Application Note 432

Title: **EMDB410 RFID READER**

Description of the firmware protocol

Product Family: **RFID Solutions**

Part Number: EMDB410

Keywords: EM4294 - EM4233 – ISO15693 - ISO14443 Type A, B

Date: November 11, 2009

1. Introduction

The EMDB410 RFID READER is a 13.56MHz proximity coupler that supports the transponders compliant with the ISO15693 & 14443 standards, and the EM4006 read only transponder. This application note describes the EMDB410 RFID Reader firmware communication protocol.

Family	Source name tree	Last Release	Description
161	EMDB410_firmware	0.6 (29.9.2009)	EMDB410 RFID READER Firmware (ISO15693)
162			EMDB410 RFID READER Firmware (ISO14443)

Table 1: Existing firmware families

Transponder family	Command set support	Communication Speed support
EM4233	Complete except 256b Active EAS	All
ISO14443 Type A	Select sequence flowchart according to ISO14443-3 for single tag	106kb/s data rate
ISO14443 Type B	REQB command according to the ISO14443-3 for single tag	106kb/s data rate
ST SR176	Complete	106kb/s data rate

Table 2: Family 80 supported command set and features

1.1. EMDB410 versus EMDB408 Difference

EMDB410 Reader was derived from the EMDB410 Reader**.** EMDB410 firmware is close to the EMDB408 firmware in terms of the firmware code functions and source architecture. Most of the EMDB408 commands work in the same way as the EMDB410 commands.

EMDB410 Reader differs from EMDB408 Reader in the following features;

- EMDB410 is USB powered only while EMDB408 requires the power supply adapter
- EMDB410 uses EM4294 AFE that incorporates both the EM4094 IC (RF front-end) and the EMTG56 IC (SIM crypto) while the EMDB408 uses separate EM4094 IC (RF front-end) and external EMTG56 IC (SIM crypto)
- EMDB410 uses Atmel ATMega16 microcontroller while EMDB408 uses ATMega64 microcontroller. EMDB410 firmware is split into two different firmware binary streams compiled from the unified source files (one for ISO15693+EM4006 communication, one for ISO14443 communication), EMDB408 firmware is a single unified binary stream only.

2. Abbreviations

ACK – Acknowledge status byte
AFE – Analogue Front End
ASK – Amplitude Shift Keying
ATR – Answer To Reset
CRC – Cyclic Redundancy Check
DES – Data Encryption Standard
EAS – Emergency Alert System
EGT – Extra Guard Time

EOF – End Of Frame
ETU – Elementary Time Unit
FSK – Frequency Shift Keying
PPS – Protocol and Parameter Selection
SOF – Start Of Frame
UART – Universal Asynchronous Receiver/Transmitter
USB – Universal Serial Bus

3. Description of the communication protocol

The commands and their responses are transmitted on USB line. The microcontroller does not integrate a USB port directly. USB to serial line converter is used to translate the USB packets to the serial line (UART) of the microcontroller.

The firmware main loop periodically analyze the UART receive buffer and performs particular actions on valid messages. All performed actions or detected errors emit a response message. The UART data reception is performed asynchronously. No next message analysis is generated until the response on previous action is sent out.

3.1. Communication parameters

The communication parameters are unified. For more information, please, refer to the firmware sources Readme.txt (description serial line communication parameters).

3.2. Message format

All messages follow the next rules:

- Byte[0] = STX = 02h
- Byte[1] = index of checksum byte = last-1
- Byte[2] = command/response identification
- Byte[3..last-2] = payload
- Byte[last-1] = XOR checksum = Byte[1] XOR Byte[2] XOR .... XOR Byte[last-2]
- Byte[last] = ETX = 03h

3.3. PC to reader (Command message)

Each command sent by the reader to the tag is initialized by a supplied command from the PC software application. The PC command set comprises three groups:

1. ISO15693 mandatory, optional, custom and proprietary commands of the EM Microelectronic transponder ICs (available in firmware family 161 only)
2. ISO14443 Type A and B commands (available in firmware family 162 only)
3. Reader Control commands

The reader firmware can utilise the arbitrary data part (data byte items) of the command to further perform a specific operation (e.g., the flag byte is always used in ISO 15693 commands, the command byte, and other special information).

The commands supported by the reader firmware are described in the following table:

PC to reader	**Serial Data Bytes sent on UART**											
Command	0	1	2	3	4	...	...	...	...	XX	XX+1	
ISO 14443 Type A and B commands (available in firmware family 162 only)												
Transparent Type B commands	02h	XXh	63h	Resp. length	1st data byte	…	...	...	...	Last data byte	CHK	03h
Type B commands (REQB,ATTRIB, SR commands)	02h	XXh	65h	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
Type A Commands	02h	XXh	66h	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
Type B commands	02h	XXh	67h	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
Arbitrary Type A commands	02h	XXh	69h	Resp. length	LSB Delay time MSB		1st data byte	...	Last data byte	CHK	03h	
Type A Get UID (Ignore proprietary coding flags)	02h	03h	6Ah	CHK	03h							

PC to reader	Serial Data Bytes sent on UART											
Type A Get UID	02h	03h	6Bh	CHK	03h							
SR Initiate Command	02h	XXh	6Ch	Resp. length	RF Reset	1st data byte	...	...	Last data byte	CHK	03h	
SR Write & Verify Command	02h	XXh	6Dh	Resp. length	wr delay	1st data byte	...	...	Last data byte	CHK	03h	

PC to reader	Serial Data Bytes sent on UART											
Command	0	1	2	3	4	...	...	...	...	XX	XX+1	
ISO 15693 Commands and EM tag commands (available in firmware family 161 only)												
1TS Inventory with self-tuning	02h	XXh	80h	1st data byte	...	...	...	...	Last data byte	CHK	03h	
1TS Inventory with RF reset with self tuning	02h	XXh	81h	RF Reset	1st data byte	...	...	...	Last data byte	CHK	03h	
1TS Inventory with RF reset	02h	XXh	82h	RF Reset	1st data byte	...	...	...	Last data byte	CHK	03h	
1TS Inventory	02h	XXh	83h	1st data byte	...	...	...	...	Last data byte	CHK	03h	
SIM Card generic command	02h	XXh	84h	Direct-ion	SIM Resp. length	SIM time out	1st data byte	...	Last data byte	CHK	03h	
Stay Quiet	02h	XXh	85h	1st data byte	...	...	...	...	Last data byte	CHK	03h	
General Read	02h	XXh	88h	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
HW Authentication	02h	XXh	89h	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
Startup Inventory	02h	XXh	8Bh	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
HW Authentication w/o Selection	02h	XXh	8Ch	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	

PC to reader	Serial Data Bytes sent on UART											
Signed General Read	02h	XXh	8Eh	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
General Read with response length in bytes	02h	XXh	8Fh	Resp. size	1st data byte	...	...	...	Last data byte	CHK	03h	
General Write	02h	XXh	90h	Resp. length	LSB Delay time MSB		1st data byte	...	Last data byte	CHK	03h	
Signed General Write	02h	XXh	91h	Resp. length	LSB Delay time MSB		1st data byte	...	Last data byte	CHK	03h	
Signed General Write without signed response	02h	XXh	92h	Resp. length	LSB Delay time MSB		1st data byte	...	Last data byte	CHK	03h	

PC to reader	Serial Data Bytes sent on UART											
Command	0	1	2	3	4	...	...	...	...	XX	XX+1	
Reader Control commands												
Fwd Pulse Tuning (AB) (family 162 only)	02h	06h	EAh	idx	LSB val MSB		CHK	03h				
Customer level generic command	02h	XXh	EFh	1st data byte	…	…	…	…	Last data byte	CHK	03h	
RF Reset	02h	04h	F0h	RF Reset	CHK	03h						
Direct SPI Write	02h	04h	F1h	LSB Configuration word MSB			CHK	03h				
SPI Write with RF reset	02h	04h	F2h	RF Reset	LSB Configuration word MSB			CHK	03h			
Bootloader Mode	02h	04h	F3h	CHK	03h							
Send Debug Data	02h	04h	F6h	CHK	03h							
Get Raw Data (family 162 only)	02h	04h	F7h	CHK	03h							
Get Capture Data	02h	03h	F8h	CHK	03h							
Toggle Debug Mode	02h	04h	F9h	Dbg mode	CHK	03h						
Fwd Pulse Tuning (family 161 only)	02h	06h	FAh	idx	LSB val MSB		CHK	03h				
Reader Status	02h	03h	FDh	CHK	03h							
Switch Coil On/Off	02h	04h	FEh	coil	CHK	03h						

Note:

1. All values are in a hexadecimal format
2. FwdLink = 01h for 1 out of 4 forwardlink type
3. Coil = <0,3>, bit 0 controls MOD_PIN output, bit 1 controls EN output
4. Resp.length = number of response bits in case of positive response
5. Resp.size = number of response bytes in case of positive response
6. Delay time = <0,65535>, delay timing for write commands
7. RF Reset = <0,255>, time delay between Field OFF and Field ON
8. Dbg = 00h for Off, 01h for Raw mode, 02h for Decoded mode
9. Configuration word - see configuration word in EM4294 data sheet
10. Idx = <0,8>, index to forwardlink delay tables
11. val = <0,65535>, negative delay value
12. LSB, MSB – low endian bytes ordering
13. 4006_scale = <7,13> - see EM4006 chapter
14. Direction = <0,2> - SIM command direction (0-reset, 1-send, 2-read)
15. SIM Resp. length = number of SIM card response bytes
16. SIM timeout = SIM card timeout in number of ETU (+ tolerance)
17. RF Reset = <0,255>, time delay between Field OFF and Field ON
18. wr_delay = <1,255>additional write delay

3.4. Reader to PC (Response)

The response result is specified by ACK item of the response (see chapter 3.5). Each command has a specific set of possible ACK values.

