
PS	Display program status
ST	Display basic unit status
WARM	Perform warm start
Y	Overwrite value of a variable with a value to be entered
Z	Display status of variables
ZD	Display and continually update status of variables
ZZ	Display only the values of variables
Configuration commands	
KONFS	Display/change operating modes (English/German). German has to be used with the 907 AC 1131 programming software.
PASS	 Password: Enable or disable the password Value: 4 hexadecimal numbers The value 0000 is not allowed. If the program is protected by a password, the user does not have access to the following commands: AEND, D, DEEP, FREI, N, NOP, O, S, V (display or modification of programs).
MAIL	Configuration of CS31 remote units
UHR	Display time and date
UHRS	Set time and date (no syntax check)

3.1 List of functions S40..50

Note:

If a different call exists for IL in the PLC than for FBD/LD in 907 PC 331 it is additionally specified in the row AWL.

1) This function is generated in the IL by a sequence of commands and/or blocks.

Description	CE identifier in 907 PC 331		907 AC 1131		Runtime in µs
	FBB	AWL	Identifier	Realised through	
Digital functions					
AND	&	1)	AND	Operator	5.1
OR	/	1)	OR	Opeartor	6
Exclusive OR	=1	1)	XOR	Operator	8.8
Allocation	=	1)	ST	Opeartor	3.9
Negation	=N	1)	NOT	Operator	
Memory functions					
Allocation, set memory	=S	1)	S	Operator	4.45
Allocation, reset memoryr	=R	1)	R	Operator	4.4
Set memory, dominating	RS	1)	SR	BASE_S40_V41	8.8
Reset memory, dominating	SR	1)	RS	BASE_S40_V41	8.8
Arithmetic functions, word					
Addition / word	+	1)	ADD	Operator	12.8
Subtraction / word	-	1)	SUB	Operator	13.8
Multiplication / word	*	1)	MUL	Operator	31.8
Division / word	:	1)	DIV	Operator	142
Multiplication with division / word	*:	MULDI	MULDI	BASE_S40_V41	186
Mult.by 2 to the power of N / Wort	MUL2N		MUL2N	BASE_S40_V41	36.2
Absolute value generator	BETR		ABS	Operator	23.3
Allocation word	=W	1)	ST	Operator	8.1
Allocation direct constant to word variable	ZUDKW		-	Internal in 907 AC 1131	17
Square root / word	SQRT	SQRT	SQRTW	Base_S40_V41	572
Arithmetic funktions, double word					
Addition / double word	+D	ADDD	ADD	Base_S40_V41	114
Subtraction / double word	-D	SUBD	SUB	Base_S40_V41	116
Multiplication / double word	*D	MULD	MUL	Base_S40_V41	380
Division / double word	:D	DIVD	DIV	Base_S40_V41	504
Square root / double word	SQRT	SQRT	SQRTD	Base_S40_V41	572
Comparision word functions					
Greater than	>	1)	GT	Operator	12.4
Greater than or equal to	>=	1)	GE	Operator	13.1
Equal	=?	1)	EQ	Operator	13.3
Unequalh	<>	1)	NE	Operator	13.3
Less than	<	1)	LT	Operator	13.1
Less than or equal to	<=	1)	LE	Operator	12.4

Description	CE identifier in 907 PC 331		907 AC 1131		Runtime in µs
	FBB	AWL	Identifier	Realised through	
Digital functions					
Comparision double word functions					
Equal, double word	=?D	VGLD	EQ	Operator	110
Greater than, double word	>D	VGRD	GT	Operator	108
Less than, double word	<D	VKLD	LT	Operator	107
Timer functions					
ON delay	ESV		ESV	BASE_S40_V41	124
OFF delay	ASV		ASV	BASE_S40_V41	124
Monostable element „abort“	MOA		MOA	BASE_S40_V41	167
Monostable element „constant“	MOK		MOK	BASE_S40_V41	170
Timer functions with time visualization					
OFF delay	TOF		TOF	BASE_S40_V41	167
ON delay	TON		TON	BASE_S40_V41	210
Monostable element „constant“	TP		TP	BASE_S40_V41	208
Monostable element „abort"	MOAT		MOAT	BASE_S40_V41	213
Counter functions					
Up/down counter / word	VRZ		VRZ	BASE_S40_V41	190
Counter	CTU		CTU	BASE_S40_V41	430
Counter for encoder inputs	CTUH		CTUH	BASE_S40_V41	560
Program control functions					
Conditional jump to label	SPBM	SPRUNG	SPRUNG	Internal in 907 AC 1131	
Target label	MRK	MR	LABEL	Internal in 907 AC 1131	
Program end	PE		-	Will be automatically generated at the end of program	
Conditinal program end	=PE	1)	-	Can be realized with jump	100
Subroutine call for an assembler program	CALLUP		-	Internal use in the MODBUS blocks	
Sub-program call	CAL_FB		-	Is generated in 907 AC 1131 for call sub-program	
Definition sub-program	DEF_FB		-	Is generated in 907 AC 1131 at the begin of a sub-program	
Return command	RET		-	Is generated in 907 AC 1131 at the end of a sub-program	
Release Interrupt Task	VTASK		VTASK	BASE_S40_V41	
Definition Interruptroutine	TASK		-	Is generated in 907 AC 1131 at the begin of an interrupt-routine	
Direct reading of an input	DIN	DI	DIN	BASE_S40_V41	
Direct writing of an output	DOUT	DO	DOUT	BASE_S40_V41	

