AEG ID

Manual

ACM 9

ARE i9

1. Introduction 5
2. ACM 9 6
2.1 Hardware 6
2.1.1 Dimensions 6
2.1.2 Protection Class 6
2.1.3 Connectivity 7
2.1.4 Grounding 9
2.2 Firmware 10
2.2.1 Industrial Ethernet description files 10
2.2.2 Output data telegram from Industrial Ethernet master 10
2.2.3 Input data telegram to Industrial Ethernet master 10
2.2.4 ACM 9 error messages 11
2.2.5 Error messages coming through from ARE i9 12
3. ARE i9 13
3.1 Hardware 13
3.1.1 Dimensions 13
3.1.2 Protection Class 13
3.1.3 Connectivity 14
3.1.4 Transponder orientation relative to ARE i9 15
3.1.5 Read range for SEMI application 16
3.1.6 Read range for miscellaneous applications 17
3.2 Firmware ARE i9 LF hdx - SEMI Industry 18
3.2.1 Instruction Set 18
3.2.2 General format of instruction set 18
3.2.3 VER 18
3.2.4 GT 19
3.2.5 TOR 19
3.2.6 NID 20
3.2.7 CID 20
3.2.8 CN 21
3.2.9 RD 21
3.2.10 WD 21

3.2.11 LD 22
3.2.12 VSAVE 22
3.2.13 INIT 22
3.2.14 Error messages 23
3.3 Firmware ARE i9 LF 24
3.3.1 Instruction Set 24
3.3.2 General format of instruction set 24
3.3.3 VER 24
3.3.4 GT 25
3.3.5 TOR 25
3.3.6 NID 26
3.3.7 CID 26
3.3.8 CN 27
3.3.9 RD 27
3.3.10 WD 27
3.3.11 VSAVE 28
3.3.12 INIT 28
3.3.13 Error messages 29
3.3.14 ALGO 29
3.3.15 LOG (EM4305 chip specific) 30
3.3.16 PWD (EM4305 chip specific) 30
3.3.17 LD (EM4305 chip specific) 31
3.4 Firmware ARE i9 HF 32
3.4.1 Instruction Set 32
3.4.2 General format of instruction set 32
3.4.3 VER 32
3.4.4 GT 33
3.4.5 TOR 33
3.4.6 CID 34
3.4.7 CN 34
3.4.8 RD 35
3.4.9 WD 35
3.4.10 VSAVE 35

3.4.11 INIT 36

3.4.12 Error messages 36

3.5 LED instruction set... 37

3.5.1 LED Standby (LSTB) 38

3.5.2 LED Reading (LGT) 38

3.5.3 LED Transponder number successfully read (LRD)... 39

3.5.4 LED No Read (LNRD)... 39

3.5.5 LED Return to standby (LRT) 39

3.5.6 LED Error (LERR) 40

3.5.7 LED Process active 40

3.5.8 LED Process status... 41

3.5.9 LED Setup help (FLED) 41

3.5.10 LED (De)activate LED functionality (LED)... 42

4. System implementation 43

4.1 Power supply... 43

4.2 Grounding... 43

4.3 Mounting on metal... 43

4.4 Frequency converters... 43

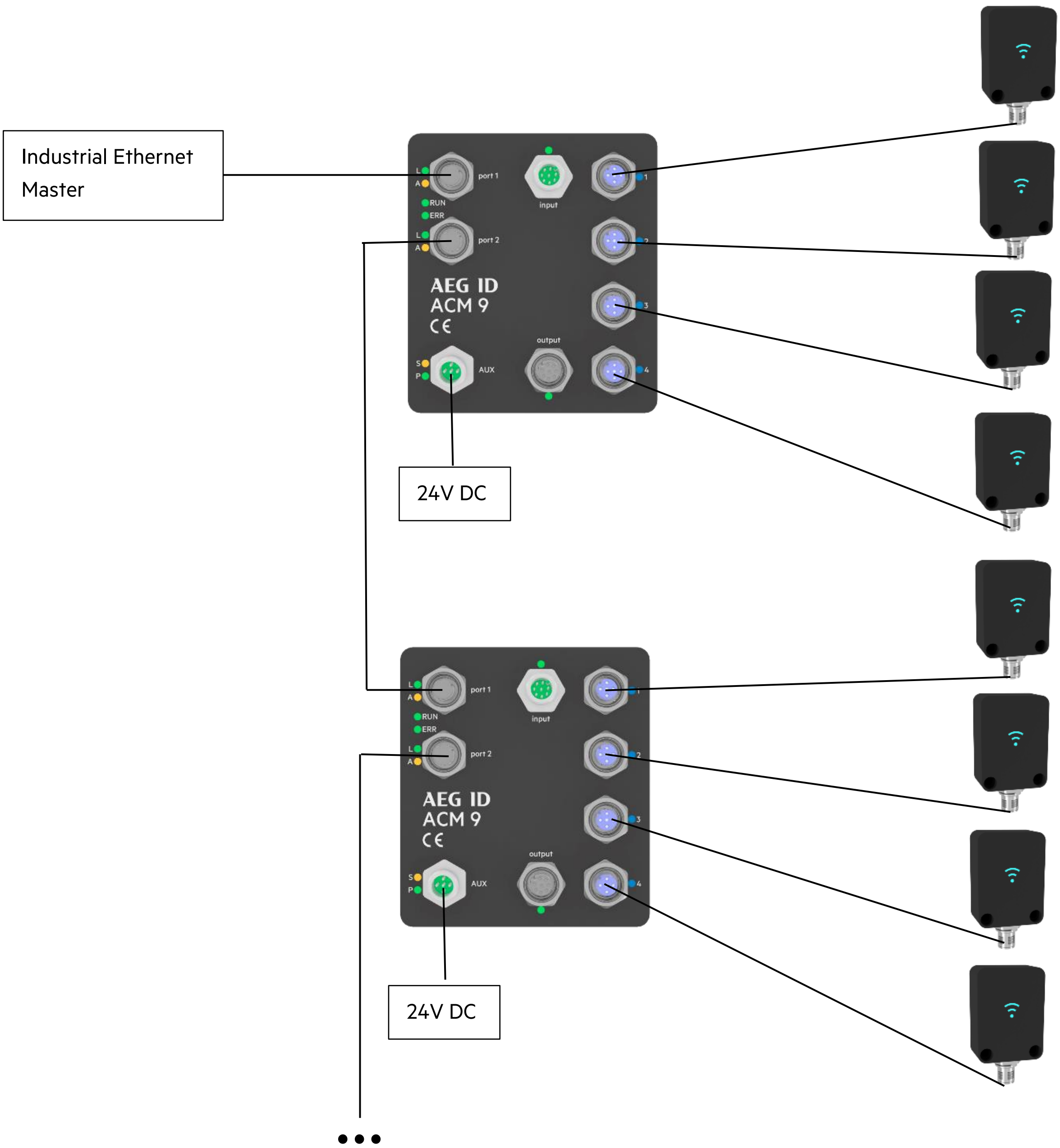
5. Release, Change Protocol 43

1. Introduction

ACM 9 Industrial Ethernet (EtherCAT®, EtherNet/IP™, Profinet®), AEG ID Communication Module, provides Industrial Ethernet connectivity for ARE i9, AEG ID RFID reader family. Both units are dedicated to each other, none of the two is meant to be used without the other. Four ARE i9 units can be hooked up to one ACM 9.

EtherCAT® is a registered trademark of Beckhoff Automation GmbH, Verl,. Germany
EtherNet/IP™ is a trademark of ODVA (Open DeviceNet Vendor. Association, Inc)
PROFINET® is a registered trademark of PROFIBUS International, Karlsruhe.

