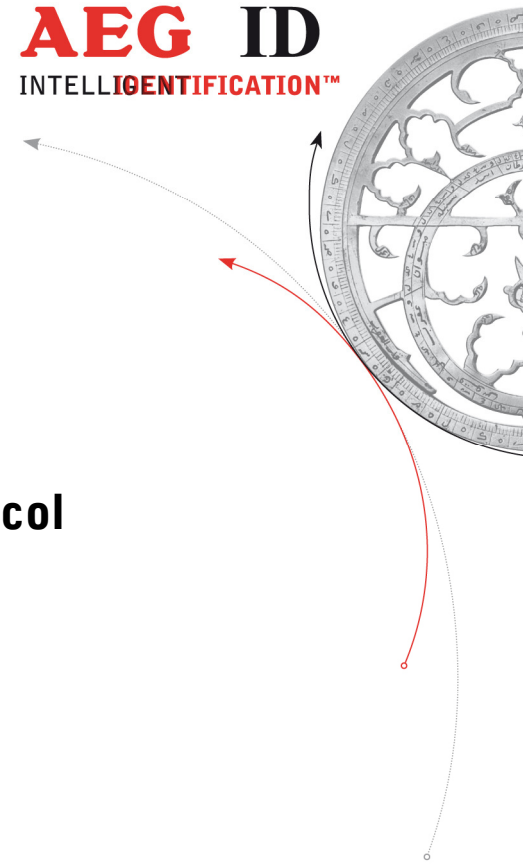
AEG ID
INTELLIGENT IDENTIFICATION™



Data Transmission Protocol

for the ARE H5

Geschäftsführer | Managing Director
Reiner Wagner

Sitz der Gesellschaft | Headquarter
Ulm, Germany

Amtsgericht | Local Court
Ulm, Germany HRB 3423
USt-IdNr. DE 183095060
Steuer-Nr. | **Tax No.** 88001/11616

Bankverbindung | Banking account
Sparkasse Ulm
BLZ: 630 500 00 | Kto.-Nr.:21072952
SWIFT Code: SOLADES1ULM
IBAN-Nr.: DE31 6305 0000 0021 0729 52

AEG Identifikationssysteme GmbH
Hörvelsinger Weg 47, D-89081 Ulm
P: +49 (0)731 140088-0
F: +49 (0)731 140088-9000
E-mail: info@aegid.de, **www.aegid.de**

1 Introduction

This document describes the protocol which is used for data exchange to and from the hand held reader ARE H5. It is applicable from firmware Version 607[1] onward.
Important note: Communication with the ARE H 5 is possible only, when it is in the operating mode "Database / PC".

2 Set of commands

Following commands are available:

ET:	Check, if database memory is empty
EC:	Clear database memory
RP:	Set dataset pointer to first stored dataset
RN:	Read stored dataset and increase dataset pointer
RL:	Read stored dataset again without increasing the pointer = repeat the previous reading (RN)
WP:	Set dataset pointer to the next free position in the memory
W:	Write dataset into database to the actual dataset pointer position and increase pointer
SV:	Read software version
XT:	Stop communication, abort operation mode "Database / PC"
R:	Set date and time
T:	Associate text (up to 14 characters) with attribute 'A' to 'Z' (display 'Stable' in stead of 'A' e.g.)

A detailed description of these commands is given in chapter 7 and 8.

[1] valid for equipment shipped after 15. Nov 2000.

3 Interface parameters

The data is transferred in the following format 19200baud, 8 data bits, no parity check, 1 stop bit, no hardware handshake (RTS, CTS), no control of data flow (e.g. Xon/Xoff)

4 Protocol, Control character

All the commands in operation mode "Database / PC" have the following structure:

STX, "command in ASCII", "CRC in ASCII", ETX

The following answers are possible:

A record with the structure given above
ACK
BEL
NAK

Except STX and ETX, the total content of the commands are composed in ASCII characters.
This procedure has been chosen explicitly, as a large number of host Operating Systems react with their own functions, when they see characters outside of 0x20 to 0x7F.

