

Modsoft AKF → Micro
Type: MICROAKF
Version 2.0 – 2.1
Short Form Guide MICROAKF

User Instruction
DOK-704106.33-1197

Translation of the German Description DOK-703410

Belongs to software kit E-Nr. 424 704 703

Documents in the software package

Documentation	Area of application
Installation User Instruction DOK-704104	Explains the usage and installation of the dis- kettes included.
MICROAKF for Beginners User Instruction DOK-704105	Serves to introduce new customers to MICROAKF. The user learns how to use the software in small steps.
Short Form Guide User Instruction DOK-704106	Tables for validity ranges and system mar- kers, SFB-Formal operands for quick use on- site.
SFB Block Library DOK-704572	Description of the Standard Function Blocks.
Configuration User Instruction DOK-704107	Contains the new features of the current ver- sion and explains the functions of the indivi- dual software menus for the configurer.
Masterindex User Instruction DOK-704108	Index of all documentation.

Notes

Application Note



Caution The relevant regulations must be observed for control applications involving safety requirements. For reasons of safety and to ensure compliance with documented system data, repairs to components should be performed only by the manufacturer.

Training

Schneider Automation offers suitable training that provides further information concerning the system (see addresses).

Data, Illustrations, Alterations

Data and illustration are not binding. We reserve the right to alter our products in line with our policy of continuous product development. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us by using the form on the last page of this publication.

Addresses

The addresses of our Regional Sales Offices, Training Centers, Service and Engineering Sales Offices in Europe are given at the end of this publication.

Copyright

All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying, processing or any information storage, without permission in writing by the Schneider Automation. You are not authorized to translate this document into any other language.

Trademarks

All terms used in this user manual to denote Schneider Automation products are trademarks of the Schneider Automation.

© 1997 Schneider Automation GmbH. All rights reserved.

Terminology



Note This symbol emphasizes very important facts.



Caution This symbol refers to frequently appearing error sources.



Warning This symbol points to sources of danger that may cause financial and health damages or may have other aggravating consequences.



Expert This symbol is used when a more detailed information is given, which is intended exclusively for experts (special training required). Skipping this information does not interfere with understanding the publication and does not restrict standard application of the product.



Path This symbol identifies the use of paths in software menus.

Figures are given in the spelling corresponding to international practice and approved by SI (Système International d' Unités).

I.e. a space between the thousands and the usage of a decimal point (e.g.: 12 345.67).

Abbreviations

ABS	absolute addressing
Adr.	Address (signal address)
AKF	Instruction List, Ladder Diagram, Function Block Diagram
IL	Instruction List
AWP	User program
DB 0 ... 9	Data block(= SYM/COM block)
D-Word	Double word
I/O	Input / Output signals (e.g. of a module)
FB	Function block
FBD	Function Block Diagram
G-Word	Floating point word
HW	Hardware (e.g. PLC)
IB	Initialization block
LD	Ladder diagram
OB	Organisation block
PB	Program block
PADT	Programming and debugging tool (= Programming panel)
React.	Reaction during descriptions of steps (on screen)
SFB	Standard function block
PLC	Programmable controller system (= programmable controller)
SSP	Signal memory
SW	Software
SYM	Symbolic addressing
SYM/COM	Symbols and comments
SA	Node
<Return>	actuate the Return key
<Esc>	actuate the Esc key
<Ctrl>+<Alt>+	actuate the keys Ctrl, Alt and Del simultaneously (begin with Ctrl and end with Del)

Objectives

The Short Form Guide gives an overview of the most important data for the software Modsoft AKF → Micro.

Arrangement of This Guide

- Chapter 1** Contains the validity ranges of the software.
- Chapter 2** contains a short description in table form of the system markers of the software Modsoft AKF → Micro.
- Chapter 3** contains the value ranges of the operands and an overview of the possible operations.
- Chapter 4** contains a list of the operating times of the individual operations and the cycle times of user programs.
- Chapter 5** contains general information to configure the modules ADU 205 with SFBs.
- Chapter 6** contains the description of all Standard Function Blocks (SFBs) available for Micro in MICROAKF Version 2.0 – 2.1. The SFBs are ordered alphabetically.

Validity Note

This instruction manual is valid for the software MICROAKF, version 2.0 – 2.1

Table of Contents

Chapter 1	Validity Ranges of the Software for Micro	1
1.1	Validity Ranges of the Software Modsoft AKF → Micro	2
Chapter 2	Tables of System Markers	5
2.1	System Markers	6
2.1.1	Definition/Use	6
2.2	System Markers (Bit)	7
2.3	System Markers (Bit) of I/O Peripherals	9
2.4	System Markers (Byte)	10
2.5	System Marker (Byte) of the I/O periphery	11
2.6	System Markers (Word)	12
2.7	System Markers (Double Word)	12
Chapter 3	Operands and Operations	13
3.1	Operands	14
3.1.1	Value Ranges	16
3.2	Operations	18
Chapter 4	Run Times	31
4.1	Run times for operations	32

Chapter 5 Configuration using SFBs 43

Index 47

Chapter 1

Validity Ranges of the Software for Micro

This chapter contains the validity ranges of the software.

