UHF Protocol Guide

for metraTec UHF Readers and Modules

Date: November 2011

UHF Protocol Guide 1.9

For Firmware-Versions:
- DeskID UHF 2.1
- DwarfG2 2.1
- Pulsar 2.1

Contact / Support:

Email: support@metratec.com

Web: http://www.metratec.com

Tel.: +49 (0)391 251906-50

Fax: +49 (0)391 251906-01

1. Introduction

This document describes the metraTec firmware protocol for all metraTec UHF RFID readers. This includes the DeskID UHF, DwarfG2 and the PulsarMX reader. All HF readers by metraTec use a similar firmware with the same Reader-PC-Communication protocol. Even though there are some differences, espeacially supported functions (GPIOs for example), default parameter (number of tag best read, power saving mode) and power.

The target audience for this document are programmers, who need to communicate with the reader but don't have to option of using our free .NET DLL because they either use a different programming language than the ones supported by Visual Studio (such as Java, Delphi, ANSI-C, …) or need to program for a different operating system such as Linux or even an embedded system or SPS. It might also make sense to use this low-level protocol if your goal is maximum speed and you can't wait for a DLL to process your data stream.

The reader firmware offers an ASCII based programming interface. The instructions are identified by an easy to remember, three character string usually followed by mandatory parameters and/or optional parameters. The response format depends on the type and result of an instruction.

Instructions (as well as this document) are divided into two main groups:

- Reader Instructions, divided into
    - Reader Control Instructions
    - Reader Configuration Instructions
- Tag Manipulation Instructions

All instructions have possible error codes which are described in section 5. In default mode (VBL=1) any data send to the reader causes some answer (including parsing errors, communication errors etc.).

1.1. *Basic Gen2 Informations*

A GEN2-Tag contains up to 4 memory areas called memory banks. This 4 memory banks are:

EPC, TID, USR and RES (Electronic Product Code, TagID, USeR, REServed (for passwords))

The EPC bank contains a 1 word (16 bits) CRC, a 1 word PC (Protocol Control) and a number of words as EPC (Electronic Product Code). The CRC is a CRC 16, computed over PC and EPC. Not the complete membank needs to be used as EPC. The length of the EPC is saved in the 5 MSBs of the PC in words. So 5 Bit contain a maximum of 31 words or 62 Byte of EPC data. A length of zero will be an empty EPC. The PC can be written by the user as can the EPC and the

CRC. Additionally, the CRC is recalculated after every re-powering of the tag. Keep in mind that a wrong CRC will cause problems, so when writing the EPC membank also change the CRC or re-power the tag (move it from field or switch the RF field off and on).

CRC	PC			EPC
WORD 0	WORD1			WORDS 2:(1+Length)
CRC16	Bit 0:4 EPC Length in WORD. MAX value: 31	Bits 5:6 Reserved (00)	Bits 7:F NSI	EPC

The TID (Tag IDentifier) membank contains an ID for the tag. They are often unique but don't need to be unique.

Bits 0:7 (1 Byte)	Bits 8:31 (3 Byte)
ISO/IEC 15963 allocation class identifier	Additional identification to uniquely identify custom commands and optional features

The USR (user) membank can be used to store data on the tag. It is optional, so not every UHF-tag has usable memory in this bank. Its size is unlimited by the EPC Class 1 Gen 2 Protocol but the firmware „only“ supports addressing up to eight Gbyte (unsigned32 variable, 2 bytes per level). Later versions might change this, if needed.

The RES (reserved) membank contains the Access Password (2 word) and the Kill Password (2 word). Passwords can be written to and read from the tag. They can be protected separately by the lock command. If required, they can become permanently unwritable and unreadable.

Bits 00:1F (2 WORDs)	Bits 20:3F (2 WORDs)
Kill Password	Access Password

All memory banks may contain additional data following to mandatory data. This is manufacturer specific.

1.2. Further Documentation

To fully understand all commands and the response of the UHF tag it might be helpful to read the EPC UHF Class 1 Gen 2 Specification, which can be downloaded for free at http://www.epcglobalinc.org/standards/uhfc1g2

Additionally, reading the tag IC datasheet of the tag you use is also a good idea if you want to use advanced feature of the tag which are defined by the manufacturer.

2. Communication Principle

The communication is based on ASCII strings. Each string is terminated with a carriage-return and will be transmitted with MSB fist.

If you use a serial connection to communicate with the reader (USB via Com-Port-Emulator, RS232, etc.) you need to use a baudrate of 115.200 Baud, 8 data bits, no parity, 1 stop bit (8,N,1). Firmware older than the actual one might use 460.800Baud instead. (DeskID <=1.6, Dwarf<=1.4, Pulsar=1.0)

NOTE

Please make sure that you really send a carriage-return character as the last character – not more and not less. Many programs (including Hyperterm and some Unix/Linux programs) use carriage-return + line-feed as end of line character which leads to problems after the first command, since the LF is treated as first character of the following command which results in the error „Unknown COmmand" (UCO) or Command Receive Timeout (CRT). For highest comfort use the free metraTerm application that can be found on the metraTec website (http://www.metratec.com/en/support/downloads/software.html).

2.1. Host to Reader Communication Details

General line:

Instruction<SPACE>Parameter<SPACE>Parameter<CR>

Example without Parameter:

```
REV<CR>
```

in ANSI C:

char Rev[4] = {'R','E','V',13};

The first values which will be sent is 'R' (52h), followed by 45h, 56h, 0dh. Some instructions may be specified with parameters, which are separated by a space (20h).

Example with Parameter

```
INV<SPACE>SSL<CR>
```

char Inv[8] = {'I','N','V',0x20,'S', 'S', 'L' ,0x0D};

2.2. Reader to Host Communication

In general, the communication is the same as above but the response from the reader may comprises more than one line. Also it depends on parameters set by the user. All answers shown in this documentation refer to default settings (VBL=1, COF, SET EPC OFF and so on).

2.3. Helpful Tools

For debugging purpose it may be helpful to use a program to "sniff" the communication between the host and the reader. Depending on the type of communication and hardware you use, this could be:

- A easy to use terminal program (we use our own metraTerm as it can compute CRC, parses special ASCII-Chars (<CR>, <EOF> and others) and is flexible. In case of a slow PC it may be to slow, especially for CNR commands.
- If you communicate via a (real or virtual) COM-Port: a Com-Port Monitor (several free version available in the net)
- If you use Ethernet or other TCP/IP-based communication, like WiFi: a packet sniffing tool, e.g. wireshark/ethereal, which is available for almost every platform
- If you use a direct UART connection or something at a similar low level: a hardware logic analyzer

3. Reader Instructions

This list gives an overview of all the existing Reader Instructions. The commands often have possible answers. General error codes (URE, CRT, BOD...) and parsing errors (UCO, EDX, EHX, NOR...) are always possible, even if no possible error is named. If used correctly, parsing errors should not occur. Possible parsing errors depend on the command and its parameters.