Typical system structure

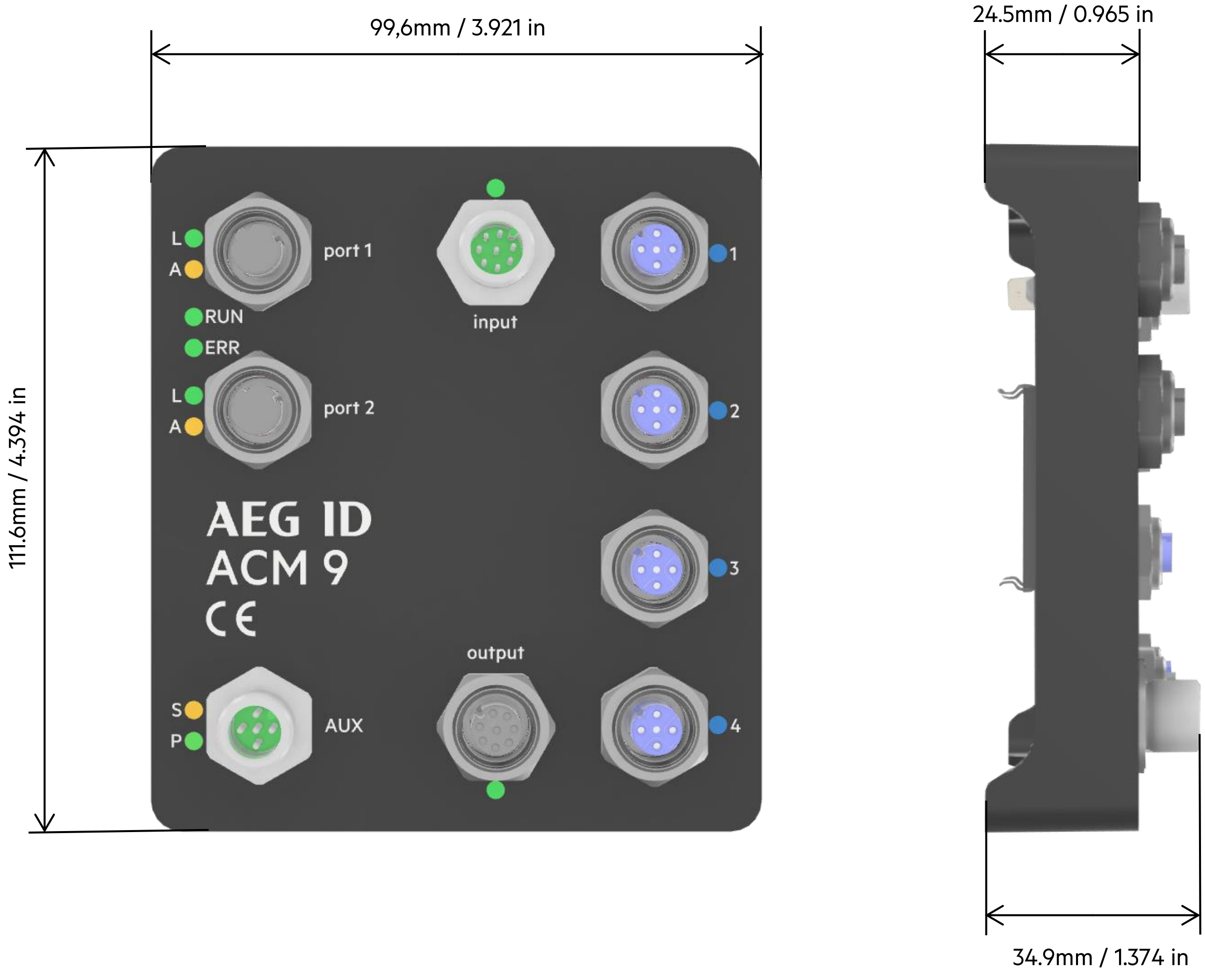


2. ACM 9

ACM 9 Industrial Ethernet communication module supports Industrial Ethernet connectivity. EtherCAT, EtherNet/IP and Profinet are implemented (only one at a time). ACM 9 has 2 Industrial Ethernet ports, 4 serial ports dedicated to ARE i9 reader family, 4 digital inputs and 4 digital outputs, both linked to their respective reader port. The auxiliary port provides power as well as a serial system interface. ACM 9 supports 32 byte cyclic input data and 32 byte cyclic output data.

2.1 Hardware

2.1.1 Dimensions

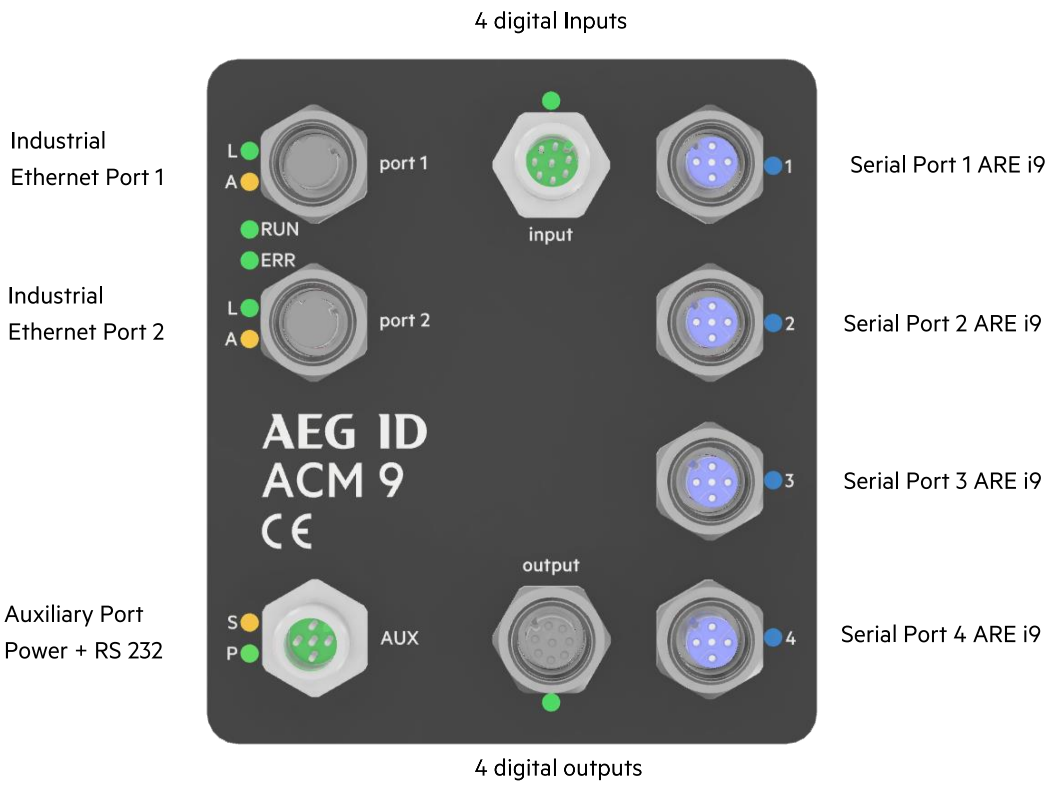


2.1.2 Protection Class

Protection class is IP65 assuming all cables and / or dummy caps are mounted.

IP67 available upon request.

2.1.3 Connectivity



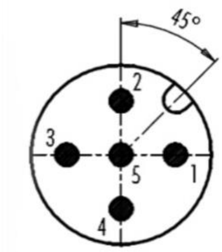
Auxiliary Port (AUX)

M12 socket 5 Pin male A-coded.

PIN 1 – nc
PIN 2 – GND
PIN 3 – TX (from ACM 9 point of view)
PIN 4 – RX (from ACM 9 point of view)
PIN 5 – +24V DC / 3A

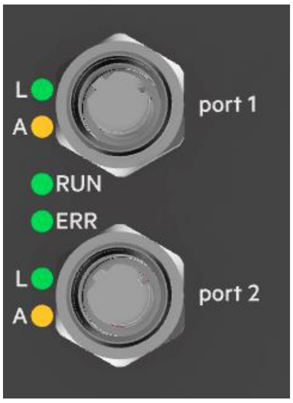
LED S: System interface LED
LED P: Power LED

Cable: M12, 5 Pin A-coded, socket to open end

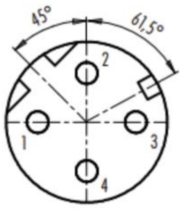


Industrial Ethernet Port (Port 1, Port 2)

M12 socket 4 Pin female D-coded (standard Industrial Ethernet)



PIN 1 – TD+
PIN 2 – RD+
PIN 3 – TD-
PIN 4 – RD-



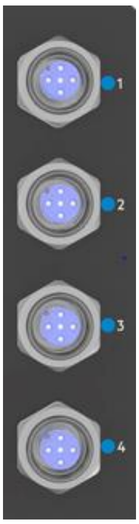
LED L: Link
LED A: Activity

Profinet	EtherNet/IP	EtherCAT
LED SF: Collective failure	LED MS: Mod Status	LED RUN: system status
LED BF: Bus failure	LED NS: Net Status	LED ERR: failure LED

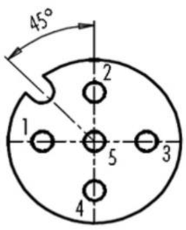
Cable: M12 4-Pin D-coded, plug to RJ45

Serial Ports for ARE i9 (1-4)

M12 socket 5 Pin female A-coded



PIN 1 – +7V
PIN 2 – GND
PIN 3 – TX
PIN 4 – RX
PIN 5 – nc



LED 1-4: Activity LED, cyan when in use

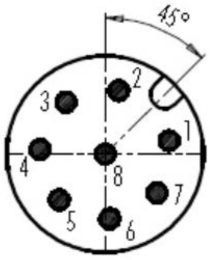
Cable: M12, 5-Pin A-coded, plug to socket

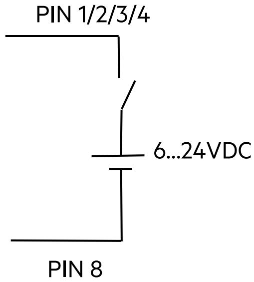
Digital Inputs (input)

