Softrónica

Ing. Electrónica Software y Comunicaciones

SOFTRÓNICA GENERIC PROTOCOL FOR RFID

Protocol Description

EHAG
ELECTRONIC HARDWARE AG
Industriestr. 8 CH-8618 Oetwil a/S.
T: +41 43 844 94 00 info@ehag.ch
F: +41 43 844 94 01 www.ehag.ch

Cod.: Rev.: Draft

May 2001

Índice

1 OVERVIEW

This document describes a new protocols developed by Softrónica in order to build a technology-independent communication interface, suitable for RFID multi-protocol or even multi-frequency readers.

The benefit of this implementation will be that a change in technology from one air-interface protocol to other will not mean any change in the host application, as the low-level details of the air-interface will be handled by the reader/encoder, making all those details transparent to the software engineer developing the application.

2 PROTOCOL MODES

Two different modes have been implemented. The command set will be the same, but the protocol structure of the first one (BINARY mode) is better suited for applications requiring large ammount of data transfer, like for instance, when dealing with many tags or technologies with big memory depths. It will support network operation, with one address byte.

The second one (ASCII mode) is an ASCII protocol for simple applications, like short range, few tags, serial number only, ... It will be possible to handle a device from a simple ASCII terminal, just typing the commands, and looking for the answer. It will optionally accept also address and checksum, if more security or network operation is required.

The protocol is also independent of the network, so it can be used in point to point communication interfaces, as in RS232, or in network operation via RS485, or in more complex structures. For instance, it will be possible to use it through a TCP/IP network. In this case, the Softronica messages will be embedded inside the frames or packet structures of the network to be used.

The protocol doesn´t define any baud rate, or data encoding scheme, as this will change according to the different kind of network used.

3 SOFTRONICA PROTOCOL

3.1 PROTOCOL OVERVIEW

The reader will accept commands sent in any of the two modes. It will answer in the same format as the received command. The first character of the packet will define the protocol mode, and the presence or absence of an address or access key for network operation.

3.2 DATA TYPES IN ASCII MODE

In this protocol, all the characters are «printable» characters, so they can be sent from a simple «ASCII» terminal just typing in the keyboard, and the answer will be represented on the screen.

The different data types encoding scheme for this mode of operation is defined next.

Literal characters are sent as they appear on the document. For instance, some commands will start with the literal character «$», so a single byte with the corresponding ASCII code will be sent. They will be described as **«LITERAL»**

Characters are sent as they appear on the command description. They are always printable characters. For instance, the command read (capital letter R) will be sent as a single byte corresponding to the ASCII character «R». They will be described as **«CHAR»**

8 bit data are sent as 2 digits corresponding to the ASCII representation of the number in hexadecimal format. For instance, the decimal number 25 (hexadecimal 19), will be sent as the two ASCII caracteres «1» «9». This will be described as **«BYTE»** data in the rest of the document.

16 bit data will be sent as 4 bytes in hexadecimal format, with the high order byte first. For instance, the decimal number 554 (hexadecimal 22A) will be sent as the 4 ASCII characters «0» «2» «2» «A». This will be described in the rest of the document as **«WORD»** data.

3.3 DATA TYPES IN BINARY MODE

Different data types will be defined in this document to be used in this protocol.

Literal characters are sent as they appear on the document. For instance, some commands will start with the literal byte «:», so a single byte with the corresponding ASCII code will be sent. They will be described as **«LITERAL»**

