ifm electronic

Reader Protocol

Version 1.24

Communications protocol for interfacing
the UHF RFID reader DTE8xx / DTE9xx
to a higher level

Change history

Version	Change	Date (YYYY-MM-DD)
1.0	New document	2008-04-02
1.1	Commands and command IDs revised, command structure added	2008-04-18
1.2	Commands “GetInitialQValue”, “SetInitialQValue”, “GetMaxAirCommErrors” and “SetMaxAirCommErrors” added	2008-05-19
1.3	Command “ASyncBulkGetEPCs” removed	2008-05-26
1.4	- tRRUI4TagErrorCodes-Enum extended - Commands “GetActiveParamset”, “SetActiveParamset”, “GetASyncObserved-ListParameters” and “SetASyncObservedListParameters” added - Command “SaveConfiguration” renamed as “SaveActiveParamset” - lower power figure in chapter 4.2.6 corrected - Commands “GetParameterById” and “SetParameterById2” added - mode “RRUI4PCSW_MODE2” removed from the Enum “tRRUI4ETSIPort ChannelSwitchingMode” (see also chapter 4.2.17 and chapter 4.2.18) - Commands “GetAntennaMode” and “SetAntennaMode” modified so that antenna mode can be set separately for synchronous and asynchronous commands (see chapter 4.2.9 and chapter 4.2.10)	2008-07-01
1.5	- Communication standard “RRUI4COMMSTANDARD_AUTODETECT” added - Commands “GetDefaultParamset” and “SetDefaultParamset” added - Command “LoadFactoryDefaults” added - Commands “GetCableLossAndAntennaGain” and “SetCableLossAndAntennaGain” added	2008-08-11
1.6	- Chapter 3 revised	2008-08-22
1.7	- Enum “tRRUI4TagErrorCodes” expanded by the value “RRUI4TEC_VERIFYFAILED” - Naming error in the command IDs rectified (“RRUI4CMD_SyncWriteDataAny”)	2008-11-11
1.8	- Functions “GetSelectFilterOnOff” and “SetSelectFilterOnOff” expanded by the parameter “Filter number” - Functions “GetSession” and “SetSession” renamed as “GetSessionAndTarget” and “SetSessionAndTarget” and expanded	2008-11-17
1.9	- Functions “GetSelectFilterData” and “SetSelectFilterData” expanded by the parameter “Target” - Functions “GetSessionAndTarget” and “SetSessionAndTarget” renamed as “GetSelSessionAndTarget” and “SetSelSessionAndTarget” and expanded by the parameter “Query-Sel”	2008-11-24
1.10	- Chapter 4.1.10 revised - Chapter 4.2.26 revised - Chapter 4.6.1 revised	2008-11-28
1.11	- Functions “GetLicenseCode” and “SetLicenseCode” renamed as “GetLicenseKey” and “SetLicenseKey” - List of parameters saved in the EEPROM by “SaveActiveParamset” (chapter 4.2.31) modified - Missing brackets ([ ]) for the return values of the function “GetCableLossAndAntennaGain” (chapter 4.2.13) added - Frequency information for the function “SetFrequency” (chapter 4.5.3) now in kHz - Structure “tRRUI4ResultFlag” expanded by the entry “RRUI4RESULTFLAG_ WRONGCFM” and this return value added for the relevant functions	2008-12-10

