# SOFTWARE DATA SHEET

## MCHD XX.X.X

## READER COMMUNICATION PROTOCOL
## SETUP COMMAND LIST

- Description of our **reader communication** protocol:
  - Message syntax between reader and connected equipment
    - Tag ID
    - Value of tag RSSi (Receive signal strength)
    - Reader ID
    - DATA from the sensor integrated into the tag
  - Operating mode
    - "ON LINE" mode (PUSH)
    - "CONTEXTUAL" mode (PULL)
  - How to add characters to message syntax

- Description of our **reader setting** commands:
  - Change PUSH or PULL Mode
  - Change Output type
  - Change RSSI threshold (distance detection)
  - Change Output Baud rates
  - Change reader ID
  - Tag Id filter
  - Activate automatic relay output
  - How to add characters to message syntax
  - …

- Reader types:

| Reference | Reader descriptionn |
|---|---|
| SCIEL_CARD | OEM Reader |
| SCIEL_READER_R | RS 232/ RS 485 / USB / D&C/ Wiegand reader |
| SCIEL_READER_USB | USB Reader |
| SCIEL_READER_IP | TCP/IP Reader |
| SCIEL_READER_WF | Wi-Fi Reader |

# 1. SUBJECT

This document describes the syntax of messages and commands used by ELA Innovation RFID readers (MCHDxx.xx versions).

## 1.1. Readers covered by this document

All ELA INNOVATION Active RFID reader series (MCHDxx.x.x versions).

## 1.2. Special cases of USB / TCP/IP / Wi-Fi readers

These readers don't have a direct RS-232 output. However, the concept of these readers is as follows:

- RFID hardware with a TTL 232 output common to all ELA Innovation readers.
- Connected to RS-232 converter hardware linked to another type of reader output (USB / TCP/IP / Wi-Fi).

The instructions described in this document can be used with all these readers. Some of the readers require a driver, which you will find on our website: http://www.rfid-ela.eu/

## 1.3. Default serial output parameters

| Speed | Dat | Stop | Parity | Hardware |
|---|---|---|---|---|
| 9600 | 8 | 1 | none | none |

**Table 1: default serial output parameters.**

Baud rates (9600bds by default) can be changed.

## 1.4. Supported commands and related firmware versions

| Read Param. | Write Param. | Description | Supported Firmware version | Default Value |
|---|---|---|---|---|
| 01 | 02 | Operating Mode (On-line (PUSH), Contextual PUSH or PULL) | 01 | On line |
| 03 | 04 | Output type (RS232, DATA/CLOCK, WIEGAND) | 01 | RS232 |
| 05 | 06 | Timing after Out-of-range detection | 01 | 6s |
| 07 | 08 | Tag list sending frequency in contextual PUSH mode | 01 | 60s |
| 09 | 19 | Reader ID | 06 | 01 |
| 0A | – | Sending Request of Contextual tag list | 01 | |
| 0B | 0C | RSSI threshold (Software Threshold) | 01 | 254 |
| 10 | 11 | Tag Frame Format (must be included in tag parameter) | 01 | 24 bits |
| 12 | 13 | RS232 speed | 06 | 9600 bds |
| 14 | 15 | RS232 message format (ASCII, binary) | 06 | ASCII |
| 16 | 17 | Acknowledgment delay | 06 | 0 ms |
| 1A | 1B | DATA/CLOCK output with or with alarm bits | 07 | OFF |
| 1C | 1D | Redundancy activation in "contextual" mode | 08 | OFF |
| 1E | 1F | Wireless protocol (must be included in tag parameter) | 08 | HD |
| 20 | 21 | Activation of tag filtering with the strongest radio level | 13.0.5 | OFF |
| 22 | 23 | NA | 13.2.2 | IR |
| | | Checksum length (must be included in the tag parameter) | 13.2.2 | 16 bits |
| | | "Direct Control of Relay" mode | 13.4.0 | OFF |
| | | Activation for sensor based tag filtering | 13.5.0 | OFF |
| | | Activation for automatic clearing of contextual stack after | 13.5.0 | OFF |
| | | Activation for replacement of the weakest or the oldest tag | 13.5.0 | ON |
| 24 | 25 | RW Protocol Bit length | 13.2.4 | 165 |
| 97 | – | Sending request of AD converter value – Radio noise level | 05 | |
| 98 | – | Sending request of firmware version | 04 | |
| 99 | – | Sending request of reader's configuration - On line Help | 04 | |
| – | 9A | Erase all tags in stack | 13.5.0 | |
| A0…A | A8…A | Tag ID filtering | 12c | OFF |
| C0 | C1 | Proprietary CKS or CRC key (must be included in the tag | 02 | 00 |
| 26 | 27 | Number of tag in stack for relay activation | 13.6.0 | 01 |
| C8 | C9 | NA (Calibration parameter of RSSI signal) | 13.6.2 | 00 |
| 2A | 2B | Modification of "Contextual" stack size | 30.0.0 | 20 |
| B0 | B1 | Activation and settings for a hardware analog RSSI | 30.0.3 | OFF |
| 70 | 71 | Add user characters before ELA messages | 30.0.4 | OFF |
| 72 | 73 | Add user characters after ELA messages | 30.0.4 | OFF |
| 2E | 2F | Tag stack format in contextual mode | 30.0.5 | 00 |
| 28 | 28 | Redundancy Timing | 30.0.5 | 00 |
| CA | CB | Half-duplex mode | 30.1.7 | 00 |
| 2C | CD | Radio statistics | 30.1.7 | |

## 1.5. Compatibility Matrix MCHD firmware versus SCIEL READER devices

| Reader type | Current Delivered Firmware version |
|---|---|
| SCIEL READER R | 30.1. |
| SCIEL READER RU | 30.1. |
| SCIEL READER RM | 30.1. |
| SCIEL READER R24 | 30.1. |
| SCIEL READER Lite | 30.1. |
| SCIEL READER SU | 30.1. |
| SCIEL READER IP | 13.6. |
| SCIEL READER IP2 | 30.1. |
| SCIEL READER WF | 13.6. |
| SCIEL READER WF2 | 30.1. |
| SCIEL CARD | 30.1. |
| SCIEL REPEATER | 13.6. |

# 2. COMMUNICATION PROTOCOL

## 2.1Message Syntax

The tag's identification is automatic and periodic: each tag is identified by the reader at each tag's transmission period: RFID tag identification is based upon the following data frame: [AAxxxxxxLL] with:

- "[" and "]" are used to delimit the frame
- AA encoded on 2 ASCII characters (or 1 binary byte in binary mode) defines a byte which corresponds to RFID tag wireless signal strength level (RSSI). This value varies between 110 (decimal), for a very close tag, and 200 (decimal), for a more remote tag (with respect to the reader).
- xxxxxx encoded on 6 (4,8) ASCII characters (or 3 binary bytes in binary mode) defines 3 bytes that correspond to RFID tag ID. The ID length is variable depending on the radio reception method (16, 24 or 32 bits).
- LL encoded on 2 ASCII characters (or 1 binary byte in binary mode) defines a byte that corresponds to reader ID.