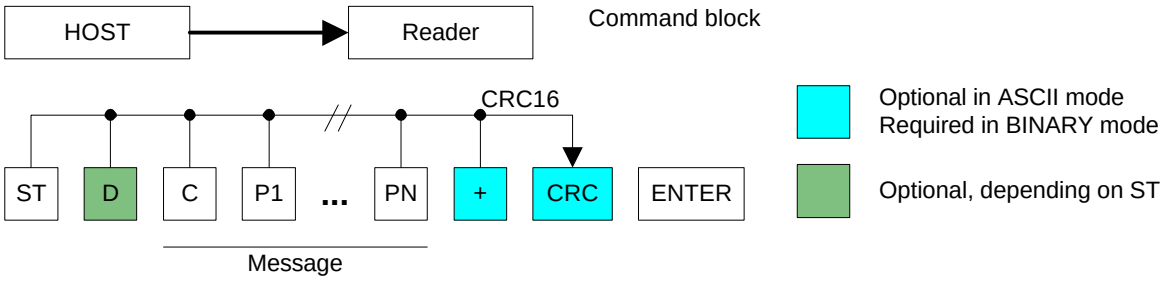
Characters are sent as they appear on the command description. They are always printable characters. For instance, the command read (capital letter R) will be sent as a single byte corresponding to the ASCII character «R». They will be described as **«CHAR»**

8 bit data are sent as 1 byte with the corresponding binary information. For instance, the decimal number 25 (hexadecimal 19), will be sent as a single byte containing the hexadecimal number 19. This will be described as **«BYTE»** data in the rest of the document.

16 bit data will be sent as 2 bytes in hexadecimal format, with the high order byte first. For instance, the decimal number 554 (hexadecimal 22A) will be sent as 2 bytes containing the hexadecimal numbers 02 and 2A. This will be described in the rest of the document as **«WORD»** data byte.

3.4 COMMANDS

The structure of the protocol is described in the following diagrams. First the communication block sent from the host to the reader (command block) is depicted:



Where:

ST **CHAR** It indicates the mode of the command and the presence or absence of address or access key for network operation, accorcing to the following coding:

- **$** Command in ASCII mode without address
- **%** Command in ASCII mode with address
- **:** Command in BINARY mode without address
- **;** Command in BINARY mode with address

D **BYTE** Address or access key of the device to wich the command is sent. If it is 0, the operation is a broadcast operation, so the information will be accepted for all the devices in the network. The address range is from 01 to FF hexadecimal.

C **CHAR** Command code corresponding to the operation to be performed, as described in the COMMAND LIST section

P1..PN **ANY** Parameters of the command. Their meaning and type are described in the COMMAND LIST section.

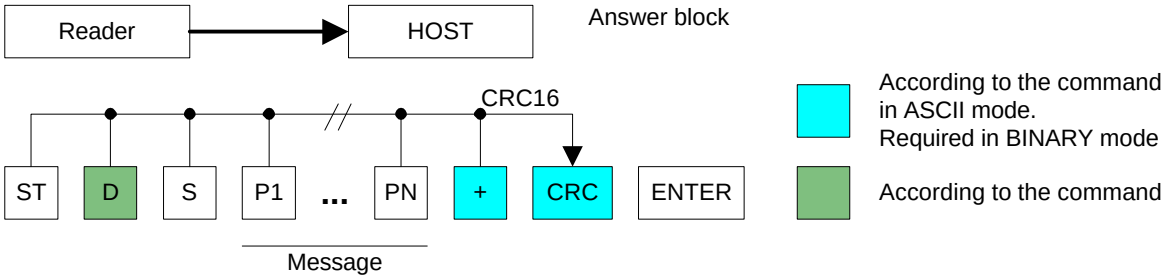
+ **LITERAL** It indicates that the CRC16 follows. It is optional in ASCII mode, but required in BINARY mode.

CRC **WORD** CRC, it is calculated as the CRC16 of all the data in the packet

ENTER **LITERAL** Character «Carriage Return» (Decimal 13) to mark the end of the packet

3.5 ANSWER

The answer from the reader is depicted in the following diagram:



Where:

ST **CHAR** It indicates the mode of the command and the presence or absence of address or access key for network operation. It has to be the same as in the command received , accorcing to the following coding:

- **$** Command in ASCII mode without address
- **%** Command in ASCII mode with address
- **:** Command in BINARY mode without address
- **;** Command in BINARY mode with address

D **BYTE** Address or access key of the device that is answering. It has to be the same that was sent in the command. If the command is a broadcast command, it will not require any inmediate answer.