M12 socket 8-Pin male A-coded

PIN 1 – Input for serial port 1
PIN 2 – Input for serial port 2
PIN 3 – Input for serial port 3
PIN 4 – Input for serial port 4
PIN 5 –7: not connected
PIN 8 - Input GND
LED : on, when any input is active

Input voltage 6..24VDC
Cable: M12, 8-pin A-coded socket to plug

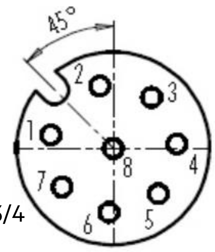


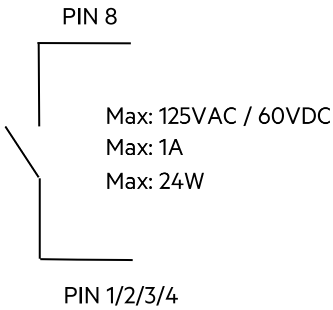


Digital outputs (output)

M12 socket 8-Pin female A-coded

PIN 1 – Output for serial port 1
PIN 2 – Output for serial port 2
PIN 3 – Output for serial port 3
PIN 4 – Output for serial port 4
PIN 5 – 7: not connected
PIN 8 – External output voltage
connected through PIN 1/2/3/4
LED : on, when any output is active
Switched voltage max: 125VAV / 60VDC
Switched current max: 1A | power max: 24W
Cable: M12, 8-pin A-coded plug to socket





2.1.4 Grounding

Grounding ACM 9 is important for proper functionality.

Grounding can be achieved in two ways:

1. Grounding via DIN hat rail clamp
   The metal DIN hat rail clamp is connected to internal ground of ACM 9. Grounding DIN hat rail in machine will connect internal ground to machine ground.
2. Use grounding pin on upper left side (viewed from back).
   Hook up machine ground to grounding pin

2.2 Firmware

ACM 9 uses 32 Byte cyclic input data and 32 Byte cyclic output data.

2.2.1 Industrial Ethernet description files

Profinet:
File ACM9_PNS_V1_1_0.xml contains Profinet description for ACM 9.
EtherCAT:
File ACM9_ECS_V1_1_0.xml contains EtherCAT description for ACM 9.
EtherNet/IP:
File ACM9_EIS_V1_1_0.eds contains EtherNet/IP description for ACM 9.

2.2.2 Output data telegram from Industrial Ethernet master

The telegram starts with a change counter byte. This byte has to be changed, typically incremented, for every new telegram. The second byte of the telegram is the serial interface port of the RFID reader. It can be '1', '2', '3' or '4' (0x31, 0x32, 0x33, 0x34). Starting from byte 3 the instruction text (instruction and parameters) for the reader is put in the following bytes. The supported commands are documented in section ARE i9. The telegram must be closed with a carriage return (0x0D). The complete telegram can be up to 32 byte long.

EXAMPLE:
The output data of the bus master to read a transponder code using ARE i9 on serial port 2 looks this way:

Hex:	01	32	47	54	0D	00	00	...		
ASCII:		'2'	'G'	'T'	<CR>					
Meaning:	Change counter	Serial interface port	Instruction Text		End character					

2.2.3 Input data telegram to Industrial Ethernet master

Above output telegram has been sent to ACM 9. ACM 9 sends the command to the appropriate ARE i9, in this case to the ARE i9 hooked up to serial interface port 2. ARE i9 reads the transponder code (UID of transponder) and sends it to ACM 9. ACM 9 puts the transponder UID into the input data telegram frame for Industrial Ethernet master.

The telegram starts with a change counter byte. This byte is changed, typically incremented, for every new telegram ACM 9 puts on the bus. The second byte of the telegram is the serial interface port of the RFID reader. It can be '1', '2', '3' or '4' (0x31, 0x32, 0x33, 0x34). Starting from byte 3 the answer from ARE i9 (in this case the transponder UID) is put in the following bytes. The telegram closes with a carriage return (0x0D). The complete telegram can be up to 32 byte long.

EXAMPLE:
The input data telegram for the Industrial Ethernet master is set by ACM 9 for example this way (UID = 1234567812345678):

Hex:	1F	32	31	32	...	38	0D	00	...	
ASCII:		'2'	'1'	'2'	...	'8'	<CR>			
Meaning	Change counter	Serial interface port	Transponder code				End character			

2.2.4 ACM 9 error messages

ACM 9 itself has the following error messages.

Wrong channel number (Error code #65)

A wrong channel number is set to the master output data. Valid channel numbers are from '1' to '4'.

Hex:	02	35	47	54	0D	00	00	...		
ASCII:		'5'	'G'	'T'	<CR>					
Meaning:	Change counter	Serial interface port	Instruction Text		End character					

ACM 9 sets master input data to:

Hex:	20	00	15	23	36	35	0D	00	...	
ASCII:			<NAK>	'#'	'6'	'5'	<CR>			
Meaning	Change counter	Serial interface port	Error Text				End character			

Missing end character (Error code #66)

Missing end character (0x0D) in master output data. Each telegram has to be terminated by 0x0D (<CR>).

Hex:	03	34	47	54	00	00	00	...		
ASCII:		'4'	'G'	'T'						
Meaning:	Change counter	Serial interface port	Instruction Text							

ACM 9 sets master input data to:

Hex:	21	00	15	23	36	36	0D	00	...	
ASCII:			<NAK>	'#'	'6'	'6'	<CR>			
Meaning	Change counter	Serial interface port	Error Text				End character			

Forbidden ARE i9 instruction (Error code #67)

Some ARE i9 instructions are not supported in an Ethernet bus environment. Those commands are not transferred to the reader, but generate error #67.

Hex:	04	34	4D	44	20	30	0D	00	...	
ASCII:		'4'	'M'	'D'	<SP>	'0'	<CR>			
Meaning:	Change counter	Serial interface port	Instruction Text				End character			

ACM 9 sets master input data to:

Hex:	23	00	15	23	36	37	0D	00	...	
ASCII:			<NAK>	‘#’	‘6’	‘7’	<CR>			
Meaning	Change counter	Serial interface port		Error Text			End character			

2.2.5 Error messages coming through from ARE i9

Other error messages can be generated by the reader, if the reader command is wrong. See section ARE i9 for a detailed description of those error messages. Below are sample error codes, see section ARE i9 for full list.

Wrong reader instruction (Error code #00)
A wrong reader instruction triggers error code #00.

Hex:	04	32	41	42	43	0D	00	...		
ASCII:		‘2’	‘A’	‘B’	‘C’	<CR>				
Meaning:	Change counter	Serial interface port	Instruction Text			End character				

ACM 9 sends this master input data:

Hex:	22	32	15	23	30	30	0D	00	...	
ASCII:		‘2’	<NAK>	‘#’	‘0’	‘0’	<CR>			
Meaning	Change counter	Serial interface port		Error Text			End character			

Wrong reader instruction parameter (Error code #02)
A wrong reader instruction triggers error code #02.

Hex:	04	32	41	4C	47	4F	20	30	0D	
ASCII:		‘2’	‘A’	‘L’	‘G’	‘O’	<SP>	‘0’	<CR>	
Meaning:	Change counter	Serial interface port	Instruction Text						End character	

ACM 9 sends this master input data:

Hex:	22	32	15	23	30	32	0D	00	...	
ASCII:		‘2’	<NAK>	‘#’	‘0’	‘2’	<CR>			
Meaning	Change counter	Serial interface port		Error Text			End character			

3. ARE i9

ARE i9 is a small industrial RFID reader that is available as a LF - SEMI Industry, LF and HF version.
ARE i9 LF hdx - SEMI Industry works with LF hdx transponders.
ARE i9 LF works with (all) low frequency transponders ASK, PSK, FSK, @ 125 KHz, 128kHz and 134.2 kHz (ISO 11784/11785).
ARE i9 HF works with all ISO 15693 compliant transponders.

3.1 Hardware

3.1.1 Dimensions

Recommended Screws for mounting: Cylinder head screw M4 x 30mm 

3.1.2 Protection Class

Protection Class is IP 67, assuming cable or dummy cap is mounted.

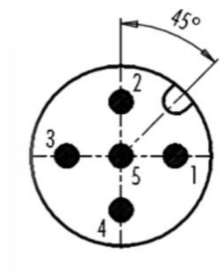
3.1.3 Connectivity

ARE i9 is connected via its M12, 5-Pin male A-coded plug. Power supply as well as communication is provided by ACM 9, communication module. Do not use any other means of powering or communicating to ARE i9 other than an ACM 9 module, otherwise ARE i9 will be damaged. Do only use specified cables. ARE i9 uses a LED lit RFID symbol on its front side to visually communicate its various states (standby, reading, successful read, no read, error, and so on…). When ARE i9 is hooked up to ACM 9, the internal LED is switched to standby color. LED colors can be set by the user.