*Example: reception of the following frames:* `[7801234501][948831A501]correspond successively to tag ID 012345 with a level of 78(hex) and tag 8831A5 with a level of 94(hex).`

Transmission mode (PUSH or PULL) for these ID frames can be set in "Operating mode":

*Exception:* `In "contextual" mode, when a tag disappears from stack memory, the message sent is: ]AAxxxxxxLL[`

**Starting with firmware version MCHD 30.0.1, a specific mode for sending compact frames is available. See "0A" order command.**

**Starting with firmware version MCHD 30.0.5, the user could add prefix and suffix which will be added to all the frames sent by the readers. See "70" to "73" order message.**

**Below, compatibility table for 24/32 bits:**

| Model | Reference | 24 bits | 32 bits | Remarks |
|---|---|---|---|---|
| THINLINE IR | IDP0231 | YES | YES | |
| THINLINE MOV | IDP0251 | YES | **NO** | |
| SLIM ID | IDF0348 | YES | **NO** | |
| COIN ID | IDF1034 | YES | YES | |
| COIN ID OEM | IDFOM34 | YES | YES | |
| COIN T | IDF1044 | YES | YES | |
| COIN T MiniLog | IDF1073 | YES | YES * | *24 bits : 8 bits ID - 16 bits DATA<br>32 bits : 16 bits ID - 16 bits DATA |
| COIN RH | IDF1050 | YES | **NO** | |
| COIN MOV | IDF1062 | YES | YES | |
| COIN MAG | IDF1064 | YES | YES | |
| ITEMS IR | IDF0431 | YES | YES | |
| ITEMS MOV | IDF0455 | YES | **NO** | |
| ITEMS EMOV | IDF0951 | YES | **NO** | |
| ITEMS TD | IDF0943 | YES | YES | |
| ITEMS AD | IDF0942 | YES | **NO** | |
| ITEMS DG | IDF0941 | YES | **NO** | |
| PUCK ID | IDF2573 | YES | YES | |
| PUCK ID OEM | IDFOM73 | YES | YES | |
| PUCK T | IDF2574 | YES | YES | |
| PUCK RHT | IDF2572 | YES | YES | |
| WATCH ID WHITE | IDF2735 | YES | YES | |
| WATCH ID BLUE | IDF2734 | YES | YES | |
| COIN ID Ex | IDF1036 | YES | YES | |
| COIN T Ex | IDF1037 | YES | YES | |
| COIN MOV Ex | IDF1061 | YES | YES | |
| COIN MAG Ex | IDF1040 | YES | YES | |
| TAG DOT | IDP2470 | YES | YES | |
| PUCK DOT | IDP2570 | YES | YES | |

## 2.2 DATA from sensor integrated in the Active RFID tag.

Some RFID tag contain an integrated sensor. In this case, the frame format is the same as that used by a tag that provides only identification information. Only 12-bit LSBs are reserved for the reading value.

***Example:*** *ID 801tag is a COIN_T type (with a Temp. sensor). Reception of the following frame: [7880112301] corresponds successively to tag ID 801 with an RSSI level of 78(hex)*

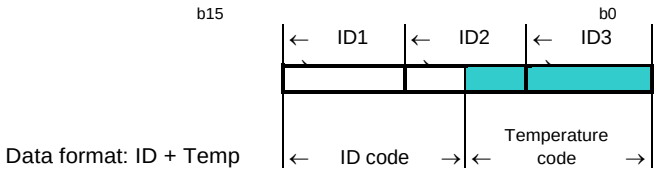
*and Temp. value of 123 (hex.). See product sheet COIN_T in order to calculate real values. (123(hex) = 291(dec.)= 18°C (291 * 0.0625°C).*

### 2.2.1 Temperature sensor

The value 7FF is reserved for alarm information about low battery level.
The Hex data is "2's complement signed data". The data must be converted into decimal format, signed and multiplied by the sensor resolution of 0.0625° in order to calculate the real value.

**Table of Active RFID max temperature coding**

| Temperature in ° C | | | Temperature code | | | | Conversion method |
|---|---|---|---|---|---|---|---|
| | | | Binary | | Hexadecimal | Decimal | |
| | | | b11 b10 b9 b8 | b7 b6 b5 b4 | b3 b2 b1 b0 | | |
| | | | S 6 5 4 | 3 2 1 0 | -1 -2 -3 -4 | | |
| Max. reading range for integrated sensor beyond COIN T specifications | Max + | 125.0000 | 0 1 1 1 | 1 1 0 1 | 0 0 0 0 | 7D0h | 2000 | **Positive temperatures**<br>[decimal value] /16<br>Example: 478/16 = +29.8750 |
| | | … | | | | … | … | |
| | | 29.8750 | 0 0 0 1 | 1 1 0 1 | 1 1 1 0 | 1DEh | 478 | |
| | | 5.0000 | 0 0 0 0 | 0 1 0 1 | 0 0 0 0 | 050h | 80 | |
| | | 0.0625 | 0 0 0 0 | 0 0 0 0 | 0 0 0 1 | **001h** | 1 | |
| | | 0.0000 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 000h | 0 | |
| | | -0.5000 | 1 1 1 1 | 1 1 1 1 | 1 0 0 0 | FF8h | 4088 | **Negative temperatures**<br>[2's complement 2] / 16<br>Example:<br>E6Fh ⇒ 190h (1's complement 1)<br>190h+1 ⇒ 191h (2's complement)<br>191h ⇒ 401 decimal<br>401/16 = -25.0625 |
| | | -10.1250 | 1 1 1 1 | 0 1 0 1 | 1 1 1 0 | F5Eh | 3934 | |
| | | … | | | | … | … | |
| | | -25.0625 | 1 1 1 0 | 0 1 1 0 | 1 1 1 1 | E6Fh | 3695 | |
| | | … | | | | … | … | |
| | Max - | -55.0000 | 1 1 0 0 | 1 0 0 1 | 0 0 0 0 | C90h | 3216 | |
| | | Alarm code for low battery (equivalent value: +127.9375°C) in alternated emission with ID code | 0 1 1 1 | 1 1 1 1 | 1 1 1 1 | 7FFh | *2047* | |



### 2.2.2 Relative humidity sensor

The value FFF is reserved for alarm information about low battery level.
The hexadecimal data is non-signed data. This data must be converted into a decimal value (V) and used in the following formula in order to get the value in humidity %.
%RH = - 2.0468 + 0.0367 V – 1.5955E-6 V²

*Example:*
*Value provided by the tag: V = 4DE (hexadecimal) meaning V = 1246 (decimal)*
%RH = - 2.0468 + 0.0367 V – $1.5955^{E-6}$ V² *meaning %RH = 41.19 %*

### 2.2.3 Movement sensor

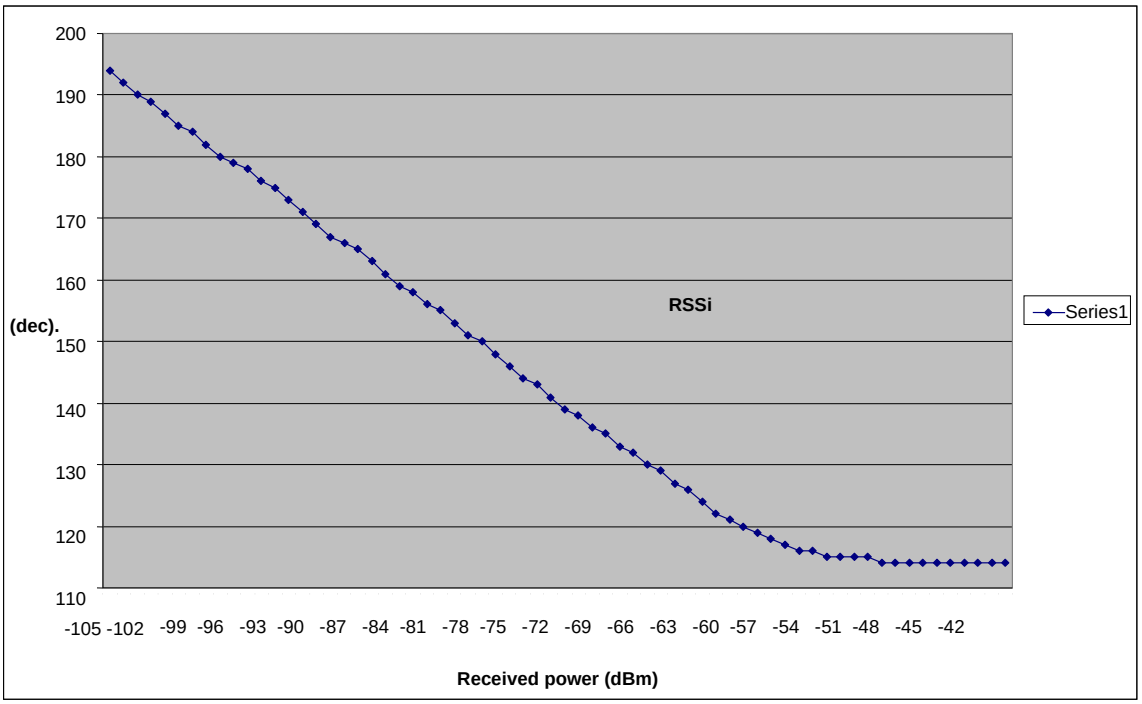
The value FFF value is reserved for alarm information about the low level of battery.
The hexadecimal data is a non signed data. This data is a relative value and is not linear to the movement: FFE for significant movement; 000 for no movement or less than the sensor's sensitivity.