S **CHAR** Answer code. It will indicate communication status. It will be the received command if there is not error or «E» if there is a communication error. In this case, the error code will be the only data in the answer field.

P1..PN **ANY** Data sent to the host. Their meaning and type are described in the COMMAND LIST section. If there is a communication error (as described in the S field), a single BYTE will be send with a code as described in the COMMUNICATION ERROR CODES annex.

+ **LITERAL** It indicates that an CRC16 follows. It will be sent only when the corresponding command was send with an address at the beginning.

CRC **WORD** CRC, it is calculated as the CRC16 of all the data in the packet

ENTER **LITERAL** Character «Carriage Return» (Decimal 13) to mark the end of the packet

An inmediate answer (see section on DELAYED ANSWER) will be sent after a command received.

3.6 DELAYED ANSWER

Some commands will not need any inmediate answer. For example, a broadcast command, to be accepted by all the readers on a network.

In other cases, a device will answer without any inmediate command from the host. For instance, in a «contnious read» operation, where the reader will send the data of any tag present in the field when it appears.

In both cases, the answer will be considered «delayed», and the format will be the same as described in a normal answer command. The use of CRC or address will correspond to the characteristics of the last command received by the device.

3.7 ASCII MESSAGE FRAME

The packets, in ASCII mode, will start by a literal «$» or «%» character, depending on the need for an address or access key to the device for network (RS485) operation.

The rest of the data is sent, according to the rules specified in section 3.2. DATA TYPES IN ASCII MODE. After the message, the end of the packet can be marked on three different ways:

- If CRC checking is used, it is marked by the Literal «+» followed by the WORD (4 ASCII characters) corresponding to the CRC16, and the literal character ENTER or Carriage Return (Decimal 13)
- If CRC checking is not required, just the ENTER character is sent.
- If CRC checking is required in the answer, but not in the command, just the literla + followed by the ENTER character is sent.

3.8 BINARY MESSAGE FRAME

The packets, in BINARY mode, will start by a silent interval of at least 3.5 character time. The first character is then transmitted, and can be either a literal «:» or a «;», depending on the need for an address or access key to the device for network (RS485) operation.

The rest of the data is sent, according to the rules specified in section 3.3 DATA TYPES IN BINARY MODE. The end of the packet is marked by the Literal «+» followed by the WORD (2 BINARY characters) corresponding to the CRC16, and a Literal Carriage Return (Decimal 13).

The entire message frame must be transmitted as a continous stream. If a silent interval of more than 1.5 character time occurs, the packet is either complete, or an error has occurred. To discriminate between the two conditions, the +, CRC16, and ENTER sequence must be checked.

Similarly, if a new message begins earlier than 3.5 character times following a previous message, the receiving devide will consider it a continuation of the previous message. This will set an error, as the CRC value will not be correct at the end of the packet.

Both the command and the answer will follow those rules for the interchange of information.

3.9 CRC CALCULATION

The 16 bit cyclic redundancy check character (CCITT-CRC16) is calculated as described in the following routine.