Change history

Version	Change	Date (YYYY-MM-DD)
1.12	- Structure “tRRUI4ResultFlag” expanded by the entry “RRUI4RESULTFLAG_ NOHANDLE” and this return value added for the relevant functions - Tag PC added to the return values for the function “TxGen2CmdACK” (chapter 4.5.16) - Support for the recommissioning bits (see EPCglobal specification V1.2.0 page 68) added to all Kill commands - Function “GetParameterById” (chapter 4.2.32) now also returns the transferred ID in the response data - Function “SetParameterById” (chapter 4.2.33) now also returns the transferred ID in the response data - Function “TxGen2CmdCustomCmd” (chapter 4.5.29) revised - Functions “SyncBlockEraseAny”, “SyncBlockEraseSpecific” and “ASyncBlockEraseAny” added (Attention: the IDs in the structure “tRRUI4HostCommCmds” (chapter 4.1.1) have changed as a result!)	2008-12-19
1.13	- Functions “GetPortPower”, “SetPortPower”, “GetCableLossAndAntennaGain”, “SetCableLossAndAntennaGain”, “GetETSIPortChannelList” and “SetETSIPortChannelList” now also return the antenna port number in the response - Functions “GetIOCardHwConfig”, “SetIOCardHwConfig”, “GetIOCardPrtklConfig” and “SetIOCardPrtklConfig” now also return the I/O card number in the response - Functions “GetSelectFilterOnOff”, “SetSelectFilterOnOff”, “GetSelectFilterData” and “SetSelectFilterData” now also return the filter number in the response	2009-01-13
1.14	- Functions “GetCommStandard” and “SetCommStandard” expanded by the communication standards for “China” and “Thailand” - “Word count” is included in the command response for the functions “SyncReadDataAny”, “SyncReadDataSpecific” and “ASyncReadDataAny” - Command “SyncRetransmitResultData” removed from the documentation - Error in the allocation of the asynchronous command responses rectified - Tag counter value included in the “Going message” for asynchronous commands (see chapter 4.6.2)	2009-01-28
1.15	- Command “ASyncGetTagrate” added	2009-02-02
1.16	- Function “CarrierOnOff” expanded by the return value “RRUI4RESULTFLAG_NOPROFILE” - Error in the descriptive text for the commands “GetPortPower” and “SetPortPower” rectified - Brackets error in the description of the block structure of the function “SetETSIPortChannelList” rectified - Note modified to suit the function “SetSelSessionAndTarget” on the reader firmware v1.45.00.	2009-12-03

Change history

1.17	**Specification version valid from reader firmware v2.00.00** - Number of select filters increased to 32 - the following chapters were converted to variable EPC length: • “SyncGetEPCs” • “SyncBulkGetEPCs” • “SyncReadDataAny” • “SyncReadDataSpecific” • “SyncWriteDataAny” • “SyncWriteDataSpecific” • “SyncWriteMaskedDataAny” • “SyncWriteMaskedDataSpecific” • “SyncBlockEraseAny” • “SyncBlockEraseSpecific” • “SyncLockAny” • “SyncLockSpecific” • “SyncKillAny” • “SyncKillSpecific” • “SyncWriteEPCToSingleTag” • “ASyncGetEPCs” • “ASyncReadDataAny” • “ASyncWriteDataAny” • “ASyncWriteMaskedDataAny” • “ASyncBlockEraseAny” • “ASyncLockAny” • “ASyncKillAny” • “OpenNoninventoriedTag” • “OpenSpecificTag” • “TxGen2CmdACK” • “Transmission of tag data”	2009-12-08
1.18	- IDs for the additional commands to support NXP-UCODE commands added in the chapter “Command / response IDs” - Following chapters were added: • “SyncNXPG2XReadProtectAny” • “SyncNXPG2XReadProtectSpecific” • “SyncNXPG2XResetReadProtectAny” • “SyncNXPG2XResetReadProtectSpecific” • “SyncNXPG2XChangeEASAny” • “SyncNXPG2XChangeEASSpecific” • “SyncNXPG2XEASAlarm” • “ASyncNXPG2XReadProtectAny” • “ASyncNXPG2XResetReadProtectAny” • “ASyncNXPG2XChangeEASAny” • “ASyncNXPG2XEASAlarm”	2009-12-16