1.1 Validity Ranges of the Software Modsoft AKF → Micro

- 500 blocks are possible. The division is carried out in the following manner:
There is one organization block. It is the OB1.
In total, a maximum of 499 program blocks or function blocks are possible.
The division between PB and FB is carried out by the user himself. The naming can be carried out independently of the number of blocks: PB1 to PB999 and FB1 to FB999.
- A maximum of 999 networks per block is permitted in the organization block OB1, in the program blocks (PBs) and in the function blocks (FBs). The number of the possible networks is dependent on the complexity of the individual instructions. If no comments, labels or parameters are used, it is limited by the maximum number of instructions per block (approx. 2 000).
- A user-function block can contain up to 45 input or 45 output parameters, with the sum of input and output parameters not to exceed 64.
- In a user function block the maximum number of instructions and moved bits is $(\text{DBB CNTxxx}) \leq 1.000$.
 $\Sigma \text{ instructions} + \Sigma \text{ xxx} \leq 1.000$
- A maximum of 16 outputs, all however with the same potential, is possible in ladder diagram and function block diagram.
- Program block calls and function block calls in LD/FBD can only be present in a network on their own.
- Networks can be up to 255 IL lines long.
- The maximum nesting depth for subexpressions is 13, therefore up to 13 brackets can be opened simultaneously.
- Up to 24 jump markers per network are permitted within a user-function block.

- ❑ A program block can be called several times. Recursive calling is also possible, with a maximum nesting depth of 10. For this reason, in a recursive call a program block can only be called conditionally.



Caution The user-programmed logic part in the program block must control this condition and provide for an abort condition within 10 calls. If this is not observed, the program is stopped automatically and system marker SM18 is set. Before the program can be re-started, it must be debugged.

- ❑ A data block DB can process a maximum of 1 800 symbolic names.
- ❑ The length of the signal comment is limited to 48 characters in total. 8 characters are intended as symbol names and up to 40 characters for the comment. A maximum of 1800 comments per DB is possible.
- ❑ In the ladder diagram/function block diagram there are no jumps, only block calls. After processing the block called, the program returns to the next instruction. Theoretically, 16 signals per network are interlockable in parallel and 7 (plus output) are interlockable serially in the ladder diagram.
- ❑ A maximum of 6 simple elements, horizontally aligned, are possible in the ladder diagram, and up to 12 elements may be vertically aligned with two inputs each.
- ❑ Outputs may not be negated.
- ❑ Up to 72 signals may be controlled and set (with the exception of system markers and time actual value).
- ❑ It is possible to force up to 72 signals

Chapter 2

Tables of System Markers

The following chapter contains a short description of the software Modsoft AKF → Micro system markers in table form.

2.1 System Markers

2.1.1 Definition/Use

The system markers are information in the form of bits, bytes, words and double words, which the system generates and controls.

The system markers may only be read and interlocked. Assignments on system markers are not possible.

Condition of the bit system markers

The system markers show a 0 signal in the non-fulfilled condition (i.e. normal operation).

The system markers show a 1 signal in the fulfilled condition (e.g. in case of fault).

It is distinguished between general System Markers and Participant Dependent Markers.

- General System Markers

- They are used for disturbance analysis and for evaluation of system data in the instruction list.

- Participant Dependent System Markers

- They are used for disturbance analysis and for evaluation of disturbances and errors of the I/O modules and intelligent function modules in the instruction list.

2.2 System Markers (Bit)

Operand	Meaning
SM1	User Program is running
SM2	Start-up marker (for hot restart and SM2=1: initialisation)
SM3	Collective error complete system (slot address fault)
SM4	Collective error slot address failed
SM5	Collective error slot address shut down
SM6	Reserved
SM7	V24-interface inactive
SM8	Bitbus-interface inactive
SM9	Reserved
SM10	Undervoltage BATTERY 1
SM11	Reserved
SM12	Flashing pulse 1 (1 HZ)
SM13	Flashing pulse 2 (2.5 HZ)
SM14	Flashing pulse 3 (5 HZ)
SM15	Flashing pulse 4 (10 HZ)
SM16	1: Restart-operation/0: New start-operation
SM17	1: FLASH-operation / 0: RAM-operation
SM18	1: Stack-overflow due to incorrect recursive call
SM19	Arithmetic overflow
SM20	Division by zero
SM21	AKF-Marker Status (set at 1 for one cycle duration, for active status function)
SM22	AKF-Marker Set (set at 1 for one cycle duration, for active set function)
SM23	AKF-Marker Force (set at 1 while Forcing remains active)
SM24	AKF-Marker Film
SM25	Power On
SM26	Actual value overflow in U/D counter
SM27	Group error SFB (for SFBs without AF)
SM28	Reserved
SM29	Reserved
SM30	Reserved

Operand	Meaning
SM31	Single Error Slot Address 1 Shut Down
SM32	Single Error Slot Address 2 Shut Down
SM33	Single Error Slot Address 3 Shut Down
SM34	" " 4 "
SM35	" " 5 "
SM36	" " 6 "
SM37	" " 7 "
SM38	" " 8 "
SM39	" " 9 "
SM40	" " 10 "
SM41	" " 11 "
SM42	" " 12 "
SM43	" " 13 "
SM44	" " 14 "
SM45	" " 15 "
SM46	" " 16 "
SM47	Single Error Slot Address 17 Shut Down
SM48	Single Error Slot Address 18 Shut Down
SM49	Is immediately set to logic 1 (without filter) when an interrupt is triggered on the INT1 input (on the master micro). The system marker will be logic 0 again, at the earliest 100 ms after the interrupt triggering, depending whether the interrupt is still there. When the interrupt is still there after 10 ms then the system marker remains logic 1 so long as INT1 is still there.
SM50	Is immediately set to logic 1 (without filter) when an interrupt is triggered on the INT2 input (on the master micro). The system marker will be logic 0 again, at the earliest 100 ms after the interrupt triggering, depending whether the interrupt is still there. When the interrupt is still there after 10 ms then the system marker remains logic 1 as long as INT2 is still there.