Command	Name	Description
RST	Reset	Resets the reader
REV	Revision	Returns hard architecture and firmware version
STB	Standby	Sends the reader into standby/sleep mode for power saving
WAK	Wake Up	Ends standby/sleep mode
RIP	Read Input Pin	Reads the state of an input pin
WOP	Write Output Pin	Writes the state of an output pin
RRP	Read Reflected Power	Gives you information regarding antenna mismatch, broken cables as well as missing antennas.
SIR	Set Inventory Retry	Sets the maximal number of tag search retries. The number of search rounds is IR + 1. Higher value brings more tags. Also, the Q matching works better (Q changes at the end of every round). The change of Q is not saved in any way (no adaption)
RIR	Read Inv. Retry	Gets the IR (set by SIR). Default value of IF is two.
SQV	Set Q value	Sets the start Q value used for tag search according to EPC_Gen2. The Q value sets the number of slots to 2^Q. The number of slots should be at least the max. number of tags in field
RQV	Read Q value	Gets the Q value set by SQV. The default value is 4 (16 slots)
CFG	Configuration	Configuration of RFID frontend parameters
SET	Settings	Lets you configure different settings for the reader
MOD	Mode	Sets the mode of the reader
SRI	Set RF Interface	Configures the RF Interface
CON	CRC on	Turns on CRC checking of computer / reader communication
COF	CRC off	Turns off CRC checking of computer / reader communication
EOF	End of Frame	Adds an 0x0A ASCII (Linefeed) after every answer block
NEF	No End of Frame	Stops EOF mode
VBL	Verbosity Level	Use this to set different amounts of responses
RSN	Read Serial Number	Returns the serial number of the device

Command	Name	Description
RHR	Read Hardware Revision	Returns the hardware revision of the device

3.1. Reader Control Instruction

The Reader Control Instructions are a set of commands that control basic functions of the reader that are not directly connected to the HF part of the reader (such as setting digital IOs) as well as request basic information.

3.1.1. Reset (RST)

The reset command resets the reader. The Reset command has no parameters. After sending the RST command the HF power is turned off, the reader is reset to its default values and has to be initialized again. RST works even in sleep mode and CNR mode. Any parameter change (except for EEPROM saved data, they need to be reloaded) will be forgotten. The next commands are accepted 20ms after the "OK!" answer.

Instruction:

```
RST<CR>
```

Response:

```
OK!<CR>
```

3.1.2. Revision (REV)

The revision command requests the device type and hardware architecture and firmware revision of the reader. The reader returns its device type and its hardware architecture version (i.e. type of CPU used) - and firmware revision. The Revision command has no parameters.

Instruction:

```
REV<CR>
```

Response:

```
PRODUCT_NAME<SPACE>HW_revision[4bytes]FW_revision[4bytes]<CR>
```

15 Bytes product name (filled with spaces) + 4 bytes HW-Architecture-Revision + 4 Bytes Firmware-Revision + <CR>

Example for a response:

```
DESKID_UHF<5 Times Space>01000100<CR>
```

Interpretation:	Product name:	DESKID_UHF
	Hardware-Architecture-Revision:	01.00
	Firmware-Revision:	01.00

3.1.3. Standby (STB)

The standby command sets the reader in a power save mode. The RF power is turned off. This means that all tags that might be in the field will also be depowered. STB returns GN8 ("Good Night"). The reader will not accept any commands except RST until a Wake Up Command (WAK) is received. Standby has no parameters.

Instruction:

```
STB<CR>
```

Response:

```
GN8<CR>
```

3.1.4. Wake Up (WAK)

The wake up command ends the power save mode. Reader will restore its last state prior to the standby. If successful it returns GMO ("Good Morning"). Wake up has no parameters.

Instruction:

```
WAK<CR>
```

Response, if successful:

```
GMO<CR>
```

Possible Error Response:

`DNS<CR>` ("Did not sleep": if not in Standby-Mode)

3.1.5. Read Input Pin State (RIP)

This command is used to read the current state of an input pin (if the reader has input pins). It takes one parameter, which is the hex-coded, zero-based number of the input pin to be read. The possible parameter range depends on the number of input pins of the reader.

If successful, it returns either HI! or LOW depending on whether the input pin is high or low.

Instruction:

```
RIP<SPACE>Pin_No<CR>
```

e.g. (to read the first input pin): RIP 00<CR>

Response, if successful:

`HI!<CR>` for High-State

`LOW<CR>` for Low-State

Possible Error Response:

`NOS<CR>` (Not Supported (DeskID_UHF has no GPIOs))

3.1.6. Write Output Pin State (WOP)

This command is used to set the state of an output pin either to high or to low (if the reader has output pins). It takes two parameters. The first parameter is the hex-coded, zero-based number of the output pin to be written to. The second parameter is either "HI" or "LOW" to set the according pin to high or low respectively. The possible parameter range is reader dependend.

Instruction:

```
WOP<SPACE>Pin_No<SPACE>PIN_Setting<CR>
```

e.g. Set pin 0 high: `WOP<SPACE>00<SPACE>HI<CR>`

e.g. Set pin 0 low: `WOP<SPACE>00<SPACE>LOW<CR>`

Response, if successful:

`OK!<CR>`

Possible Error Response:

`NOS<CR>` (Not Supported (DeskID_UHF has no GPIOs))

3.1.7. Read Reflected Power (RRP)

This command is used to read the reflected power value from the reader IC. It returns a complex number as two decimal coded values (A-Channel and B-Channel), which correspond to the i and q value of the reflected power.

The Pin value is computed as follows:

$$Pin(dBm) = 20 \cdot \log\left(\frac{\sqrt{A^2+B^2}}{G}\right) dBm$$

with G depends on the used chip and electronic environment. For the PulsarMX it is about 7, for the DeskID UHF about 10 and for the DwarfG2 about 2. The used power level is about 12dBm for Pulsar, DwarfG2 and DeskID UHF.

Instruction:

```
RRP<CR>
```

Response:

```
A<SPACE>B<CR>
```

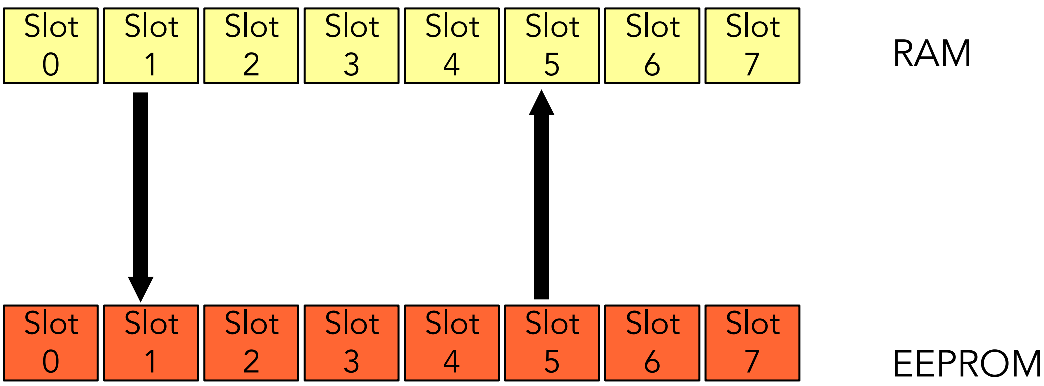
3.2. *Reader Configuration Instructions*

The Reader Configuration Instructions include all the commands that configure the behavior of the reader for all following commands. Usually they are set once at the beginning of every program and remain unchanged afterward (except when switching between different types of tags).

3.2.1. Config Profile Command (CFG)

The reader can be configured in great detail with regard to communication parameters. These options can be used to get the optimal performance from the device for each type of tag, antenna and general situation. Each set of configuration options make up a profile.