### 2.2.4 Analog input

The value FFF is reserved for alarm information about low battery level.
The hexadecimal data is non-signed data. This data must be converted into a decimal value (V) and used in the following formula in order to get the value in Volts.
Van = 3*V/4096

## 2.3Tag RSSI level

The tag's Received Signal Strength Indication is always returned and associated with the tag's ID. The next table shows the variation between the information (in decimal) returned by the receiver to the connected equipment in function of the RSSI level expressed in dBm.




## 2.4"Online" or "Contextual" Mode

- **"ON LINE" mode**: this is a PUSH mode between reader and the connected equipment. As soon as a tag's identification frame is received, the reader transmits this information on the RS232 port. This emission is always repeated as long as the tag remains in the reader's detection area, with the RFID's tag emission cycle.

- **"CONTEXTUAL" mode:** this is a PUSH mode between reader and the connected equipment. Each tag's ID is memorized in a stack call "Contextual Stack". ID information is transmitted by the reader only 2 times: when the tag is identified for the first time (the tag enters in the reader's detection area) and when the tag leaves this detection area. The tag will be identified one more time, only after being gone out of the detection area during a time greater than the user-defined "TIME OUT" or "Exit Timing" (see paragraph on these points) AND after new detection in the area. Note that string delimiters are inverted in order to indicate output from the stack: **]**AAxxxxxxLL**[**
- **"PERIODIC CONTEXTUAL" mode:** this mode is similar to the "CONTEXTUAL" mode, but the list of tags in the "Contextual stack" is pushed with periodic cycle. This parameter can be defined by user. Each transmission of stack is finished by the next string [NNLL] with NN = Number of tags in the stack and LL = Reader ID.

- **"ON DEMAND CONTEXTUAL" mode:** the reader is in **PULL** mode. The reader doesn't transmit any ID information by default. Reader sends the list of tags only when it receives a specific command: [0A00LL]. Each stack transmission is finished by the next string [NNLL] with NN = Number of tags in the stack and

LL = Reader ID. Please see paragraph 3.7 "Contextual Stack Query".

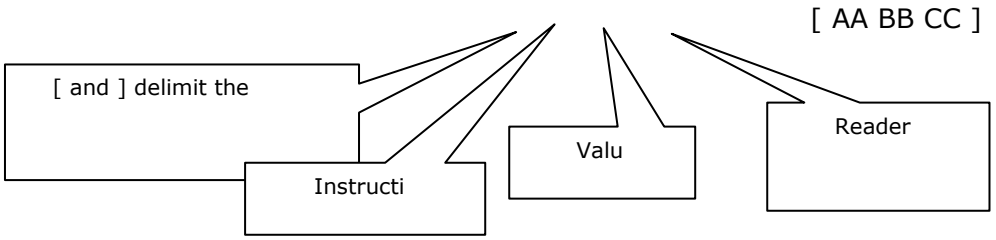
**Stack length is limited.**
**The stack's limit is 20 simultaneous tags before the MCHD 30.0.0 version.**
**The stack's limit is 250 simultaneous tags starting with the MCHD 30.0.0 version and can be changed by user.**

# 3. Description of instructions

The command list enables you to change the reader's operating settings.

## 3.1. Instruction syntax and "on line" help menu

Command format:




You can use the value CC= 00. In this case, all readers will accept this instruction (Broadcast Mode). Some instructions accept values based on 2 bytes or more. ([AA BB BB CC])

"On Line" HELP menu:
This menu is only available starting with the MCHD 30.0.5 version (in English). If you need help with an instruction, send the command:

[AA BB CC?

Example:
*Send normal instruction to the reader: [120001] –> normal answer of the reader could be: [120401] Need help on this instruction? Send: [120001? –> Answer from the reader will be:*

```
-------RS232 Serial Speed-------
get: [12xx01]
set: [13xx01]
-00h 9600 Bauds
-01h 19200 Bauds
-02h 38400 Bauds
-03h 57600 Bauds
-04h 115200 Bauds
------------------------------
```

## 3.2. Operating mode (01/02)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read value | [01xxLL] | [01VVLL] | LL: Reader ID<br>VV: operating mode<br>xx: Ignored |
| Write value | [02VVLL] | [OK02VVLL] | |

Coding of operating mode (VV):

| Valu | Operating mode |
|---|---|
| 01 | On-line |
| 02 | Contextual |
| 03 | Periodic Contextual |
| 04 | On-demand Contextual |

***Example:***
*If a reader has an ID 01, sending the instruction [010000] (only one reader should be present) will enable to know which operating mode is being used. An answer of [010101]*

*means that the reader is using "ONLINE" mode and its ID is 01. Sending instruction [020401] makes the reader work in "ON DEMAND CONTEXTUAL" mode.*

## 3.3. **Output type (03/04)**

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| **Read value** | [03xxLL] | [03VVLL] | LL: Reader ID VV: Output type |
| **Write value** | [04VVLL] | [OK04VVLL] | |

Coding of output type (VV):

| Valu | Output type |
|---|---|
| 01 | RS232 |
| 02 | WIEGAND |
| 03 | DATACLOCK 10 bit |
| 04 | DATACLOCK 13 bit |

*Example:*
*If we have a reader that is identified as 01, sending the instruction [030001] will enable to know which output type is currently being used. An incoming frame [030101] means that output type is 232 only. Sending the instruction [040201] will select the output type RS232 + Wiegand from the reader.*

## 3.4. **"TIMEOUT" or Out of Range Timing (05/06)**

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| **Read value** | [05xxLL] | [05VVLL] | LL: Reader ID<br>VV: Out of Range Timing<br>xx: Ignored |
| **Write value** | [06VVLL] | [OK06VVLL] | |

**Coding of Out of Range Timing (VV):** the value is the output timing of the reader's stack in second (hex value). When tags disappear from the reader's detection field, the tag's ID is memorized in the reader's stack for VV seconds.
A VV value = 01 is not recommended.

**Only for Contextual mode, Periodic Contextual mode, and on-demand Contextual mode.**

The "Time out" is the minimum lead-time to validate the output or the lack of a tag, which enables to limit the reading of this tag when this one is still present in the detection field. Parameter value by default is 6 seconds. If the tag goes out of the detection field and comes into this same field during the 6 seconds, this tag's output won't be considered as a valid output.