1)

1) Hardware equipment mounting of slot x (node x) does not match the Equipment list.

2.3 System Markers (Bit) of I/O Peripherals

Operand	Meaning
SM1.1	Slot Address 1 Failed
SM2.1	Slot Address 2 Failed
SM3.1	Slot Address 3 Failed
SM4.1	Slot Address 4 Failed
SM5.1	Slot Address 5 Failed
SM6.1	Slot Address 6 Failed
SM7.1	Slot Address 7 Failed
SM8.1	Slot Address 8 Failed
SM9.1	Slot Address 9 Failed
SM10.1	Slot Address 10 Failed
SM11.1	Slot Address 11 Failed
SM12.1	Slot Address 12 Failed
SM13.1	Slot Address 13 Failed
SM14.1	Slot Address 14 Failed
SM15.1	Slot Address 15 Failed
SM16.1	Slot Address 16 Failed
SM17.1	Slot Address 17 Failed
SM18.1	Slot Address 18 Failed

Module-dependent²⁾

2) Individual functions of module x (node x) are inoperative. For error description see the module description for each inoperative module.

2.4 System Markers (Byte)

Operand	Meaning
SMB1	Division remainder byte
SMB2	Number of actual-value normalizing cycles
SMB3	High-byte of Byte-mul (If Overflow)
SMB4	Reserved
SMB5	Weekday (0 = Sunday)
SMB6	Date-time (Century)
SMB7	Date-time (Year)
SMB8	Date-time (Month)
SMB9	Date-time (Day)
SMB10	Date-time (Hour)
SMB11	Date-time (Minute)
SMB12	Date-time (Second)
SMB13	Reserved
SMB14	Reserved
SMB15	Reserved
SMB16	Reserved
SMB17	Reserved
SMB18	Reserved
SMB19	Reserved
SMB20	Reserved

2.5 System Marker (Byte) of the I/O periphery

Operand	Meaning
IBx.y ³⁾	Gives a detailed description of the fault in the module, see the appropriate module description
SMB1.1	Slot Address 1 Error-code
SMB2.1	Slot Address 2 Error-code
SMB3.1	Slot Address 3 Error-code
SMB4.1	Slot Address 4 Error-code
SMB5.1	Slot Address 5 Error-code
SMB6.1	Slot Address 6 Error-code
SMB7.1	Slot Address 7 Error-code
SMB8.1	Slot Address 8 Error-code
SMB9.1	Slot Address 9 Error-code
SMB10.1	Slot Address 10 Error-code
SMB11.1	Slot Address 11 Error-code
SMB12.1	Slot Address 12 Error-code
SMB13.1	Slot Address 13 Error-code
SMB14.1	Slot Address 14 Error-code
SMB15.1	Slot Address 15 Error-code
SMB16.1	Slot Address 16 Error-code
SMB17.1	Slot Address 17 Error-code
SMB18.1	Slot Address 18 Error-code

Error coding⁴⁾
module dependent

3) for COS: IB1.1 ... IB3.128
for ADU 206: IB1.1 ... IB18.1

4) Table of error codes see respective module description of the faulty module

2.6 System Markers (Word)

Operand	Meaning
SMW1	Division remainder word
SMW2	Operating time user program in 10 μ s units (last cycle)
SMW3	Timebase 1000 ms
SMW4	Timebase 100 ms
SMW5	Timebase 10 ms
SMW6	High-word of word-MUL (at overflow)
SMW7	Number of counting pulses on CNT input (on the master micro) since power on
SMW8	Reserved
SMW9	Reserved
SMW10	Reserved

2.7 System Markers (Double Word)

Operand	Meaning
SMD1	Long time counter (counts up 1 every 10 ms; counts at stopped PLC, too)
SMD2	Division remainder double word
SMD3	High-DWORD of DWORD-MUL (at overflow)
SMD4	Reserved
SMD5	Reserved

Chapter 3

Operands and Operations

This chapter contains the address ranges of the operands and an overview about the possible operations.

3.1 Operands

Operand	Meaning	Range of Values**
I	Bit input	I1.1 ... I18.16
IB	Byte input	5)
IW	Word inputs	IW1.1 ... IW18.16
ID	Double word inputs	ID1.1 ... ID18.16
Q	Bit output	Q1.1 ... Q18.16
QB	Byte output	6)
QW	Word output	QW1.1 ... QW18.16
QD	Double word output	QD1.1... QD18.16
M	Marker bit	M1.1***... M125.2
MB	Marker byte	MB1 ... MB3970
MW	Marker word	MW1 ... MW1985
MD	Marker double word	MD1 ... MD992
SM	System marker bit	SM1 ... SM50
SM	System marker bit (I/O peripherals)	SM1.1 ... SM18.1
SMB	System marker byte	SMB1 ... SMB20
SMB	System marker byte I/O (Peripherals)	SMB1.1 ...SMB18.1
SMW	System marker word	SMW1 ... SMW10
SMD	System marker double word	SMD1 ... SMD5
V	Constants (decimal, +/-)	V-2147483648 ... V2147483647
VO	Constants (octal)	VO20000000000 ... VO17777777777
VH	Constants (hexadecimal)	VH80000000 ... VH7FFFFFFF
VM	Constants, samples (single bit representation)	
VC	Constants, characters (ASCII)	