There are eight profiles stored in the internal RAM memory of the reader. They can be changed using the commands named in this chapter. All eight profiles also have a matching eeprom memory slot. This can be used to save profiles and reload them after a reset by hardware reset or by RST command. After every reset the profiles will be reset to their default values (not the EEPROM value).



There will always be at least one profile active. If all profiles become inactive by any user command (e.g. CFG PRO OFF ALL) profile 0 will become active.

NOTE

The reader ships with a set of useful default profiles. If you get a good read performance there is no need to change any profile.

It is possible to activate more than one profile at one time and also to use tag manipulation commands with more than one. This will cause the reader to execute the given command (for example INV) for every profile. The tags found may differ depending on the used parameters in the active profiles. Low power profile will not find tags far away for example.

The answers will look each like with one profile except:
Starting with "PROFILE X:<CR>"
Only the last profile answer will end with <LF> if EOF is set.
Example:

```
CFG<SPACE>PRO<SPACE>OFF<SPACE>ALL<CR>

//This will activate only profile 0

CFG<SPACE>PRO<SPACE>ON<SPACE>1<CR>     //profile 1 active

EOF<CR>  //activate linefeed

INV<CR>
```

Response:

```
OK!<CR>

OK!<CR>

OK!<CR><LF>

PROFILE 0:<CR>

E0000001<CR>        //Tag EPC

E0000002<CR>        //Tag EPC

IVF 002<CR>

PROFILE 1:<CR>

E0000003<CR>        //Tag EPC

E0000001<CR>        //Tag EPC

IVF 002<CR><LF>
```

3.2.1.1. Print Profile (PRP)

The Print Profile command will show the following parameters.

Name	Command /Short	Range	Explanation
Active Profile	PRO	ON/OFF	The profile will be used to find a tag. You can use any profile combination. Please keep in mind: using multiple profiles increases the time needed for any tag command. Also see SET GTO command in this case. After a PRP command this entry also shows the profile number (e.g. PRO 1 ON if profile one is shown and it is current active)
Rx Wait Time	RWT	0-255	Time to wait before Receiver activated. Multiplier is 6.4 µs
No Response Wait Time	NRW	0-255	Time to wait before a tag is "not answering" so the next slot will be used. Tag dependent, should be as low as possible for the tag(s) in use. Multiplier is 25.6 µs
Rx Gain	RXG	-3 to +3	The receiver gain. Multiplier is 3dB
Digitizer Hysteresis Setting	DHS	0 - 7	Hysteresis of the digitizer, multiplier is 3 dB. Testings result in 0-2 being the most useful settings.
Low Pass Filter	LPF	0 - 7	Depends on Link Frequency. May be 0 for 640kHz, 4 for 320 kHz, 6 for 256 kHz and 7 for 160kHz and 40kHz
High Pass Filter	HPF	0 - 7	Depends on Link Frequency. For 320 kHz values from 0 to 4 work best for most tags
TRCal	TRC	0-4095	Multiplier is 0.1µs
TxPower	PWR	Reader dependent	The Tx Power in dBm, DeskID UHF and DwarfG2 range -2 to 17 dBm, PulsarMX range 12 to 27 dBm.
Rx Encoding	MIL (for Miller)	0 - 3	0: FMO, 1: MILLER2, 2: MILLER4, 3: MILLER8 MILLER8 is usually best on tags supporting MILLER8
Tari	TAR	1 - 2	1: Tari=12.5µs , 2: Tari=25µs
Rx Link Frequency	LKF	0 – 15, but not all	0: 40kHz, 6: 160kHz, 9: 256kHz, 12: 320kHz, 15: 640kHz
Differential Mixer Gain Increase	DMG	ON/OFF	10 dB Gain if ON
RX Settling Speed Up	SSU	ON/OFF	Should make reading a bit faster for most tags
Mixer Input Attenuation	MIA	ON/OFF	5dB Attenuation on DwarfG2 and DeskID UHF, 8 dB on PulsarMX
PR-ASK	PAS	ON/OFF	PR-ASK OFF: Modulation is ASK, ON: Modulation is PR-ASK

Any Profile can be active or inactive. Use more than one profile at once if you have different tag types at one reader or to get better reliability.

Instruction:

```
CFG<SPACE>PRP<SPACE><SLOT><CR>
```

Response:

```
OK!<CR> <PARAMETER>
```

Example:

Show profile 1

```
CFG<SPACE>PRP<SPACE>1<CR>
```

Response:

```
OK!<CR>

PRO 1 OFF<CR>

DMG OFF<CR>

MIA OFF<CR>

SSU ON<CR>

PAS OFF<CR>

MIL 03<CR>

TAR 01<CR>

LKF 12<CR>

TRC 0667<CR>

NRW 11<CR>

RWT 01<CR>

LPF 04<CR>

HPF 00<CR>

PWR  09<CR>

DHS 00<CR>

RXG -2<CR>
```

3.2.1.2. Save Profile to EEPROM (SPE)

There are eight profiles stored in the internal RAM of the reader. Any Profile can be saved to EEPROM to be used after a reset.

Instruction:

```
CFG<SPACE>SPE<SPACE><SLOT><CR>
```

Response:

```
OK!<CR>
```

Example:

Save profiles 0

```
CFG<SPACE>SPE<SPACE>0<CR>
```

3.2.1.3. Load Profile from EEPROM (LPE)

Use the LPE command to load a profile from the EEPROM to the RAM of the reader.

Instruction:

```
CFG<SPACE>LPE<SPACE><SLOT><CR>
```

Response:

```
OK!<CR>
```

Example:

Load profile 7

```
CFG<SPACE>LPE<SPACE>7<CR>
```

3.2.1.4. Hardware Communication Parameters / Profile Elements

Any profile element can be changed by using the command printed by the CFG PRP command or the command given in the overview.

Instruction:

```
CFG<SPACE><PROFILEPARAMETER><SPACE><VALUE><SPACE><SLOT><CR>
```

Response:

`OK!<CR>`

Example:

Set RxGain to -3 for all profiles

`CFG<SPACE>RXG<SPACE>-3<SPACE>ALL<CR>`

Activate Profile 1

`CFG<SPACE>PRO<SPACE>ON<SPACE>1<CR>`

3.2.2. Settings Command (SET)

The settings command lets you configure the behaviour of the reader in a number of ways. If successful it returns OK! All settings are reset with reader reset.

3.2.2.1. MSK Parameter (Mask)

Most tags manipulation instruction can be limited to a population of tags with certain data values, e.g. tags that start with a certain EPC, a certain TID or even contain certain data in the user memory. This is done via a mask given with each command. Using this feature you can address certain tags in the field with directly accessing each tag via its TID or EPC.

This is especially handy if you only want to read from or write to tags of a certain product type (coded in the EPC).

To use this feature you have to tell the reader by which data field you want to filter and what the starting address should be to compare your mask. The mask is always given in full bytes. By default the mask is not set, so all tags will answer to your commands.