PIN 1 – +7V
PIN 2 – GND
PIN 3 – RX
PIN 4 – TX
PIN 5 – CGND

LED: Status indication

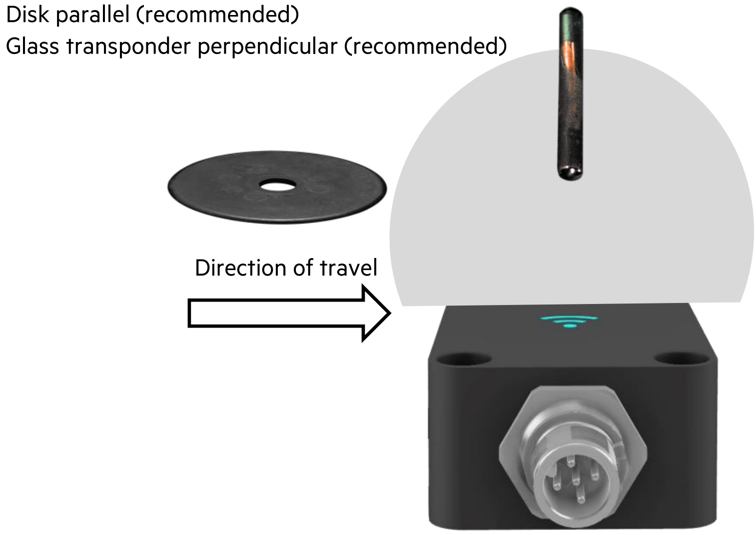
Cable: M12, 5-Pin A-coded, plug to socket



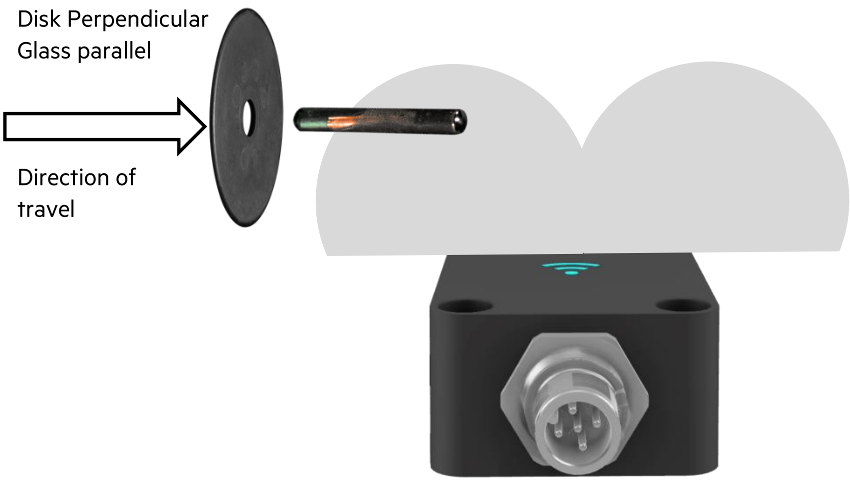
ARE i9 uses an air core coil. There is air core coil transponders like disks and ferrite core coil transponders like glass tube transponders. It is important to understand the impact of orientation of transponders relative to ARE i9. Optimum orientation is parallel to the front side of the reader for disks and perpendicular for glass tube transponders. In this orientation, the highest read range can be achieved.

If it is not possible to ensure such orientation, the Disk transponders can be oriented perpendicular to the front side of ARE i9 and glass tube transponders can be oriented parallel. This will result in some decrease of read range, but in most cases this is acceptable.

3.1.4 Transponder orientation relative to ARE i9



The highest read range is achieved right above the center of ARE i9 front side.



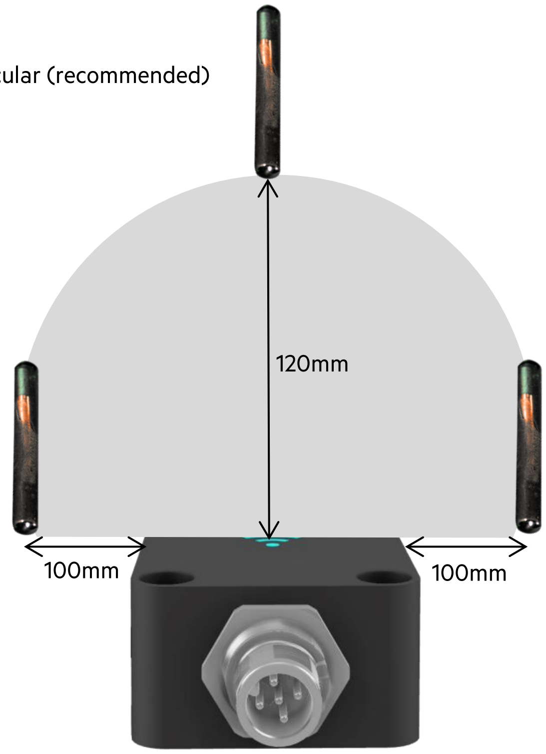
In this orientation both transponder types are read best just outside the perimeter of ARE i9. There is a significantly reduced read range in the center of ARE i9. This is no problem in a dynamic situation. Please ensure to start reading before the transponder is above ARE i9 and keep reading until the transponder is beyond ARE i9.

Reading distance depends a lot on the particular installation. Absolute values only make sense based on a particular transponder. Absolute values make no sense for transponder types, because the values will vary too much. Above is the guiding principles to achieve the best possible read range.

3.1.5 Read range for SEMI application

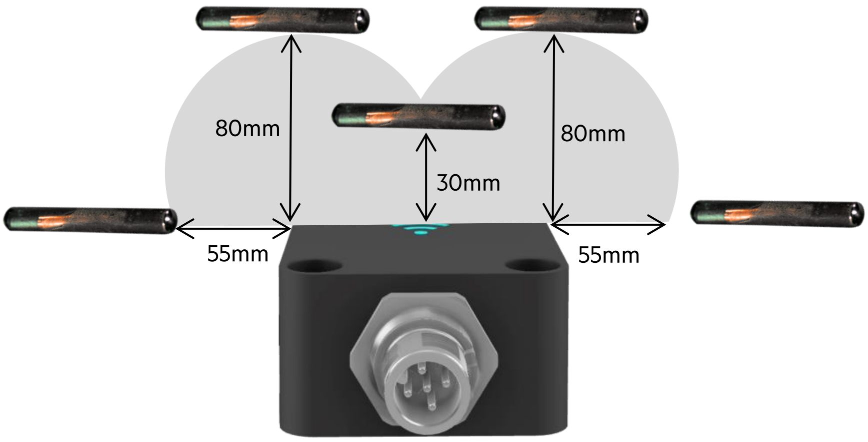
Glass transponder Texas Instruments RI-TRP-DR2B

Glass transponder perpendicular (recommended)



The highest read range is achieved right above the center of ARE i9 front side.

Glass parallel



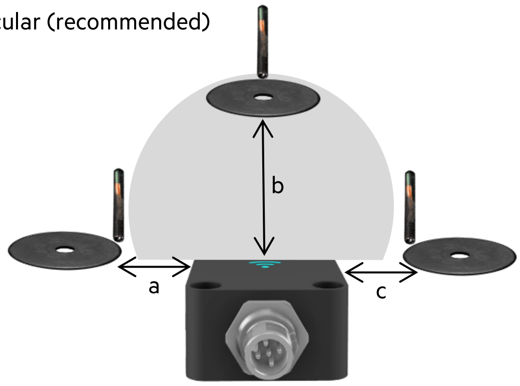
The highest read range is achieved right at the perimeter of the reader housing.

*note: only one transponder in the field at a time. Above illustration only shows possible read ranges.

3.1.6 Read range for miscellaneous applications

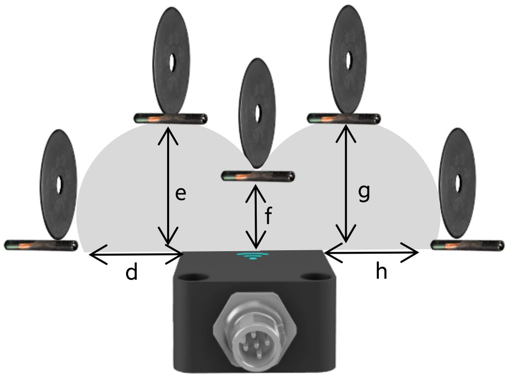
Glass transponder and Disk transponder with different transponder chips (see table at the bottom)

Disk transponder parallel (recommended)
Glass transponder perpendicular (recommended)



The highest read range is achieved right above the center of ARE i9 front side.

Disk perpendicular
Glass parallel



The highest read range is achieved right at the perimeter of the reader housing.
*note: only one transponder in the field at a time. Above illustration only shows possible read ranges.