5) for KOS: IB1.1 ... IB3.128
for ADU 206: IB1.1 ... IB18.1

6) for KOS: QB1.1 ... QB3.128
for ADU 206: QB1.1 ... QB18.1
for ADU 214: QB1.1 ... QB18.8

Operand	Meaning	Range of Values**		
T	Timer (timer)	T1	...	T567
TAW	Timer actual value	TAW1	...	TAW567
TSW	Timer setpoint value	TSW1	...	TSW567
C	Counter	C1	...	C794
CAW	Counter actual value	CAW1	...	CAW794
CSW	Counter setpoint value	CSW1	...	CSW794
TN	Node-No.	TN2	...	TN18

** dependant on the type of module; the maximum values are specified here

*** 1.1 ... 1.32, 2.1 ... 2.32 etc.

3.1.1 Value Ranges

Table 1 Operand Value Ranges

Operand	Element Type	Values* decimal (min. ... max.)
I, Q, M, SM	Bit	0 ... 1
IB, QB, MB, SMB	Byte	-128 ... +127 resp. 0 ... 255
EW, AW, MW, SMW	Word	-32 768 ... +32 767 resp. 0 ... 65 535
TSW, TIW	Word	-32 768 ... +32 767 resp. 0 ... 65 535
ZSW, ZIW	Word	-32 768 ... +32 767 resp. 0 ... 65 535
ED, AD, MD, SMD	Doubleword	-2 147 483 648 ... +2 147 483 647 resp. 0 ... 4 294 967 295
ZB (time base)	Constant	10ms, 100ms, 1000ms, 10000ms 60000ms
T	Constant	0 ... 1
C	Constant	0 ... 1
TN (Participant No.)	Constant	Participant Dependent
V	Constant (decimal)	-2 147 483 648 ... +2 147 483 647 resp. 0 ... 4 294 967 295
VO	Constant (octal)	-2 147 483 648 ... +2 147 483 647 resp. 0 ... 4 294 967 295
VH	Constant (hexadecimal)	-2 147 483 648 ... +2 147 483 647 resp. 0 ... 4 294 967 295
VM	Constant Sample (Single Bit)	0 ... 1
VC	Constant Character (ASCII)	-128 ... +127 bzw. 0 ... 255

* The value ranges are dependent on the operations used. With some operations no signs are allowed, the value range is then expanded within the positive range.

3.2 Operations

Operation	Meaning	Bit					Byte				
		I	Q	V**	M	SM	IB	QB	V**	MB	SMB
LOGIC											
logic operations											
A	AND, arithmetic										
	AND	+	+	+	+	+	+	+	+	+	+
AN	AND negated	+	+	-	+	+	-	-	-	-	-
A(AND with RLO of brackets	-	-	-	-	-	-	-	-	-	-
AN(AND negated with RLO of brackets	-	-	-	-	-	-	-	-	-	-
O	OR, arithmetic										
	OR	+	+	+	+	+	+	+	+	+	+
ON	OR negated	+	+	-	+	+	-	-	-	-	-
O(OR with RLO of brackets	-	-	-	-	-	-	-	-	-	-
ON(OR negated with RLO of brackets	-	-	-	-	-	-	-	-	-	-
X	EXCLUSIVE-OR, arithmet. XOR	+	+	+	+	+	+	+	+	+	+
XN	EXCLUSIVE-OR negated	+	+	-	+	+	-	-	-	-	-
X(EXCL.-OR with RLO of brackets	-	-	-	-	-	-	-	-	-	-
XN(EXCL.-OR negated with RLO of brackets	-	-	-	-	-	-	-	-	-	-
)	Termination of the brackets operation	-	-	-	-	-	-	-	-	-	-

** V means constant: V, VO, VH, VM

+ Operands marked with this sign may be used for these operations.

IW	QW	Word				ID	Double word			Additional operands
		V**	MW	SMW	TSW		QD	V**	MD	
TAW					CSW					
CAW										

+	+	+	+	+	+	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
+	+	+	+	+	+	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-

Operation	Meaning	Bit					Byte				
		I	Q	V**	M	SM	IB	QB	V**	MB	SMB
Setting/Resetting											
S	Set input, counter, Flip-Flop	-	+	-	+	-	-	-	-	-	-
R	Reset input, counter, Flip-Flop	-	+	-	+	-	-	-	-	-	
SU*	Set counter unconditionally, Flip-Flop	-	+	-	+	-	-	-	-	-	
RU*	Reset counter conditionally, Flip-Flop	-	+	-	+	-	-	-	-	-	
Counter/Timer											
TP	Pulse (Monoflop)	-	-	-	-	-	-	-	-	-	
TEP	Extended pulse	-	-	-	-	-	-	-	-	-	
TON	On-delay	-	-	-	-	-	-	-	-	-	
TS	Memorizing on-delay	-	-	-	-	-	-	-	-	-	
TOF	Off-delay	-	-	-	-	-	-	-	-	-	
CTU	Counter up	-	-	-	-	-	-	-	-	-	
CTD	Counter down	-	-	-	-	-	-	-	-	-	
C+	Count input up, up/down-counter	-	-	-	-	-	-	-	-	-	
C-	Count input down, combi-counter	-	-	-	-	-	-	-	-	-	
DTB	Defined time basis for timer	-	-	-	-	-	-	-	-	-	
Edge detection											
ED	Detection of a pos. or negative edge	-	+	-	+	-	-	-	-	-	
EDP	Detection of a pos. edge	-	+	-	+	-	-	-	-	-	
EDN	Detection of a neg. edge	-	+	-	+	-	-	-	-	-	

* currently not available

** V means constant: V, VO, VH, VM

+ Operands marked with this sign may be used for these operations.