Change history

1.19	- IDs for the “ActivateBuzzer” and “ASyncGetRawEPCs” commands added in the chapter “Command / response IDs” - Chapter “ActivateBuzzer” added - Chapter “ASyncGetRawEPCs” added - Chapter “I/O card types” added - Chapter “I/O communications protocols” added - Chapters “GetIOCardHwConfig” and “SetIOCardHwConfig” modified to suit reader firmware v2.00.00 - IDs for the GPIO commands added in the chapter “Command / response IDs” - Chapter “GPIOGetIOData” added - Chapter “GPIOSetOutput” added - Chapter “GPIOAddActionToActionlist” added - Chapter “GPIOClearActionlist” added - Chapter “GPIOAssignInputToActionlist” added - Chapters “FwUpdPrepare”, “FwUpdPutData” and “FwUpdFlash” extended by the missing return value “RRUI4RESULTFLAG_NOLICENSE” - ID range of the NXP command moved in the chapter “Command / response IDs” - Enum extended in the chapter “Results flags” so that the Tag Protocol Control Words can also be transmitted - Chapters “GetExtResultFlag” and “SetExtResultFlag” modified to suit the changed “Results flags” - Following chapters were extended by XPC functionality: • “SyncGetEPCs” • “SyncBulkGetEPCs” • “SyncReadDataAny” • “SyncReadDataSpecific” • “SyncWriteDataAny” • “SyncWriteDataSpecific” • “SyncWriteMaskedDataAny” • “SyncWriteMaskedDataSpecific” • “SyncBlockEraseAny” • “SyncBlockEraseSpecific” • “SyncLockAny” • “SyncLockSpecific” • “SyncKillAny” • “SyncKillSpecific” • “SyncWriteEPCToSingleTag” • “SyncNXPG2XReadProtectAny” • “SyncNXPG2XReadProtectSpecific” • “SyncNXPG2XResetReadProtectAny” • “SyncNXPG2XResetReadProtectSpecific” • “SyncNXPG2XChangeEASAny” • “SyncNXPG2XChangeEASSpecific” • “ASyncGetRawEPCs” • “ASyncGetEPCs”	2010-01-13

Change history

	• “ASyncReadDataAny” • “ASyncWriteDataAny” • “ASyncWriteMaskedDataAny” • “ASyncBlockEraseAny” • “ASyncLockAny” • “ASyncKillAny” • “ASyncNXPG2XReadProtectAny” • “ASyncNXPG2XResetReadProtectAny” • “ASyncNXPG2XChangeEASAny” • “OpenNoninventoriedTag” • “OpenSpecificTag” • “TxGen2CmdACK” • “Transmission of tag data” - IDs for the BlockPermalock commands extended in the chapter “Command / response IDs” - Chapters “SyncBlockPermalockAny”, “SyncBlockPermalockSpecific” and “ASyncBlockPermalockAny” added	
1.20	- Structure of return values added in the following chapters: • “SyncReadDataSpecific” • “SyncWriteDataSpecific” • “SyncWriteMaskedDataSpecific” • “SyncBlockEraseSpecific” • “SyncLockSpecific” • “SyncKillSpecific” • “SyncBlockPermalockSpecific” • “SyncNXPG2XReadProtectSpecific” • “SyncNXPG2XResetReadProtectSpecific” • “SyncNXPG2XChangeEASSpecific” - Missing asynchronous response IDs extended in the chapter “Command / response IDs”	2010-02-02
1.21	- Value “RRUI4CG_DATACHANGING” added in the chapter “Coming / going flag” - Bracketing for the command response block structure corrected in the chapters “GetIOCardHwConfig”, “SetIOCardHwConfig”, “GetIOCardPrtklConfig” and “SetIOCardPrtklConfig” - IDs for the EN302208TestmodulationOnOff command added in the chapter “Command / response IDs” - Chapter “EN302208TestmodulationOnOff” added	2010-03-08
1.22	**Specification version valid from reader firmware v2.03.00** - Minor corrections in the chapters “GetETSIPortChannelList”, “SetETSIPortChannelList”, “GPIOGetIOData”, “GPIOSetOutput” and “GPIOAssignInputToActionlist” - Note on the transmission sequence of the masked data in the chapter “SetSelectFilterData” added - New command “SyncWriteEPCSpecific” added - Return value “RRUI4RESULTFLAG_OUTOFRANGE” extended for the command “SyncWriteEPCToSingleTag”	2010-07-14
1.23	- Command “GetDeviceSerialNmbr” added	2010-08-25
1.24	- dAddendum to “LoadFactoryDefaults”: transfer of a number of a set of parameters - Command “GetDeviceTypeNmbr” added - Chapter “Ethernet transmission” added	2010-12-17

1. Contents

1. Contents

1. Contents

2. Introduction

The Reader Protocol – referred to as KBRP in the following – is used to interface the UHF RFID reader DTE8xx / DTE9xx and its successors to a higher level. It is possible to transfer system commands, data commands and diagnostics commands, as well as their responses.