The CRC is applied only on the characters of the "command", STX and ETX are not included.
The 4 nibbles of the CRC are transmitted in form of 4 ASCII characters. By doing so, the CRC is in conformance with the ASCII standard as well.

Control characters	Hex code	Designation	Function
STX	0x02	Start Of Text	Begin of a telegram
ETX	0x03	End Of Text	End of a telergram
BEL	0x07	Buzzer (bell)	Signaling a specific condition, e.g. if the interrogated memory site is not occupied
CR	0x0D	Carriage return	End of line
ACK	0x06	Acknowledge	The command has been executed successfully
NAK	0x15	Negative Acknowledge	The command was not recognized and consequently has not been executed (syntax error)

5 Checksum CRC

The checksum (CRC=cyclic redundancy check) is generated using the ISO (or CCITT) standardized polynomial:
0x1021; $P(X) = X^{16} + X^{12} + X^5 + 1$.

CRC-CCIT Polynomial	0x1021
CRC order	16 Bit
Start value CRC	0x0000
Data stream	Every data byte is mirrored (from LSB to MSB)
CRC	Mirror CRC result before final XOR

Example in ANSI C:
The CRC checksum is realized as inverse CRC-CCITT in this program example:

```
// ******************************************************************
// Function to calculate the CRC from a protocol buffer with
// the given length
// ******************************************************************
unsigned int build_crc(unsigned char length, unsigned char* protocol)
{
        // the initial CRC value
        #define CRC_PRESET 0x0000
        // the reverse CRC-CCIT pollynomial
        #define CRC_POLYNOM         0x8408

        unsigned char i,k;
        unsigned int crc;
        unsigned char crc_in;

        crc CRC_PRESET;                 // initial value
        for(i=0;i< length;i++)          // loop trough the protocol
        {
                crc_in = protocol[i];   // get next protocol byte
                for (k=0;k<=7;k++)      // loop trough one byte LSB to MSB
                {
                        // test each Bit for CRC calculation
                        if((((crc_in>>k)&0x01)^(crc&0x0001))==1)
                        { crc=crc>>1; crc=crc^0x8408; }
                        else
                        { crc=crc>>1;}
                }
        }
        return(crc);
}
```

The checksum is always composed out of 4 hex characters (values from 0x0000 to 0x FFFF).
These 4 hex characters are transmitted in ASCII code from '0' .. '9' and 'A' to 'F'.

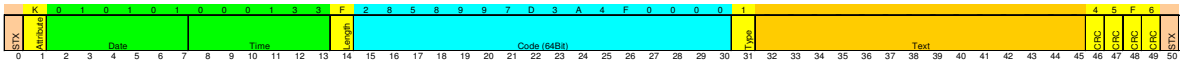
Example:
The checksum 0E2A is transmitted as chr(0x30), chr(0x45), chr(0x32), chr(0x41).

6 Telegram structure

1 byte for attribute (# = no attribute, 'A' ... 'Z') - transmitted as 1 ASCII-character
6 byte for date/clock in BCD-format - transmitted as 12 ASCII- characters
1 byte for code length (1 ... 16) - transmitted as 1 ASCII- character '0' .. '9', 'A' .. 'F'
Example: length = 16; length – 1 = 15; hex value = 0xF;
ASCII-character = 'F' = chr(0x46)
8 byte for the code (up to 16 nibbles, starting left) - transmitted as 16 ASCII- characters
1 byte for type of transponder transmitted as 1 ASCII- character:

0 = unknown type
1 = ISO-Fdx
2 = Marin, ASK 64 Bit
3 = Trovan
4 = Datamars
5 = Destron
6 = ISO-Hdx
7 = Hitag 1, Hitag S
8 = Hitag 2
9 = Pontech
A = PSK 2
B = PSK 1
C = Diehl Aircabin
D = BDE Fdx
E = BDE Hdx
F = ISO 14443A 4 Byte
G = ISO 14443A 7 Byte
H = ISO 15693
U = EM 4305