Operation	Meaning	Bit					Byte				
		I	Q	V**	M	SM	IB	QB	V**	MB	SMB
Shifting/rotating											
SHL	Shift register to the left	-	-	-	-	-	-	-	+	-	-
SHR	Shift register to the right	-	-	-	-	-	-	-	+	-	-
ROL	Rotate register to the left	-	-	-	-	-	-	-	+	-	-
ROR	Rotate register to the right	-	-	-	-	-	-	-	+	-	-
Increment/decrement											
DEC	Decrement register	-	-	-	-	-	-	-	-	-	-
INC	Increment register	-	-	-	-	-	-	-	-	-	-
Transfer/load											
=	Transfer from register (assignment)	-	+	-	+	-	-	+	-	+	-
=C	Transfer into connector (register contents continued processing)	-	+	-	+	-	-	+	-	+	-
LBW	Load bit string into the word register	+	+	-	+	+	+	+	-	+	+
TBW	Transfer bit string from the word register	-	+	-	+	-	-	+	-	+	-
LBB	Load bit string in byte register	+	+	-	+	+	-	-	-	-	-
TBB	Transfer bit string from byte register	-	+	-	+	-	-	-	-	-	-
DBB	Number of bits (for loading/transfer of bit string)-	-	-	-	-	-	-	-	-	-	-
LBS*	Bit selector of register	-	-	-	-	-	-	-	+	-	-

* currently not available

** V means constant: V, VO, VH, VM

+ Operands marked with this sign may be used for these operations.

IW	QW	Word				ID	Double word				Additional operands	
		V**	MW	SMW	TSW		QD	V**	MD	SMD		
TAW					CSW							
CAW												

-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	+	-	+	-	+	-	+	-	+	-	-	-
-	+	-	+	-	+	-	+	-	+	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	CNT xx
-	-	-	-	-	-	-	-	-	-	-	-	-

Operation	Meaning	Bit					Byte				
		I	Q	V**	M	SM	IB	QB	V**	MB	SMB
LLD	Load byte/word into LOW-word/double word	-	-	-	-	-	+	+	-	+	+
TLD	Transfer byte/word from LOW-word/double word	-	-	-	-	-	-	+	-	+	-
LHD	Load byte/word into HIGH-word/double word	-	-	-	-	-	+	+	-	+	+
THD	Transfer byte/word from HIGH-word/double word	-	-	-	-	-	-	+	-	+	-
Jumps											
JL	Unconditional jump	-	-	-	-	-	-	-	-	-	-
JT	Conditional jump in logic 1	-	-	-	-	-	-	-	-	-	-
JF	Conditional jump in logic 0	-	-	-	-	-	-	-	-	-	-
JN*	Conditional jump in register, unequal to 0	-	-	-	-	-	-	-	-	-	-
JE*	Conditional jump in register larger than 0	-	-	-	-	-	-	-	-	-	-
JL*	Conditional jump in register smaller than 0	-	-	-	-	-	-	-	-	-	-
JO*	Conditional jump in overflow	-	-	-	-	-	-	-	-	-	-
Block call											
BC	Unconditional block call	-	-	-	-	-	-	-	-	-	-
BCC	Conditional block-call (in log. "1")	-	-	-	-	-	-	-	-	-	-
BCZ	Conditional block-call (in log. "0")	-	-	-	-	-	-	-	-	-	-
Blank operations											
NOP	Blank operation	-	-	-	-	-	-	-	-	-	-
FREE	place holder for released outputs (e.g. SR Flip Flop)	-	-	-	-	-	-	-	-	-	-

* currently not available

** V means constant: V, VO, VH, VM

+ Operands marked with this sign may be used for the operation.

IW	QW	Word				ID	Double word				Additional operands	
		V**	MW	SMW	TSW		QD	V**	MD	SMD		
TAW					CSW							
CAW												

+	+	-	+	+	+	-	-	-	-	-	-	-
-	+	-	+	-	+	-	-	-	-	-	-	-
+	+	-	+	+	+	-	-	-	-	-	-	-
-	+	-	+	-	+	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	=
-	-	-	-	-	-	-	-	-	-	-	-	=
-	-	-	-	-	-	-	-	-	-	-	-	=
-	-	-	-	-	-	-	-	-	-	-	-	=
-	-	-	-	-	-	-	-	-	-	-	-	=
-	-	-	-	-	-	-	-	-	-	-	-	=
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-

Operation	Meaning	Bit					Byte				
		I	Q	V**	M	SM	IB	QB	V** VC	MB	SMB
ARITHMETICAL											
Loading											
LD	Load with sign in register	-	-	-	-	-	+	+	+	+	+
Arithmetic operation											
ADD	Addition	-	-	-	-	-	+	+	+	+	+
SUB	Subtraction	-	-	-	-	-	+	+	+	+	+
MUL	Multiplication	-	-	-	-	-	+	+	+	+	+
DIV	Division	-	-	-	-	-	+	+	+	+	+
ADD(*Addition (register + bracket result)	-	-	-	-	-	-	-	-	-	-
SUB(*	Subtraction (register - bracket result)	-	-	-	-	-	-	-	-	-	-
MUL(*	Multiplication (register x bracket result)	-	-	-	-	-	-	-	-	-	-
DIV(*	Division (register : bracket result)	-	-	-	-	-	-	-	-	-	-
ABS*	Taking the absolute value	-	-	-	-	-	-	-	-	-	-
SIG*	Sign test	-	-	-	-	-	-	-	-	-	-
Complementation											
OCW*	Forms one's complement of the register contents	-	-	-	-	-	-	-	-	-	-
TCW*	Form two's complement of the register contents	-	-	-	-	-	-	-	-	-	-

* currently not available

** V means constant: V, VO, VH, VM

+ Operands marked with this sign may be used for these operations.