3. Transmission methods

3.1 UART transmission (RS232, RS422, RS485 or similar)

3.1.1 Physical layer

A full or half-duplex connection such as RS232, RS422 or RS485 is used for the physical layer

3.1.2 Data link layer

Transmission is in frames and blocks. A block comprises a maximum of 256 frames. A frame comprises a maximum of 256 bytes, of which a maximum of 250 bytes can be user data. The result is a maximum block size of 64000 bytes of user data.
The data link layer is used to safeguard the data between sender and recipient. The sender receives a response from the recipient for each frame received. If the sender does not receive a response from the recipient within a time window of 350 milliseconds after sending a frame, the frame sent is repeated until the error counter signals the cancellation of the transmission.

3.1.2.1 Structure of a frame

5A LL SS FF DD ... DD P1 P2

- 5A: Start code for synchronisation
- LL: Number of bytes in the frame not including the start code
- SS: Status byte
- FF: Frame number
- DD: User data
- P1: 16-bit checksum low byte
- P2: 16-bit checksum high byte

3.1.2.2 Start code and synchronisation

The start code is used to synchronise the recipient to the sender. It further allows the receiver to synchronise to the start of a frame when no data have been received for 15 milliseconds.

3.1.2.3 Start code and synchronisation

The status byte has the following significance:

- 50: Data packet
- A0: Response: “OK”
- A1: Response: “Memory error” (the receiver was unable to allocate any memory for the data block received)

A response is only 3 bytes long and is not CRC checked.
“OK” response: 5A 02 A0
“Memory error” response: 5A 02 A1

3.1.2.4 Frame number

The frame number defines how many more frames there are in this data block. Only the first frame in a data block can be shorter than 256 bytes. Each additional frame must have a length of 256 bytes (length byte LL is FF).
It is therefore possible to calculate the block size from the first frame number.

For example:

A block with 700 bytes of user data is to be transmitted. For this purpose the block is divided into three frames.

1st frame: 5A CD 50 02 – there now follow 200 bytes of user data – P1 P2

2nd frame: 5A FF 50 01 – there now follow 250 bytes of user data – P1 P2

3rd frame: 5A FF 50 00 – there now follow 250 bytes of user data – P1 P2

The receiver can use the frame number of the first frame (here 02) and its length byte to calculate the block size (block size = frame number * 250 bytes + length byte -5) (here in the example: 2 * 250 bytes + 205 bytes - 5 bytes = 700 bytes), and reserve an appropriate amount of memory for the data.

3.1.2.5 User data

User data are the bytes in a frame that flow into the block transmitted.

3.1.2.6 Checksum

The checksum is calculated using the polynomial x^16 + x^12 + x^5 + 1 with a pre-initialisation of 0x0000 from the start code to the last user data byte.

3.1.3 Network layer

As the KBRP is a point-to-point protocol, there is no network layer.

3.1.4 Transport layer, session layer, presentation layer

Do not exist.

3.1.5 Application layer

The application layer transmits data blocks from 1 to a maximum of 64000 bytes.

3.2 Ethernet transmission

A data transmission layer is used for communication to the reader over Ethernet, just like in serial communication.

The data transmission layer over Ethernet is a much more simple solution here since the TCP/IP protocol already has a data security layer. As TCP/IP is a stream protocol, only the packet beginning and packet end is needed.

3.2.1 Frame set-up

A frame looks like the following:

Start + data block + end

The start is made up of 0xAA 0xBB 0x01 0x01, whereby the first 1 is the Datetransmit byte and the second 1 is a Stuffbyte. The end is made up of 0xAA 0xCC. If the byte 0xAA appears in the KBRP frame, it must be doubled (0XAA -> 0xAA 0xAA).