14 byte for text (up to 14ASCII's in the range 0x20 ... 0x7F) - transmitted as 14 ASCII- characters



Example: description of a record – (sequence of characters at the interface):

STX 'K010101000133F2858997D3A4F00001_____46F6' ETX

The meaning is as follows:
K = Attribute 'K'
010101 = 1. January 2001
000133 = 00:01:33 (hh:mm:ss)
F = Code length = 16
2858997D3A4F0000 = Transponder code
1 = Type of transponder: ISO-FDX
_____ = 14 characters for transponder code associated text (not used = '_')
46F6 is the relevant CRC-checksum

7 Simplified read out procedure, without programming the CRC-routine

In order to get a quick output of data from the hand held reader, programming the CRC may be omitted. In this case the following values have to be used:

Command	CRC in Hex
'ET'	0x2C7F
'EC'	0x 4841
'RP'	0x B2C2
'RN'	0x4B3D
'RL'	0x682F
'WP'	0xCC7A
'SV'	0xCE2C
'XT'	0x0996

Example:
In order to read the software version out of the hand held reader, the following command has to be sent from the PC to the reader: STX 'SVCE2C' ETX. In case of a valid result, the answer is for example STX '610CE8E' ETX, where '610' stands for the Version and 'CE8E' for the check sum belonging to it. The disadvantage of this method is: The command 'W' can not be executed because the check sum is computed out of the entire Record (including the command and the code number). The consequence is, that every command in conjunction with a code number has is own check sum.

8 The command structure in detail (Syntax, function)

8.1 Is the database memory empty?

Command syntax:

STX, "ET", "CRC", ETX

Possible answers of the reader:

ACK = Database contains data sets
BEL = Database is empty
NAK = Error

With this command it can be checked if there are data sets in the reader database.

8.2 Clear database memory

Command syntax:

STX, "EC", "CRC", ETX

Possible answers of the reader:

ACK = Command executed
NAK = Error

This command erases all data sets out of the database memory of the ARE H5.

8.3 Set pointer to first full data set

Command syntax:

STX, "RP", "CRC", ETX

Possible answers of the reader:

ACK = Command executed
NAK = Error

This command sets the data set pointer to the first full data set in database memory.
It is absolutely necessary, that this command is executed, before a read out or clearing command is given. Otherwise the position of the Pointer is undefined.

8.4 Read new data set

Command syntax:

STX, "RN", "CRC", ETX

Possible answers of the reader:

STX, "Data set in ASCII", "CRC in ASCII", ETX
NAK = Error

This command asks for the data set on actual pointer position. After reading the data set pointer is increased. This way next "RN" command reads the next data set.

8.5 Read last data set

Command syntax:

STX, "RL", "CRC", ETX

Possible answers of the reader:

STX, "Data set in ASCII", "CRC in ASCII", ETX
NAK = Error

The reader repeats the output of the data set that was transmitted with the previous "RN" command. The position of the pointer remains unchanged (as set by the previous "RN" command).

8.6 Set pointer to first empty data set in database memory

Command syntax:

STX, "WP", "CRC", ETX

Possible answers of the reader:

ACK = Command executed
NAK = Error

It is absolutely necessary, that this command is executed, before a new data set is stored in the reader. Otherwise the position of the pointer is undefined and already stored records may be overwritten.

8.7 Write data set

Command syntax:

STX, "W", "Data set in ASCII", "CRC in ASCII", ETX

Possible answers of the reader:

ACK = Command executed
NAK = Error

This command writes the data set into the database memory, provided that length, syntax, and CRC have been identified as true. After completion, the pointer is incremented. Consequently at the next "W" command the new record will be written into the next position.