IW	QW	Word				ID	Double word				Additional operands
		V**	MW	SMW	TSW		QD	V**	MD	SMD	
TAW					CSW						
CAW											

+	+	+	+	+	+	+	+	+	+	+	-
---	---	---	---	---	---	---	---	---	---	---	---

+	+	+	+	+	+	+	+	+	+	+	-
+	+	+	+	+	+	+	+	+	+	+	-
+	+	+	+	+	+	+	+	+	+	+	-
+	+	+	+	+	+	+	+	+	+	+	-

-	-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---	---

-	-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---	---

-	-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---	---

-	-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---	---

-	-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---	---

-	-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---	---

-	-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---	---

Operation	Meaning	Bit					Byte				
		I	Q	V**	M	SM	IB	QB	V** VC	MB	SMB
Comparators											
>	Greater than register	-	-	-	-	-	+	+	+	+	+
>=	Greater than or equal to register	-	-	-	-	-	+	+	+	+	+
=	Equal to register	-	-	-	-	-	+	+	+	+	+
<	Smaller than register	-	-	-	-	-	+	+	+	+	+
<=	Smaller than or equal to register	-	-	-	-	-	+	+	+	+	+
<>	Not equal to register	-	-	-	-	-	+	+	+	+	+
> (*	register greater than bracket result	-	-	-	-	-	-	-	-	-	-
>= (*	register greater than or equal to bracket result	-	-	-	-	-	-	-	-	-	-
= (*	register equals bracket result	-	-	-	-	-	-	-	-	-	-
< (*	register smaller than bracket result	-	-	-	-	-	-	-	-	-	-
<= (*	register smaller than or equal to bracket result	-	-	-	-	-	-	-	-	-	-
<>(*	register not equal to bracket result	-	-	-	-	-	-	-	-	-	-
End of block											
***	End of network	-	-	-	-	-	-	-	-	-	-
BE	Unconditional block end	-	-	-	-	-	-	-	-	-	-
BEC	Conditional block end for logic 1	-	-	-	-	-	-	-	-	-	-
BEZ	Conditional block end for logic 0	-	-	-	-	-	-	-	-	-	-
**	Optional one-line comment	-	-	-	-	-	-	-	-	-	-

* currently not available

** V means constant:V, VO, VH, VM

+ Operands marked with this sign may be used for these operations.

Chapter 4

Run Times

The following chapter contains a list of the runtimes of the individual operations and the cycle times of user programs.

4.1 Run times for operations

The times stated are dependent on the processor and are for information purposes only. The overflow treatment during the operations ADD, SUB, MUL (not when MUL with double-word) require 3 ms to be added for each 10 Bytes.

The command lengths of the operations for CPU 311/411 correspond with those stated for CPU 512/612.

For the run times on CPU 311/411 the run time stated for CPU 512/612 has to be doubled.

Function (in IL)	CPU 512 / 612	
	Byte	min. μ s

Logic

A bit, byte	4	1.4
AN bit	8	2.4
AN bit, parameter	10	3.8
A bit, byte, constant	2	0.5
A bit, byte, parameter	6	2.7
A word	4	1.7
A word, constant	3	0.8
A word, parameter	6	2.9
A(..) bit, byte	4	3.0
AN(..) bit	6	3.4
A(..) word	4	3.0
O bit, byte	4	1.4
ON bit	8	2.4
ON bit, parameter	10	3.8
O bit, byte, constant	2	0.5
O bit, byte, parameter	6	2.7
O word	4	1.7
O word, constant	3	0.8
O wort, parameter	6	2.9
O(..) bit, byte	4	3.0
ON(..) bit	6	3.4
O(..) word	4	3.0

* Values in IL after network end *** or (

Function (in IL)	CPU 512 / 612	
	Byte	min. μ s
X bit, byte	4	1.4
XN bit	8	2.4
XN bit, parameter	10	3.8
X bit, byte, constant	2	0.5
X bit, byte, parameter	6	2.7
X word	4	1.7
X word, constant	3	0.8
X word, parameter	6	2.9
X(..) bit, byte	4	3.0
XN(..) bit	6	3.4
X(..) word	4	3.0
ED	10	4.3
ED, parameter	10	4.8
EDP	12	4.4
EDP, parameter	12	5.0
EDN	14	4.6
EDN, parameter	14	5.2
SHL byte	4	upto 1.1
SHL word	4	upto 1.6
SHR byte	4	upto 1.1
SHR word	4	upto 1.6
ROL byte	4	upto 1.1
ROL word	4	upto 1.6
ROR byte	4	upto 1.1
ROR word	4	upto 1.6
DEC register	2	0.5
DEC word register	1	0.3
INC register	2	0.5
INC word register	1	0.3