Type	Description	a	b	c	d	e	f	g	h
ID 1002/11/P/13	Disk, 13mm, PSK1	10	45	10	10	25	1	25	10
ID 1002/11/P/20	Disk, 20mm, PSK1	25	65	25	25	35	10	35	25
ID 1002/11/P/30	Disk, 30mm, PSK1	40	80	40	40	40	10	40	40
ID 1002/11/P/50	Disk, 50mm, PSK1	60	115	60	60	50	10	50	60
ID 102/11/P/3	Glass, 3mm, PSK1	1	35	1	1	25	1	25	1
ID 1002/0/A/13	Disk, 13mm, ASK	10	45	10	10	25	1	25	10
ID 1002/0/A/20	Disk, 20mm, ASK	25	65	25	25	35	10	35	25
ID 1002/0/A/30	Disk, 30mm, ASK	40	80	40	40	40	10	40	40
ID 1002/0/A/50	Disk, 50mm, ASK	60	115	60	60	50	10	50	60
ID 102/0/A/3	Glass, 3mm, ASK	1	35	1	1	25	1	25	1

Read range is given in mm.
Read range is typical read range. Actual read range may vary due to individual circumstances.

3.2 Firmware ARE i9 LF hdx - SEMI Industry

ARE i9 LF hdx - SEMI Industry reads low frequency hdx transponders, typically in glass transponder format 4mmx34mm.

3.2.1 Instruction Set

Communication with ARE i9 LF SEMI Industry is based on a simple ASCII text based protocol. The host sends text based telegrams to ARE i9 LF SEMI Industry and receives text based telegrams back containing the answer to the query. Communication to ARE i9 LF SEMI Industry is always triggered by the host.

3.2.2 General format of instruction set

The protocol format is as follows

Instruction <SP> **parameter** <CR>

The space character <SP> separates commands from parameters and the <CR> character acts as command line terminator.

For commands without parameter values (e.g. GT) the <SP> character and parameter values are omitted. The command line is as short as this:

Instruction <CR>

3.2.3 VER

VER – Reader firmware version

VER is used to get the actual reader firmware version. .

Input format: VER <CR>

Hex:	56	45	52	0D
ASCII:	‘V’	‘E’	‘R’	<CR>

Output (example): ARE i9 V_1.011 <CR>

Hex:	21	00	15	...	...	31	0D
ASCII:	‘A’	‘R’	‘E’	...	...	‘1’	<CR>

3.2.4 GT

GT – Get Tag

GT is used to retrieve the transponder UID.

Input format: GT<CR>

Hex:	47	54	0D
ASCII:	'G'	'T'	<CR>

Output (example): 1234567812345678 <CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	'1'	'2'	'3'	...	...	'8'	<CR>

3.2.5 TOR

TOR – Timeout Reading

After a read is triggered by GT, TOR is a time during which ARE i9 continuously tries to read a transponder UID without the need to be triggered by the host again. This limits bus traffic considerably. Once a successful read is performed, continuous reading stops immediately regardless of time and the transponder UID is transmitted to the host. If reading is not successful, a no read (XXXXXXXXXXXXXXXX) is sent to the host after TOR time has expired.
The chosen parameter for TOR is sent as acknowledgement.

Input format: TOR<SP>50<CR>

Hex:	54	4F	52	20	35	30	0D
ASCII:	'T'	'O'	'R'	<SP>	'5'	'0'	<CR>

Output (example): 50 <CR>

Hex:	35	30	0D
ASCII:	'5'	'0'	<CR>

Parameter:

PARAMETER	FUNCTION
0	limits the reading process duration to exactly one reading cycle
1	limits the reading process duration to maximum 1 times 100ms
2	limits the reading process duration to maximum 2 times 100ms
...	
255	limits the reading process duration to maximum 255 times 100ms

A TOR value of 50 equals 50 x 100ms = 5000ms = 5 sec.
It is recommended to set TOR value to the amount of time it takes in a dynamic situation for the transponder to travel over ARE i9. This maximizes the number of possible reads, in order to compensate for EMV noise in the vicinity.

3.2.6 NID

NID – Double reading of UID to ensure consistency in EMV polluted environment.

NID is used to double read a transponder UID to ensure consistency in an EMV polluted environment. The transponder UID is transmitted only after two consecutive reads of the same UID

Parameters: 0 – every UID is transmitted | 1 – UID only transmitted if read twice consecutively

Input format: NID<SP>1<CR>

Hex:	4E	49	44	20	31
ASCII:	'N'	'I'	'D'	<SP>	'1'

Output (example): 1<CR>

Hex:	31	0D
ASCII:	'1'	<CR>

3.2.7 CID

CID – Filter same UID numbers to transmit only once via interface

CID is used to filter multiple read transponder UID to transmit only once via interface. There needs to be one different Transponder UID read before the same number will be transmitted again.

Parameters: 0 – no filter function | 1 – filter same chip UID as previously read

Input format: CID<SP>1<CR>

Hex:	43	49	44	20	31
ASCII:	'C'	'I'	'D'	<SP>	'1'

Output (example): 0x1<CR>

Hex:	31	0D
ASCII:	'1'	<CR>

3.2.8 CN

CN – Filter no read from being transmitted via interface.

CN is used in those cases, where no read information 'XXXXXXXXXXXXXXXX' is not to appear on the interface. Only valid transponder UID will be transmitted.

Parameters: 0 – no filter function | 1 – filter no read information from being transmitted

Input format: CID<SP>1<CR>

Hex:	43	4E	20	31
ASCII:	'C'	'N'	<SP>	'1'

Output (example): 0x1<CR>

Hex:	31	0D
ASCII:	'1'	<CR>

3.2.9 RD

RD – Read transponder memory page

RD is used to read an individual memory page from a transponder in the field.

Input format: RD<SP> 1<CR>

Hex:	52	44	20	31	0D
ASCII:	'R'	'D'	<SP>	'1'	<CR>

Output (example): 1234567812345678<CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	'1'	'2'	'3'	...	...	'8'	<CR>

3.2.10 WD

WD – Write transponder memory page

WD is used to write to individual memory page from a transponder in the field.

Input format: WD<SP> 5<SP> 1234567812345678<CR>

Hex:	57	44	20	35	20	31	...	38	0D
ASCII:	'W'	'D'	<SP>	'5'	<SP>	'1'	...	'8'	<CR>

Output (example): 1234567812345678<CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	'1'	'2'	'3'	...	...	'8'	<CR>

3.2.11 LD

LD – lock memory page

LD is used to lock a particular memory page from a transponder in the field.

Input format: LD<SP> 1<CR>

Hex:	4D	44	20	31	0D
ASCII:	'L'	'D'	<SP>	'1'	<CR>

Output (example): 1234567812345678<CR> (content of locked memory page)

Hex:	31	32	33	...	...	38	0D
ASCII:	'1'	'2'	'3'	...	...	'8'	<CR>

If there is an error during locking, the answer will be XXXXXXXXXXXXXXXX<CR>

3.2.12 VSAVE

VSAVE – Save parameter permanently in ARE i9 flash memory

VSAVE is used to save parameters permanently in flash memory of ARE i9 to be available after power on.

Input format: VSAVE <CR>

Hex:	56	53	41	56	45	0D
ASCII:	'V'	'S'	'A'	'V'	'E'	<CR>

Output (example): OK<CR>

Hex:	4F	4B	0D
ASCII:	'O'	'K'	<CR>

3.2.13 INIT

INIT – Restore standard parameters. Command needs to be followed up by VSAVE in order to permanently store the parameters.

Input format: INIT<CR>

Hex:	49	4E	49	54	0D
ASCII:	'I'	'N'	'I'	'T'	<CR>

Output (example): OK<CR>

Hex:	4F	4B	0D
ASCII:	'O'	'K'	<CR>

The following parameters are set:

TOR 50	LRD 01001
MD 2	LNRD 10001
CID 0	LERR 10011
CN 0	LED 1
LSTB 01101	LRT 30
LGT 01111	LPA 00000

3.2.14 Error messages

Error messages and protocol errors are acknowledged by ARE i9 using an error code. The format is described below:

<NAK> '#' <error code> <CR>

Example error #02 (wrong parameter)

Hex:	15	23	30	32	0D
ASCII:	<NAK>	'#'	'0'	'2'	<CR>

The error code is comprised of a two digit ASCII coded number. Please note that for communication through ACM 9, the appropriate reader number is preceding the error message.

The following table displays possible error messages:

Error code	Meaning
"00"	Unknown instruction
"02"	Wrong parameter

3.3 Firmware ARE i9 LF

ARE i9 LF works with (all) low frequency transponders in ASK, PSK and FSK modulation. Please see chapter 3.3.13 for details on which transponder chips are implemented. Depending on the selected algorithms, not all instructions below make sense so only those which do work accordingly (e.g. write command ‘wd’ does not work for a read only transponder).