Reader to PC	**Serial Data Bytes sent on UART**											
Response	0	1	2	3	4	5	6	7	8	…	YYh	YYh+1
					ISO 14443 Type A and B commands							
Transparent Type B Command	02h	YYh	63h	ACK	1st data byte	...	...	...	...	Last data byte	CHK	03h
Type B commands	02h	YYh	65h	ACK	1st data byte	...	...	...	...	Last data byte	CHK	03h
Type A Commands	02h	YYh	66h	ACK	1st data byte	...	...	...	...	Last data byte	CHK	03h
Type B commands	02h	YYh	67h	ACK	1st data byte	...	...	...	...	Last data byte	CHK	03h
Arbitrary Type A Commands	02h	YYh	69h	ACK	1st data byte	...	...	...	...	Last data byte	CHK	03h

Reader to PC	Serial Data Bytes sent on UART											
Type A Get UID (Ignore proprietary coding flags)	02h	03h	6Ah	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
Type A Get UID	02h	03h	6Bh	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
SR Initiate Command	02h	YYh	6Ch	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
SR Write & Verify Command	02h	YYh	6Dh	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h

Reader to PC	Serial Data Bytes sent on UART											
Response	0	1	2	3	4	5	6	7	8	…	YYh	YYh+1
ISO 15693 Commands and EM tag commands												
1TS Inventory with self-tuning	02h	YYh	80h	ACK	MSByte EM4294 Configuration word LSByte				1^{st} data byte ... Last data byte		CHK	03h
1TS Inventory with RF reset and self-tuning	02h	YYh	81h	ACK	MSByte EM4294 Configuration word LSByte				1^{st} data byte ... Last data byte		CHK	03h
1TS Inventory with RF reset	02h	YYh	82h	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
1TS Inventory	02h	YYh	83h	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
SIM Card generic command	02h	YYh	84h	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
Stay Quiet	02h	04h	85h	ACK	CHK	03h						
General Read	02h	YYh	88h	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
HW Authentication	02h	YYh	89h	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
Startup Inventory	02h	YYh	8Bh	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
HW Authentication w/o Selection	02h	YYh	8Ch	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h

Reader to PC	Serial Data Bytes sent on UART											
Signed General Read	02h	YYh	8Eh	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
General Read with response length in bytes	02h	YYh	8Fh	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
General Write	02h	YYh	90h	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
Signed General Write	02h	YYh	91h	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h
Signed General Write with unsigned response	02h	YYh	92h	ACK	1^{st} data byte	...	...	...	...	Last data byte	CHK	03h

Reader to PC	Serial Data Bytes sent on UART											
Response	0	1	2	3	4	5	6	7	8	…	YYh	YYh+1
Reader Control commands												
Fwd Pulse Tuning (AB)	02h	04h	EAh	ACK	CHK	03h						
Customer level generic command	User defined											
RF Reset	02h	04h	F0h	ACK	CHK	03h						
Direct SPI Write	02h	04h	F1h	ACK	CHK	03h						
SPI Write with RF reset	02h	04h	F2h	ACK	CHK	03h						
Bootloader Mode	02h	04h	F3h	ACK	CHK	03h						
Send Debug Data	02h	04h	F6h	CHK	Debug Data .... CHK 03h							
Get Raw Data	02h	04h	F7h	CHK	Raw Data .... CHK 03h							
Get Capture Data	02h	YYh	F8h	ACK	Capture Data+Valid .... CHK 03h							
Toggle Debug Mode	02h	04h	F9h	ACK	CHK	03h						
Fwd Pulse Tuning	02h	04h	FAh	ACK	CHK	03h						
Reader Status	02h	08h	FDh	ACK	Release	Date		Version	CHK	03h		
Switch Coil On/Off	02h	04h	FEh	ACK	CHK	03h						
General error	02h	04h	00h	ACK	CHK	03h						

Note:

1. All values are in hexadecimal format
2. data = decoded data if ACK= {00h, 0Eh}, raw data if ACK = 0x10h, otherwise Capture Data + Valid bits

3.5. ACK byte

If the ACK is set to 00h, it signalises a successful execution of the command, otherwise non-zero ACK values signalise errors or other information.

UART errors are common to all the commands. They indicate a problem during PC<->Reader communication or protocol errors.

Generally, the commands, that communicate with the tags successfully (i.e., ACK = 00h), return a data bytes already decoded.

The commands, that fail to communicate with the tags, usually return a Raw Data or Capture Data with the contents of internal communication buffer. Data part of failing command is not strictly bound to the ACK value and should be treated as specific to each command.

The response data type can be also conditioned by a debug or raw mode activation. This feature is also specific to each command.

ACK value	Name	Fault from part (Description)
00h	UART_MESSAGE_OK	All parts (OK)
01h	ERR_ASIC_ANTENNA_FAULT	ASIC + Reader (timeout during capture process, invoked by watchdog)
04h	ERR_UART_ERROR_FLAG	UART (none or wrong STX, parity error)
05h	ERR_UART_OVERFLOW	UART (UART buffer overflow)
06h	ERR_UART_WRONG_ICMD	Reserved
07h	ERR_UART_BAD_CRC	UART (Wrong CHK)
08h	ERR_UART_UNKNOWN_CMD	UART (Unknown command)
09h	ERR_UART_NO_ETX	UART (No ETX)
0Ah	ERR_UART_INTERBYTE_ERR	UART (message length is out of range, message length is wrong with this command)
0Bh	ERR_EM4035_WRONG_LEN	Reader (Wrong Response Length, not enough data, wrong demodulation parameters)
0Ch	ERR_EM4035_NO_EOF	Reserved (No EOF)
0Dh	ERR_WRONG_DR	Reader (4006_scale out of range)
0Eh	ERR_ISO_ERROR_MSG	Reader (tag error - flag byte is not 00h)
0Fh	ERR_EM4035_BAD_CONF_DATA	Reserved
10h	ERR_RAW_DATA	Reader (response contains raw data)
11h	ERR_CAPT_DATA	Reader (response contains captured data pairs)
12h	ERR_NO_TAG	Reader (no EM4006 uid found)
13h	ERR_BAD_CRC	Reader (bad response CRC)

ACK value	Name	Fault from part (Description)
14h	ERR_INV_BUFFER_OVERFLOW	Reader (not enough memory to track all the tags)
15h	ERR_NO_SOF	Reader (no response received at all)
16h	ERR_EM4035_SELECT_FAILED	Reader (HW Authentication Select command failed)
17h	ERR_EM4035_A1_NO_SOF	Reader (HW Authentication Step 1 no response SOF)
18h	ERR_EM4035_A1_CRC_ERROR	Reader (HW Authentication Step 1 bad response CRC)
19h	ERR_EM4035_A1_FAILED	Reader (HW Authentication Step 1 wrong response data)
1Ah	ERR_EM4035_A2_NO_SOF	Reader (HW Authentication Step 2 no response SOF)
1Bh	ERR_EM4035_A2_CRC_ERROR	Reader (HW Authentication Step 2 bad response CRC)
1Ch	ERR_EM4035_A2_FAILED	Reader (HW Authentication Step 2 wrong response data)
1Dh	ERR_EM4035_AUTH_FAILED	Reader (HW Authentication failed, wrong password)
1Eh	ERR_EM4035_SIM_SELECT_FAILED	Reader(SIM card authentication - select key failed)
1Fh	ERR_EM4035_SIM_A1_FAILED	Reader (SIM card authentication - send A1 phase failed)
20h	ERR_EM4035_SIM_A2_FAILED	Reader (SIM card authentication - get A2 phase failed)
21h	ERR_EM4035_SEND_G_FAILED	Reader (SIM card authentication - send G phase failed)
22h	ERR_EM4035_SIM_SIGN_FAILED	Reader (SIM card signing - operation failed)
30h	ERR_EM4035_TAG_NORMAL	Reader (Tag already in normal mode)
31h	ERR_EM4035_NM_READ_FAILED	Reader (Normal mode read failed - Read Lock block failed)
32h	ERR_EM4035_NM_AUTH_FAILED	Reader (Normal mode write failed - Write Lock block failed)
33h	ERR_EM4035_NM_CRC_ERROR	Reader (Normal mode CRC error)
40h	ERR_SIM_NOT_DETECTED	Reader (SIM Card not detected)
41h	ERR_SIM_NOT_ENOUGH_DATA	Reader (SIM card response is too short)
42h	ERR_SIM_PROCEDURE_BYTE	Reader (SIM Card Wrong procedure byte)
43h	ERR_SIM_NOT_INITIALISED	Reader (SIM Card Reset sequence was not performed yet)
44h	ERR_SIM_PPS_FAILED	Reader (SIM Card PPS sequence not supported or failed)
50h	ERR_A_GETUID_REQA_FAILED	Reader (Get UID REQA failed)
51h	ERR_A_GETUID_REQA_NA	Reserved
52h	ERR_A_GETUID_SEL0_FAILED	Reader (Get UID - Sel of 1st cascade failed)
53h	ERR_A_GETUID_SEL1_FAILED	Reader (Get UID - Sel of 2nd cascade failed)
54h	ERR_A_GETUID_SEL2_FAILED	Reader (Get UID - Sel of 3rd cascade failed)

ACK value	Name	Fault from part (Description)
55h	ERR_A_GETUID_SELECT0_FAILED	Reader (Get UID - Select of 1st cascade failed)
56h	ERR_A_GETUID_SELECT1_FAILED	Reader (Get UID - Select of 2nd cascade failed)
57h	ERR_A_GETUID_SELECT2_FAILED	Reader (Get UID - Select of 3rd cascade failed)
58h	ERR_A_GETUID_SELECT0_CRC	Reader (Get UID - Select of 1st cascade CRC failed)
59h	ERR_A_GETUID_SELECT1_CRC	Reader (Get UID - Select of 2nd cascade CRC failed)
5Ah	ERR_A_GETUID_SELECT2_CRC	Reader (Get UID - Select of 3rd cascade CRC failed)
FEh	(ERR_)INVENTORY_FINISHED	Reader (No other tags - 1TS Inventory algorithm finished)

3.6. Antenna fault (01h) error

The Antenna fault (01h) error is common to all the commands. Antenna fault error is generated automatically on microcontroller watchdog time-out after 2.1s. It signalises that the operation could not be terminated standard way.