Possible Flags:

- OFF – No mask is set and all tags will answer (Default)
- EPC/TID/USR this Flags use the corresponding membank as masking target
- Mask Value (Hexadecimal, max. 31Byte)
- Optional: Start Address in Bit (hex, max unsigned32). Default = 0
- Optional: Length in Bit (hex coded. Max is 31 Byte=0xF8 Bits). Default = Length of Mask Value (full Nibbles). If a Length is given the [Length] first bits will be used. This is only useful if you want to use a mask that is not a full word length.

```
SET<SPACE>MSK<SPACE>Membank<SPACE>MaskValue<SPACE>Optional
Start<SPACE>Optional Bit Length<CR>
```

Response:

```
OK!<CR>
```

Example:

Kill all tags that have FF at the second byte of their EPC (and the same kill password)

```
SET<SPACE>MSK<SPACE>EPC<SPACE>FF<SPACE>08<CR>
```

```
KIL<SPACE>KILLPASSWORD<CR>
```

Inventory all tags that have 101 at the start of the forth nibble (begin with bit twelve (0x0C), length is 3 bit) of their TID membank.

```
SET<SPACE>MSK<SPACE>TID<SPACE>A0<SPACE>0C<SPACE>03<CR>
```

```
INV<CR>
```

3.2.2.2. ACP (AccessPassword), APS (AccessPasswordSave), APL(AcPwLoad)

Use this parameters to set the access mode. If access is „OFF" all tags with an access password other than zero will be in open state (instead of secured state) so some commands might not work. For further details on this topic please refer to the EPC Gen 2 Protocol Description and the lock command. ACP with password and APL command will activate access/secured mode. APS save the given password to EEPROM and will not change the password in RAM, APL loads the password to the RAM (same result as ACP XXXXXXXXXXXXXXXX). The Eeprom load command is also useful for higher security as the password is not send via Uart. There are eight slots in EEPROM to save access passwords.

```
SET<SPACE>ACP<SPACE>OFF<CR> (default)
```

```
SET<SPACE>ACP<SPACE>xxxxxxxx<CR>
```

```
SET<SPACE>APS<SPACE>xxxxxxxx<SPACE><SLOT><CR>
```

```
SET<SPACE>APL<SPACE><SLOT><CR>
```

Response:

```
OK!<CR>
```

Example:

```
SET<SPACE>ACP<SPACE>01234567
```

```
SET<SPACE>APS<SPACE>01234567<SPACE>1<CR>
```

```
SET<SPACE>APL<SPACE>1<CR>
```

3.2.2.3. KLP (KillPassword), KPS (KillPassword Save), KPL(KillPwLoad)

Use this parameters to set the kill password. For details on this topic please refer to the EPC Gen 2 Protocol Description and the kill command. KPS saves the given password to EEPROM and will not change the password in RAM, KPL load the password to the RAM (same result as KLP XXXXXXXXXXXXXXXX). The Eeprom load command is also useful for higher security as the password is not send via Uart. There are eight slots in EEPROM to save kill passwords.

```
SET<SPACE>KLP<SPACE>xxxxxxxx<CR>
SET<SPACE>KPS<SPACE>xxxxxxxx<SPACE><SLOT><CR>
SET<SPACE>KPL<SPACE><SLOT><CR>
```

Response:

```
OK!<CR>
```

Example:

```
SET<SPACE>KLP<SPACE>DEADC0DE<CR>
SET<SPACE>KPS<SPACE>DEADC0DE<SPACE>1<CR>
SET<SPACE>KPL<SPACE>1<CR>
```

3.2.2.4. EPC (EPC added)

By setting this parameter the user activates the EPC-ADD mode. If active, every tag command will add the EPC after the answer (read data, „OK!“, etc.) of the function but before RSSI (SET TRS ON). This makes it easier to identify which tag is actually responding to a command like e.g. Write Data or Read Data.

```
SET<SPACE>EPC<SPACE>OFF<CR> (default)
SET<SPACE>EPC<SPACE>ON<CR>
```

Response:

```
OK!<CR>
```

3.2.2.5. GTO (Global Timeout)

Any tag command (INV, RDT etc.) starts a global timeout. If the function does not terminate – either successful or returning an error – the function will be killed and TOE error code printed. SET GTO changes the timeout value. It is given in decimal milliseconds.

```
SET<SPACE>GTO<SPACE>400<CR> (default for DwarfG2 and DeskID_UHF)

SET<SPACE>GTO<SPACE>1000<CR> (default for Pulsar_MX)

SET<SPACE>GTO<SPACE>1800<CR> (max value)

SET<SPACE>GTO<SPACE>100<CR> (suitable for many tag types if only
one or two tags are in field (and Q-Value and IR-Value are low)
```

Response:

```
OK!<CR>
```

3.2.2.6. TRS Parameter (Transponder Receive Signal)

Sometimes you want to know the received signal strength when communicating with a transponder. With the TRS setting the reader will automatically add the RSSI to responses from a tag. The value is always negative in a range from -25 to -70 with -25 being the best case (strongest signal). The value will be printed in a new line following the answer of the tag operation for each tag and the EPC if "SET EPC ON" is set.

Possible Flags:

- OFF – No RSSI values given (default)
- ON – A RSSI value is added to every response that communicates with a tag

Instruction:

```
SET<SPACE>TRS<SPACE>Flag<CR>
```

Response:

```
OK!<CR>
```

Example:

```
INV<CR>
```

Answer with TRS OFF:

```
0011223344556677889910011<CR>

00112233445566778899AABB<CR>
```

```
IVF 002<CR>
```

The same Answer with TRS ON:

```
0011223344556677889910 11<CR>
```

```
-48<CR>
```

```
00112233445566778899AABB<CR>
```

```
-56<CR>
```

```
IVF 002<CR>
```

3.2.3. Mode Command (MOD)

MOD is the base command for the choosing of the Rf-Standard used by the reader. More mode parameters for other regions might be added in later versions of the firmware.

NOTE

You ***have*** to set the regional standard before you can start any operation that activates the RF field to be sure to comply with local frequency regulations.

Currently the only valid parameter and option for the Mode command is:

- STD – "Standard": selects a regional standard: "ETSI"

On PulsarMX the MOD STD ETSI command will check also if a matching antenna is attached to the reader. If not the RF LED will start to blink and ARH error code is sent. The blinking LED will switch to normal after resending MOD STD ETSI with a working antenna connected to the reader. The antenna check may be overridden (if an working antenna is attached which reflective values are not as expected) by adding AWI (antenna warning ignore).

Instruction:

```
MOD<SPACE>STD<SPACE>ETSI<CR>
```

```
MOD<SPACE>STD<SPACE>ETSI<SPACE>AWI<CR>
```

Response, if successful:

```
OK!<CR>
```

Error Response:

```
ARH<CR>, NOS<CR>
```

3.2.4. Set Q-Value (SQV)

The Set Q-Value Command is used to change the used Q-value. The Q-Value indicates the starting number of Slots for the tag searching used in every tag command. The number of slots is 2^Q, so Q=0 is used for only one tag (see also SSL, which will set Q=0 temporarily). The maximal value is 15, even though this will result in a timeout error normally. The default value is four (16 Slots) for DeskID_UHF and DwarfG2 and 6 (64 slots) for Pulsar_MX, which is fine for up to 8 tags / 50 tags. For more tags change Q. Lower Q-Values will fasten the search, so for 2 Tags Q=2 will be fine (in general: the number of channels should be much bigger (about two times bigger) than the expected number of tags. The value is given as decimal number.

Instruction:

```
SQV<SPACE><Q-Value><CR>
```

```
Q-Value: 0<=Q<=15
```

Response:

```
OK!<CR>
```

3.2.5. Read Q-Value (RQV)

Gets the Q-Value set by SQV command as two-digit decimal number. For more information look at SQV command.