3.3.1 Instruction Set

Communication with ARE i9 is based on a simple ASCII text based protocol. The host sends text based telegrams to ARE i9 and receives text based telegrams back containing the answer to the query. Communication to ARE i9 is always triggered by the host.

3.3.2 General format of instruction set

The protocol format is as follows

Instruction <SP> **parameter** <CR>

The space character <SP> separates commands from parameters and the <CR> character acts as command line terminator.

For commands without parameter values (e.g. GT) the <SP> character and parameter values are omitted. The command line is as short as this:

Instruction <CR>

3.3.3 VER

VER – Reader firmware version

VER is used to get the actual reader firmware version. .

Input format: VER <CR>

Hex:	56	45	52	0D
ASCII:	‘V’	‘E’	‘R’	<CR>

Output (example): ARE i9 V_1.011 <CR>

Hex:	21	00	15	...	...	31	0D
ASCII:	‘A’	‘R’	‘E’	...	...	‘1’	<CR>

3.3.4 GT

GT – Get Tag

GT is used to retrieve the transponder UID.

Input format: GT<CR>

Hex:	47	54	0D
ASCII:	‘G’	‘T’	<CR>

Output (example): 12345678 <CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	‘1’	‘2’	‘3’	...	...	‘8’	<CR>

3.3.5 TOR

TOR – Timeout Reading

After a read is triggered by GT, TOR is a time during which ARE i9 continuously tries to read a transponder UID without the need to be triggered by the host again. This limits bus traffic considerably. Once a successful read is performed, continuous reading stops immediately regardless of time and the transponder UID is transmitted to the host. If reading is not successful, a no read (XXXXXXXX) is sent to the host after TOR time has expired.
The chosen parameter for TOR is sent as acknowledgement.

Input format: TOR<SP>50<CR>

Hex:	54	4F	52	20	35	30	0D
ASCII:	‘T’	‘O’	‘R’	<SP>	‘5’	‘0’	<CR>

Output (example): 50<CR>

Hex:	35	30	0D
ASCII:	‘5’	‘0’	<CR>

Parameter:

PARAMETER	FUNCTION
0	limits the reading process duration to exactly one reading cycle
1	limits the reading process duration to maximum 1 times 100ms
2	limits the reading process duration to maximum 2 times 100ms
...	
255	limits the reading process duration to maximum 255 times 100ms

A TOR value of 50 equals 50 x 100ms = 5000ms = 5 sec.
It is recommended to set TOR value to the amount of time it takes in a dynamic situation for the transponder to travel over ARE i9. This maximizes the number of possible reads, in order to compensate for EMV noise in the vicinity.

3.3.6 NID

NID – Double reading of UID to ensure consistency in EMV polluted environment.

NID is used to double read a transponder UID to ensure consistency in an EMV polluted environment. The transponder UID is transmitted only after two consecutive reads of the same UID

Parameters: 0 – every UID is transmitted | 1 – UID only transmitted if read twice consecutively

Input format: NID<SP>1<CR>

Hex:	4E	49	44	20	31
ASCII:	'N'	'I'	'D'	<SP>	'1'

Output (example): 1<CR>

Hex:	31	0D
ASCII:	'1'	<CR>

3.3.7 CID

CID – Filter same UID numbers to transmit only once via interface

CID is used to filter multiple read transponder UID to transmit only once via interface. There needs to be one different Transponder UID read before the same number will be transmitted again.

Parameters: 0 – no filter function | 1 – filter same chip UID as previously read

Input format: CID<SP>1<CR>

Hex:	43	49	44	20	31
ASCII:	'C'	'I'	'D'	<SP>	'1'

Output (example): 1<CR>

Hex:	31	0D
ASCII:	'1'	<CR>

3.3.8 CN

CN – Filter no read from being transmitted via interface.

CN is used in those cases, where no read information ‘XXXXXXXX’ is not to appear on the interface. Only valid transponder UID will be transmitted.

Parameters: 0 – no filter function | 1 – filter no read information from being transmitted

Input format: CID<SP>1<CR>

Hex:	43	4E	20	31
ASCII:	‘C’	‘N’	<SP>	‘1’

Output (example): 1<CR>

Hex:	31	0D
ASCII:	‘1’	<CR>

3.3.9 RD

RD – Read transponder memory page

RD is used to read an individual memory page from a transponder in the field.

Input format: RD<SP> 1<CR>

Hex:	52	44	20	31	0D
ASCII:	‘R’	‘D’	<SP>	’1’	<CR>

Output (example): 12345678<CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	‘1’	‘2’	‘3’	...	...	‘8’	<CR>

3.3.10 WD

WD – Write transponder memory page

WD is used to write to individual memory page from a transponder in the field.

Input format: WD<SP> 5<SP> 12345678<CR>

Hex:	57	44	20	35	20	31	...	38	0D
ASCII:	‘W’	‘D’	<SP>	‘5’	<SP>	‘1’	...	’8’	<CR>

Output (example): 12345678<CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	‘1’	‘2’	‘3’	...	...	‘8’	<CR>

3.3.11 VSAVE

VSAVE – Save parameter permanently in ARE i9 flash memory

VSAVE is used to save parameters permanently in flash memory of ARE i9 to be available after power on.

Input format: VSAVE <CR>

Hex:	56	53	41	56	45	0D
ASCII:	‘V’	‘S’	‘A’	’V’	‘E’	<CR>

Output (example): OK<CR>

Hex:	4F	4B	0D
ASCII:	‘O’	‘K’	<CR>

3.3.12 INIT

INIT – Restore standard parameters. Command needs to be followed up by VSAVE in order to permanently store the parameters.

Input format: INIT<CR>

Hex:	49	4E	49	54	0D
ASCII:	‘I’	‘N’	‘I’	’T’	<CR>

Output (example): OK<CR>

Hex:	4F	4B	0D
ASCII:	‘O’	‘K’	<CR>

The following parameters are set:

TOR 50	LRD 01001
MD 2	LNRD 10001
CID 0	LERR 10011
CN 0	LED 1
LSTB 01101	LRT 30
LGT 01111	LPA 00000

3.3.13 Error messages

Error messages and protocol errors are acknowledged by ARE i9 using an error code. The format is described below:

<NAK> '#' <error code> <CR>

Example error #02 (wrong parameter)

Hex:	15	23	30	32	0D
ASCII:	<NAK>	'#'	'0'	'2'	<CR>

The error code is comprised of a two digit ASCII coded number. Please note that for communication through ACM 9, the appropriate reader number is preceding the error message.

The following table displays possible error messages:

Error code	Meaning
"00"	Unknown instruction
"02"	Wrong parameter

3.3.14 ALGO

ALGO is used to activate one particular LF algorithm to be used in a particular installation.

Input format: ALGO<SP>ALGO#<CR>
Example: ALGO<SP>1<CR>

Hex:	41	4C	47	4F	20	31	0D
ASCII:	'A'	'L'	'G'	'O'	<SP>	'1'	<CR>

Output (example): 1<CR>

Hex:	31	0D
ASCII:	'1'	<CR>

Above example activates algorithms 1

Implemented LF algorithms:

1 - PSK1, Trovan
4 - ASK 64 Bit Manchester
5 - ISO 11784/85
6 - Hitag1/HitagS
8 - Hitag2
14 - EM4305
23 - HDX

3.3.15 LOG (EM4305 chip specific)

EM 4305

EM 4305 is a multi purpose chip from EM microelectronic Marin.
It features 512 bit memory and can be configured to transmit in ASK 64-bit Manchester, PSK1, Trovan, ISO 11784/85 fdx-b, pigeon mode among others or work as a simple memory chip.
In addition to above commands the chip uses the following chip specific commands.

LOG is used to log into a password protected chip. (see chip data sheet for details). Standard password is 0x00000000.

Input format: LOG<SP>password<CR>
Example: LOG<SP>00000000<CR>

Hex:	4C	4F	47	20	30	30	...	30	0D
ASCII:	‘L’	‘O’	‘G’	<SP>	‘0’	‘0’	...	‘0’	<CR>

Output (example): ACK<CR>

Hex:	41	43	4B	0D
ASCII:	‘A’	‘C’	‘K’	<CR>

This answer is sent if everything went ok.
Output (example): NAK<CR>

Hex:	4E	41	4B	0D
ASCII:	‘N’	‘A’	‘K’	<CR>

This answer is sent if login failed.

3.3.16 PWD (EM4305 chip specific)

PWD is used to change the password for the chip. Please make sure to log into the transponder first using the LOG command and the current password. Only then can the password be changed. Standard password is 0x00000000.

Input format: PWD<SP>password<CR>
Example: PWD<SP>01234567<CR>

Hex:	50	57	44	20	30	31	...	37	0D
ASCII:	‘P’	‘W’	‘D’	<SP>	‘0’	‘1’	...	‘7’	<CR>

Output (example): ACK<CR>

Hex:	41	43	4B	0D
ASCII:	‘A’	‘C’	‘K’	<CR>

This answer is sent if everything went ok.
Output (example): NAK<CR>

Hex:	4E	41	4B	0D
ASCII:	‘N’	‘A’	‘K’	<CR>

This answer is sent if password change failed.