The known operation that could not be terminated standard way is a data capture process of communication commands that uses an interrupt. Due to the interrupt priority scheduling and with a certain type of input data signals, the data capture process stop condition has less priority to be executed. Therefore, watchdog is used to interrupt the data capture process, and Antenna Fault (01h) error has to be treated as standard result.

In current firmware release, ISO15693 capture routines are driven by interrupt events, causing this problem when a tag with EAS on is placed into the RF field.

4. Command Description

Following sub-chapters describe each command's behaviour and its possible errors. *Requires* field defines expected data format structure, *Accessed items* defines what data items are accessed by the firmware, and *Errors* lists possible error results (UART communication errors are common to all the commands and are omitted here, antenna fault (01h) error is mentioned only when it is used to signalise the operation timeout). *Supports* field describes a limitation of the command. *Availability* filed defines a compile conditional code. *Example* field shows an example command.

4.1. 1TS Inventory with RF reset and self-tuning (81h)

Current one time slot inventory algorithm searches active tags present in the RF field resetting the RF field first. The RF reset is performed so that all the tags are switched to the Ready state. See 83h command for details.

4.2. 1TS Inventory (83h)

Current one time slot inventory algorithm searches active tags present in the RF field. 1 time slot ASK is supported only (one single sub-carrier mode).

Response of the command contains one tag UID (actually the response of Inventory command), data item is valid only if ACK = 0. If two or more tags are detected successfully, the same number of responses is generated. The last response contains ACK = FEh only.

Current inventory routine is not based on a detection of the collision position. It performs a binary tree search.

Inventory routine starts with zero mask length. The routine sends ISO15693 Inventory command with current mask and mask length settings. According to the result of the response received, the routine updates the mask and the mask length until timeout is reached

1. If a single UID is found, mask is "stepped back"
2. If UID buffer overflows, the inventory process is terminated with ERR_INV_BUFFER_OVERFLOW (14h) error. UID stack storage buffer is dimensioned for 8 UIDs.
3. If a collision is found, mask is "stepped forward" by 1 and further Inventory command is sent
4. If no response is received, mask is "stepped back". If no response is received at all 7 times, the mask length is zeroed causing the search to "fast restart"

For more information, please, refer to the higher level Inventory routine diagram in Figure 1 Inventory flow.

There are three operations over the mask and mask length used:

1. "Slow step back" is a modification of current mask and mask length so that all branches are traversed. If the last mask bit is '0', it is toggled to '1'. Otherwise, (the last mask bit is '1'), mask length is decremented by 1 and "slow step back" is repeated
2. "Slow step forward" is a modification of current mask and mask length so that all the '0' sub-tree is traversed first. The mask length is incremented by 1 and this last mask bit is set to '0'.
3. "Fast restart" is a reset of mask length to '0'.

Inventory_step routine performs a single inventory query. Inventory_step routine builds an ISO 15693 Inventory command according to the current flag byte, AFI value, mask, and mask length.

It sends it to the tags in the RF field and receives a response. If a clean single tag UID is received, it stores the tag UID into the array of found UIDs and builds and sends a Stay Quiet command to this tag.

Requires: ISO15693 unaddressed ASK 1 time slot inventory command with mask_length = 0, supplied by the application software. AFI field is optional.

Example: ISO15693 1TS ASK Inventory (ISO15693 packet in **bold**) - `02 08 83` **`26 01 00 F6 0A`** `50 03`

Accessed items: flag byte, AFI value

Errors: 14h, FEh

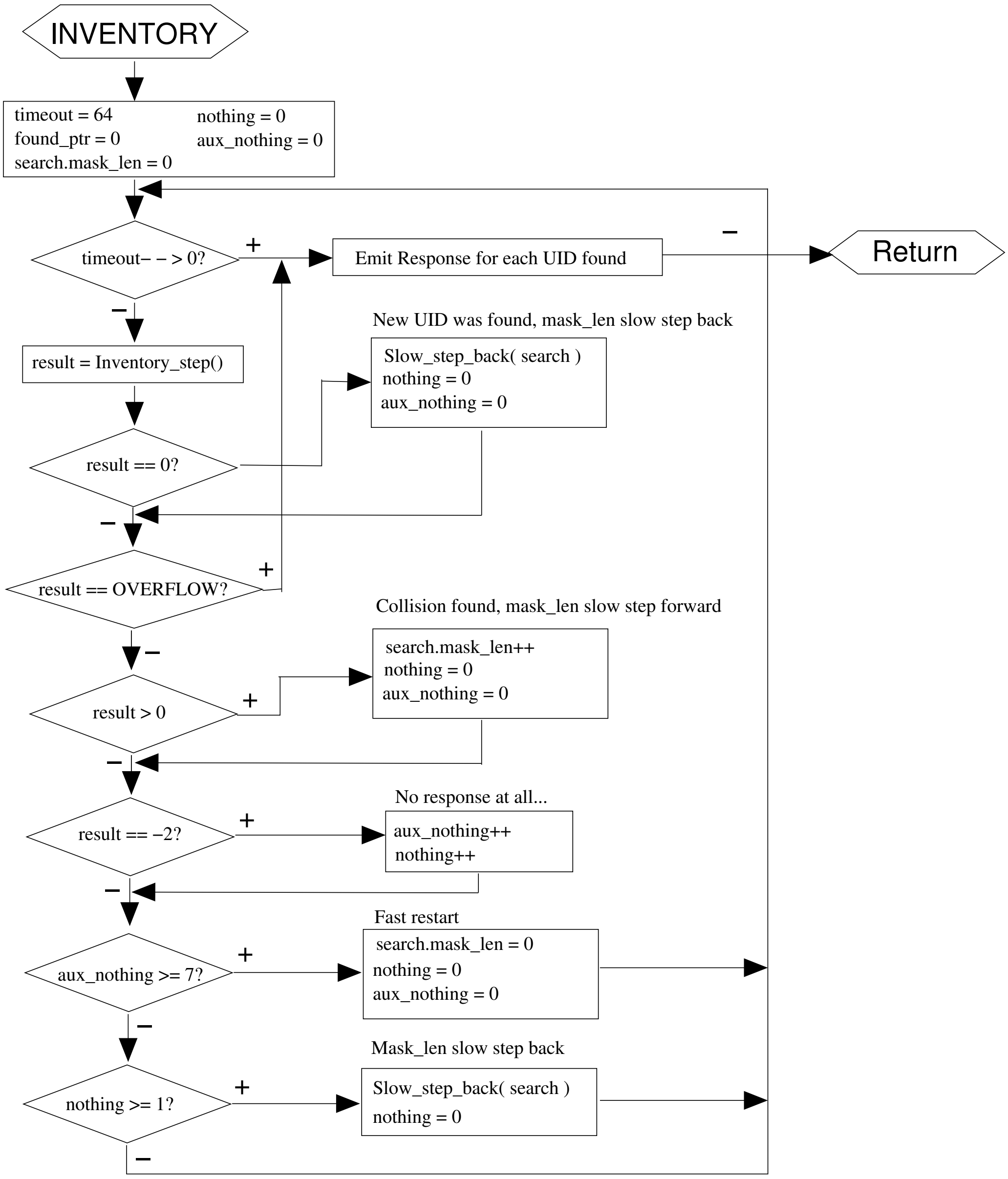

Figure 1 Inventory flow

4.3. 1TS Inventory with RF reset (82h)

Current one time slot inventory algorithm searches active tags present in the RF field reseting the RF field first. The RF reset is performed so that all the tags are switched to the Ready state. See 83h command for details.

4.4. 1TS Inventory with self-tuning (80h)

1TS Inventory command with self-tuning (80h) is another implementation of 1TS Inventory command (83h). Self-tuning inventory routine is started with current EM4294 configuration word. During the Inventory routine loop, if the aux_nothing counter value exceeds its limit or if the noise is detected, the EM4294 configuration is changed to one of three hard coded EM4294 constants (see **Figure 2 Inventory flow with self-tuning**). These three EM4294 configuration words were selected to have the best data reception level either for ISO card size transponders either for small size transponders. At the end, the current (i.e. before start) EM4294 configuration word is restored.