8.8 Software Version

Command syntax:

STX, "SV", "CRC", ETX

Possible answers of the reader:

STX, "Software version in ASCII", "CRC in ASCII", ETX
NAK = Error

This command asks for the software version of the reader.

8.9 Exit communicatio mode

Command syntax:

STX, "XT", "CRC", ETX

Possible answers of the reader:

ACK = Command executed
NAK = Error

This command aborts the operation mode "Database / PC" of the reader.

9 Parameterization commands

9.1 Reader settings

In operation mode "Database / PC" all parameters of the reader are accessible. Some of them can be changed in the menu of the reader. All other parameters can only be changed by the PC with the communication described here.
For writing a parameter the command "s" is used. For reading "S" is the right command. Every reader parameter has its address and range of values. It is transmitted as follows:

Command syntax writing:

STX, "s", "Address = 3xASCII-chr. ", "Value = 2xASCII-chr.", "CRC = 4xASCII-chr.", ETX
Address = 12 bit, Parameter value = 8 bit

Possible answers of the reader:
ACK = Command executed
NAK = Error (Command or parameter)

Example: Timeout Reading (Address 0x010) to 25 cycles = set to approx. 2,25 seconds (25 = 0x19):
<STX>s01019C872<ETX>
CRC = 0xC872

Command syntax reading:

STX, "S", "Address = 3xASCII-chr.", "CRC = 4xASCII-chr.", ETX
Address = 12 bit

Possible answers of the reader:
STX, "Wert 2xASCII-chr.", "CRC 4xASCII-chr.", ETX
NAK = Fehler

Example: read out Timeout Reading (Address 0x010):
<STX>S010E88C<ETX>
CRC = 0xE88C
Answer:
<STX>328E5B<ETX>
Value = 0x32 = 50 cycles = aprox. 4,5 seconds
CRC = 0x8E5B

Address [hex]	Name	Meaning	Default [hex]	Range [hex]
0x000	Attribute	0x00 = No attribute 0x01 = ‚A' 0x02 = ‚B' ... 0x1A = ‚Z'	0x00	0x00 ... 0x1A
0x001	Operation mode	0x00 = Standard 0x01 = Daten -> RS232 0x02 = Read / Transfer 0x03 = Database / PC 0x04 = Data -> Bluetooth [2]	0x00	0x00 ... 0x04
0x002	Interface	0x00 = RS232, USB or Bluetooth 0x01 = IRDA	0x00	0x00 ... 0x01
0x003	Language	0x00 = English 0x01 = German	0x01	0x00...0x01
0x004	Code format	0x00 = Hexadecimal 0x01 = ISO Animal 0x02 = ISO Industry 0x03 = BDE (german waste management format) 0x04 = TRUTEST 0x05 = ISO Animal original [3]	0x01	0x00...0x05
0x005	Lock up reader	0x00 = Off 0x01 = On	0x00	0x00 ... 0x01
0x006	Reserved		0x02	0x00 ... 0x03
0x007	Reserved		0x00	0x00..0x01
0x008	Multiple data sets	0x00 = Off, data sets have to differ at least in attribute to be saved in database 0x01 = On, the same transponder numbers are saved multiple in database	0x00	0x00 ... 0x01
0x009	Type output	0x00 = Off 0x01 = On, in operation mode "Data -> RS232", "Read / Transfer" and "Data -> Bluetooth" a 3 character short cut for the transponder type and a space character is sent in front of transponder code via serial interface (if the code format is not "TRUTEST") Possible short cuts are: FDX = ISO full duplex ASK = ASK 64 Bit TRO = Trovan DAT = Datamars DES = Destron HDX = ISO half duplex PK2 = PSK 2 PK1 = PSK 1	0x00	0x00 ... 0x01

[2] In ARE H5 software versions, that support Bluetooth interface, e.g. 6.35 or B2.00003
[3] Display attribute and text in stead of retagging counter and additional information, since ARE H5 software version 6.35