## 3.5. **Stack sending cycle (07/08)**

**Coding of sending cycle of the reader stack (VV):** sending cycle value in seconds (hex value). This value is only used in "PERIODIC CONTEXTUAL" mode.

## 3.6. **Reader ID (09/19)**

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| **Read value** | [09xxLL] | [09VVLL] | LL: Reader ID VV: Reader ID |
| **Write value** | [19VVLL] | [OK19VVLL] | |

**Coding of reader ID (VV):** the value is the reader's ID that user wants to allocate to the reader (hex value).

***Example:***
*If we have a reader which is identified as 01 (default value), sending instruction [190201] allows selecting value 02 for reader ID. [OK190201] will be sent by the reader in acknowledgment.*

**Remarks**:

- The reading instruction is useful to determinate an unknown reader ID. In this case, you must send the reading instruction in BROADCAST mode (LL=00). Example: [090000].
- In the writing case, reader ID must be specified, it mean that this instruction is not allowed in BROADCAST mode (LL=00).

## 3.7. **Contextual stack query (0A)**

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| **Read value** | [0AxxLL] | * | LL: Reader ID xx: Ignored |
| **Write value** | - | - | |

**Remarks:**
The response to a contextual stack interrogation instruction is composed of several frames: one for each tag in the reader's stack, followed by the sent frame counter.

*Example:*
*[0A0000] → response: [7801234501][948831A501][7821234501][0301]*

MCHD 30.0.0 firmware version, a new syntax message is available:

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| **Read value** | [0AvvLL] | * | LL: Reader ID vv: Value |
| **Write value** | - | - | |

| VV | Format | Syntax |
|---|---|---|
| 00 | Normal | [AA xxxxxx LL] [AA xxxxxx LL] …[yy LL] with:<br>[ and ] are the message delimiters.<br>AA is the tag's RSSI level.<br>xxxxxx Id Tag (16, 24 or 32 bits).<br>yy number of tags in the stack.<br>LL Id reader |
| 01 | Compact without RSSI | [xxxxxx xxxxxx xxxxxx yy LL]<br>xxxxxx ID tag (16, 24 or 32 bits).<br>yy number of tags<br>LL Id reader |
| 02 | Normal with tag select with increasing RSSI | |
| 03 | Normal with tag select with decreasing RSSI | |
| 04 | Normal with tag select with increasing ID | |
| 05 | Normal with tag select with decreasing ID | |
| FF | Only number of tags | [yy LL] with:<br>[ and ] are the message delimiters. yy number of tags in the stack.<br>LL Id reader |

## 3.8. **Reading threshold (0B/0C)**

| Instruction | Syntax | Acknowledge | Comme |
|---|---|---|---|
| **Read value** | [0BxxLL] | [0BVVLL] | LL: Reader ID<br>VV: Reading threshold<br>xx: Ignored |
| **Write value** | [0CVVLL] | [OK0CVVLL] | |

**Coding Reading threshold (VV):** the value is the threshold of the RSSI level (hex value). This instruction allows programming a filtering threshold on tag reading. Only tags with a reception level lower than the threshold will be read.

**Remarks:** the shape of the detection area cannot be modified. To change this shape, it is necessary to choose the right type of antenna (directional or omni-directional).

**This threshold is a software threshold after having received the tag ID. This filtering operation has no impact on anti-collision management, in case there are a lot of tags in the reader's detection area.**

*Example: Default reception level is FE (default value) and reader regularly transmits the next frames: [7801234501][948831A501] matching successively a tag whose ID is 012345 with a reception level 78 (hex) and a tag whose ID is8831A5 with a reception level 94(hex).Sending the instruction [0C9001] allows changing the reader threshold to 90(hex). [OK0C9001] will be sent by the reader in acknowledgment. The reader will filter 8831A5 and will transmit the frame [7801234501] regularly.*

## 3.9. **Tag Id length (10/11)**

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| **Read value** | [10xxLL] | [10VVLL] | LL: Reader ID<br>VV: Radio reception method<br>xx: Ignored |
| **Write value** | [11VVLL] | [OK11VVLL] | |

Coding of Tag ID length (VV):

| Valu | Radio reception method |
|---|---|
| 01 | 16 bits ID tag Reception |
| 02 | 24 bits ID tag Reception |
| 03 | 32 bits ID tag Reception |

## 3.10. **RS232 speed (12/13)**

| Instruction | Syntax | Acknowledge | Comme |
|---|---|---|---|
| **Read value** | [12xxLL] | [12VVLL] | LL: Reader ID<br>VV: 232 communication speed<br>xx: Ignored |
| **Write value** | [13VVLL] | [OK13VVLL] | |

Coding of 232 communications speed (VV):

| Valu | 232 speed | Allowed version |
|---|---|---|
| 00 | 9600 bauds | all |
| 01 | 19200 bauds | all |
| 02 | 38400 bauds | > MCHD 20.x.x |
| 03 | 56700 bauds | > MCHD 20.x.x |
| 04 | 115200 bauds | > MCHD 20.x.x |

## 3.11. ASCII or Binary transmission format (14/15)

| Instruction | Syntax | Acknowledge | Comme |
|---|---|---|---|
| Read value | [14xxLL] | [14VVLL] | LL: Reader ID<br>VV: Transmission Format<br>xx: Ignored |
| Write value | [15VVLL] | [OK15VVLL] | |

Coding of 232 transmission format (VV):

| Valu | Transmission Format |
|---|---|
| 00 | ASCII |
| 01 | Binary |

**Remarks:** in binary mode, frames delimiting characters are the same: "[ and ]" matching in binary mode at 5B and 5D.

## 3.12. Delay before acknowledgment (16/17)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read value | [16xxLL] | [16VVLL] | LL: Reader ID<br>VV: Delay<br>xx: Ignored |
| Write value | [17VVLL] | [OK17VVLL] | |

**Coding of Delay before Acknowledgment from an external instruction frame:**
value shows delay in milliseconds (hex value). Default value is 0 (no delay).
A delay in reader acknowledgment can be useful when you need to send a query command in Broadcast mode to several readers at the same time. In this case, we suggest applying 20ms between each reader at 9,600 bds or 2ms between each reader at 115,200 bds.

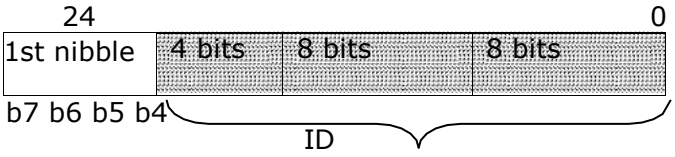
## 3.13. Activation of Alarm Information on DATA/CLOCK output (1A/1B)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read value | [1AxxLL] | [1AVVLL] | LL: Reader ID<br>VV: Sending alarm status<br>xx: Ignored |
| Write value | [1BVVLL] | [OK1BVVLL] | |

Coding of sending alarm status (VV):

| Valu | Sending alarm status |
|---|---|
| 00 | Sending alarm OFF |
| 01 | Sending alarm ON |

**Definition of tag's alarm in 24-bit mode:**
Low battery: voltage < 2.1 V
Breakout ( µswitch ): Tag unglued
**Coding structure:**
Tag ID is composed of 6 hex characters, on 24 bits
First (or last) ID's nibble changes in function of the activated alarms.