* Values in IL after network end *** or (

Function (in IL)	CPU 512 / 612	
	Byte	min. μ s
<hr/>		
S memory	9	4.3
S memory, parameter	10	4.9
R memory	11	4.5
R memory, parameter	12	5.1
CTU	8	25.8
CTU, parameter	9	26.8
CTD	8	25.8
CTD, parameter	9	26.8
C+	21	41.8
C+, parameter	22	42.8
C-	21	41.8
C-, parameter	22	42.8
S CTU	8	18.0
S CTU, parameter	9	19.0
S CTD	8	18.0
S CTD, parameter	9	19.0
S C+, C-	8	18.0
S C+, C-, parameter	9	19.0
R counter	17	19.8
R counter, Parameter	18	18.8
R time	8	13.0
R time, parameter	9	14.0
TP	8	37.4
TP, parameter	9	38.4
TEP	8	37.4
TEP, parameter	9	38.4
TON	8	37.4
TON, parameter	9	38.4
TS	8	37.4
TS, parameter	9	38.4
TOF	8	37.4
TOF, parameter	9	38.4

Function (in IL)	CPU 512 / 612	
	Byte	min. μ s

TB time base	5	1.0
CSW/TSW	4	1.6
CSW/TSW, constant	3	0.6
CSW/TSW, parameter	6	3.0

Load/transfer

LD byte	3	1.1
LD byte, constant	2	0.5
LD byte, parameter	6	2.3
LD word	3	1.4
LD word, constant	3	0.8
LD word, parameter	6	2.6
LD double word	7	2.8
LD Double word, constant	6	1.9
LD Double word, parameter	9	3.6
LLD byte \rightarrow word	4	1.4
LLD byte \rightarrow word, parameter	7	2.6
LLD word \rightarrow double word	4	1.6
LLD word \rightarrow double w., parameter	7	2.8
LHD byte \rightarrow word	4	1.5
LHD byte \rightarrow word, parameter	6	2.7
LHD word \rightarrow double word	4	1.5
LHD word \rightarrow double w., parameter	6	2.7

Function (in IL)	CPU 512 / 612	
	Byte	min. μ s
≠C bit, byte	3	1.1
≠C bit, byte, parameter	6	2.8
≠C word	3	1.4
≠C word, parameter	6	3.0
≠C double word	7	3.2
≠C double word, parameter	9	4.6
TLD word → byte	4	1.4
TLD word → byte, parameter	7	3.1
TLD double word → word	4	1.6
TLD double word → word param.	7	3.2
THD word → byte	4	1.6
THD word → byte, parameter	6	3.1
THD double word → word	4	1.8
THD double word → word param.	6	3.4
LBB DBB bits → byte register	11	upto 13.9
LBB DBB bits → byte register param.	13	upto 15.1
LBW DBB bits → word register	11	upto 20.6
LBW DBB bits → word register param.	13	upto 21.8
LBW DBB Byte → word register	9	3.0
LBW DBB bytes → word register param.	16	6.6
TBB DBB byte register → bits	11	upto 28.3
TBB DBB byte register → bits param.	13	upto 29.5
TBW DBB word register → bits	11	upto 41.5
TBW DBB word register → bits param.	13	upto 42.7
TBW DBB word register → bytes	15	4.0
TBW DBB word register → bytes param.	19	7.0

Arithmetical

ADD byte	4	1.4
ADD byte, constant	2	0.5
ADD byte, parameter	6	2.7
ADD word	4	1.7
ADD word, constant	3	0.8
ADD word, parameter	6	2.9
ADD double word	8	3.5
ADD double word, constant	7	1.9
ADD double word, parameter	9	4.4
SUB byte	4	1.4
SUB byte, constant	2	0.5
SUB byte, parameter	6	2.7
SUB word	4	1.7
SUB word, constant	3	0.8
SUB word, parameter	6	2.9
SUB double word	8	3.5
SUB double word, constant	7	1.9
SUB double word, parameter	9	4.4
MUL byte	8	4.0
MUL byte, constant	8	3.6
MUL byte, parameter	10	5.3
MUL word	8	5.1
MUL word, constant	9	4.7
MUL word, parameter	10	6.3
MUL double word	25	64.6
MUL double word, constant	23	64.6
MUL double word, parameter	26	64.6
DIV byte	14	7.4
DIV byte, constant	14	6.9
DIV byte, parameter	16	8.5

Function (in IL)	CPU 512 / 612	
	Byte	min. μ s

DIV word	14	8.4
DIV word, constant	15	8.0
DIV word, parameter	16	9.6
DIV double word	26	236.3
DIV double word, constant	24	236.3
DIV double word, parameter	27	242.0
> byte	10	3.4
> byte, constant	8	2.4
> byte, parameter	12	4.8
> word	10	3.4
> word, constant	9	2.6
> word, parameter	12	4.8
> double word	20	5.8
> double word, constant	19	4.2
> double word, parameter	21	7.2
>= byte	10	3.4
>= byte, constant	8	2.4
>= byte, parameter	12	4.8
>= word	10	3.4
>= word, constant	9	2.6
>= word, parameter	12	4.8
>= double word	14	4.6
>= double word, constant	13	3.0
>= double word, parameter	15	6.0
== byte	10	3.4
== byte, constant	8	2.4
== byte, parameter	12	4.8
== word	10	3.4
== word, constant	9	2.6
== word, parameter	12	4.8
== double word	16	5.0
== double word, constant	15	3.4
== double word, parameter	17	6.6