Instruction:

```
RQV<CR>
```

Response:

```
<Q-Value><CR>
```

3.2.6. Set Inventory Retry (SIR)

Sets the number of retries in tag searching algorithm. Depending on the number of tags in the field and the number of slots (Q-Value) there is the chance of a tag collision. Also, the tag detection communication might be corrupted. Both cases leave a tag undetected but in detectable state (arbitrate). In a new round the tag might be found. IR-Value sets how often a new round can is started (it will not if there is no sign of a missing tag at all). The default value is 2 for DwarfG2 and DeskID_UHF and it's 3 for Pulsar_MX.

Instruction:

```
SIR<SPACE><IR-Value><CR>
```

```
IR-Value: 0<=IR<=10
```

Response:

```
OK!<CR>
```

3.2.7. Read Inventory Retry (RIR)

Gets the IR-Value set by SIR command as a two-digit decimal number. For more information look at SIR command.

Instruction:

```
RIR<CR>
```

Response:

```
<IR-Value><CR>
```

3.2.8. Set RF-Interface (SRI)

The Set RF-Interface Command is used to control the RF Output of the reader. This command is used to switch the RF field on (the reader manages frequency hopping automatically to conform with all ETSI regulations) or off (to reset tags e.g.).

Valid parameters for the command are:

- ON – Turns the RF field on
- OFF – Turns the RF field off
- TIM<SPACE> [Time in ms] – Timer controlled RF field reset. Turns the field off, waits for the specified number of ms and then turns the field back on. Can be useful to reset all tags in the field without managing everything. Time is given as decimal value.
- SPM<SPACE><ON/OFF> - SavePowerMode switched on (default on DwarfG2) or off (default for all other UHF-Reader). This mode will switch off the power amplifier automatically after every tag operation starts (either directly user called or CNR-mode) reducing the power consumtion nearly as much as STB but does not need to be woke up by WAK command. Also all reader commands are usable with disabled amplifier.

Instruction:

```
SRI<SPACE>Parameter[in case of TIM:<SPACE>Value, SPM:
<SPACE>ON/OFF]<CR>
```

Response, if successful:

```
OK!<CR>
```

Examples:

Turn RF field off:

`SRI<SPACE>OFF<CR>`

Turn SPM on:

`SRI<SPACE>SPM<SPACE>ON<CR>`

Possible Error Response:

`NSS<CR>`

3.2.9. Cyclic Redundancy Check On (CON)

This commands turns on the Cyclic Redundancy Check (CRC) of the computer-to-reader communication. This is used to detect transmission errors between the reader and the host (computer, embedded PC, microcontroller, etc.). In general this feature is not necessary except in scenarios where you have noise on the communication bus (e.g. when using USB communication in the vicinity of electric motors) or you encounter any other problems with communication errors.

If this feature is activated (default is off), the reader firmware expects a CRC16 (4 hex numbers) between all commands to the reader and the respective <CR>. Between the command and the CRC there is a space character which is included in the CRC calculation. All answers from the reader will also be extended accordingly. The CRC used uses the 0x8408 polynomial, starting value is 0xFFFF. This command will work with or without the (optional) CRC.

The command returns OK! plus the according CRC of "OK! ".

Appendix 1 shows a function in C, C# & Java to calculate the correct CRC16.

Instruction:

`CON<CR>`

or:

`CON 819E<CR>, con 2EC5<CR>`

Response:

`OK! 9356<CR>`

3.2.10. Cyclic Redundancy Check Off (COF)

This command turns off the Cyclic Redundancy Check (CRC) of the computer-to-reader communication. This is the default setting. The command will only work in CRC mode if a CRC is added obviously.

Instruction:

```
COF<CR>
```

Response, if successful:

```
OK!<CR>
```

3.2.11. End of Frame Mode (EOF)

This command turns on the End of Frame Delimiter (EOF). This means that after every complete message (frame) the last CR will be followed by an additional line feed (LF, 0x0A). This allows the user to build simpler parsers since it is clear when no to expect any further message from the reader. The EOF returns on the end of any answer including any CNR mode answer.

Please keep in mind: In case of a watchdog reset you get a SRT error code after the reset. This SRT is without the EOF because of the reset!

Instruction:

```
EOF<CR>
```

Response, if successful:

```
OK!<CR><LF>
```

3.2.12. No End of Frame Mode (NEF)

This command turns off the End of Frame Delimiter (See EOF command for more informations). Now all messages from the reader are only signaled by a CR at the end.

Instruction:

```
NEF<CR>
```

Response:

```
OK!<CR>  (no <LF>)
```

3.2.13. Verbosity Level (VBL)

The VBL command gives you the possibility to set the amount of data coming back from the reader to the level you need. The default level is 1, 0 reduces the informations, 2 increases them.

Instruction:

```
VBL<SPACE>[Level]<CR>
```

```
0<=Level<=2
```

The different possible levels are:

0: Any Tag answer except the correct answer is suppressed. No tag manipulation error codes are given. The number of tags (IVF XXX) is not send. Parsing and hardware errors remain unchanged.

1 (default): Default, secured tag communication errors are in.

2: Additionally, all tag communication answers including RXE and CRE normally indicating a collision or TNR (indication no Tag in this slot) for all slots in every round are send additionally to the answers in VBL=1. Also, the answer on NAK and Select are printed. This is nearly unparsable and should only be used for debugging purposes. For high Q-values or high IR (inventory retry) the number of answers might be more than 1000! for a single INV. Many intrnal errorcodes printed in VBL=2 are not named at the command description.

3.2.14. Read Serial Number (RSN)

Returns the serial number of the reader. The serial number is an ASCII string of 16 characters.

Instruction:

```
RSN<CR>
```

Answer:

```
XXXXXXXXXXXXXXXX<CR>
```

3.2.15. Read Hardware Revision (RHR)

Returns the hardware revision of the reader which corresponds to the PCB layout version printed on the board. The number is an ASCII string of four characters.

Instruction:

```
RHR<CR>
```

Answer:

```
XXXX<CR>
```

4. Tag Manipulation Instructions

This list gives an overview of the existing Tag Manipulation Instructions.

Command	Name	Description
INV	Inventory	Sends the Inventory request to the tag(s)
RDT	Read Direct	Read data from tag (EPC, user memory, TID, etc.)
WDT	Write Direct	Write data to tag (EPC, user memory, TID, etc.)
LCK	Lock	Locks a specific area of the tag memory
KIL	Kill	"Kills" the tag
DRC	Direct Command	Allows to send a free configured command to the open or secured tag. Handle can be placed freely
CNR	Continuous Repeat	Prefix for continuous / automatically repeated requests. Usable with all commands in this table except CNR and BRK)
BRK	Break	Stops continuous operation (ends a CNR prefix operation)

Every Reader instruction can be called with a masking before (except CNR and break, which are no direct instructions but commands to use/stop the continuous repeat mode (CNR). CNR-ed commands will use the mask every time they are executed. Every tag command supports the SSL parameter which sets the "Q"-value to 0 and IR-Value to 0, for fastest read rates when using only a single tag. SSL is identical to using SQV 0 and SIR 0, but only for one command / the full CNR time. The „old" value remains.

The difference between Reader Instructions and Tag Instructions is: The target of the instruction is the reader or the tag itself. Since RFID is mostly about tags and the data stored on tags, the Tag Manipulation Instructions are used extensively in almost any program. Among these commands the CNR command has a special role, since it is not strictly a command by itself but a prefix which changes the way the command following CNR is interpreted by the reader.