Generator Polynom: $X^{16} + X^{12} + X^5 + 1$ ⇒ CRC_POLYNOM = 8408 Hex

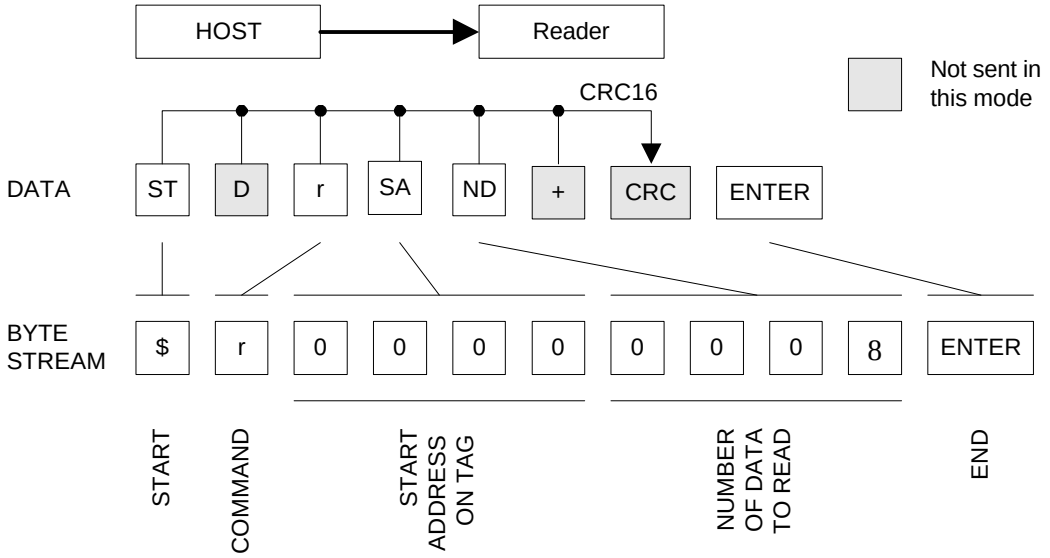
Preset Value: ⇒ CRC_PRESET = FFFF Hex

Calculation algorithm (C example)

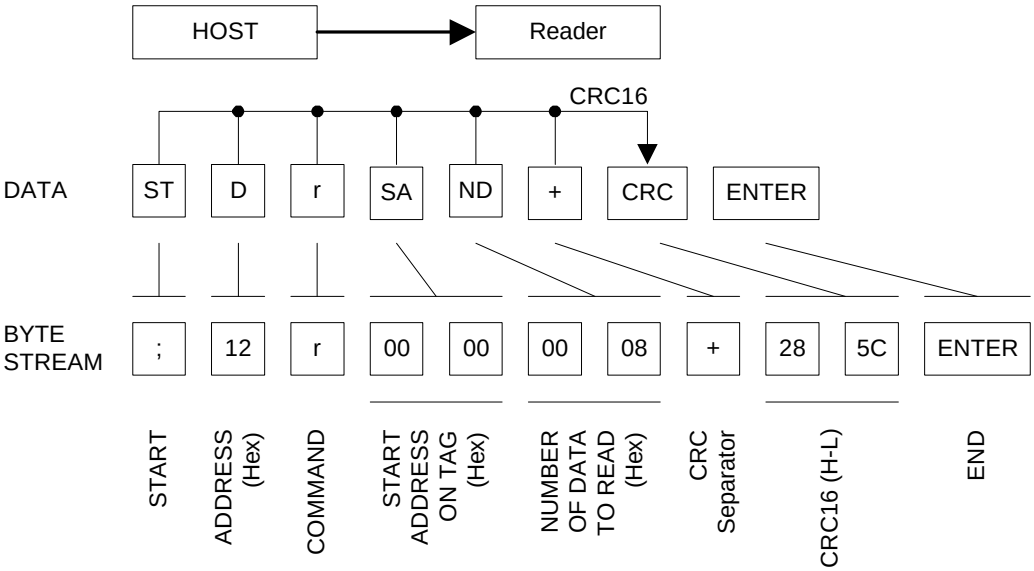
```
unsigned int crc=CRC_PRESET;

for (i=0;i<cnt;i++)          /* cnt = length of the packet up to  */
                             /*       and including the + literal */
  {
  crc^=Data[i];
  for (j=0;j<8;j++)
       {
       if (crc & 0x0001)
            crc=(crc>>1)^CRC_PLOYNOM;
       else
            crc=(crc>>1);
       }
  }

/* Command */
Data[i+1]=crc & 0xFF;        /*CRC16 Low Byte*/
Data[i]=crc >> 8;            /*CRC16 High Byte*/
```

3.10 EXAMPLES

SINGLE READ COMMAND IN ASCII MODE WITH NO CRC AND NO ADDRESS



SINGLE READ COMMAND IN BINARY MODE WITH CRC AND ADDRESS



4 COMMAND SET

The command set is independent of the protocol mode, so only the format will change between the Softronica ASCII and the Softronica BINARY protocols, as described in the corresponding sections.