**Remarks: The bit b7 of the high byte is reserved to indicate a tag with data: ID + DATA (Temp., RH%,...)**

| Example ID no.: 050123 | | | | |
|---|---|---|---|---|
| (b6) | Low Bat | uSwitch (b4) | ID Frame hex | Alarm type |
| 0 | 0 | 0 | **0**50123 | - |
| 0 | 0 | 1 | **1**50123 | Breakout |
| 0 | 1 | 0 | **2**50123 | Low Bat |
| 0 | 1 | 1 | **3**50123 | Breako ut Low |

**Table 2: management of tag's alarm in 232 mode**

**Coding of Data Clock mode:**
10 font sizes: XX 0 ID
XX: alarm type code in 2 decimal
digit 0: complementary digit
(always 0)
ID: complete identification code in 7decimal digits

*Remarks**:** when reader's settings have no alarm management, the tag's ID takes different decimal values depending on the type of alarm sensors activated (please see the examples given in Table 3).*

| Example no. ID: 0105A0 | | | | Data Clock output 10 decimal characters | |
|---|---|---|---|---|---|
| | | | | Alarm management | |
| (b6) | Low Bat (b5) | uSwitch (b4) | Hex frame ID | Without (parameter | With (parameter [1B01yy]) |
| 0 | 0 | 0 | **0**105A0 | 000 0066976 | 00 0 0066976 |
| 0 | 0 | 1 | **1**105A0 | 000 1115552 | 01 0 0066976 |
| 0 | 1 | 0 | **2**105A0 | 000 2164128 | 02 0 0066976 |
| 0 | 1 | 1 | **3**105A0 | 000 3212704 | 03 0 0066976 |

**Table 3: TAG alarm management in D&C 10char mode**

13 character format: 000 XX 0 ID
000: complementary header (always 000) XX: Alarm type code (decimal) on two-digits
0: complementary digit (always 0)
ID: decimal tag ID code on seven digits

*Remarks**:** when reader's settings have no alarm management, the tag's ID takes different decimal values depending on the type of alarm sensors activated (please see the examples in Table 4).*

| Example no. ID: 0105A0 | | | | Data Clock output 13 decimal | |
|---|---|---|---|---|---|
| | | | | Alarm management | |
| (b6) | Low Bat (b5) | uSwitch (b4) | Hex ID frame | Without (parameter | With (parameter |
| 0 | 0 | 0 | 0105A0 | 000 000 | 000 00 0 |
| 0 | 0 | 1 | 1105A0 | 000 000 | 000 01 0 |
| 0 | 1 | 0 | 2105A0 | 000 000 | 000 02 0 |
| 0 | 1 | 1 | 3105A0 | 000 000 | 000 03 0 |

**Table 4: TAG alarm management in D&C 13char mode**

## 3.14. Redundancy Activation (1C/1D)

**Before MCHD 30.0.5 firmware version:**

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read value | [1CxxLL] | [1CVVLL] | LL: Reader ID<br>VV: Redundancy status<br>xx: Ignored |
| Write value | [1DVVLL] | [OK1DVVLL] | |

Coding of redundancy status (VV):

| valu | Redundancy effect |
|---|---|
| 00 | Redundancy off |
| 01 | Redundancy on (2 successive identifications required) |

Redundancy activation allows waiting for an identification acknowledgment (two successive ID frames) during the timeout lead-time before being memorized in the reader.

**Only for Contextual, Periodic Contextual, and Contextual on-demand mode.**

*Example: If the timeout delay is 3s and a tag has a 2s transmit period, if redundancy is activated, the tag will never be memorized by the reader.*

**Starting with the MCHD 30.0.5**

Coding of redundancy status (VV):

| Valu e | Redundancy effect | Number of successive identifications in order to |
|---|---|---|
| 00 | Redundancy Off | 1 (first identification is accepted) |
| 01 | Redundancy On | 2 (2 successive identifications are |
| 02 | Redundancy On | 3 (3 successive identifications are |
| 03 | Redundancy On | 4 (4 successive identifications are |

**If redundancy is activated, each successive identification has to be done in a "Time In" time** (see instruction [28xxLL] ). User can change this "Time In" time. **We recommend programming "Time In" time equal to 3 * redundancy value * tag transmission period.**

## 3.15. Radio Reception Method (1E/1F)

| Instruction | Syntax | Acknowledge | Comme |
|---|---|---|---|
| Read value | [1ExxLL] | [1EVVLL] | LL: Reader ID<br>VV: Radio reception method<br>xx: Ignored |
| Write value | [1FVVLL] | [OK1FVVLL] | |

Coding of radio reception method (VV):

| Valu | Radio reception procedure |
|---|---|
| 00 | 16 or 24 bits with CKS at 433 MHz |
| 01 | 16 or 24 bits with CKS at 868 MHz |
| 02 | 16 bit without CKS |
| 03 | 24 bit without CKS (S) |
| 10 | X4 mode |

Remarks: the communication speed of our RFID tags is optimized in order to get the best compromise between a tag's range and its lifecycle. All the information about the range and the autonomy is based upon this optimized protocol (called HD).

Nevertheless, in some conditions, you may use a quick communication protocol (X4 mode), in order to increase the tag's lifecycle to the detriment of a reduced range. Please contact our technical department in order to find out which products support this new protocol.

## 3.16. **Activation of tag filtering with the strongest radio level (20/21)**

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read value | [20xxLL] | [20VVLL] | LL: Reader ID<br>VV: Value<br>xx: Ignored |
| Write value | [21VVLL] | [OK21VVLL] | |

Coding of the tag with the strongest radio level statue (VV):

| Valu | Sorting the tag with the strongest |
|---|---|
| 00 | OFF |
| 01 | ON |

The activation of sort of the tag with the strongest radio level allows to limit the contextual stack to a unique tag: the one which gets the highest RSSI level (the lowest value).

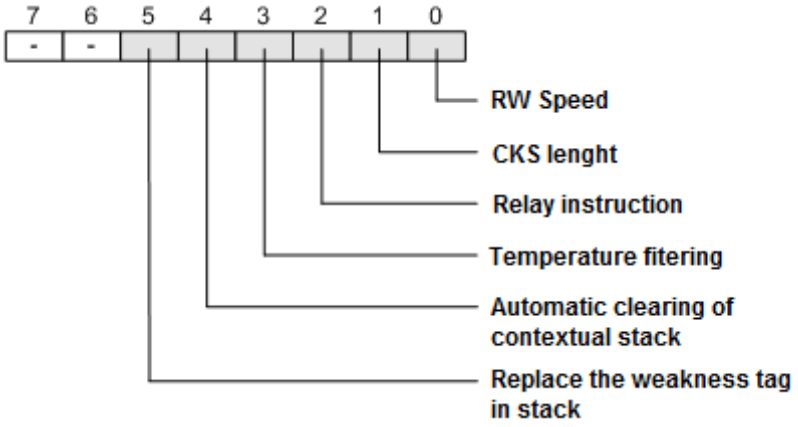
**Only for on-demand Contextual mode.**
*Example: from those tags received by the reader: [7801234501][948831A501], only [7801234501] will be kept in the contextual stack.*

## 3.17. Tag programmer speed (22/23)

Depending on the device used, two protocol speeds are available to program tags: The first one is a low speed protocol used for a magnetic setting tool (tag version 8, 9, 10, and 20) The second one is faster, used for an infrared setting tool (tag version 11, 15, 16, 21 and higher).

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read value | [22xxLL] | [22VVLL] | LL: Reader ID<br>VV: Value<br>xx: Ignored |
| Write value | [23VVLL] | [OK23VVLL] | |

These commands allow reading and writing many parameters. The coding of these parameter's value is the following:




The available values of the programming speed are:

| B0 Bit value of byte flag (22 and | Programming speed |
|---|---|
| 0 | SLOW (ILS) |
| 0 | FAST (IR) |

## 3.18. Tag checksum length

Two lengths can be used for the tag checksum: 8- and 16-bit. A reader set with a 16-bit rw protocol bit length will only see tag frames with a 16-bit rw protocol length. On the other hand, readers set with an 8-bit rw protocol bit length will see all tag frames (with 8- and 16-bit rw protocol lengths).