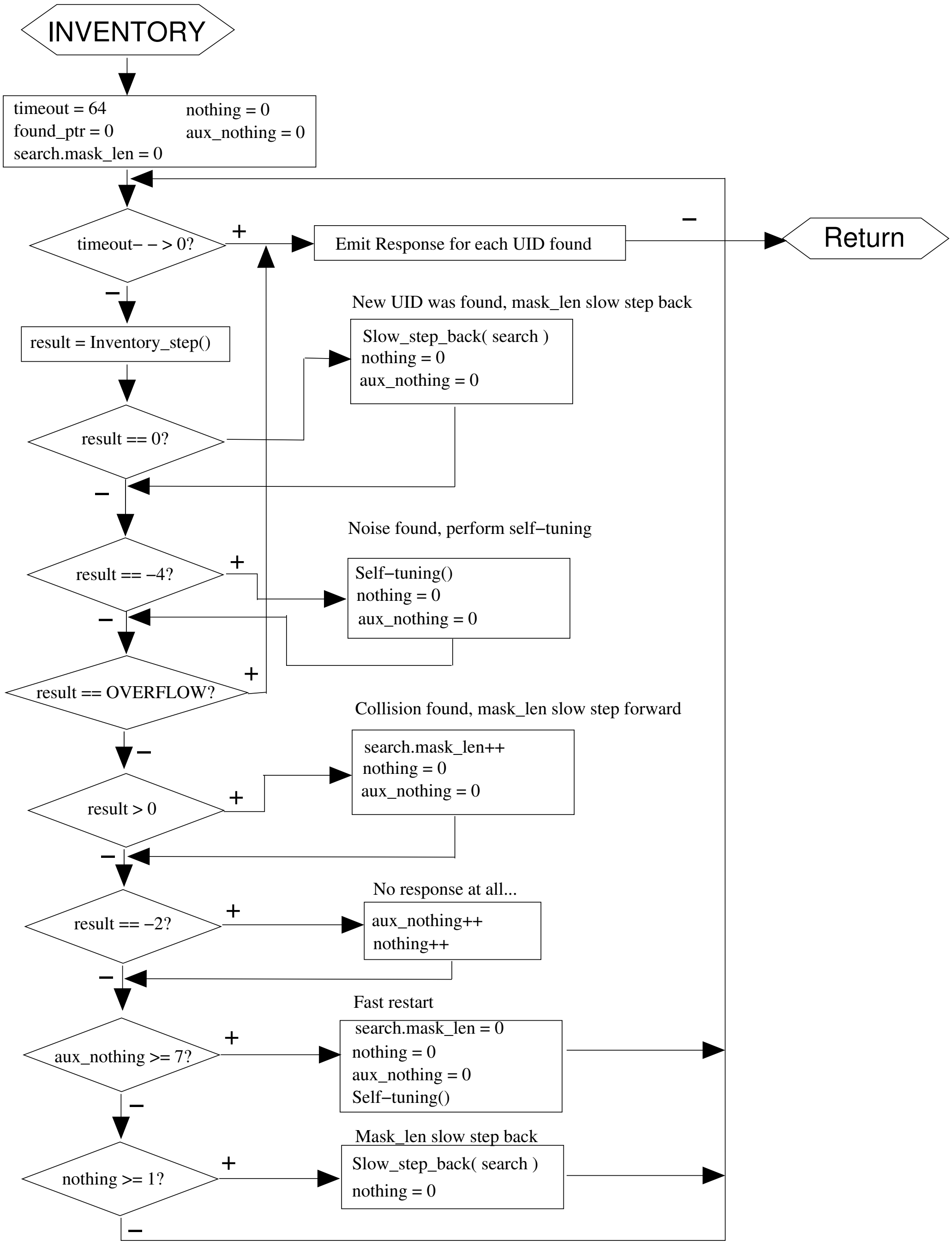

Figure 2 Inventory flow with self-tuning

As a result, each UID response contains actual EM4294 configuration it was found with.

The last response contains ACK = FEh only. Generally, the following actions that are required before further communication result from the following cases;

- All the tags were found with a single (and default) configuration word – no actions are not necessary
- All the tags were found with a single (non default) configuration word – the received configuration word has to be used (see F1h command) to communicate with these tags
- All the tags were found with more than one configuration word – the inventory process encountered a low reception level during the communication with all the tags. It is recommended to remove some tags from the RF field and repeat the inventory process with self-tuning.
- No tag was found – no information can be stated

Actual EM4294 configuration word constants derivation:

Along the currently set EM4294 configuration word, the three more constants are derived from this configuration word. Current firmware derives three constants that differ in the Gain settings only so that the most feasible reception settings are used during the inventory process. By means of such derivation, either the OOK or the ASK uplink modulation can be achieved.

4.5. SIM Card generic command (84h)

SIM Card generic command performs either the SIM card reset or SIM card read/write operation following the ISO 7816-3 norm (T=0 communication mode). The SIM card IC is already built-in the EMDB410 Reader as the part of EM4294 IC. The SIM card firmware is expected to support PPS command.

If Direction parameter is set to **0**, the SIM card reset is performed with a firmware limited timeout.

If Direction parameter is set to **1**, the firmware sends a SIM card header (first 5 bytes of data) to the SIM card and listens for a procedure byte acknowledge from the SIM card. If the procedure byte does not equal the expected one, ERR_SIM_PROCEDURE_BYTE (42h) error is returned. After receiving correct procedure byte (i.e.; the procedure byte matches the INS [2nd] byte of the header), the firmware continues to send the rest of data and then it listens for status bytes (status bytes normal ending is indicated by ‘9000’). SIM resp. length specifies a number of response bytes including a procedure byte without a status bytes.

If Direction parameter equals **2**, the firmware sends a SIM card header (first 5 bytes of data) to the SIM card and listens for a procedure byte acknowledge.If the procedure byte does not equal the expected one, ERR_SIM_PROCEDURE_BYTE (42h) error is returned. After receiving correct procedure byte the data from the SIM card is received including status bytes. SIM resp. length specifies a number of response bytes including a procedure byte plus 1.

If the SIM card reset is not yet performed or it fails, ERR_SIM_NOT_DETECTED (40h) error is returned. If there is not enough data in the response, ERR_SIM_NOT_ENOUGH_DATA (41h) error is returned.

If Direction parameter is set to **3**, the SIM card reset is performed with a firmware limited timeout. If the SIM card is detected successfully and its ATR response reports the TA(1) = 0x95, the PPS command with PPS1 = 0x95 is sent to the card. If correct PPS response is obtained from the SIM card, the further SIM card communication is performed with the speed corresponding to the FI = 9 and DI = 5 (i.e. Fi = 512 and Di = 16 => 1 ETU = Fi / (Di * f) = 512 / (16 * f) ~ 32 SIM clocks per 1 ETU). In case of any error, the communication speed remains set to 372 SIM clocks per 1 ETU.

Before the transmission of Direction = **2** and **3** commands, the SIM card auto detection including the PPS command is performed automatically if the SIM card reset has not been performed yet.

Requires: SIM card crypto engine command

Errors: 40h, 41h, 42h, 43h, 44h

4.6. Stay Quiet (85h)

Stay Quiet command sends the command supplied by the application software. The command is sent, no response is analysed at all.

Requires: Command formed by the application software

Errors: -

4.7. General Read (88h)

General Read command sends the command supplied by the application software, waits for T1 time (~318 us), and tries to capture the response. Response of expected Resp. length is analysed after T1 time. The response check (format, data, and CRC check) is left on the application software. General Read process if following:

1. If the command is Secure Read EM4035 Proprietary command (0xE2), the CRC of supplied command is signed by the crypto engine
2. Send the command, wait for T1 time and capture response
3. If there is no response at all, return ERR_NO_SOF (15h) error
4. Find and extract the decoded data
5. If the number of adjacent valid response data bits is higher than 32, and the first byte is not zero, consider the message to be an error message, return response data and ERR_EM4035_ERROR_MSG (0Eh) error. If the response is signed (optionFlag of the Secure Read command was set to '1'), unsign the last two bytes of the response so that the crypto engine is synchronized even if there is either the negative response or any other problem.
6. If the number of response data bits is less than expected Resp. length, return decoded response data and ERR_EM4035_WRONG_LEN (0Bh) error. No crypto engine synchronization is done because the response is corrupted.

7. Clamp the response to expected Resp. length divided by 8 byte boundary. If the response is signed (optionFlag of the Secure Read command was set to '1'), unsign the last two bytes of the response so that the crypto engine is synchronized. Return response data and ACK = 00h (no error)

Requires: Command formed by the application software

Example: Unaddressed read (ISO15693 packet in **bold**) - `02 0A 88 58` **`02 23 0C 00 57 80`** `20 03`

`Supports:` Up to 3 64bit blocks or up-to 6 32bit blocks can be read at once.

Accessed items: flag byte, command byte

Errors: 0Bh, 0Eh, 10h, 11h, 15h

4.8. HW Authentication (89h)

HW Authentication command performs the complete authentication to one EM4233 2k tag in High Security mode or EM4035 tag using the crypto engine. This command requires the input data containing the addressed Authentication Step 1 Proprietary command. HW Authentication process is following:

- Form the Select ISO command to move the tag to selected state
- Send the Select ISO command and capture the response
- If there is no response, return ERR_EM4035_SELECT_FAILED (16h) error
- If the response length does not equal 3 bytes or the response flags byte is not 00h or the response CRC is bad, return decoded response data and ERR_EM4035_SELECT_FAILED (16h)
- Form the selected Authentication Step 1 Proprietary command reusing ICMfg, key number, flag byte from the input data
- Send the selected Authentication Step 1 Proprietary command and capture response
- If there is no response, return ERR_NO_SOF (15h) error
- If the response contains no valid data byte, return decoded response data and ERR_EM4035_A1_FAILED (19h) error
- If the response CRC is bad, return decoded response data and ERR_EM4035_A1_CRC_ERROR (18h) error
- If the response length is different from 10 bytes or the response flag byte is not zero, return response and ACK = 00h
- If the SIM card was not reset yet, perform the SIM card reset operation including the PPS command
- Reselect the crypto engine key
- Send A1 constant to crypto engine
- Receive A2 constant, 8 dummy bits, and f() constant from crypto engine
- Form the selected Authentication Step 2 Proprietary command reusing ICMfg, key number, flag byte from the input data, A2 constant, and f(). There is no delay loop to match Tprnd + Twee time as the SIM card communication is very slow
- Send the selected Authentication Step 2 Proprietary command and capture response
- If there is no response, return ERR_EM4035_A2_NO_SOF (1Ah) error
- If the response contains no valid data byte, return decoded response data and ERR_EM4035_A2_FAILED (1Ch) error
- If the response CRC is bad, return decoded response data and ERR_EM4035_A2_CRC_ERROR (1Bh) error
- If the response flag byte is not zero, return response and ACK = 00h
- Send a g() constant to the crypto engine. If authentication passes, . In such case, the crypto engine state is synchronized with the tag state, return response and ACK=00h, otherwise return decoded response data and ERR_EM4035_AUTH_FAILED (1Dh) error

There are two key sets available in the crypto engine (see chapter 6), user key set is numbered as defined in EM4035 datasheet. Initialisation key set number is the EM4035 key number value shifted 4 bits to the left. The firmware recognizes the initialisation keys and remaps them into EM4035 key number value before sending it to the EM4035 tags. EM4233 2k tag has only one key when configured in High Security mode, the firmware always uses the key no.2 of the crypto engine.