4.1 SINGLE READ

It will try to read only one tag in the field.

Command format

```
r DI ND
```

Answer

```
P N D1 D2... DN
```

Description

r **LITERAL** Command

DI **WORD** Starting address. The address of the first data to send inside the tag memory

ND **WORD** Number of data to be sent from the memory of the tag

The reader will try to read ND number of data from the memory of the tag read, starting at address DI.

Answer

P **CHAR** Corresponding to the protocol of the tag read. See LIST OF PROTOCOLS annex.

N **WORD** Number of read data. It will correspond to ND unless the tag does not support such number of data. 0 will indicate error on the air interface, or no tag present. In this case, an error code will be the only data sent after. See AIR INTERFACE ERROR CODES annex.

D1..DN **BYTE** N data read from the tag memory. If there is an error on the air interface, N will be 0, and only one error code will be sent here. See AIR INTERFACE ERROR CODES annex.

4.2 ANTICOLLISION READ

It will try to read several tags in the field with the appropriate anticollition protocol.

Command format

```
R DI ND TE
```

Answer

```
T

P1 N1 D11 D21...DN11

P2 N2 D12 D22...DN22

...

PT NT D1T D2T...DNTT
```

Description

R **LITERAL** Command

DI **WORD** Starting address. The address of the first data to send inside the tag memory

ND **WORD** Number of data to be sent from the memory of the tag. If ND=0, it will only select the tags, which can be useful in certain protocols for posterior writing.

TE **BYTE** Max. Number of tags expected in the field. If it is 0, the reader will automatically try to read all the tags present.

The tag will try to read ND number of data from the memory of the tag read, starting at address DI. The maximum number of tags expected is TE, and is used in certain protocols to allow for more tags to be read at once, but taking more time to finish the reading operation.

Answer

T **BYTE** Number of tags actually read (with or without error)

Px **CHAR** Protocol corresponding to every tag of the T detected. See LIST OF PROTOCOLS annex.

Nx **WORD** For every tag detected, number of read data. It will correspond to ND unless the tag does not support such number of data. 0 will indicate error on the air interface, or no tag present. In this case, an error code will be the only data sent after for this tag. See AIR INTERFACE ERROR CODES annex.

D1x..DNx **BYTE** For each tag, Nx data read from the tag memory. If there is an error on the air interface, Nx will be 0, and only one error code will be sent here. See AIR INTERFACE ERROR CODES annex.

The answer is formed by a number of tags read T and a concatenation of the T individual responses as described in the SINGLE READ command.

4.3 STATUS

It will send the internal status information of the reader

Command format

S M

Answer

CE

or

CE PD PR T

Depending on the M parameter

Description

S **LITERAL** Command

M **CHAR** Character «S» if simple status is required or «D» if detailed status information is needed.

The reader will send some information depending on the mode selected.

Answer

CE **BYTE** Status code, according to the STATUS CODES LIST annex

PD **BYTE** Direct or incident power sent by the reader (in Watts)

PR **BYTE** Reflected power measured by the reader (in Watts)

T **BYTE** Internal temperature in Celsius degrees, as measured inside the reader

4.4 EAS DETECTION

It will try to detect the presence of a tag on the field (EAS mode)

Command format

D

Answer

CD

Description

D **LITERAL** Command

Depending on the tag protocol, this will mean EAS detection or any other similar feature if it is available.