**Format**
See previous section for the format of commands 22 and 23.

The values of checksum's length are:

| B1 Bit value of byte flag (22 and | CKS length |
|---|---|
| 0 | 8 bits |
| 0 | 16 bits |

## 3.19. Direct relay control mode

Relays are directly integrated in some readers. Two modes are available to control the relay:

a) Bysending the instruction: the relay is turned on for 4 seconds when the reader receives the instruction [OK0000LL].

b) By contextual stack length (number of tags in the detection field): the relay is turned on until the number of tags in the detection field is higher or equal to the number of expected tags (see commands 26 and 27).

**Format**

See previous section for the format of commands 22 and 23. The available values to select the direct relay control mode are:

| B2 Bit value of byte flag (22 and | Relay control mode |
|---|---|
| 00 | [OK0000LL] |
| 01 | On Contextual stack |

## 3.20. Activation of sensor filtering

Some tags integrate sensors. A tag set in a data mode (12b ID + 12b DATA) sends its data by the 3 lowest nibbles of its ID frame; the issue is that the tag will be added in the reader's stack several times when the temperature varies. To avoid this, we introduce *data filtering mode*.

Data filtering mode enables you to hide sensor information during scanning and stack update operations. In this manner, only one record in the contextual stack is made for a sensor-based tag. Only the last value received by the reader is saved in the reader's contextual stack.

**Remarks:** this mode can be also used for DG, RH tags (mode: 12b ID + CNT, 12b ID + TOR,…) and every tag that get its high bit at 1.

**Format**

See previous section for the format of commands 22 and 23.
The available values to select the sensor's data filtering mode are:

| B3 Bit value of byte flag (22 and | Data filtering |
|---|---|
| 0 | Off (no filter) |
| 0 | Filter is activate |

## 3.21. Activation of automatic contextual stack erasing after reading mode

This operation mode enables you to activate automatic erasing of the contextual stack when sending the instruction [0A], after receiving the answer.

Stack erasing after reading is used when number of tags that can be received is higher than the contextual stack length. In this case, a regular erase of the stack makes it possible to add other tags in the contextual stack between two reading operations.

**Format**

See the chapter 3.17 Read/Write protocol speed for the format of commands 22 and 23. The available values to erase the contextual stack automatically are:

| B4 Bit value of byte flag (22 and 23 | Automatic stack erasing |
|---|---|
| 0 | Off |
| 0 | On |

## 3.22. Full Contextual stack option

In normal mode, when the contextual stack is full, a new tag replaces the oldest identified tag. When you activate this bit, a new tag replaces the tag with the lowest RSSI when contextual stack is full.

| B5 Bit value of byte flag (22 and | Full Stack option |
|---|---|
| 0 | Oldest Id tag is erased |
| 0 | Lowest Id tag is erased |

## 3.23. Programmer IR pulse length (24/25)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read Value | [24xxLL] | [24VVLL] | LL: Reader ID<br>VV: Value<br>xx: 00 |
| Write Value | [25VVLL] | [OK25VVLL] | |

This value is reserved for ELA INNOVATION programming options (value from 164 to 167).

## 3.24. Sending request for Radiofrequency Noise Measurement (97)

| Instruction | Syntax | Acknowledgment | Commen |
|---|---|---|---|
| Read Value | [97xxLL] | [97VVLL] | LL: Reader ID<br>VV: Value<br>xx: 00 |
| Write Value | - | - | |

This instruction gets a measurement of the radiofrequency noise in the reader's frequency and checks whether or not this band is polluted.

**Remarks:** a value between 185 and 210 is correct (noise is low). The lower the value, the higher the external disturbances. A noise measurement below 185 is also very significant.

## 3.25. Sending request for firmware version (98)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read Value | [98xxLL] | [VEvvv…vvv] | LL: reader ID<br>vvv…vvv: Firmware version<br>xx: 00 |
| Write Value | - | - | |

**Coding of firmware version (vvv…vvv):** the returned value is a character chain in ASCII format.
*Example: [98xxLL] → Response: [VEMCHD v13.5.1]*

## 3.26. Sending request for reader parameters (99)

| Instruction | Syntax | Acknowledge | Comme |
|---|---|---|---|
| Read Value | [99xxLL] | * | LL: reader Id<br>xx: 00: simple output<br>01: Commented parameters |
| Write Value | - | - | |

Encoding the response to the drive configuration request consists of multiple data. Each data corresponds to a parameter and is sent in the same format as that of the response to the corresponding command for reading the parameter.
*Example:*

```
[990001] → reader answer:
[010101]
[030101]
[050601]
[073C01]
[0BFE01]
[100201]
[120001]
[140001]
[160001]
[1A0001]
[1C0001]
[1E0001]
[200001]
[220301]
[24A501]
[260101]
[280001]
[2A1401]
[2E0001]
[B00001]
[C00001]
[C80001]
[CA0001]
[VEMCHD v30.1.7]
```

*Example: [990101], then reader returns:*

```
+-------------------------------------------------------------+
| Ela Innovation - RFID Reader MCHD v30.1.7                   |
+-------------------------------------------------------------+
| Parameter               Get/Set Value(Hexa)     Comment     |
+-------------------------------------------------------------+
| Operating Mode          [01/02]  01  On Line                |
| Output Type             [03/04]  01  RS232 Only             |
| Tag Stack Timeout       [05/06]  06  6 seconds              |
| Tag Stack Send Period   [07/08]  3C  60 seconds             |
| Reader ID               [09/19]  01                         |
| Contextual Stack        [0A/--]  --  Send Tag List          |
| Software Threshold      [0B/0C]  FE  254 (no units)         |
| Radio Frame Size        [10/11]  02  24 Bits                |
| RS232 Serial Speed      [12/13]  00  9600 Bauds             |
| Data Format             [14/15]  00  ASCII Format           |
| Answer Delay            [16/17]  00  0 milliseconds         |
| Data Clock Alarm        [1A/1B]  00  Disabled               |
| Redundancy              [1C/1D]  00  Disabled               |
| Radio Format            [1E/1F]  00  HD Format              |
| Sort Strongest Tag      [20/21]  00  Disabled               |
| Config Flags            [22/23]  03  see [22xx01? for details|
| RW Bit Width            [24/25]  A5  165 (no units)         |
| Min Tags In Stack       [26/27]  01  1 Tags                 |
| Tag Stack Timein        [28/29]  00  0 seconds              |
| Stack Size              [2A/2B]  14  20 Tags                |
| Radio Stats             [2C/2D]  --  see [2Cxx01? for details|
| Contextual Stack Format [2E/2F]  00  Normal Mode            |
| Encapsulation (header)  [70/71]  --  see [70xx01? for details|
| Encapsulation (trailer) [72/73]  --  see [72xx01? for details|
| Radio Noise Level       [97/--]  --                         |
| Software Version        [98/--]  --                         |
| Factory Settings        [--/9C]  --  see [9Cxx01? for details|
| Filter range (low)      [A0/A1]  --  see [A0xx01? for details|
| Filter range (high)     [A2/A3]  --  see [A2xx01? for details|
| Filter mask             [A4/A5]  --  see [A4xx01? for details|
| Hardware Threshold      [B0/B1]  00  Disabled               |
| Delta Checksum          [C0/C1]  00  0 (no units)           |
| Attenuation             [C8/C9]  00  0 (no units)           |
| Half Duplex Mode        [CA/CB]  00  Disabled               |
+-------------------------------------------------------------+
```

## 3.27. Erase Contextual Stack (9A)

| Instruction | Syntax | Acknowledge | Comme |
|---|---|---|---|
| | - | - | LL: Reader ID<br>xx: 00 |
| **Write Value** | [9AxxLL] | [OK9AxxLL] | |

This command allows you to erase the Contextual Stack

**Remarks**: BROADCAST (with LL=00) message are not supported for this instruction

## 3.28. Tag Id Filter (Ar/Aw)

Instructions A0 to AF allow you to define a filtering interval for tag ID. All tags in this interval are accepted; all others are not taken into account for the reading process. This interval is defined as described below:

The lowest ID is composed 3 bytes ID1i, ID2i and ID3i. Example: IDi = 012345 (hex).
The highest ID is composed of 3 bytes ID1f, ID2f and ID3f. Example: IDf = 6789AB (hex).