3.2.2 Port

The TCP communication port is the port 4007.

3.2.3 Example

The frame “ASyncGetEPCs” is shown here as an example. The ID for this command is “0x0111”, which makes the frame look like this:

0xAA 0xBB 0x01 0x01 0x11 0x01 0xAA 0xCC

4. Command and response transmission

Commands and their responses are transmitted in individual data blocks via UART. The same applies to asynchronous responses from the reader (Responses without immediately preceding command.).

Structure of the data blocks:

I1 I2 DD ... DD

I1: Command / response ID low byte
I2: Command / response ID high byte
DD: Command / response data

4.1 Data types

4.1.1 Command / response IDs

```
typedef enum
{
    // RRUI4 commands
    RRUI4CMD_GetMode = 0x0001,                          // 0x0001
    RRUI4CMD_SetMode,                                   // 0x0002
    RRUI4CMD_GetCommStandard,                           // 0x0003
    RRUI4CMD_SetCommStandard,                           // 0x0004
    RRUI4CMD_GetPortPower,                              // 0x0005
    RRUI4CMD_SetPortPower,                              // 0x0006
    RRUI4CMD_GetCarrierFollowUpTime,                    // 0x0007
    RRUI4CMD_SetCarrierFollowUpTime,                    // 0x0008
    RRUI4CMD_GetAntennaMode,                            // 0x0009
    RRUI4CMD_SetAntennaMode,                            // 0x000A
    RRUI4CMD_GetPortMultiplexSequenceAndExposureTime,   // 0x000B
    RRUI4CMD_SetPortMultiplexSequenceAndExposureTime,   // 0x000C
    RRUI4CMD_GetCableLossAndAntennaGain,                // 0x000D
    RRUI4CMD_SetCableLossAndAntennaGain,                // 0x000E
    RRUI4CMD_GetETSIPortChannelList,                    // 0x000F
    RRUI4CMD_SetETSIPortChannelList,                    // 0x0010
    RRUI4CMD_GetETSIPortChannelSwitchingMode,           // 0x0011
    RRUI4CMD_SetETSIPortChannelSwitchingMode,           // 0x0012
    RRUI4CMD_GetProfileList,                            // 0x0013
    RRUI4CMD_GetProfile,                                // 0x0014
    RRUI4CMD_SetProfile,                                // 0x0015
    RRUI4CMD_GetModulationType,                         // 0x0016
    RRUI4CMD_SetModulationType,                         // 0x0017
    RRUI4CMD_GetExtResultFlag,                          // 0x0018
    RRUI4CMD_SetExtResultFlag,                          // 0x0019
    RRUI4CMD_GetErrorStatus,                            // 0x001A
    RRUI4CMD_GetDefaultParamset,                        // 0x001B
    RRUI4CMD_SetDefaultParamset,                        // 0x001C
    RRUI4CMD_GetActiveParamset,                         // 0x001D
    RRUI4CMD_SetActiveParamset,                         // 0x001E
    RRUI4CMD_SaveActiveParamset,                        // 0x001F
    RRUI4CMD_GetParameterById,                          // 0x0020
    RRUI4CMD_SetParameterById,                          // 0x0021
    RRUI4CMD_GetIOCardHwConfig,                         // 0x0022
```

```
RRUI4CMD_SetIOCardHwConfig,                          // 0x0023
RRUI4CMD_GetIOCardPrtklConfig,                       // 0x0024
RRUI4CMD_SetIOCardPrtklConfig,                       // 0x0025
RRUI4CMD_GetTime,                                    // 0x0026
RRUI4CMD_SetTime,                                    // 0x0027
RRUI4CMD_GetTemp,                                    // 0x0028
RRUI4CMD_GetSWVersion,                               // 0x0029
RRUI4CMD_GetHWVersion,                               // 0x002A
RRUI4CMD_GetLicenseKey,                              // 0x002B