3.3.17 LD (EM4305 chip specific)

After the chip is configured correctly, it may be necessary to lock specific memory blocks of EM 4305. Memory blocks from 0 to 13 can be locked. Memory pages 14 and 15 serve as lock data (see chip data sheet for details).

page 14 and 15 - protection word																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	bit#
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	bit content binary
0				0				0				0				8				0				7				2				bit content hexadecimal
page 14 and 15 protection word																																memory page chip
not used - always 0																protection word status bit	protection word protection bit	potection word 13	potection word 12	potection word 11	potection word 10	potection word 9	potection word 8	potection word 7	potection word 6	potection word 5	potection word 4	potection word 3	potection word 2	potection word 1	potection word 0	bit function

Protection word 1 is factory set, as it contains the chip UID. Memory pages 0, 2-13 can be locked by the user. This is OTP, so once locked, it can not be undone.

Input format: LD<SP>00008072<CR>

Hex:	4C	44	20	30	...	30	37	32	0D
ASCII:	'L'	'D'	<SP>	'0'	...	'0'	'7'	'2'	<CR>

Output (example): ACK<CR>

Hex:	41	43	4B	0D
ASCII:	'A'	'C'	'K'	<CR>

This answer is sent if lock data went ok.

Output (example): NAK<CR>

Hex:	4E	41	4B	0D
ASCII:	'N'	'A'	'K'	<CR>

This answer is sent if something went wrong during lock data.

Above example locks memory pages 4, 5 and 6. Memory page 1 is factory set, as is the protection status bit.

3.4 Firmware ARE i9 HF

3.4.1 Instruction Set

Communication with ARE i9 is based on a simple ASCII text based protocol. The host sends text based telegrams to ARE i9 and receives text based telegrams back containing the answer to the query. Communication to ARE i9 is always triggered by the host.

3.4.2 General format of instruction set

The protocol format is as follows

Instruction <SP> **parameter** <CR>

The space character <SP> separates commands from parameters and the <CR> character acts as command line terminator.

For commands without parameter values (e.g. GT) the <SP> character and parameter values are omitted. The command line is as short as this:

Instruction <CR>

3.4.3 VER

VER – Reader firmware version

VER is used to get the actual reader firmware version. .

Input format: VER <CR>

Hex:	56	45	52	0D
ASCII:	'V'	'E'	'R'	<CR>

Output (example): ARE i9 V_1.011 <CR>

Hex:	21	00	15	...	...	31	0D
ASCII:	'A'	'R'	'E'	...	...	'1'	<CR>

3.4.4 GT

GT – Get Tag

GT is used to retrieve the transponder UID.

Input format: GT<CR>

Hex:	47	54	0D
ASCII:	'G'	'T'	<CR>

Output (example): 12345678 <CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	'1'	'2'	'3'	...	...	'8'	<CR>

3.4.5 TOR

TOR – Timeout Reading

After a read is triggered by GT, TOR is a time during which ARE i9 continuously tries to read a transponder UID without the need to be triggered by the host again. This limits bus traffic considerably. Once a successful read is performed, continuous reading stops immediately regardless of time and the transponder UID is transmitted to the host. If reading is not successful, a no read (XXXXXXXX) is sent to the host after TOR time has expired.
The chosen parameter for TOR is sent as acknowledgement.

Input format: TOR<SP>50<CR>

Hex:	54	4F	52	20	35	30	0D
ASCII:	'T'	'O'	'R'	<SP>	'5'	'0'	<CR>

Output (example): 50 <CR>

Hex:	35	30	0D
ASCII:	'5'	'0'	<CR>

Parameter:

PARAMETER	FUNCTION
0	limits the reading process duration to exactly one reading cycle
1	limits the reading process duration to maximum 1 times 100ms
2	limits the reading process duration to maximum 2 times 100ms
...	
255	limits the reading process duration to maximum 255 times 100ms

A TOR value of 50 equals 50 x 100ms = 5000ms = 5 sec.
It is recommended to set TOR value to the amount of time it takes in a dynamic situation for the transponder to travel over ARE i9. This maximizes the number of possible reads, in order to compensate for EMV noise in the vicinity.

3.4.6 CID

CID – Filter same UID numbers to transmit only once via interface

CID is used to filter multiple read transponder UID to transmit only once via interface. There needs to be one different Transponder UID read before the same number will be transmitted again.

Parameters: 0 – no filter function | 1 – filter same chip UID as previously read

Input format: CID<SP>1<CR>

Hex:	43	49	44	20	31
ASCII:	'C'	'I'	'D'	<SP>	'1'

Output (example): 0x1<CR>

Hex:	31	0D
ASCII:	'1'	<CR>

3.4.7 CN

CN – Filter no read from being transmitted via interface.

CN is used in those cases, where no read information 'XXXXXXXX' is not to appear on the interface. Only valid transponder UID will be transmitted.

Parameters: 0 – no filter function | 1 – filter no read information from being transmitted

Input format: CID<SP>1<CR>

Hex:	43	4E	20	31
ASCII:	'C'	'N'	<SP>	'1'

Output (example): 0x1<CR>

Hex:	31	0D
ASCII:	'1'	<CR>

3.4.8 RD

RD – Read transponder memory page

RD is used to read an individual memory page from a transponder in the field.

Input format: RD<SP> 1<CR>

Hex:	52	44	20	31	0D
ASCII:	'R'	'D'	<SP>	'1'	<CR>

Output (example): 12345678<CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	'1'	'2'	'3'	...	...	'8'	<CR>

3.4.9 WD

WD – Write transponder memory page

WD is used to write to individual memory page from a transponder in the field.

Input format: WD<SP> 5<SP> 12345678<CR>

Hex:	57	44	20	35	20	31	...	38	0D
ASCII:	'W'	'D'	<SP>	'5'	<SP>	'1'	...	'8'	<CR>

Output (example): 12345678<CR>

Hex:	31	32	33	...	...	38	0D
ASCII:	'1'	'2'	'3'	...	...	'8'	<CR>

3.4.10 VSAVE

VSAVE – Save parameter permanently in ARE i9 flash memory

VSAVE is used to save parameters permanently in flash memory of ARE i9 to be available after power on.

Input format: VSAVE <CR>

Hex:	56	53	41	56	45	0D
ASCII:	'V'	'S'	'A'	'V'	'E'	<CR>

Output (example):OK<CR>

Hex:	4F	4B	0D
ASCII:	'O'	'K'	<CR>

3.4.11 INIT

INIT – Restore standard parameters. Command needs to be followed up by VSAVE in order to permanently store the parameters.

Input format: INIT<CR>

Hex:	49	4E	49	54	0D
ASCII:	'I'	'N'	'I'	'T'	<CR>

Output (example): OK<CR>

Hex:	4F	4B	0D
ASCII:	'O'	'K'	<CR>

The following parameters are set:

TOR 50	LRD 01001
MD 2	LNRD 10001
CID 0	LERR 10011
CN 0	LED 1
LSTB 01101	LRT 30
LGT 01111	LPA 00000

3.4.12 Error messages

Error messages and protocol errors are acknowledged by ARE i9 using an error code. The format is described below:

<NAK> '#' <error code> <CR>

Example error #02 (wrong parameter)

Hex:	15	23	30	32	0D
ASCII:	<NAK>	'#'	'0'	'2'	<CR>

The error code is comprised of a two digit ASCII coded number. Please note that for communication through ACM 9, the appropriate reader number is preceding the error message.

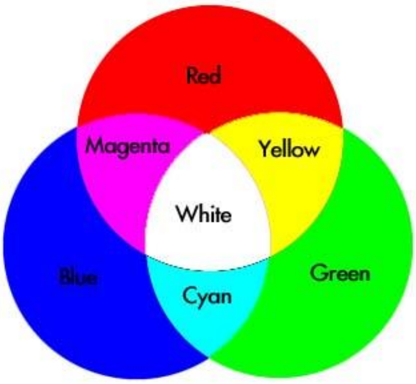
The following table displays possible error messages:

Error code	Meaning
"00"	Unknown instruction
"02"	Wrong parameter

3.5 LED instruction set

ARE i9 employs a multi-color LED to signal different modes.

Basically below colors can be created:



The user can choose any color apart from white. This color is reserved for setup help functionality as described below.

The following modes use a distinct color each.

- Standby (LSTB)
- Reading (LGT)
- Transponder number successfully read (LRD)
- No Read (LNRD)
- Error (LERR)
- Process active (LPA)
- Process status (LPS

In addition, the user can choose to switch on the LED permanently or flashing.

The following instruction set is used:

Mode<SPACE>RGBFX<CR>

R – Red

G – Green

B – Blue

F - Flash

X – LED functionality ON or OFF for this mode

Allowed parameters are 1 (on) or 0 (off)

Default colors are shown with the instructions.