Before MCHD 30.0.5 firmware version, the reading and writing commands are:

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| **Read Value** | [ARxxZZ] | [ARYYZZ] | R/W: Read or write byte limits number. See Example<br>YY: Value<br>ZZ: Reader ID<br>xx: 00 |
| **Write Value** | [AWYYZZ] | [OKAWYYZZ] | |

**Exception for 16 bits tag format:**
In this case only the two first byte limits are used: $ID1_i$ $ID2_i$ and $ID1_f$ $ID2_f$.
Alarms are not available in this specific format.

**Alarm management:**
In the case of 24-bit format for tag ID, the first nibble contains the alarm coding. For these tags, the filtering process has to be done by ignoring the alarm status (mask of first nibble containing the alarm status).

**Activation / deactivation commands for alarm filtering process:**
Activation: [AE01ZZ]
Deactivation: [AE00ZZ]

***Example:*** *Read value → $ID_i$ = 0x000000 and $ID_f$ = 0xffffff.*

```
Instruction → Reader answer:

[A00001] → [A00001]
[A10001] → [A10001]
[A20001] → [A20001]
[A30001] → [A3FF01]
[A40001] → [A4FF01]
[A50001] → [A5FF01]
```

***Example:*** *Change Tag Id filter limits: from $ID_i$ = 0x123456 to $ID_f$ = 0x789ABC.*

```
Instruction → Reader answer:

[A81201] → [OKA81201]
[A93401] → [OKA93401]
[AA5601] → [OKAA5601]
[AB7801] → [OKAB7801]
[AC9A01] → [OKAC9A01]
[ADBC01] → [OKADBC01]
```

Instructions changed starting with the MCHD 30.0.5 firmware version:

**Reading & writing command for the lowest tag's ID:**

| Instructi | Syntax | Acknowledgme | Comments |
|---|---|---|---|
| Read | [A0xxxxxxZZ] | [A0yyyyyyZZ] | R/W: read/write. See example<br>YY : value for reading or writing<br>ZZ: Reader ID<br>xx: 0..0 |
| Write Value | [A1yyyyyyZZ] | [OKA1yyyyyyZZ] | |

yy is defined on 16, 24, or 32 bits.

**Reading & writing command for the highest Tag's ID:**

| Instruction | Syntax | Acknowledgme | Comments |
|---|---|---|---|
| Read Value | [A2xxxxxxZZ] | [A2yyyyyyZZ] | R/W: read/write. See example<br>YY : value for reading or writing<br>ZZ: Reader ID<br>xx: 0..0 |
| Write Value | [A3yyyyyyZZ] | [OKA3yyyyyyZZ] | |

***Example:*** *write a Tag Id interval from* $ID_i$ *= 0x123456 to* $ID_f$ *= 0x789ABC.*

```
Instruction → Reader answer:

[A112345601] → [OKA112345601]
[A3789ABC01] → [OKA3789ABC01]
```

## 3.29. Filtering Mask (A4/A5)

This command is only available starting with the version 30.1.5. The masking is a logical operation that preserves or forces a group of bit to 0.

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read Value | [A4xxxxxxZZ] | [A4yyyyyyZZ] | YY: Filtering Mask ZZ: Reader ID |
| Write Value | [A5yyyyyyZZ] | [OKA5yyyyyyZZ] | |

*Example: masking write:*

```
Instruction → Reader answer:

[A5FF00FF01] → [OKA5FF00FF01]
[A400000001] → [A4FF00FF01]
```

If an "ABCDEF" tag is received by the reader after application of the masking, as seen in the example, the tag' value shorten by the mask becomes "AB00EF".
Tags are stored in the stack and / or transmissions in RS-232, depending on the operating mode.

Note: The masking operation can be used only after setting a filter range, control A1 and A3 (cf. 3.28).

## 3.30. Reader's CRC / TAGS Delta CHECKSUM (C0/C1)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read Value | [C0xxLL] | [C0VVLL] | LL: Reader ID<br>VV: Checksum value<br>xx: 0..0 |
| Write Value | [C1VVLL] | [OKC1VVLL] | |

VV is hex value of delta checksum.
This parameter is a private communication key between readers and tags. They must have the same key value to identify each other.

## 3.31. Contextual stack size for relay activation mode (26/27)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read Value | [26xxLL] | [26VVLL] | LL: Reader ID<br>VV: Stack length for relay activation mode<br>xx: 0..0 |
| Write Value | [27VVLL] | [OK27VVLL] | |

If the option for direct relay commands is activated (commands 22 and 23), and if the reader has an internal relay, it is activated as long as the number of tags detected is greater or equal to the contextual stack's size for relay activation mode.

## 3.32. Reader RSSI calibration (C8/C9)

| Instruction | Syntax | Acknowledgment | Comments |
|---|---|---|---|
| Read Value | [C8xxLL] | [C8VVLL] | LL: Reader ID<br>VV: Calibration value<br>xx: 0..0 |
| Write Value | [C9VVLL] | [OKC9VVLL] | |

VV is the calibration value for the reader's RSSI. This factory-calibrated parameter is used to get the same RSSI variation for all the readers.
Remarks: attenuation can be positive or negative, from -32 to +32.
**The format of negative values is in 2's complement!**
These instructions are only available starting with the MCHD version 13.6.2!

**This command is strictly reserved for ELA INNOVATION use.**

## 3.33. Contextual stack length modification (2A/2B)

| Instruction | Syntax | Acknowledgment | Comments |
|---|---|---|---|
| Read Value | [2AxxLL] | [2AVVLL] | LL: Reader ID<br>VV: Contextual stack length<br>xx: 0..0 |
| Write Value | [2BVVLL] | [OK2BVVLL] | |

VV is hex value for contextual stack length. By default this value is 0x14 (20 tags).
**Important remarks:** all the firmware versions before 30.0.0 version DO NOT have this feature. For these, contextual stack length cannot be longer than 20 tags.

## 3.34. **Hardware threshold for reader's RSSI (B0/B1)**

| Instruction | Syntax | Acknowledgment | Comments |
|---|---|---|---|
| Read Value | [B0xxLL] | [BOVVLL] | LL: Reader ID<br>VV: Analog value for hardware RSSI threshold<br>xx: 0..0 |
| Write Value | [B1VVLL] | [OKB1VVLL] | |

This command enables you to program a hardware threshold (analog value) on tag detection. Only tags that have an RSSI level lower than this threshold will be read.