Description	CE identifier in 907 PC 331		907 AC 1131		Runtime in µs
	FBB	AWL	Identifier	Realised through	
Digital functions					
Format conversions					
BCD to digit conversion, word	BCDDUAL	BCDBIN	BCDDUAL	BASE_S40_V41	72.5
Digital to BCD conversion, word	DUALBCD	DUALBIN	DUALBCD	BASE_S40_V41	107
Pack digital values in word	PACK		PACK4 (8, 16)	BASE_S40_V41	355,650, 1220
Unpack a word in digital variables	UNPACK		UNPACK4 (8,16)	BASE_S40_V41	325,615, 1200
Word to double word conversion	WDW		WDW	BASE_S40_V41	
Double word to word conversion	DWW		DWW	BASE_S40_V41	
Conversion words into time	W_TIME		W_TIME	BASE_S40_V41	
Conversion time into words	TIME_W		TIME_W	BASE_S40_V41	
Pulse					
Pulse (rising edge)	I+	1)	I_PLUS	BASE_S40_V41	
Pulse (falling edge)	I-	1)	I_MINUS	BASE_S40_V41	
Logical functions with word values					
AND combination, word	WAND		WAND	BASE_S40_V41	22.7
OR combination, word	WOR		WOR	BASE_S40_V41	22.7
Exclusive OR combination, word	WXOR		WXOR	BASE_S40_V41	22.6
Negation, word	NEG	=-	NEGW	BASE_S40_V41	10.6
Logical functions with double word values					
AND combination, double word	DWAND		DWAND	BASE_S40_V41	38
OR combination, double word	DWOR		DWOR	BASE_S40_V41	39
Exclusive OR combination, double word	DWXOR		DWXOR	BASE_S40_V41	38
Access to physical addresses					
Copying memory areas	COPY		COPY	BASE_S40_V41	258
Read word with enabling	WOL		WOL	BASE_S40_V41	21.5
Higher order functions					
Digital selection gate	AWTB		AWTB / SEL	BASE_S40_V41 / Operator	38.4
Selection gate, word	AWT		AWT / SEL	BASE_S40_V41 / Operator	22
Maximum value generator, word	MAX		MAX	Operator	426
Minimum value generator, word	MIN		MIN	Operator	430
Limiter, word	BEG		BEG	BASE_S40_V41	
Digital value change monitor	BMELD		BMELD8 (16,32,64, 127)	BASE_S40_V41	1430
Configuration of analog channels	CONFIO		CONFIO1 (4, 8)	BASE_S40_V41	
Function generator	FKG		FKG2 (4, 16, 32, 64, 256)	BASE_S40_V41	
List allocator	LIZU		LIZU	BASE_S40_V41	139
Read word variable, indexed	IDLm	IDL	IDLm	BASE_S40_V41	27.4
Write word variable, indexed	IDSm	IDS	IDSm	BASE_S40_V41	38.6
Read binary variable, indexed	IDLB		IDLB	BASE_S40_V41	209
Write binary variable, indexed	IDSB		IDSB	BASE_S40_V41	201

Description	CE identifier in 907 PC 331		907 AC 1131		Runtime in µs
	FBB	AWL	Identifier	Realised through	
Digital functions					
Clock read/set	UHR		UHR	BASE_S40_V41	430
Impulse generator for stepping motor drive	NPULSE		NPULSE	BASE_S40_V41	386
Automatic control functions					
Proportional integral controller	PI		PI	BASE_S40_V41	1600
PIDT1 control	PIDT1		PIDT1	BASE_S40_V41	1600
Pulse duration modulator	PDM		PDM	BASE_S40_V41	640
Communication via serial interface					
Initialization and configuration of the serial interfaces	SINIT		SINIT	BASE_S40_V41	100
Output of ASCII characters and hexadecimal values via the serial interface COM1	DRUCK		Inline Pragma	S40 Inline Pragma	
Receive characters on COM1	EMAS		Inline Pragma	S40 Inline Pragma	
MODBUS master	MODBUS	CALLUP	MODBUSB MODBUSW MODMASTB MODMASTW	BASE_S40_V41	
CS31 functions					
Configure AC31 remote units	CS31CO		CS31CO	BASE_S40_V41	180
CS31 acknowledge errors	CS31QU		CS31QU	BASE_S40_V41	27.5