Requires: Addressed Authentication Step 1 Proprietary command

Accessed items: flag byte, command byte, key number, ICMfg byte

Errors: 15h, 16h, 18h, 19h, 1Ah, 1Bh, 1Ch, 1Dh

4.9. Startup Inventory (8Bh)

Startup Inventory command switches the RF field off for 64 ms, then it switches the RF field on and performs General Read command (88h). This command can be used to obtain the uid of the tag which has the EAS enabled.

Requires: Command formed by the application software (e.g.; ISO15693 Inventory command)

Accessed items: flag byte, command byte

Errors: 0Bh, 0Eh, 10h, 11h, 15h

4.10. HW Authentication w/o Selection (8Ch)

HW Authentication w/o Selection command performs an authentication process using crypto engine. This command uses an unaddressed mode for all its steps, unlike the HW Authentication (89h) command.

Requires: Unaddressed Authentication Step 1 Proprietary command

Accessed items: flag byte, command byte, key number, IC Manufacturer byte

Errors: 15h, 16h, 18h, 19h, 1Ah, 1Bh, 1Ch, 1Dh

4.11. Signed General Read (8Eh)

Signed General Read command signs the CRC of the command supplied by the application software, sends it to the tag and waits for T1 time (~318 us), then it tries to capture the response. Response of expected Resp. length is analysed after T1 time. Valid response CRC checksum is unsigned. The response check (format, data, and unsigned CRC check) is left on the application software. General Read process if following:

1. The CRC of supplied command is signed by the crypto engine
2. Send the command, wait for T1 time and capture response
3. If there is no response at all, return ERR_NO_SOF (15h) error
4. Find and extract the decoded data
5. If the number of adjacent valid response data bits is higher than 32, and the first byte is not zero, consider the message to be an error message, return response data and ERR_EM4035_ERROR_MSG (0Eh) error. Unsign the last two bytes of the response so that the crypto engine is synchronized even if there is either the negative response or any other problem.
6. If the number of response data bits is less than expected Resp. length, return decoded response data and ERR_EM4035_WRONG_LEN (0Bh) error. No crypto engine synchronization is done because the response is corrupted.
7. Clamp the response to expected Resp. length divided by 8 byte boundary. Unsign the last two bytes of the response so that the crypto engine is synchronized. Return response data and ACK = 00h (no error)

Requires: Command formed by the application software

Example: Unaddressed read (ISO15693 packet in **bold**) - 02 0A 8E 58 **02 23 0C 00 57 80** 20 03

Supports: Up to 3 64bit blocks or up-to 6 32bit blocks can be read at once.

Accessed items: flag byte

Errors: 0Bh, 0Eh, 10h, 11h, 15h

Note: this command shall be used for the read-like commands of EM4233 tag in High Security mode after the successful authentication.

4.12. General Read with response length in bytes (8Fh)

General Read command with response length in bytes works the same way as the General Read command except that the Resp.size parameter defines the expected response length in bytes (includes flag byte and two CRC16 bytes). This command allows reading up to 14 64bit words at once while General Read command allows reading of 3 64bit words only at once. This command supports the EM4035 Secure Read access, allowing a read of 12 {64b+Security byte} at once.

Requires: Command formed by the application software

Accessed items: flag byte

Example of reading 12 words of EM4035 starting from block no.13:

Command: 02 0A 8E 63 02 23 0D 0B 5C 27 BB 03 (Note: Expected response length is 12*8+3=99=0x63 bytes.)

Response: 02 67 8E 00 00 45 23 01 00 00 00 0B 00 00 00 00 00 00 00 00 00 EF BE BE BA FE CA CD AB 00 00 00 00 00
00 00 00 22 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 56 3B DC 03

Note: this command is useful for EM4233 in non authenticated mode. It cannot be used for commands to authenticated tags in High Security mode.

4.13. General Write (90h)

The General Write command sends the command supplied by the application software, wait for T1 time (~318 us), and tries to capture the response of the Delay time duration, the expected response length is defined by Resp. length value. The response check (format, data, and CRC check) is left on the application software. General Write process if following:

1. If the command is Secure Write EM4035 Proprietary command (0xE3), the CRC of supplied command is signed by the crypto engine
2. Send the command
3. If the Option flag is set and write command is one of Write block, Lock block, and Write AFI, wait for a supplied delay time, send single EOF and capture the response; otherwise, wait for T1 time and capture response during supplied Delay time period
4. If there is no response at all, return ERR_NO_SOF (15h) error
5. Find and extract the decoded data
6. If the number of response data bits is higher than 32, and the first byte is not zero, consider the message to be an error message, return response data and ERR_EM4035_ERROR_MSG (0Eh) error. If the response is signed (optionFlag of the Secure Read command was set to '1'), unsign the last two bytes of the response so that the crypto engine is synchronized even if there is either the negative response or any other problem.
7. If the number of response data bits is less than expected Resp. length, return decoded response data and ERR_EM4035_WRONG_LEN (0Bh) error. No crypto engine synchronization is done because the response is corrupted.
8. Clamp the response to expected Resp. length divided by 8 byte boundary. If the response is signed (optionFlag of the Secure Read command was set to '1'), unsign the last two bytes of the response so that the crypto engine is synchronized. Return response data and ACK = 00h (no error)

Requires: Command formed by the application software

Accessed items: flag byte, command byte

Errors: 0Bh, 0Eh, 10h, 11h, 15h

4.14. Signed General Write (91h)

Signed General Write command signs the CRC checksum of the command supplied by the application software, sends it to the tag and waits for T1 time (~318 us), then it tries to capture the response of the Delay time duration, the expected response length is defined by Resp. length value. Valid response CRC is unsigned. The response check (format, data, and CRC check) is left on the application software. General Write process if following:

1. The CRC of supplied command is signed by the crypto engine
2. Send the command
3. If the Option flag is set and write command is one of Write block, Lock block, and Write AFI, wait for a supplied delay time, send single EOF and capture the response; otherwise, wait for T1 time and capture response during supplied Delay time period
4. If there is no response at all, return ERR_NO_SOF (15h) error
5. Find and extract the decoded data
6. If the number of response data bits is higher than 32, and the first byte is not zero, consider the message to be an error message, return response data and ERR_EM4035_ERROR_MSG (0Eh) error. Unsign the last two bytes of the response so that the crypto engine is synchronized even if there is either the negative response or any other problem.
7. If the number of response data bits is less than expected Resp. length, return decoded response data and ERR_EM4035_WRONG_LEN (0Bh) error. No crypto engine synchronization is done because the response is corrupted.
8. Clamp the response to expected Resp. length divided by 8 byte boundary. Unsign the last two bytes of the response so that the crypto engine is synchronized. Return response data and ACK = 00h (no error)

Requires: Command formed by the application software

Accessed items: flag byte, command byte

Errors: 0Bh, 0Eh, 10h, 11h, 15h

Note: this command shall be used for write-like commands to communicate with EM4233 tag in High Security mode after the successful authentication.

4.15. Signed General Write without signed response (92h)

Signed General Write command without signed response signs the CRC checksum of the command supplied by the application software, sends it to the tag and waits for T1 time (~318 us), then it tries to capture the response of the Delay time duration, the expected response length is defined by Resp. length value. The response check (format, data, and CRC check) is left on the application software. General Write process if following:

9. The CRC of supplied command is signed by the crypto engine
10. Send the command
11. If the Option flag is set and write command is one of Write block, Lock block, and Write AFI, wait for a supplied delay time, send single EOF and capture the response; otherwise, wait for T1 time and capture response during supplied Delay time period
12. If there is no response at all, return ERR_NO_SOF (15h) error
13. Find and extract the decoded data
14. If the number of response data bits is higher than 32, and the first byte is not zero, consider the message to be an error message, return response data and ERR_EM4035_ERROR_MSG (0Eh) error. Unsign the last two bytes of the response so that the crypto engine is synchronized even if there is either the negative response or any other problem.
15. If the number of response data bits is less than expected Resp. length, return decoded response data and ERR_EM4035_WRONG_LEN (0Bh) error. No crypto engine synchronization is done because the response is corrupted.
16. Clamp the response to expected Resp. length divided by 8 byte boundary. Return response data and ACK = 00h (no error)

Requires: Command formed by the application software

Accessed items: flag byte, command byte

Errors: 0Bh, 0Eh, 10h, 11h, 15h

Note: this command shall be used for EM4233 write-like commands that always return unsigned response (EM4233 tag in High Security mode after the successful authentication).