0x00A	Noread output	0x00 = Off 0x01 = On, in operation mode "Data -> RS232", "Read / Transfer" and "Data -> Bluetooth" the text "NoRead" is sent after a failed reading attempt via serial interface	0x00	0x00 ... 0x01
0x00B	Reserved		0x01	0x00 ... 0x01
0x00C	Algorithm	0x00 = All Algorithms are deactivated 0x01 = Only Trovan is on 0x02 = Only ASK 64 Bit on 0x04 = Only Datamars on 0x08 = Only Destron on 0x10 = Only ISO HDX on 0x20 = Only ISO FDX on 0x40 = Only PSK 2 on 0x80 = Only PSK 1 on 0x03 = Trovan and ASK 64 Bit on ... 0xFF = All Algorithms are activated	0xFF	0x00 ... 0xFF
0x00D	Reserved		0xFF	0x00 ... 0xFF
0x00E	TimeOut (Main) [4]	After this time the ARE H5 is switched off, if there is no user intervention, and the device is not in menu mode, in seconds, default 10s, range 1 ... 255s	0x0A	0x01...0xFF
0x00F	TimeOut (Menu) [5]	After this time the ARE H5 is switched off, if there is no user intervention, and the device is in menu mode, in seconds, default 20s, range 1 ... 255s	0x14	0x01...0xFF
0x00E 0x00F	Time Out [6]	After this time the ARE H5 is switched off, if there is no user intervention, in seconds, default 12s, range 1 ... 65535	0x0C 0x00	0x01 ... 0xFF 0x00 ... 0xFF
0x010	TimeOut (Reading)	Number of reading attempts, one attempt takes approx. 90ms, default 50 cycles	0x32	0x01...0xFF
0x011	Reserved		0x01	0x00 ... 0x01
0x012	Buzzer	0x00 = Off 0x01 = On	0x01	0x00...0x01
0x013	Time output [7]	0x00 = Off 0x01 = On, in operation mode "Data -> RS232", "Read / Transfer" and "Data -> Bluetooth" actual time stamp and a space character is sent in front of transponder code via serial interface (if the code format is not "TRUTEST") e.g. "24.12.10 11:55:00"	0x00	0x00 ... 0x01

[4] Until ARE H5 software version 6.22 and 6.35
[5] Until ARE H5 software version 6.22 and 6.35
[6] Since ARE H5 software version V2.00005 and B2.00003
[7] In ARE H5 software Bluetooth versions since 6.35

0x014	Handshake RS232 [8]	0x00 = Off, in operation mode "Data -> RS232" and "Read / Transfer" a read data set is sent via serial interface, if the telegram is received by the communication partner is not verified 0x01 = On, the transmission of a data set begins with a empty telegram (<STX> <ETX>), if the communication partner is answering (<STX> <ACK> <ETX>), data is sent, reception is verified again by an answer of the communication partner (<STX> <ACK> <ETX>).	0x00	0x00 ... 0x01
0x015	Bluetooth Role [9]	0x00 = Master, the Bluetooth connection is initiated by the ARE H5 0x01 = Slave, the Bluetooth partner cares for connection establishment	0x00	0x00 ... 0x01
0x016	Handshake Bluetooth [10]	0x00 = Off 0x01 = On Same function as "Handshake RS232" for wireless communication	0x01	0x00 ... 0x01

9.2 Text association to attributes

The reader provides 27 attributes. They are characterized with "#" (= no attribute) and the capital letters "A" to "Z". To every attribute a text, up to 14 characters, can be allocated:

Example:
If "Stable" is allocated to "A", "Pasture" to "B" and "Forage" to "C", in the ARE H5 menu "Set Attribute" the followings attributes are available:

"#" -> "Stable"-> "Pasture" -> "Forage" -> "D" -> "E" -> ... -> "Z"

The allocated attribute text is shown in the lower ARE H5 display line, right-aligned, after reading a transponder. Please be aware of the transponder type or a transponder number allocated text is also shown in the lower display line, left-aligned. The display line has 16 characters, the transponder text has maximal 14 characters (left-aligned) and the attribute text has maximal 14 characters, maximal 12 of them are displayed (right-aligned). This way attribute text can overlap transponder text.