%

%IW1062.15 S40..50 26
%IW1062.8-%IW1062.14 S40..50 26
%MD4000.0 S40..50 23
%MW1254.0-%MW1255.15 S40..50 25
%MW3000.0 S40..50 13
%MW3000.1 S40..50 14
%MW3000.10 S40..50 22
%MW3000.11 S40..50 22
%MW3000.2 S40..50 14
%MW3000.3 S40..50 15
%MW3000.4 S40..50 15
%MW3000.5 S40..50 15
%MW3000.6 S40..50 16
%MW3000.7 S40..50 21
%MW3000.9 S40..50 21
%MX255.0-%MX255.3 S40..50 24
%MX255.10-%MX255.14 S40..50 25
%MX255.15 S40..50 25
%MX255.6 S40..50 24
%MX255.7 S40...50 24
%MX255.8 S40...50 25
%MX255.9 S40...50 25

A

Addressing of input/output variables S40..50 40
Adressing CS31 system bus S40..50 37
ASCII mode S40..50 16

B

Backup of data areas S40..50 28
Buffering of data areas S40..50 27
Bus refresh times S40..50 36

C

Changing application mode S40..50 13
Clock S40..50 26
cold start S40..50 27
Communication mode of serial interface COM1 S40..50 16
Configuration S40..50 52
CS31 status word S40..50 26
CS31CO S40..50 52
Cycle time S40..50 33, 34
cycle timet S40..50 23

D

Data exchange with CS31 system bus S40..50 37
Diagnosis examples S40..50 66

E

Error messages from the PLC S40..50 25
EW 62,15 S40..50 26
EW62,08-EW62,14 S40..50 26

F

Fehlermeldungen der SPS S40..50 25
FK3 errors S40..50 21
Functional diagram S40..50 31

I

Import file for operands S40..50 24
Importfile for operands S40..50 4, 12
Initialization of binary flags S40..50 14
Initialization of chain steps S40..50 15
Initialization of data areas S40..50 29
Initialization of double word flags S40..50 15
Initialization of historical values S40..50 15
Initialization of word flags S40..50 14
Initialization S40..50 30
Initialization with the factory settings S40..50 30

K

KD 00,00 S40..50 23
KW 00,00 S40..50 13
KW 00,01 S40..50 14
KW 00,02 S40..50 14
KW 00,03 S40..50 15
KW 00,04 S40..50 15
KW 00,05 S40..50 15
KW 00,06 S40..50 16
KW 00,07 S40..50 21
KW 00,09 S40..50 21
KW 00,10 S40..50 22
KW 00,11 S40..50 22

L

List of functions S40..50 85

M

M255,0-M255,3 S40..50 24
M255,10-M255,14 S40..50 25
M255,15 S40..50 25
M255,6 S40..50 24
M255,7 S40..50 24
M255,8 S40..50 25
M255,9 S40..50 25
MODBUS S40..50 16

N

Network communication with the incorporated MODBUS® interface S40..50 68

O

Oscillators S40..50 24

P

PLC application mode SPS S40..50 13
Program execution S40..50 32
Programming mode S40..50 16
Project files S40..50 3

R

Receive area size of the slave PLC S40..50 22
Reply time S40..50 36
Restart detection S40..50 25
RTS signal COM1 S40..50 17, 18
RTS signal COM1 with S40 18
RTS Signal COM1 with S50 19

S

Starting up the AC3 system serie 50 21
Switch MODBUS/Active mode S40..50 24, 25
System constants S40..50 12
system flag words S40..50 24
system flags S40..50 24

T

Terms S40..50 27
Transmit area size of the slave PLC S40..50 22

W

Warm start S40..50 27
Watchdog MODBUS S40..50 24, 25

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

ABB Control
10, rue Ampère Z.I. - B.P. 114
F-69685 Chassieu cedex France

Telephone +33 (0) 4 7222 1722
Telefax +33 (0) 4 7222 1935
E-mail
Internet http://www.abb.com/lowvoltage