4.16. Transparent Type B Commands (63h)

The Transparent Type B command (63h) transmits arbitrary Type B command (REQB, ATTRIB, SR176 Select, SR176 Read, SR176 Completion, etc.). Then, it captures the arbitrary length response. If Type B response data formatting error is found, the captured data and ERR_CAPT_DATA (11h) error is returned, otherwise Type B decoded data is returned. Moreover, the response decoding routine also accepts the responses with SOF missing. The response check (format, data, and CRC check) is left on the application software. Delay parameter defines total response capture timeout duration in 16us steps. Resp. length parameter is just used as a flag of the response reception, zero value means no response capture is performed.

Requires: Type B command formed by the application software

Supports: 106kb/s only.

Errors: 11h.

4.17. Type B Commands (65h)

The Type B command (65h) transmits arbitrary Type B command (REQB, ATTRIB, SR176 Select, SR176 Read, SR176 Completion). Then, it captures the response of the Response length bytes. If Type B response data formatting error is found or the decoded bytes number does not equal Resp.length value, the captured data and ERR_CAPT_DATA (11h) error is returned, otherwise Type B decoded data is returned. The response check (format, data, and CRC check) is left on the application software.

Requires: Type B command formed by the application software

Supports: 106kb/s only.

Errors: 11h.

4.18. Type A Commands (66h)

Type A Commands command transmits arbitrary Type A command. Then, it captures the response of the Resp.length bytes. If Type A response data formatting error is found or the decoded bytes number does not equal Resp.length, the captured data and ERR_CAPT_DATA (11h) error is returned, otherwise Type A decoded data is returned. The response check (format, data, and CRC check) is left on the application software.

Requires: Type A Command formed by the application software

Supports: 106kb/s only.

Errors: 11h.

4.19. Type B Commands (67h)

Type B Commands (67h) command transmits arbitrary Type B command. Then, it captures the response of Resp.length bytes. If Type B response data formatting error is found or the decoded bytes number does not equal {Resp.length, 5}, the captured data and ERR_CAPT_DATA (11h) error is returned, otherwise Type B decoded data is returned. This command is an extension of ATTRIB Command (65h) to get general tag error responses in decoded format.

The response check (format, data, and CRC check) is left on the application software.

Supports: 106kb/s only.

Errors: 11h.

4.20. Arbitrary Type A Commands (69h)

Arbitrary Type A Commands command transmits arbitrary Type A command. Then, it captures the arbitrary length response. If Type A response data formatting error is found, the captured data and ERR_CAPT_DATA (11h) error is returned, otherwise Type A decoded data is returned. The response check (format, data, and CRC check) is left on the application software. Delay parameter defines additional delay to the response capture timeout. Resp.length parameter is not used. This command can be used for application part data communication with ISO14443 Type A tags.

Requires: Type A Command formed by the application software

Supports: 106kb/s only.

Errors: 11h.

4.21. Type A Get UID Command – Ignore proprietary coding flags (6Ah)

Type A Get UID Command performs an Initialisation flowchart for PCD according to the ISO 14443-3 norm assuming there is a single Type A tag in the RF field. Unlike the Type A Get UID Command (6Ah), this command ignores the proprietary coding flags and continues the execution so that the UID of tags that support standard flowchart and have proprietary coding flags set can be read.

4.22. Type A Get UID Command (6Bh)

Type A Get UID Command performs an Initialisation flowchart for PCD according to the ISO 14443-3 norm assuming there is a single Type A tag in the RF field.

The process starts with sending REQA command. If ATQA response is not received and decoded successfully, ERR_A_GETUID_REQA_FAILED (50h) error is returned. If ATQA response proprietary coding bits are set or bit frame anti-collision has not a valid value, the Get UID data and UART_MESSAGE_OK (00h) is returned (ATQA item is valid only). Then, up to three cascades are traversed to obtain a complete Type A UID, this process includes SEL and SELECT commands. If any processed response is not received and decoded successfully, appropriate error (52h-5Ah) error is returned. The cascade traversing process is terminated when a valid SAK with UID complete flag set is received.

If the application software receives UART_MESSAGE_OK (00h), the response data has following format;

Response Offset	4-5	6-10	11-15	16-20	21	22	23
Item	lsB ATQA msB	Cascade 93 UID part + BCC	Cascade 95 UID part + BCC	Cascade 97 UID part + BCC	SAK 93	SAK 95	SAK 97
Example	04 00	F2 81 3E A3 AE	XX XX XX XX BC	XX XX XX XX BC	08	XX	XX

Supports: 106kb/s only.

Errors: 11h, 50h, 52h-5Ah.

4.23. SR Initiate Command (6Ch)

The SR Initiate command initialises the SR176 tag. Before the command is sent, an automatic RF field reset is performed according to RF Reset value. Then, the firmware captures the response of Resp.length bytes. If Type B response data formatting error is found or the decoded bytes number does not equal Resp.length, the captured data and ERR_CAPT_DATA (11h) error is returned, otherwise Type B decoded data is returned. The response check (format, data, and CRC check) is left on the application software.

Requires: SR176 Initialise command formed by the application software

Example: (SR176 packet structure in **bold**) - 02 09 6C 03 01 **06 00 97 5B** AD 03

Supports: 106kb/s only.

Errors: 11h.

4.24. SR Write & Verify Command (6Dh)

The SR Write & Verify command performs a write and re-read of a single block to SR176 tag. After the supplied Write command is sent, the firmware waits about 4.5ms plus optional wr_delay ~0.5ms steps. Then, the firmware sends automatically formed Read command (**08 aa CC CC**) and captures the response of Resp.length bytes. If Type B response data formatting error is found or the decoded bytes number does not equal Resp.length, the captured data and ERR_CAPT_DATA (11h) error is returned, otherwise Type B decoded data is returned. The response check (format, data match, and CRC check) is left on the application software.

Requires: SR176 Write command formed by the application software

Example: (SR176 packet structure in **bold**) - 02 0B 6D 04 01 **09 00 34 12 EC E7** 47 03

Supports: 106kb/s only.

Errors: 11h.

4.25. Fwd Pulse Tuning (AB) (EAh)

Timing of forward link DIN signal during ISO14443 forward link communication can be set by Fwd Pulse Tuning command. New value of selected timing value can be set by this command. Actual value corresponds to the negative number of the microcontroller clocks minus interrupt overhead before the new interrupt is raised. Generally, actual timings values may differ between firmware releases. Intended for test engineer only.

Availability: If enabled during firmware compilation.

Errors: none

4.26. Customer level generic command (EFh)

This command sends the user defined data to the Customer level defined function defined in custom_level.c source file;

```
void exec_customer_command(uint8_t ui_dbl, uint8_t *p_ui_db );
```

The response format is whatever user defined.

4.27. RF Reset (F0h)

RF Reset command switches the RF field for a specified time interval. RF Reset value in range <0,255> specifies the time interval in approximately 32.7 ms steps. Valid configuration word has to be supplied (Direct SPI Write) to the firmware first before using this command. Response to this command is sent after the field OFF. The watchdog is disabled during this command.

Errors: none

4.28. Direct SPI Write (F1h)

Direct SPI Write command sends the SPI transaction to EM4294 chip. The configuration word contains the 32 bits data according to the EM4294 data sheet. This configuration word is stored internally in the microcontroller and it is used for the RF field on/off operations.

Errors: none

4.29. Direct SPI Write with RF Reset (F2h)

Direct SPI Write with RF Reset command combines RF Reset (F0h) and Direct SPI Write (F1h).

Errors: none

4.30. Bootloader Mode (F3h)

The Bootloader Mode command returns UART_MESSAGE_OK (00h) response and then switches the firmware into the bootloader mode.

Errors: none

4.31. Send Debug Data (F6h)

This debug command returns the contents of last debug buffer. Intended for test engineer only.

4.32. Get Raw Data (F7h)

This debug command returns the contents of last raw data bit stream buffer. Intended for test engineer only.

4.33. Get Capture Data (F8h)

This debug command returns the contents of last decoded bit stream buffer. Intended for test engineer only.

If the debug mode is Normal or Decoded, then the response data contains two binary arrays of the same length. The first array contains the demodulated data bits, the second array contains each demodulated data bit validity (if available). The length of this command response is variable depending on the number of bits actually captured. If there is no bit captured at all, the two arrays are zero length.

For example, response:

02 0C   XX YY FF AA 05 50   80 00 02 00   CHK 03

is interpreted as:

XX	the command/response number
YY	00h for XX = F8h, 11h for others
Data bitsFF AA 05 50	= 11111111 10101010 00000101 01010000
Valid bits	80 00 04 00 = 10000000 00000000 00000010 00000000

It results in fact that the 0th and 22th data bits were rejected by decoding routine.
Errors: none

If the debug mode is Raw the response data contains single array of raw captured data.

4.34. Toggle Debug Mode (F9h)

To check some firmware demodulation routines, Read and Write Tag Memory commands can return either raw pulse lengths (dbg = 1) or decoded bit stream (dbg = 2). Intended for test engineer only.

Errors: none

4.35. Fwd Pulse Tuning (FAh)

Timing of forward link DIN signal during ISO15693 forward link communication can be set by Fwd Pulse Tuning command. New value of selected timing value can be set by this command. Actual value corresponds to the negative number of microcontroller clocks minus interrupt overhead before the new interrupt is raised. Generally, actual timings values may differ between firmware releases. Intended for test engineer only.

Availability: If enabled during firmware compilation.

Errors: none

4.36. Reader Status (FDh)

Reader Status command response contains Version (family), Release and Release date of the firmware. Release is defined as a number in "BCD" format (e.g.: 0Ch => release 0.12). Date of the release is coded in format: year[15:10], month[9:6], day[5:0]. Year value = 0 is a year 2K.

4.37. Switch Coil On/OFF (FEh)

Switch Coil On/Off controls the MOD_PIN(DIN) and EN uC outputs to the EM4294. Bit 0 of coil parameter is set to the MOD_PIN output signal, bit 1 is set to the EN output signal, i.e. standard operating coil parameter value is 02h (MOD_PIN = ‘0’, EN = ‘1’).