Every Tag command (except BRK) will end with IVF XXX. IVF stands for Inventory Found. It's the number of tags found – as the name suggests. The IVF value is decimal coded, always 3 characters long and does not exceed 250. If a tag is found the Detect LED is switched on at this moment for 500ms. The timer will restart if another tag is found.

Any command might get a TOE error (see SET GTO command) in case of pending instruction cause by wrong, missing, or unexpected answers in UHF communication steps or just longer lasting command than timeout is set (for example INV with high Q-Value and /or IR-Value.

4.1. Inventory (INV)

The Inventory command looks for all tags in range of the reader and sends the EPCs of all tags as a number of strings back to the host. To detect multiple tags at once it uses the anti-collision algorithm defined by the EPC UHF Class 1 Gen 2 Specification. This is the most common command for any UHF RFID application. Using the CNR prefix (see below) this can be used repeatedly to find tags continuously.

The length of the answer (the length of the EPC) is decided by the Protocol Control (PC) as defined by EPC Class 1 Gen 2 Protocol.

Instruction:

```
INV<SPACE>[opt. SSL]<CR>
```

Response, if successful:

for 0 Tags: `nothing (except IVF 000<CR>)` (IVF = Inventory Found)

for 1 Tag: `001122334455<CR>` (EPC with a length of 3 words)

for 2 Tags:

```
0011223344556677889910 11<CR>
00112233445566778899AABB<CR>
```

Possible Error Responses:

```
NSS<CR>, TMT<CR>, FLE<CR>, TOE<CR>
```

4.2. Read Data from Tag Command (RDT)

To read data from the tag the Read Data from Tag Command is used. The data can be just the EPC, data from the user memory some tags have or even the access or kill password (if they are not locked against reading). You can read out only one types of memory (EPC, TID, Passwords, etc.) at the same time. You always read multiples of two bytes (one word of 16 bit).

It is possible to mask this command (using the SET MSK command before) to limit this command to a certain population of tags. This command can be combined with the CNR flag.

The IVF value is decimal coded, always 3 characters long and does not exceed 250. It is the number of found tags (not successful operations like in former descriptions).

Possible memory types:

- EPC – The epc membank. Contains CRC, PC and EPC.

- RES – Reserved membank. Contains Kill password and Access password
- TID – The tag ID of the tag (sometimes contains a unique ID, sometimes only a manufacturer code, depending on the tag type)
- USR – The optional user memory some tags have

After the memory type you have to specify the start address you want to start reading from in words (two bytes) in hexadecimal as well as the number of words (also in hex).

Instruction:

```
RDT<SPACE>[opt. SSL]<SPACE>Memory Type<SPACE>Start
Address<SPACE>No. of Words<CR>
```

Response, if successful:

`001122334455<CR>` (some data from the first tag)

`AABBCCDDEEFF<CR>` (some data from the second tag)

Examples:

1. Reading the first 3 words (4 bytes) of the EPC of the tag. Only EPCs starting with AB shall answer using the Mask Command. The EPC starts at word two (Word 0 is CRC, Word 1 is PC), 0x20 is the hex bit address for masking. Don't forget: Masking settings will not reset after the end oft RDT!

```
SET<SPACE>MSK<SPACE>EPC<SPACE>AB<SPACE>20<CR>
```

```
RDT<SPACE>EPC<SPACE>2<SPACE>3<CR>
```

2. Reading the entire TID without mask (According to EPC Gen2 a length of zero equals "All". This can be used, however the data length will often be not correct as it is unknown. Post-pulse coscillation and noise will often cause additional data of unknown length. If masking is off the SET MSK OFF command is not necessary:

```
SET<SPACE>MSK<SPACE>OFF<CR>
```

```
RDT<SPACE>TID<SPACE>0<SPACE>0<CR>
```

3. Reading 8 words (16 byte) from the user memory starting at the second word (since it is zero based you have to use the value 1) as well as the kill password:

```
RDT<SPACE>USR<SPACE>1<SPACE>8<CR>
```

Possible Error Response:

```
NRF<CR>, TCE<CR>, RXE<CR>, PDE<CR>, HBE<CR>, HBE XX<CR>,
CER<CR>, RDL<CR>, TOR<CR>, FLE<CR>, TOE<CR>
```

4.3. Write Data to Tag Command (WDT)

To write data to the tag the Write Data to Tag Command is used. This data can be the EPC, data from the user memory some tags have or even the access or kill password (if they are not locked against writing). Don't forget to re-calculate their new CRC16 by writing it to the tag or by re-powering the tag (move it out of the field or switch it off). Without this all following inventory operations would result in CRC errors.

It is possible to mask this command (using the SET MSK command before) to limit this command to a certain population of tags. If you do not use a MSK all tags in the field will execute the write command which might not be what you are expecting.

This command can be combined with the CNR flag, including its BAR option.

After the memory type you have to specify the start address you want to start writing from in words (two bytes) in hexadecimal and zero-based. Please keep in mind to write complete words.

Data can be multiples of one word. The reader will try to send all data word by word to the tag. If an error occurred it will tell you which word it occurred with and continue. Due to energy problems some tags are not able to send a correct answer during the timeout. So with some tags even an error-returning WDT command might have been successful. Please check by using the RDT command for these cases.

Instruction:

```
WDT<SPACE>[opt. SSL]<SPACE>Memory Bank<SPACE>Starting
Block<SPACE>[Data]<CR>
```

Response, if successful:

```
OK!<CR>       If a tag is correctly written
```

Examples:

1. Write AABBCCDD into the first two words of the EPC of the tag. The EPC starts at word two so the start address is 2. Only one tag should be in the field, so the SSL flag is used for faster execution of the command.

```
WDT<SPACE>SSL<SPACE>EPC<SPACE>20<SPACE>AABBCCDD<CR>
```

2. Writing 0011 into the third word of the user block of all tags in the field whose EPC starts with AABB:

```
SET<SPACE>MSK<SPACE>EPC<SPACE>20<SPACE>AABB<CR>
```

```
WDT<SPACE>USR<SPACE>02<SPACE>0011<CR>
```

3. Write AABB into the first word of the EPC of the tag and change the length of the EPC to one word only. The EPC starts at word two but the protocol control (PC) block starts at word one. In Bits the start address is 0x10. The First 5 Bits of the PC block are the length in words so the 0x0800 is required for one word EPC length assuming all remaining PC Bits are 0. Only one tag should be in the field, so the SSL flag is used for faster execution of the command.

```
WDT<SPACE>SSL<SPACE>EPC<SPACE>1<SPACE>0800AABB<CR>
```

```
SRI<SPACE>TIM<SPACE>100<CR>  //resets the tag, CRC is
recalculated
```

3. Alternative way to set length and ID:

```
WDT<SPACE>SSL<SPACE>EPC<SPACE>2<SPACE>AABB<CR>
```

```
SRI<SPACE>TIM<SPACE>100<CR>  //resets the tag, CRC is
recalculated
```

```
WDT<SPACE>SSL<SPACE>LEN<SPACE>1<CR>
```

```
SRI<SPACE>TIM<SPACE>100<CR>  //resets the tag, CRC is
recalculated
```

Possible Error Response:

```
NRF<CR>, TCE<CR>, RXE<CR>, PDE<CR>, HBE<CR>, HBE XX<CR>,
CER<CR>, RDL<CR>, TOR<CR>, FLE<CR>, TOE<CR>
```

4.4. Lock Command (LCK)

The Lock Command is used to set the access rights of the different data blocks, including the access password itself and the kill password. To use this command you have to be in the secured state.