3.5.1 LED Standby (LSTB)

Standby color is Cyan, no flash.

Input format: LSTB<SP> 01101<CR>

Hex:	4C	53	54	42	20	30	...	31	0D
ASCII:	‘L’	‘S’	‘T’	‘B’	<SP>	‘0’	...	’1’	<CR>

Output: 01101<CR>

Hex:	30	31	31	30	31	0D
ASCII:	‘0’	‘1’	‘1’	‘0’	‘1’	<CR>

Standby mode is active if no other instructions are carried out.

If Standby LED is switched off, the LED will be active for 10 seconds after reboot in its last color scheme and then it will be switched off.

3.5.2 LED Reading (LGT)

Reading color is Cyan, flashing

Input format: LGT<SP> 01111<CR>

Hex:	4C	47	54	20	30	31	...	31	0D
ASCII:	‘L’	‘G’	‘T’	<SP>	‘0’	‘1’	...	’1’	<CR>

Output: 01111<CR>

Hex:	30	31	31	31	31	0D
ASCII:	‘0’	‘1’	‘1’	‘1’	‘1’	<CR>

Reading mode is active for the duration of the TOR parameter. It will stop prematurely only to show a successful read using the respective color. At the end of the TOR parameter it will show the no read mode LED color.

3.5.3 LED Transponder number successfully read (LRD)

Successful read color is green, no flash

Input format: LRD<SP> 01001<CR>

Hex:	4C	52	44	20	30	31	...	31	0D
ASCII:	‘L’	‘R’	‘D’	<SP>	‘0’	‘1’	...	’1’	<CR>

Output: 01001<CR>

Hex:	30	31	30	30	31	0D
ASCII:	‘0’	‘1’	‘0’	‘0’	‘1’	<CR>

Successful read mode is active for LRT seconds, after which the standby mode will be active again.

3.5.4 LED No Read (LNRD)

No Read color is red no flash

Input format: LNRD<SP> 10001<CR>

Hex:	4C	4E	52	44	20	31	...	31	0D
ASCII:	‘L’	‘N’	‘R’	‘D’	<SP>	‘1’	...	’1’	<CR>

Output: 10001<CR>

Hex:	31	30	30	30	31	0D
ASCII:	‘1’	‘0’	‘0’	‘0’	‘1’	<CR>

No Read mode is active after TOR seconds for LRT seconds, after which the standby mode will be active again.

3.5.5 LED Return to standby (LRT)

Some modes require ARE i9 to go back to standby. The time until this happens is set by using the LRT command.

Input format: LRT<SP>time<CR>

Hex:	4C	52	54	20	33	30	0D		
ASCII:	‘L’	‘R’	‘T’	<SP>	‘3’	‘0’	<CR>		

Output: 30<CR>

Hex:	33	30	0D
ASCII:	‘3’	‘0’	<CR>

LRT<SP>30<CR> sets approx. 3 seconds as time for return to standby (30x100ms)

3.5.6 LED Error (LERR)

Error color is red, flashing

Input format: LERR<SP> 10011<CR>

Hex:	4C	45	52	52	20	31	...	31	0D
ASCII:	'L'	'E'	'R'	'R'	<SP>	'1'	...	'1'	<CR>

Output: 10011<CR>

Hex:	31	30	30	31	31	0D
ASCII:	'1'	'0'	'0'	'1'	'1'	<CR>

Error mode is triggered by an error of ARE i9 and is active until a correct instruction is received.

3.5.7 LED Process active

In case of multiple commands being sent to the chip (e.g. rd and wd instructions), it may be necessary to control LED functionality manually. The LED Process active instruction sets the LED to a defined color and mode. This color and mode stays on as long as the LED Process active parameter is switched on. Normal LED functionality is discontinued during the activity of this parameter. LED functionality returns to normal only when LED Process active is switched off via its X parameter.

Activating Process active

LED color is yellow, flashing

Input format: LPA<SP> 11011<CR>

Hex:	4C	50	41	20	31	...	...	31	0D
ASCII:	'L'	'P'	'A'	<SP>	'1'	...	...	'1'	<CR>

Output: 11011<CR>

Hex:	31	31	30	31	31	0D
ASCII:	'1'	'1'	'0'	'1'	'1'	<CR>

Deactivating Process active

LED color doesn't care, because parameter is switched off using X parameter

Input format: LPA<SP> 11010<CR>

Hex:	4C	50	41	20	31	...	...	30	0D
ASCII:	'L'	'P'	'A'	<SP>	'1'	...	...	'0'	<CR>

Output: 11010<CR>

Hex:	31	31	30	31	30	0D
ASCII:	'1'	'1'	'0'	'1'	'0'	<CR>

3.5.8 LED Process status

LED Process status is used to indicate the status of a process, after it is performed.

Successful Process

LED color is green, not flashing

Input format: LPS<SP> 01001<CR>

Hex:	4C	53	54	20	30	31	...	31	0D
ASCII:	'L'	'P'	'S'	<SP>	'0'	'1'	...	'1'	<CR>

Output: 01001<CR>

Hex:	30	31	30	30	31	0D
ASCII:	'0'	'1'	'0'	'0'	'1'	<CR>

Not Successful Process

LED color is red, not flashing

Input format: LPS<SP> 10001<CR>

Hex:	4C	53	54	20	31	30	...	31	0D
ASCII:	'L'	'P'	'S'	<SP>	'1'	'0'	...	'1'	<CR>

Output: 10001<CR>

Hex:	31	30	30	30	31	0D
ASCII:	'1'	'0'	'0'	'0'	'1'	<CR>

LPS stays on for LRT seconds and then returns to standby.

3.5.9 LED Setup help (FLED)

In order to locate the respective ARE i9 hooked up to a particular port of ACM 9, the instruction FLED is used.

This instruction flashes the LED in white for 10 seconds. The color can not be changed.

Input format: FLED<CR>

Hex:	4C	53	54	42	0D
ASCII:	'F'	'L'	'E'	'D'	<CR>

Output: FLED<CR>

Hex:	4C	53	54	42	0D
ASCII:	'F'	'L'	'E'	'D'	<CR>

After flashing for 10 seconds ARE i9 returns to standby mode.

3.5.10 LED (De)activate LED functionality (LED)

In order to deactivate (or activate) the LED functionality, LED instruction is used.

Input format: LED<SP>Parameter<CR>

Hex:	53	54	42	20	30	OD			
ASCII:	‘L’	‘E’	‘D’	<SP>	‘0’	<CR>			

Output: 0<CR>

Hex:	30	0D
ASCII:	‘0’	<CR>

LED<SP> 0<CR> deactivates LED functionality.

LED<SP> 1<CR> activates LED functionality (default).

Above examples represent ARE i9 default values.

4. System implementation

4.1 Power supply

SEMI industry uses hdx LF RFID technology. This particular method relies on field gaps, where the RFID field is switched off. In this gap the transponder answers with its code. This method has the advantage of high read range in laboratory conditions. However, in a EMV polluted environment, read range of hdx transponders is significantly reduced as even low noise signals have a direct impact on read range.
Therefore it is absolutely mandatory for system integration to make sure that power supply for ACM 9 and therefore ARE i9 is absolutely stable and clean with no noise. It is recommended to use linear power supplies rather than switching power supplies. All other applications benefit from this as well.

4.2 Grounding

Please make absolutely sure that ACM 9 communication module is properly grounded. This ensures proper functionality of the entire system comprising of ACM 9 and ARE i9. Please see chapter 2.1.2 for details on grounding of ACM 9.
Grounding can be achieved by grounding DIN hat rail, as clamp on backside of ACM 9 is connected to ground. Alternatively the grounding pin on the backside of ACM 9 can be used to achieve this.

4.3 Mounting on metal

ACM 9 is typically mounted on a metal DIN hat rail in a metal electrical cabinet. There is no influence of metal on performance of ACM 9 and therefore nothing to watch out for.

It is recommended to mount ARE i9 onto a non-conductive surface. However, ARE i9 is designed to work when mounted on metal as well. There is a slight decrease in read/write range when compared to mounting on non-conductive surfaces, but in most cases the read/write range will still be plenty for the application.

4.4 Frequency converters

Frequency converters used in electronic motors are a source of significant EMV noise. Make sure to stay away as far as possible from those frequency converters when designing spots where ARE i9 is to be used. Noise from frequency converters significantly reduce read range of ARE i9.

5. Release, Change Protocol

Revision:	Date:	Changes:	Author:
01	21.09.2020	Release first edition	NK
02	29.09.2020	Details added	NK
03	12.10.2020	Details added	NK
04	22.11.2020	Details added	NK

AEG Identifikationssysteme GmbH
Hörvelsinger Weg 47
89081 Ulm
Tel.: +49 731 14 00 88 – 0
Email: sales@aegid.de
Web: www.aegid.de