5. Bootloader

Current firmware provides a bootloader feature. The bootloader feature permits to the user to upload a new firmware release using USB cable and an application software that is provided with the EMDB410 RFID Reader.

Bootloader allows an upload of application part only. It is not possible to upload the bootloader itself.

Bootloader is activated either on firmware start-up either by Bootloader Mode command (F3h). Start-up firmware activation is applied so that the broken (application part) firmware or firmware which does not implement Bootloader Mode command (F3h) can be uploaded. Bootloader is not activated by watch-dog reset.

Firmware data being sent to the bootloader are synchronised in two levels; hand-shake page synchronisation = 0xD6 sent twice per page, and byte synchronisation = 0xE7 sent once per two bytes (see figure on the next page). The application may transmit a next page data only if it receives the first bootloader page synchronisation byte = 0x03 (i.e.; hand-shake), and may not send the next page synchronisation byte until it receives the second bootloader synchronisation byte = 0x03 (after the bootloader performed the eeprom_page_write operation). The byte synchronisation is not applicable as the bootloader byte processing is hidden in byte reception latency.
Current Bootloader uses the same communication parameters as the application part. However, the communication parameters may differ in future.

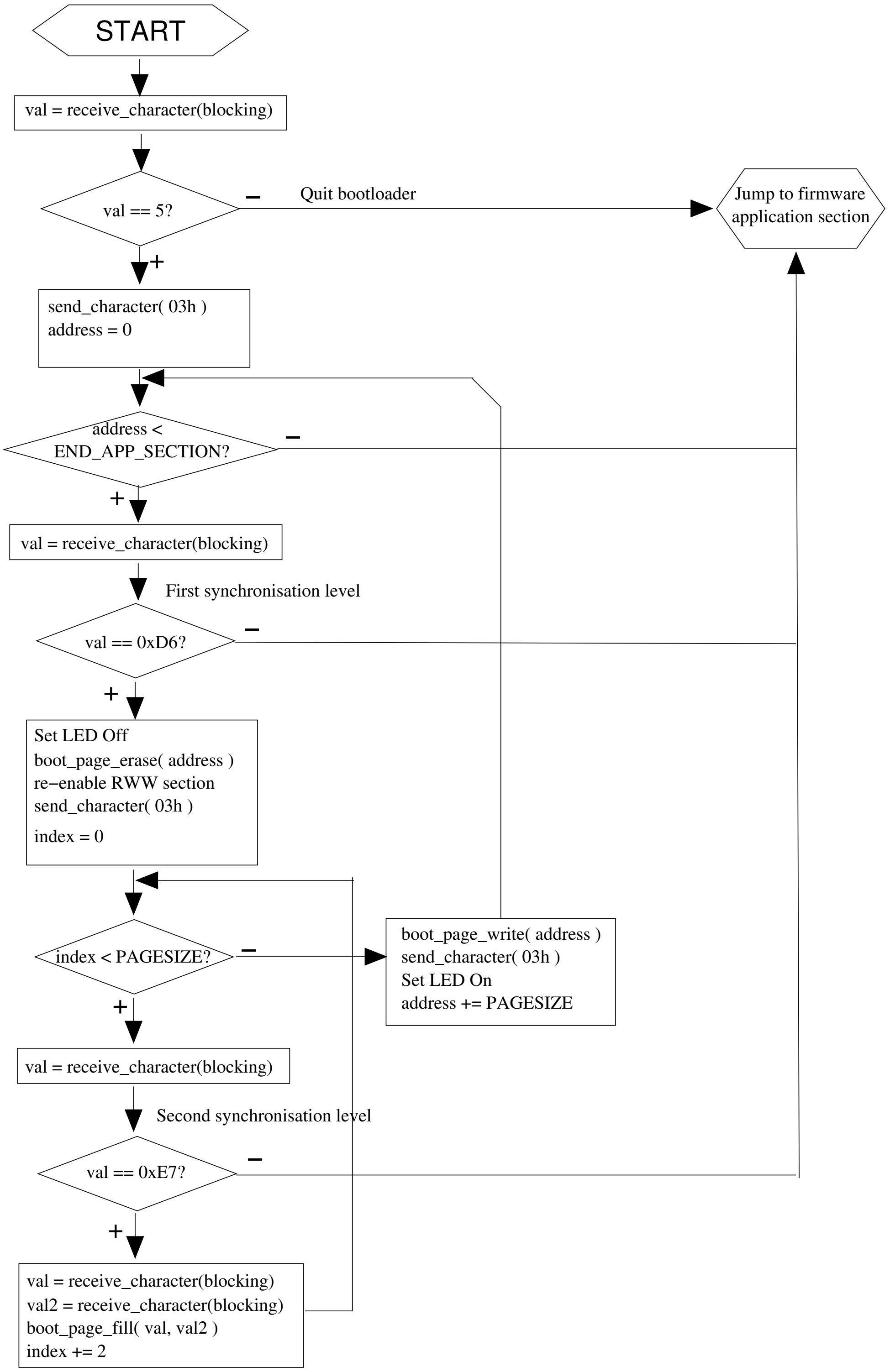
START
val = receive_character(blocking)
val == 5?
–
Quit bootloader
Jump to firmware application section
+
send_character(03h)
address = 0
address < END_APP_SECTION?
–
+
val = receive_character(blocking)
First synchronisation level
val == 0xD6?
–
+
Set LED Off
boot_page_erase(address)
re–enable RWW section
send_character(03h)
index = 0
index < PAGESIZE?
–
boot_page_write(address)
send_character(03h)
Set LED On
address += PAGESIZE
+
val = receive_character(blocking)
Second synchronisation level
val == 0xE7?
–
+
val = receive_character(blocking)
val2 = receive_character(blocking)
boot_page_fill(val, val2)
index += 2

Figure 3 Bootloader flow diagram

6. EM4294 crypto engine

EM4294 crypto firmware (version 0.6) implements the following features;

- EM4035 crypto engine to communicate with EM4035 in secure mode
- Auxiliary commands

- 3DES enciphering/deciphering algorithm (2key or 3key CBC mode ciphering with arbitrary Initial Vector)

There are 8 SIM keys available in the EM4035 crypto engine. First four keys {0, 1, 2, 3} are standard EM4035 user key set. Second four key set {4, 5, 6, 7} is introduced so that the user can use a default EM4035 key set along his user key set. Second key set should be set initial EM4035 default keys (initialisation key set).

SIM Card type	SIM Card firmware version	Reset (ATR) response
EMTG56	0.6 (sub-version 0.1)	3B 12 95 36 06

Table 3 SIM card crypto engine versions

For futher information on EM4294 crypto engine commands, please refer to the EM4294 crypto firmware description document.

6.1. SIM card reset

Command: 02 06 84 00 04 00 86 03

OK response: 02 09 84 00 **3B 12 95 36 06** 01 03

Note: The **ATR** response field differs according to the SIM card crypto engine version/release (see Table 3 SIM card crypto engine versions).

6.2. SIM card select key (01h) (EM4035 crypto command)

Crypto engine is initialised using the SIM key number [KK].

Command: 02 0B 84 02 02 05 **10 01 KK 00 01** YY 03

OK response: 02 07 84 00 **01 90 00** 12 03

6.3. SIM card send A1 (02) (EM4035 crypto command)

A1 constant [XX] is sent to the crypto engine.

Command: 02 12 84 01 01 05 **10 02 00 00 07 XX XX XX XX XX XX XX** YY 03

OK response: 02 07 84 00 **02 90 00** 11 03

6.4. SIM card get A2 (03) (EM4035 crypto command)

A2 [XX] and f() [ZZ] constants are read from the crypto engine.

Command: 02 0B 84 02 0D 50 **10 03 01 00 0B** C9 03

OK response: 02 12 84 00 **03 XX XX XX XX XX XX XX 00 ZZ ZZ ZZ 90 00** YY 03

6.5. SIM card send G (04) (EM4035 crypto command)

G() [GG] constant is sent to the crypto engine to check the authentication result.

Command: 02 0E 84 01 01 32 10 **04 01 00 03 GG GG GG** YY 03

OK response: 02 07 84 00 **04 90 00** YY 03

Error response: 02 07 84 00 **04 90 EE** F9 03

6.6. SIM card Sign (05) (EM4035 crypto command)

Two bytes [XX] (usually the CRC of EM4035 secure commands) are signed [ZZ] by crypto engine using this command.

Command: 02 0B 84 02 04 40 **10 05 XX XX 02** YY 03

OK response: 02 08 84 00 **05 ZZ ZZ 90 00** YY 03

6.7. SIM card Write Key (06) (EM4035 crypto command)

Write key command changes the SIM key number [KK] contents to the new value [XX]. SIM card reset command is required after this command is executed.

Command: 02 17 84 01 01 14 **10 06 KK 00 0C XX XX XX XX XX XX XX XX XX XX XX XX** YY 03

OK response: 02 07 84 00 **06 90 00** 15 03

Example: Write EM4035 default Super User Key value to SUK key –

02 17 84 01 01 14 10 **06 00 00 0C 56 45 52 45 4D 53 54 41 4E 44 41 52** 8B 03

6.8. SIM card login (07)

Login command with password [PP] is sent to the crypto engine. There is no error message signalised in case of incorrect login password.

Command: 02 0F 84 01 01 05 **10 07 00 00 04 PP PP PP PP** YY 03

Response: 02 07 84 00 **07 90 00** 14 03

6.9. SIM card Change Password (08)

Password of the crypto engine that secure a key change command can be changed in two steps sending a Change password command. In both steps, the new password [NN] is sent. SIM card reset command is required after this command is executed. Login with previous valid password is required.