Answer

CD **BYTE** EAS level, or similar value, from 00 to 0xFF

4.5 WRITE

Write operation on the tags

Command format

W DI ND TE M1 ... MTE D1 ... DND

Answer

T CE1 CE2 ... CET

Description

W **LITERAL** Command

DI **WORD** Starting address. The address of the first data to be written in the tag memory

ND **WORD** Number of data to be written into the tag memory

TE **BYTE** Number of tags to be written

M1..MTE **CHAR** Writing mode for each tag selected between 1 and TE. «Y» will mean writing to be performed, and «N» will mean no writing on this particular tag. If the protocol does not support this feature, the parameter will be ignored.

D1..DND **BYTE** Data to be written to the tag memory

The reader will try to write ND number of data into the memory of the tag read, starting at address DI. It will do it for TE tags, which will be written or not depending on the M1..MTE parameters. The data to be writtein is on D1..DND.

Answer

T **BYTE** Number of tags actually written (with or without error)

CEx **BYTE** Error code in the writing operation of the tag x. See ERROR CODES ON AIR INTERFACE annex.

4.6 SILENCE

Put the tag on silent or halt mode

Command format

```
Q TE M1 ... MTE
```

Answer

```
T CE1 CE2 ... CET
```

Description

Q **LITERAL** Command

TE **BYTE** Number of tags to be put into silent mode

M1..MTE **CHAR** Writing mode for each tag selected between 1 and TE. «Y» will mean putting the tag into silent mode, and «N» will mean no changing this particular tag. If the protocol does not support this feature, the parameter will be ignored.

The reader will try to put some of the TE tags on the silent mode, as described on the M1..MTE parameters.

Answer

T **BYTE** Number of tags actually put into silent mode (0 if the tag protocol does not support this feature)

CEx **BYTE** Error code in the writing operation of the tag x. See ERROR CODES ON AIR INTERFACE annex.

4.7 PROTOCOL SELECTION

Configures the device for operation with a number of protocols.

Command format

P PD M

Answer

R

Description

P **LITERAL** Command

PD **CHAR** Protocol to be configured, according to the PROTOCOL LIST annex

M **CHAR** Mode of the protocol. «Y» will enable the protocol, «N» will disable it, and «?» will ask for the state (enabled or disabled) of the protocol.

Answer

R **CHAR** The protocol is: «Y», enabled, «N» disabled, or «I» not supported in this version.

4.8 RESET TAGS

It will reset all the tags in the field to its normal operation state.

Command format

q

Answer

q

Description

q	**LITERAL**	Command

The operation actually performed will depend upon the RF protocol used. For instance, in ICODE, this will send the RESET_QUIET_BIT.

Answer

q	**LITERAL**	Same as received command

4.9 CONTINOUS READ

Configure the reader to work in continous read mode

Command format

C DI ND TE

Answer

V

Delayed Answer

T

P1 N1 D11 D21...DN11

P2 N2 D12 D22...DN22

...

PT NT D1T D2T...DNTT

Description

C **LITERAL** Command

DI **WORD** Starting address. The address of the first data to send inside the tag memory

ND **WORD** Number of data to be sent from the memory of the tag. If ND=0, it will only select the tags, which can be useful in certain protocols for posterior writing.

TE **BYTE** Max. Number of tags expected in the field. If it is 0, the reader will automatically try to read all the tags present.

The tag will try to read ND number of data from the memory of the tag read, starting at address DI. The maximum number of tags expected is TE, and is used in certain protocols to allow for more tags to be read at once, but taking more time to finish the reading operation.

Answer

V **CHAR** «Y» if the command is supported «N» if not

Delayed Answer

T **BYTE** Number of tags actually read (with or without error)

Px **CHAR** Protocol corresponding to every tag of the T detected. See LIST OF PROTOCOLS annex.