[8] In the ARE H5 software Bluetooth versions since 6.35
[9] In the ARE H5 software Bluetooth versions since 6.35
[10] In the ARE H5 software Bluetooth versions since 6.35

The related command to set attribute text is:

STX, "t", "Attribute 1xASCII-chr.", "Attribute text 3-14xASCII-chr.", "CRC 4xASCII-chr.", ETX.

Possible answers of the reader:
ACK = Command executed
NAK = Error

Format:
The attribute token is "#" or "A" to "Z", the text must be 3 to 14 ASCII characters long. If no attribute text shall be displayed but only the attribute token (default setting), the command "t" with the desired attribute token and the attribute text 3 underline characters "_" must be sent.

Example:

To replace "A" with the text "Stable", the following command has to be sent:
<STX>tAStable7F7F<ETX> with t = Command, A = Attribute, Stable = Text and CRC = 7F7F.

To delete the text of attribute "A" again, the following command has to be sent:
<STX>tA___0186<ETX> with t = Command, A = Attribute, ___ = Text (no attribute text) and CRC = 0186.

To read the attribute text from the ARE H5, the following command has to be sent:

STX, "T", "Attribute 1xASCII-chr.", "CRC 4xASCII-chr.", ETX
Answer: STX, "Attribute 1xASCII-chr.", "Value 1-14xASCII-chr.", "CRC 4xASCII-chr.", ETX

Example:

To read the attribute text of attribute "A", the following command has to be sent:
<STX>TAE71A<ETX> with T = Command and A= Attribute and CRC = E71A

Answer: <STX>A538D<ETX> with A = Attribute und CRC = 538D, no attribute allocated to "A"
or <STX>StableB90B<ETX> with Stable = Attribute and CRC = B90B

9.3 Date and time

In operation mode "Database / PC" date and time of the ARE H5 can be read and written.

To set the clock of the reader, the command "r" is used:

Command syntax writing:
STX, "r", "Date and time in ASCII BCD", "CRC 4xASCII-chr.", ETX
Date and time are transmitted in ASCII format of the BCD coding.

Possible answers of the reader:
ACK = Command executed
NAK = Error

Example:
Set the clock of the reader to 15. November 2002, 10:02:16.
<STX>r1511021002162CA5<ETX> with CRC = 2CA5.

To get date and time from the reader, the command "R" is used:

Command syntax reading:
STX, "R", "CRC 4xASCII-chr.", ETX

Possible answers of the reader:
STX, "Date and time in ASCII BCD", "CRC 4xASCII-chr.", ETX
Date and time are transmitted in ASCII format of the BCD coding.
NAK = Error

Example:
<STX>R7197<ETX> with CRC = 7197
<STX>02091008333768A0<ETX> with CRC = 68A0
ARE H5 date and time is 2. September 2010, 08:33:37 (hh:mm:ss)

10 Notification of changes

Release	Date	Changes	Author
3.1	09/10/2003		Genz
3.2	09/02/2010	New format, additional for new software version	MK
3.3	09/29/2010	Translation from the German version	MK
3.4	08/17/2012	Further data carrier types	MK
3.5	06/09/2015	Further data carrier types	MK
3.6	10/06/2017	Translation of german text in english document	MK

11 Contacts

To improve our products, as well as its documentation is our permanent effort.
For any questions, feedback or comments please call:

Tel.: ++49 (0)731-140088-0

Fax: ++49 (0)731-140088-9000

e-mail: sales@aegid.de

http:\ www.aegid.de