**Coding of RSSI Threshold (VV):** the value represents the threshold of reception level (hex value).

**The filtering process can be disabled (automatic mode) by using the value VV=00. Automatic mode is optimizing the performance of reader's max range. This mode must be used when a max detection range is expected. By default, the readers are configured in AUTO mode.**

**The filtering process is based upon a hardware analogical filter applied before the RFID tag's identification. This filtering improves the performances of tags' anti-collision management in a lot of tags are present in the detection field.**
**Analog filtering is not an adaptive filter. The filtering threshold must be significantly lower than the noise level (RSSI SCAN) measured locally. We recommend programming a threshold 20 points lower than the noise level, or staying in AUTO mode.**

*Example: Reception level is initially set to 00 (default value) and the reader periodically transmits these frames: [7801234501][948831A501] corresponding successively to tag ID 012345 with an RSSI level of 78(hex) and the tag 8831A5 with an RSSI level of 94 (hex). Sending the instruction [B19001] changes the analog reader reception threshold value to 90 (hex). Otherwise the reader will acknowledge reception with the following frame [OKB19001]. The reader will not read the tag identified 8831A5 and will send the frame [7801234501] periodically.*

## 3.35. Adding a prefix to frames (70/71)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read Value | [70ppxxLL] | [70ppccLL] | LL: Reader ID<br>pp: Position of character in the string cc: ASCII value of character string xx: 0..0 |
| Write Value | [71ppccLL] | [OK71ppccLL] | |

This instruction adds a customized prefix to all messages sent by readers.

**The 00 character is reserved: it indicates the end of the prefix**

***Example:*** *to set frame encoding for a GPRS X1 receiver, you must send the following frames:*

*For the header: $ST+MMSG=*

*[71002401] sending $*
*[71015301] sending S*
*[71025401] sending T*
*[71032B01] sending +*
*[71044D01] sending M*
*[71054D01] sending M*
*[71065301] sending S*
*[71074701] sending G*
*[71083D01] sending =*
*[71090001] prefix END STRING*

## 3.36. Adding a suffix to frames (72/73)

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read Value | [72ppxxLL] | [72ppccLL] | LL: Reader ID<br>pp: Position of character in the string cc: ASCII value of character string xx: 0..0 |
| Write Value | [73ppccLL] | [OK73ppccLL] | |

This instruction adds a customizable suffix to all messages that the reader returns.

**The 00 character is reserved: it indicates the end of the suffix**

***Example****: to set frame encoding for the GPRS X1 receiver, you must send the following frames:*

*Ending encoding frames: carriage return CR + LF*
*[73000D01] sending carriage return (CR)*
*[73010A01] sending line feed (LF)*
*[73020001] suffix END STRING*

## 3.37. Definition of frame's format in Periodic Contextual mode (2E/2F)

**Note**: format coding is only available starting with the 30.0.5 version.

| Instruction | Syntax | Acknowledgme | Comme |
|---|---|---|---|
| Read Value | [2EvvLL] | * | LL: Reader ID<br>vv: Response syntax |
| Write Value | [2FvvLL] | - | |

| value | Format | Syntax |
|---|---|---|
| 00 | Normal Format | [AA xxxxxx LL] [AA xxxxxx LL] …[yy LL]<br>with: [ and ] are characters that delimit each tag frame.<br>AA coded on 2 ASCII characters that define 1 byte corresponding to RFID tag reception level.<br>xxxxxx coded on 6 ASCII characters that define 3 bytes corresponding to RFID tag ID. ID length is variable, and depends on the length that we expect (16, 24, or 32 bits).<br>LL coded on 2 ASCII characters that defines 1 byte corresponding to reader ID.<br>**Several tags can be sent**. |
| 01 | Compacted format without RSSI level | [xxxxxx xxxxxx xxxxxx yy LL]<br>xxxxxx coded on 6 ASCII character that define 3 bytes corresponding to RFID tag ID. ID length is variable, depending on the length that we expect (16, 24, or 32 bits).<br>LL coded on 2 ASCII characters that define 1 byte corresponding to reader ID.<br>**Several tags can be sent**.<br>yy coded on 2 ASCII characters that define 1 byte corresponding to the number of tags sent. |
| 02 | Normal format, Tag RSSIs in ascending order | |
| 03 | Normal format, Tag RSSIs in descending order | |
| 04 | Normal format, Tag IDs in ascending order | |
| 05 | Normal format, Tag IDs in descending order | |
| FF | Sending frame with number of tags | [yy LL] :<br>[ et ] ] are characters that delimit each tag frame. yy coded on 2 ASCII characters that define 1 byte corresponding to the number of tags sent in contextual stack. |

## 3.38. **Redundancy Time-In (28/29)**

Remark: this command is only available starting with the version 30.0.5.

| Instruction | Syntax | Acknowledge | Comme |
|---|---|---|---|
| **Read Value** | [28xxLL] | [28ssLL] | LL: Reader ID<br>ss: Time value in seconds to validate the frame needed to return a tag in the contextual stack. |
| **Write Value** | [29ssLL] | [OK29ssLL] | |

This instruction defines the reading acknowledgement frame delay when a tag is received.
**This value is limited to 128 seconds.**
**This command is taken into account only when the redundancy option is activated, and only for Contextual, Periodic Contextual, and On-demand Contextual modes.**

Redundancy activation ([1CxxLL] command) allows you to wait for confirmation of a tag's identification (2, 3 or 4 successive identifications) in a time period shorter than the **Time In** before this tag's ID is memorized in the contextual stack.

For a tag with an emission cycle of 1.3s, and in the case that the Redundancy option is activated with 4 successive transmissions (03 mode), the tag will be memorized in the contextual stack after 3.9 seconds in the best case. We recommend a **Time In value** equal to 3*3.9s or 12s.

## 3.39. **Half-duplex mode (CA/CB)**

This parameter is used to select the communication mode of the drives.
Two possibilities: half-duplex or full-duplex

| Instruction | Syntax | Acknowledge | Comment |
|---|---|---|---|
| **Read Value** | [CAxxLL] | [CAssLL] | LL: Reader ID<br>ss: if 0 = full-duplex<br>if 01 = half-duplex<br>xx: 0..0 |
| **Write Value** | [CBssLL] | [OKCBssLL] | |

Remark: half duplex communication mode is useful in some configuration for instance with the two-wire RS485.

## 3.40. Radio statistics (2C/2D)

This parameter measures the radio success rate and thus obtain an image of the environment where system is installed.

| Instruction | Syntax | Acknowledge | Comment |
|---|---|---|---|
| **Read Value** | [2CxxLL] | [2Cxxxx/yyyyLL] | LL: Reader ID<br>xxxx: Number of frames received with a CRC identical to the reader.<br>yyyy: Total number of frames received by the reader (CRC |
| **Write Value** | [2DxxLL] | [OK2D00LL] | |

Example**:** response [2CA1F3/A54101]

***2 choices:***

- *number of frames received with a CRC identical to the reader = A1F3h = 41459d*
- *total number of frames received by the reader (CRC ignored) = A541h = 42305d*

*In this example, the reception of the frames with an identical CRC is: 98%.*

*In order to reset values, it is necessary to apply the command* [2DxxLL] *with xx = any value. The reader response "*[OK2D00LL]*" confirms and validates the reset.*

## 4. Document version

| Version | Date | Author | Changes |
|---|---|---|---|
| H_UK | 29/05/13 | PB | First UK document version |
| I_UK | 27/06/14 | PB | Compatibility Matrix between firmwares and readers<br>Clarification on alert message about the possibility to add prefix and suffix to the radio frames |
| J_UK | 25/04/16 | LA | Add instruction information: communication mode |
| K_UK | 02/11/16 | MG | Add 3.39 Radio Statistic, 3.29 Filtering Mask |
| L_UK | 24/11/16 | DR | Function table updated |
| M_UK | 31/03/17 | DR LA | Add compatibility table for 24/32 bits |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| **STATUS** | **DRAFT** | **CORRECTION** | **FINAL** |
| | | |  |
| **DISTRIBUTION LEVEL** | **CONFIDENTIAL** | **LIMITED** | **GENERAL** |
| | | |  |