Function (in IL)	CPU 512 / 612	
	Byte	min. μ s
< byte	10	3.4
< byte, constant	8	2.4
< byte, parameter	12	4.8
< word	10	3.4
< word, constant	9	2.6
< word, parameter	12	4.8
< double word	14	4.6
< double word, constant	13	3.0
< double word, parameter	15	6.0
<= byte	10	3.4
<= byte, constant	8	2.4
<= byte, parameter	12	4.8
<= word	10	3.4
<= word, constant	9	2.6
<= word, parameter	12	4.8
<= double word	20	5.8
<= double word, constant	19	4.2
<= double word, parameter	21	7.2
<> byte	10	3.4
<> byte, constant	8	2.4
<> byte, parameter	12	4.8
<> word	10	3.4
<> word, constant	9	2.6
<> word, parameter	12	4.8
<> double word	16	5.0
<> double word, constant	15	3.4
<> double word, parameter	17	6.6

Function (in IL)	CPU 512 / 612	
	Byte	min. μ s

Conversion

bit \rightarrow byte	2	0.4
Bit \rightarrow word	3	0.6
Byte \rightarrow word, without sign	2	0.4
Byte \rightarrow word, with sign	1	0.4
Byte \rightarrow double word, without sign	5	1.0
Byte \rightarrow double word, with sign	2	0.8
Word \rightarrow double word, without sign	3	0.6
Word \rightarrow double word, with sign	1	0.6

Block handling

BC	30	11.0
BCC	37	13.0
BCZ	37	13.0
BE	1	2.8
BEC	5	3.8
BEZ	5	3.8

Jump commands

Jl label	3	1.8
JT jump	7	2.8
JF	7	2.8
Parameter for FB call	4	1.8
Pass parameters for FB call	5	2.8

Formula for estimating the memory demand of blocks

- 300 OB1 (empty)
- + 66 parameter
- + 27 * Number of unconditional PB-calls
- + 31 * Number of unconditional PB-calls
- + 34 * Number of conditional PB-calls
- + 38 * Number of conditional PB-calls
- + 7 * Number of conditional Block Ends
- + 4 * number of parameter deliveries
- + 5 * number of parameter transfers

= byte complete program

Conversion times when using MICRO AKF



Note The conversion time in the Hardware User Manual (423 702 485) that is indicated with 52 ms per channel refers exclusively to hardware. This is generally a software independent time, though when using MICRO AKF there will be a time ranging from typically 100 up to 200 ms per channel, plus the AWP scan time.

Scan times for User Programs



Note The overall length of a user program is not a reliable indication of the expected scan time. Only those parts of the program which are actually processed during run time contribute something to the scan time. Thus this time can vary from one scan to the next. The byte figure refers to the executable program code, and stands in no relation to the size actually to be loaded. The size of reproductions, user comments and user data fields (data structures, initial values, etc.) as well as the components of automatically generated basic software should be added.

Times for User Program	CPU 311 / 411	CPU 512 / 612
Scan time		
Basic load (without extencion)	2.4 ms	1.2 ms
per 1 K binary instructions	7.2 ms	3.6 ms

Chapter 5

Configuration using SFBs

This chapter contains general information on configuring of specific modules using SFBs.

ADU204

Table 2 Standard Function Blocks (SFB's) for ADU 204

SFB-No.	Name
---------	------

SFB 86	PT100
--------	-------

ADU205

Table 1 Standard Function Blocks (SFB's) for ADU 205

SFB-No.	Name
---------	------

SFB 85	ADU 205
--------	---------

For a programming example see in user instruction "Configuration" on page 107 (ADU 205)

FRQ204

Table 2 Standard Function Blocks (SFB's) for FRQ 204

SFB-No.	Name
---------	------

SFB 87	FRQ 204
--------	---------

For more information see in user instruction FRQ 204 Print-No. A91M.12-703 821 (german)

TXT201

Table 3 Standard Function Blocks (SFB's) for TXT 201

SFB-No.	Name
SFB 90	TXT_SEND
SFB 91	TXT_RCV
SFB 92	TXT_WRTE1
SFB 93	TXT_WRTE2

For more information see in user instruction TXT 201 Print-No. A91M.12-701 748 (german)

ZAE201

Table 4 Standard Function Blocks (SFB's) for ZAE 201

SFB-No.	Name
SFB 31	ZAE
SFB 32	WEG

ZAE204

Table 5 Standard Function Blocks (SFB's) for ZAE 204

SFB-No.	Name
SFB 33	ZAE 204

For more information see in user instruction ZAE 204 Print-No. A91M.12-271 962 (german)

Index

C

Configuration
ADU 204, 44
ADU 205, 44
FRQ 204, 44
TXT 201, 45
ZAE 201, 45
ZAE 204, 45
Conversion time, 42

D

Data Block, DB, 3

F

Formulae memory demand, 42
Function Block Diagram, FBD, 2, 3

I

IL lines, 2

L

Ladder Diagram, LD, 2, 3

M

Module, ADU 204, 44
Modules, ADU 205, 44

N

nesting depth, 2, 3

O

Operand Value Ranges, 16
Operands, 14
Operations for IL, LD, FBD, 18–30
Organization Block, OB, 2

R

Range of values, 14, 15
recursive call, 3

S

Scan times (User Program), 42
System Markers, 6
 (bit) from center, 7–12
 (bit) of the nodes, 9–12
 (byte) of the I/O-periphery, 11
 (bytes) of the center, 10–12
 (double word) of the center, 12
 (word) of the center, 12
 definition, 6
 Short description, 5

V

Validity Ranges of the Software
Modsoft AKF – Micro, 2