Nx **WORD** For every tag detected, number of read data. It will correspond to ND unless the tag does not support such number of data. 0 will indicate error on the air interface, or no tag present. In this case, an error code will be the only data sent after for this tag. See AIR INTERFACE ERROR CODES annex.

D1x..DNx **BYTE** For each tag, Nx data read from the tag memory. If there is an error on the air interface, Nx will be 0, and only one error code will be sent here. See AIR INTERFACE ERROR CODES annex.

The delayed answer has the same format as in a ANTICOLLISION READ operation. The device will exit this mode after receiving any other valid command.

4.10 CONFIGURATION

Configure the reader´s operating parameters.

Command format

N M P V

Answer

M

Description

N	**LITERAL**	Command
M	**CHAR**	Mode. It will indicate if the value to be changed is in the EEPROM («E») or RAM («R»).
P	**BYTE**	Parameter to be configured. See OPERATING PARAMETERS LIST annex
V	**BYTE**	Value to set the parameter

The reader parameter P will be configured with value V. The change will be made either on the RAM or on the EEPROM of the unit depending on the parameter M.

Answer

M	**CHAR**	«Y» if the command is supported «N» if not

4.11 CONFIGURATION READ

Asks for the reader´s operating parameters.

Command format

n M P

Answer

V

Description

n **LITERAL** Command

M **CHAR** Mode. It will indicate if the value to be read is in the EEPROM («E») or RAM («R»).

P **BYTE** Parameter to be read. See OPERATING PARAMETERS LIST annex

The reader parameter P will be read from RAM or EEPROM according to the value of M.

Answer

V **BYTE** Value read corresponding to the parameter

4.12 MEMORY READ

Asks the tags contents read in auto-read mode and stored in the reader memory.

Command format

```
M
```

Answer

```
T

P1 N1 D11 D21...DN11

P2 N2 D12 D22...DN22

...

PT NT D1T D2T...DNTT
```

Description

M	**LITERAL**	Command

The reader will send to the host the information corresponding to all the tags detected in auto-read mode. To empty the buffer, the RESET MEMORY BUFFER command will be sent.

Answer

T	**BYTE**	Number of tags actually read (with or without error)
Px	**CHAR**	Protocol corresponding to every tag of the T detected. See LIST OF PROTOCOLS annex.
Nx	**WORD**	For every tag detected, number of read data
D1x..DNx	**BYTE**	For each tag, Nx data read from the tag memory

The answer is formed by a number of tags read T and a concatenation of the T individual responses as described in the SINGLE READ command.

4.13 RESET BUFFER

This command will empty the buffer of tags read in auto-read mode.

Command format

B

Answer

B

Description

B **LITERAL** Command

To be used after reading the buffer with the READ MEMORY command.

Answer

B **LITERAL** Same as command

4.14 RESET

This command will make a soft reset to the reader.

Command format

X

Answer

No answer

Description

X **LITERAL** Command

If there is a change in the operating parameters, this will restart the reader with the new configuration.

Answer

4.15 SAMPLE

This command will ask for samples in the analog signal received

Command format

P NS

Answer

N M1 ... MN

Description

P	**LITERAL**	Command
NS	**WORD**	Number of samples requested

This command will retrieve samples of the analog signal in the receiver, for system performance analysis.

Answer

N	**WORD**	Number of samples actually retrieved. It should be NS
M1..MN	**WORD**	Samples of signal

4.16 ACCESS CODE

To access tags with acces codes or security

4.17 TRIGGER READ

To wait for a tag to be present and then return the data read

4.18 MASTER MODE ENTER

To enter master mode inmediately after power-up

4.19 CONTROL

To control external elements, like tuning units, IO ports, ...

ANNEX 1 PROTOCOL LIST

ANNEX 2 COMMUNICATION ERROR CODES

ANNEX 3 AIR INTERFACE ERROR CODES

ANNEX 4 STATUS CODES

ANNEX 5 OPERATING PARAMETERS LIST