It is possible to mask this command (using the SET MSK command before) to limit this command to a certain population of tags. In this case the filter value has to be given (full byte only).

This command can be combined with the CNR flag, including its BAR option.

Valid parameters for the command are:

- EPC – Set Lock State for the EPC Memory
- TID – Set Lock State for the TID Memory (usually read only anyway)

- USR – Set Lock State for the User Memory
- ACP – Set Lock State for the Access Password
- KLP – Set Lock State for the Kill Password

For each type of memory you can define the access mode. Their meaning differ depending on the first parameter. (See EPC Gen 2 Specification for more details).

For the passwords (ACP and KLP):

Mode	Description
0	Password location is readable and writable from either open or secured states
1	Password location is permanently readable and writable from either the open or secured states and may never be locked
2	Password location is readable and writable from secured state but not from open state
3	Password location is permanently not readable or writable from any state

For the memory banks (EPC, TID, USR):

Mode	Description
0	Memory bank is writable from either open or secured states
1	Memory bank is permanently writable from either the open or secured states and may never be locked
2	Memory bank is writable from secured state but not from open state
3	Memory bank is permanently not writable from any state

Instruction:

LCK<SPACE>[optional: SSL]<SPACE>Parameter<SPACE>Mode<CR>

Example:

1. Lock the EPC against changes from unauthorized users (open state) but still writable from the secured state.

LCK<SPACE>EPC<SPACE>2<CR>

2. Lock the kill password permanently against access for everyone:

LCK<SPACE>KLP<SPACE>3<CR>

3. Make the TID writable for everyone permanently:

LCK<SPACE>TID<SPACE>1<CR>

4. Lock the user memory against changes from anyone permanently:

```
LCK<SPACE>USR<SPACE>3<CR>
```

Response, if successful:

```
OK!<CR>
```

Possible Error Response:

```
NRF<CR>, TCE<CR>, RXE<CR>, PDE<CR>, HBE<CR>, HBE XX<CR>,
CER<CR>, TOR<CR>, TOE<CR>
```

4.5. Kill Command (KIL)

The Kill Command can be used to disable UHF Gen2 Tags forever. To do this the kill password is necessary (four bytes). The password is given via SET command or is loaded from eeprom via set command.

It is possible to mask this command (using the SET MSK command before) to limit this command to a certain population of tags.

This command can be combined with the CNR flag, including its BAR option. „SET EPC ON" command might be helpful to keep track of killed tags.

ATTENTION

If you use this command incorrectly (esp. in combination with the CNR flag) you can kill a very big number of UHF tags in a very short time ;-)

Instruction:

```
SET<SPACE>KLP<SPACE>XXXXXXXX<CR>
```

```
KIL<SPACE>[optional: SSL]<CR>
```

Response, if successful:

`OK!<CR>` One for every successful Kill

Possible Error Response:

```
NRF<CR>, TCE<CR>, RXE<CR>, PDE<CR>, HBE<CR>, HBE XX<CR>,
CER<CR>, TOR<CR>, TOE<CR>
```

4.6. Direct Command (DRC)

The direct command is used to access tags with optional or manufacturer specific commands.

It is possible to mask this command (using the SET MSK command before) to limit this command to a certain population of tags. This command can be combined with the CNR flag, including its BAR option.

ATTENTION

If you use this command incorrect, error or completely undefined states might be set. It might be not possible to reset the tag afterwards.

Instruction:

```
DRC[optional: <SPACE>SSL][opt.: <SPACE>CHB][<SPACE><Data 1
Hex>]<SPACE><Data1 bit length>[opt.:<SPACE>H[opt.: <SPACE><Data
2 Hex>][opt.:<SPACE><Data2 bit length>]][opt.: <SPACE>CRC]<CR>
```

Parameters are:

- SSL: Single Slot (sets Q value for this round to zero)
- CHB: Check Header Bit: If this parameter is set the first bit of the answer will be seen as header bit. If it is set (=1) the answer will get a HBE<SPACE> prefix. In any case the answer is moved by one bit. So the answer is much easier to read, especially for human.
- Data 1 is not optional
- Length of Data 1 in Bit is optional
- H (Handle) is optional
- Data 2 is optional and only allowed if handle is used
- CRC is optional (often needed as CRC depends on handle)

Response, if successful:

```
[<Answer Data in ASCII hex code>]<CR>
```

Possible Error Response:

```
NRF<CR>, TCE<CR>, RXE<CR>, HBE<CR>, HBE XX<CR>, CER<CR>,
TOR<CR>, FLE<CR>, TOE<CR>
```

4.7. Continuous Repeat Prefix (CNR)

To allow the repeated / continuous execution of commands, the “CNR” prefix was implemented in the firmware. Any tag (or reader) command can be written after the “CNR” prefix and will then be repeated indefinitely or until the “BRK” command is sent (see below). This is a very powerful mechanism for unassisted and repeated operations where the reader is initialized at the beginning and then repeats the command over and over. Examples for useful continuous operations are reading tag IDs, reading data from tags or even writing and locking data on tags continuously, e.g. in a printer.

CNR has one optional parameter. BAR (break at read) lets the CNR mode be stopped when at least one tag is found one time.

Example: Read all tag IDs repeatedly until stopped

Instruction:

```
CNR<SPACE>INV<CR>
```

Response (exemplary, with two tags in the field):

```
E0040100078E3BB0<CR>

E0040100078E3BB7<CR>

IVF 02<CR>

E0040100078E3BB0<CR>

E0040100078E3BB7<CR>

IVF 02<CR>

…
```

Additional Parameter: BAR (Break At Read)

To automatically break a continuous command with the first tag found use the BAR parameter. This saves having to use BRK when having found a tag. The break is acknowledged by „BRA". If other commands than INV are used keep in mind that an unsuccessful command may trigger the break anyway. The Tag may be found but an error occurs when executing the main command (read, write, kill etc.). The tag may not answer (TOR), may return an error (HBE XX) or an internal error occurs (e.g. RDL, FLE).

Example: Wait silently for a tag to enter the field, report its ID and then stop. For this to be silent (no IVF 000 every try) , VBL should be set to 0 (see Verbosity Level).

```
VBL 2<CR>

CNR BAR INV<CR>
```

Response:

```
[TagID1]<CR>

[TagID2]<CR>
```

```
BRA<CR>
```

Example 2: With VBL 0 and an error like stated above occurs the answer might only be an BAR

```
 CNR BAR RDT<CR>
```

Response:

```
BRA<CR>
```

Example 3: With VBL 1 and an error like stated above occurs

```
 CNR BAR RDT<CR>
```

Response:

```
HBE 03<CR>     //memory address not define

BRA<CR>

IVF 001<CR>
```

4.8. *Break Command (BRK)*

To end the continuous mode entered into by the "CNR" prefix, the break command can be sent. This will lead to the complete execution of the current command iteration and will then lead to a "BRA" (break acknowledged) response. The command needs no parameter and returns no error codes.