1st command: 02 0F 84 01 01 05 **10 08 00 00 04 NN NN NN NN** YY 03

1st OK response: 02 07 84 00 **08 90 00** 1B 03

2nd command: 02 0F 84 01 01 05 **10 08 01 01 04 NN NN NN NN** YY 03

2nd OK response: 02 07 84 00 **08 90 00** 1B 03

6.10. Get Chip Supplier Serial Number (CSSN) (EM4294 crypto Auxiliary command)

8bytes EMTG56 IC serial number [SN] can be obtained by means of the Get Chip Supplier Serial Number command.

Command: 02 0B 84 02 0A 05 **20 00 00 00 08** AA 03

OK response: 02 0F 84 00 **00 SN SN SN SN SN SN SN SN 90 00** YY 03

6.11. Get Random Number (EM4294 crypto Auxiliary command)

8bytes random number [RN] generated by the EMTG56 IC internal random number generator can be obtained by means of the Get Random Number command.

Command: 02 0B 84 02 0A 05 **20 01 00 00 04** AA 03

OK response: 02 0F 84 00 **01 RN RN RN RN RN RN RN RN 90 00** YY 03

6.12. Get Sub Version (EM4294 crypto Auxiliary command)

Additional SIM card firmware sub-version number [SV], user memory size [MHML], and I/O buffer size [IO] information can be obtained by means of the Get Sub Version command.

Command: 02 0B 84 02 0A 05 **20 03 00 00 08** AA 03

OK response: 02 0B 84 00 **03 SV ML MH IO** YY 03

Example response: 02 0B 84 00 **03 00 00 97 40 90 00** YY 03

6.13. Read User Memory (EM4294 crypto Auxiliary command)

SIM card User Memory can be read by means of Read User Memory command. Minimum 1 byte and maximum 16 bytes [LL] of the user data [DD] can be read from the user memory address [AAAA] in the range from 0000h to the value obtained from Get Sub Version command.

Command: 02 0B 84 02 (LL+2) 19 **20 10 AA AA LL** YY 03

Example command: 02 0B 84 02 12 19 **20 10 00 00 10** A6 03

OK response: 02 XX 84 00 **10 DD{LL*}** YY 03

Example response: 02 17 84 00 **10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 90 00** 13 03

6.14. Write User Memory (EM4294 crypto Auxiliary command)

SIM card User Memory can be written by means of Write User Memory command. Minimum 1 byte and maximum 16 bytes [LL] of the user data [DD] can be written to the user memory at the address [AAAA] in the range from 0000h to the value obtained from Get Sub Version command.

Command: 02 XX 84 01 01 32 **20 20 AA AA LL DD{LL*}** YY 03
Example command: 02 1B 84 01 01 32 **20 20 AA AA 10 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF** 53 03
OK response: 02 07 84 00 **20 90 00** 33 03

7. Obsolete product support

EMDB410 firmware supports the transponders that are already marked as obsolete products;

Transponder family	Command set support	Communication Speed support
EM4034	Complete	All
EM4135	Complete	All
EM4035	Complete except EAS toggling feature	All
EM4006	Read UID	RF/512

Table 4: Family 161 obsolete command set and features

7.1. PC to reader (Command message)

PC to reader	Serial Data Bytes sent on UART											
Command	0	1	2	3	4	...	...	...	...	XX	XX+1	
ISO 15693 Commands and EM tag commands (available in firmware family 161 only)												
EM4034, EM4035, EM4135	These tags share relevant ISO15693 commands defined in chapter 3.3 plus the dedicated commands listed below											
Long General Read	02h	XXh	87h	word size	1st data byte	...	...	...	Last data byte	CHK	03h	
HW Toggle EAS	02h	XXh	8Ah	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
Switch to normal mode (EM4035 only)	02h	XXh	8Dh	Resp. length	1st data byte	...	...	...	Last data byte	CHK	03h	
Read EM4006 UID	02h	04h	98h	4006_ scale	CHK	03h						

Note: word size = 4 or 8 bytes per word

7.2. Reader to PC (Response)

Reader to PC	Serial Data Bytes sent on UART											
Response	0	1	2	3	4	5	6	7	8	…	YYh	YYh+ 1
ISO 15693 Commands and EM tag commands												
Long General Read	02h	YYh	87h	ACK	LSB Word 1 MSB		…		LSB Word N MSB		CHK	03h
HW Toggle EAS	02h	YYh	8Ah	ACK	1st data byte	...	...	...	...	Last data byte	CHK	03h
Switch to Normal Mode	02h	YYh	8Dh	ACK	1st data byte	...	...	...	...	Last data byte	CHK	03h
Read EM4006 UID	02h	0Eh	98h	ACK	1st data byte	...	...	...	...	Last data byte	CHK	03h

7.3. Command description

7.3.1. Long General Read (87h)

Long General Read command is designed to read whole EM4035 tag user memory by single reader command. The command expects Read Multiple Block command is supplied by the application software. The supplied command parameters are analysed and If the number of response data exceeds 2*13*8=208 bytes, two response messages are emit by the reader to read up to 36 64bit words (General Read with response length in bytes command (8Fh) code which allows reading up to 14 64b words is used).

The command response contains the requested word data, in case the Secure Read command is supplied each returned word also contains a security status byte.

Requires: Read command

Accessed items: flag byte, command code, start words, number of words

Example of addressed secure reading 36 words of EM4035 starting from block no.13:

Command: 02 0B 87 08 02 E2 16 0D 23 AE A4 56 03

Response: 02 DC 87 00 00 13 00 00 00 00 00 00 00 00 14 00 00 00 00 00 00 00 00 15 00 00 00 00 00 00 00 00 16 00 00 00 00 00 00 00 00 17 00 00 00 00 00 00 00 00 18 00 00 00 00 00 00 00 00 19 00 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00 00 21 00 00 00 00 00 00 00 00 22 00 00 00 00 00 00 00 00 23 00 00 00 00 00 00 00 00 24 00 00 00 00 00 00 00 00 25 00 00 00 00 00 00 00 00 26 00 00 00 00 00 00 00 00 27 00 00 00 00 00 00 00 00 28 00 00 00 00 00 00 00 00 29 00 00 00 00 00 00 00 00 30 00 00 00 00 00 00 00 00 31 00 00 00 00 00 00 00 00 32 00 00 00 00 00 00 00 00 33 00 00 00 00 00 00 00 00 34 00 00 00 00 00 00 00 00 35 00 00 00 00 00 00 00 00 36 00 00 00 00 00 00 00 7F 03 02 70 87 00 00 37 00 00 00 00 00 00 00 00 38 00 00 00 00 00 00 00 00 39 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 41 00 00 00 00 00 00 00 00 42 00 00 00 00 00 00 00 00 43 00 00 00 00 00 00 00 00 44 00 00 00 00 00 00 00 00 45 00 00 00 00 00 00 00 00 46 00 00 00 00 00 00 00 00 47 00 00 00 00 00 00 00 00 48 00 00 00 00 00 00 00 89 03

7.3.2. HW Toggle EAS (8Ah)

HW Toggle EAS command switches the EAS feature of the EM4034 transponder IC. This command performs the RF field reset (sends zero configuration word, waits for 64ms, sends the current configuration word loaded by Direct SPI Write command) first. Then, according to the command byte (0xE4 for EM4034), it sends the supplied login command to the EM4034. Finally, the EM403x Toggle EAS Custom command is sent before the EAS is activated.

Requires: Addressed Login EM4034 Proprietary command.

Accessed items: flag byte, command byte

Errors: 15h, 16h, 18h, 19h, 1Ah, 1Bh, 1Ch, 1Dh, 13h, 0Bh

7.4. Switch to normal mode (8Dh)

Switch to normal mode command switches single EM4035 tag in transport mode into a normal mode using crypto engine. As unaddressed commands are used, single EM4035 tag is assumed to be in the RF field.

The switch comprises following actions;

- Unaddressed authentication is performed
- If authentication fails, authentication result error is returned
- If authentication response is not correct, ERR_EM4035_NM_AUTH_FAILED (32h) is returned
- Unaddressed secure read of block 8
- If read fails, ERR_EM4035_NM_READ_FAILED (31h) or ERR_EM4035_NM_CRC_ERROR (33h) are returned
- If tag is already in transport mode, UART_MESSAGE_OK (00h) is returned
- Unaddressed secure write of normal mode settings to block 8
- If write operation fails, write result is returned
- If write response crc fails, ERR_BAD_CRC error is returned

Requires: Unaddressed Authentication Step 1 EM4035 Proprietary command with Super User Key (SUK) set

Accessed items: flag byte, command byte, key number, ICMfg byte

Errors: 15h, 16h, 18h, 19h, 1Ah, 1Bh, 1Ch, 1Dh, 31h, 32h, 33h

7.5. Read EM4006 UID (98h)

Read EM4006 UID command performs a scan for single EM4006 tag in the RF field with a data rate specified by 4006_scale value. 4006_scale value defines according to equation *data_rate = 2 ^ 4006_scale*. In case a correct EM4006 UID is captured (including CRC check), the response contains 10 bytes of EM4006 UID (see Memory Map in EM4006 datasheet, without Start and Stop bits). Otherwise, ERR_ NO_TAG (12h) error is returned.

Supports: EM4006 RF/512 actually tested.

Errors: 12h

EM Microelectronic-Marin SA cannot assume responsibility for use of any circuitry described other than circuitry entirely embodied in an EM Microelectronic-Marin SA product. EM Microelectronic-Marin SA reserves the right to change the circuitry and specifications without notice at any time. You are strongly urged to ensure that the information given has not been superseded by a more up-to-date version.