Instruction:

```
 BRK<CR>
```

Response, if successful:

```
 BRA<CR>
```

Possible Error Response:

```
 NCM<CR>
```

5. Definition of Error Codes

5.1. Transceiver Errors

Error Code	Name	Description
ARH	Antenna Reflectivity High	Reading the analog value of the answer to a raw send (no command) gives a value higher than it should be with a matching antenna. Often caused from missing or mismatched antennas.
BOD	BrownOut detected	The micro controller detected a brownout. This is a hardware error. If it occurs more than once in a special situation please contact metraTec for support.
BOF	Buffer Overflow	An UART buffer got an overflow error. The send buffer has 512 Byte to use, the receive buffer has 128 Byte.
CCE	Communication CRC Error	A CRC-Error has been detected while receiving a line from the Host system
CRT	Command Received Timeout	Parts of a command where received by the reader but the device never got a carriage return. Always close commands with <CR>.
DNS	Did not Sleep	A wake up (WAK) was sent although the reader wasn't in sleep mode
EDX	Error decimal expected	Parameter string cannot be interpreted as a valid decimal value. Common error source: Other character than '0' to '9'
EHF	Error Hardware Failure	The RF interface chip does not match or is damaged. Please try a full reset (power reset) and/or contact support.
EHX	Error hex expected	Parameter string cannot be interpreted as a valid hexadecimal number. Common error source: Other character than 0-F
NCM	Not in CNR Mode	A Break Command (BRK) was sent although the reader wasn't in CNR mode
NOR	Number Out Of Range	Parameter value exceed maximum value, minimum value or is no word (odd number of bytes)
NOS	Not supported	This command is not supported by this hardware.
NRF	No RF-Field active	Start the RF field using SRI command

Error Code	Name	Description
NSS	No Standard Selected	You did not set a regional standard before starting the RF field (SRI, INV, RDT, etc.). Use MOD STD first.
PLE	PLL Error	PLL, RF level or crystal oscillator is not stable. The PLE error code is send on VBL=1 after resetting the air interface hardware and getting a "not stable" again. With VBL=2 PLE is send on the first occurrence, too. PLE can occur at every frequency hop so even if no command is send or even if the RF is off. The reader will retry getting PLL stable at the next hop (<4s with ETSI). If PLE occurs again try a hard restart (power off). If this does not work the hardware is likely to be damaged. In this case please call support.
SRT	Hardware Reset	In case of a critical error this error might occur. If you get this error frequently, your hardware is probably damaged.
UCO	Unknown command	An invalid command has been passed to a function. Common error source: - You have a typo in your command string - Wrong firmware version
UER	Unknown Error	Bad Interrupt occurred, unknown internal tag error code returned or another "default" case occurred.
UER XX	Unknown Error number	Internal error code reaches surface (XX = error code in hex)
UPA	Unknown parameter	An invalid or missing parameter has been passed to a function. Common error source: - You have a typo in your command string - A parameter is missing
URE	UART Receive Error	The hardware UART detected an error. Data might be corrupted. The whole packet will not be used. If the <CR> at the end of command was the corrupted byte the next command will be deleted, too.
WDL	Wrong Data Length	The given data is to long or short. This might occur on commands using data of variable length.

5.2. *Tag Errors*

Error Code	Name	Description
ACE	Access Error	The Access operation was not successful. The tags answer does not match. Might be a communication error or wrong password

Error Code	Name	Description
CER	CRC error	CRC16 from tag is wrong. Common error source: - The tag left the field or is too far away - Another devices disturbed the communication - Collision
FLE	FIFO Length Error	The FIFO contains too much data, data might be corrupted.
HBE(XX)	Headerbit Error	XX is the hex code of the error code given by the tag. See EPC Gen 2 Specification for further information. The defined error codes are: 0x00: Other Error (none matches) 0x03: Memory location does not exist or PC value not supported 0x04: Memory locked 0x0B: Insufficient power to write 0x0F: Nonspecific error (tag does not support error codes) If no error Code (XX) follows the answer had an error (CRC or handle do not match)
PDE	Preamble Detect Error	Common error source: - The tag left the field or is to far away - Another devices disturbed the communication - Collision - The tag is to near, overloading the tag or missmatching the antenna. This occurs especially with WDT command
RDL	Read Data to Long	The answer on RDT command from the tag is longer than the memory buffer reserved for receiving (120 Byte).
RXE	Response Count Expected Error	The answer was longer or shorter than expected. Common error source: - The tag left the field or is too far away - Another devices disturbed the communication - Collision
TCE	Tag Communicatio n Error	General Error during tag communication Common error source: Write command returned wrong check (handle). Data might be corrupted.
TMT	Too Many Tags	The reader found more tags than it can handle (max IVF 250).
TNR	Tag not responding	No answer on query. This only occurs if VBL=2

Error Code	Name	Description
TOE	TimeOut Error	The command timed out. No answer (including errors or local timeout) detected. This happens if some flow error occurred e.g. the transmit does not succeed or and hardware FIFO error occurs. The timeout is 250ms default and can be changed using SET GTO. The IVF gives the number of tags found til the Timeout occurred. The command gets canceled.
TOR	Tag Out of Range	The tag answered on query but the following command (for example read or lock) was not successful. This might indicate a range or access problem.

6. Appendix 1: CRC Calculation

```
// this function calculates a CRC16 over a unsigned char Array with, LSB first
// @Param1 (DataBuf): An Array, which contains the Data for Calculation
// @Param2 (SizeOfDataBuf): length of the Data Buffer (DataBuf)
// @Param3 (Polynom): Value of the Generatorpolynom, 0x8408 is recommended
// for ISO15693-3 and Communication-CRC
// @Param4 (Initial_Value ): load value for CRC16, 0xFFFF is recommended
// for ISO15693-3 and Communication-CRC

unsigned short GetCrc(        unsigned char *DataBuf,
                              unsigned char SizeOfDataBuf,
                              unsigned short Polynom,
                              unsigned short Initial_Value)
{
      unsigned short Crc16;
      unsigned char Byte_Counter, Bit_Counter;

      Crc16 = Initial_Value;
      for (Byte_Counter=0; Byte_Counter < SizeOfDataBuf; Byte_Counter++)
      {
            Crc16^=DataBuf[Byte_Counter];
            for (Bit_Counter=0; Bit_Counter<8; j++)
            {
                  if (( Crc16 & 0x0001)==0) Crc16>>=1;
                  else Crc16=(Crc16>>1)^Polynom;
            }
      }
       return (Crc16);
}
```

7. Version History

Version	*Change*	*Changed by*	*Date*
1.0	created	KD	16.10.2009
1.1	Major changes for FW Version 1.0	KD	17.09.2010
1.2	Some minor corrections and layout work	KD	06.10.2010
1.3	Added some clarifications	KD	15.10.2010
1.4	Profiles added, added commands (KLP, KLS,APS,APL..) and removed (HWO, OPS)	MK	22.02.2011
1.5	Errors removed, added some clarifications	MK	04.04.2011
1.5.1	Added SQV, RST clarification	MK	12.04.2011
1.6	Added BOD error code, changed UART to 115200 Baud (since Firmware Desk1.7,Dwarf1.5,Pulsar1.1)	MK	28.04.2011
1.6.1	RRP Constant changed for Pulsar+DwarfG2, added output power level Changed PulsarMX power range	MK	14.07.2011
1.7	Added DRC command	MK	16.09.2011
1.8	Added RQV, SIR, RIR, Removed EPH, Profile Description changed, some parameter max values changed, removed Parsing errors from command error descriptions. Changed G-Value for RRP	MK	17.11.2011
1.9	NOS no longer possible error on SRI SPM	MK	28.11.2011