RRUI4CMD_SetLicenseKey,                              // 0x002C
RRUI4CMD_LoadFactoryDefaults = 0x0040,               // 0x0040
RRUI4CMD_FwUpdPrepare,                               // 0x0041
RRUI4CMD_FwUpdPutData,                               // 0x0042
RRUI4CMD_FwUpdFlash,                                 // 0x0043
RRUI4CMD_ResetSystem,                                // 0x0044
RRUI4CMD_ActivateBuzzer,                             // 0x0045
RRUI4CMD_GetDeviceSerialNmbr,                        // 0x0046
RRUI4CMD_GetDeviceTypeNmbr,                          // 0x0047
RRUI4CMD_GetSelSessionAndTarget = 0x0080,            // 0x0080
RRUI4CMD_SetSelSessionAndTarget,                     // 0x0081
RRUI4CMD_GetInitialQValue,                           // 0x0082
RRUI4CMD_SetInitialQValue,                           // 0x0083
RRUI4CMD_GetMaxAirCommErrors,                        // 0x0084
RRUI4CMD_SetMaxAirCommErrors,                        // 0x0085
RRUI4CMD_GetASyncObservedListParameters,             // 0x0086
RRUI4CMD_SetASyncObservedListParameters,             // 0x0087
RRUI4CMD_GetSelectFilterOnOff,                       // 0x0088
RRUI4CMD_SetSelectFilterOnOff,                       // 0x0089
RRUI4CMD_GetSelectFilterData,                        // 0x008A
RRUI4CMD_SetSelectFilterData,                        // 0x008B
RRUI4CMD_GPIOGetIOData,                              // 0x008C
RRUI4CMD_GPIOSetOutput,                              // 0x008D
RRUI4CMD_GPIOAddActionToActionlist,                  // 0x008E
RRUI4CMD_GPIOClearActionlist,                        // 0x008F
RRUI4CMD_GPIOAssignInputToActionlist,                // 0x0090
RRUI4CMD_SyncGetEPCs = 0x0101,                       // 0x0101
RRUI4CMD_SyncBulkGetEPCs,                            // 0x0102
RRUI4CMD_SyncReadDataAny,                            // 0x0103
RRUI4CMD_SyncReadDataSpecific,                       // 0x0104
RRUI4CMD_SyncWriteDataAny,                           // 0x0105
RRUI4CMD_SyncWriteDataSpecific,                      // 0x0106
RRUI4CMD_SyncWriteMaskedDataAny,                     // 0x0107
RRUI4CMD_SyncWriteMaskedDataSpecific,                // 0x0108
RRUI4CMD_SyncBlockEraseAny,                          // 0x0109
RRUI4CMD_SyncBlockEraseSpecific,                     // 0x010A
RRUI4CMD_SyncLockAny,                                // 0x010B
RRUI4CMD_SyncLockSpecific,                           // 0x010C
RRUI4CMD_SyncKillAny,                                // 0x010D
RRUI4CMD_SyncKillSpecific,                           // 0x010E
RRUI4CMD_SyncWriteEPCToSingleTag,                    // 0x010F
RRUI4CMD_ASyncGetRawEPCs,                            // 0x0110
RRUI4CMD_ASyncGetEPCs,                               // 0x0111
RRUI4CMD_ASyncReadDataAny,                           // 0x0112
```

```
RRUI4CMD_ASyncWriteDataAny,                          // 0x0113
RRUI4CMD_ASyncWriteMaskedDataAny,                    // 0x0114
RRUI4CMD_ASyncBlockEraseAny,                         // 0x0115
RRUI4CMD_ASyncLockAny,                               // 0x0116
RRUI4CMD_ASyncKillAny,                               // 0x0117
RRUI4CMD_ASyncStopCommand,                           // 0x0118
RRUI4CMD_ASyncGetTagrate,                            // 0x0119
RRUI4CMD_SyncWriteEPCSpecific,                       // 0x011A
RRUI4CMD_SyncBlockPermalockAny = 0x0120,             // 0x0120
RRUI4CMD_SyncBlockPermalockSpecific,                 // 0x0121
RRUI4CMD_ASyncBlockPermalockAny,                     // 0x0122
RRUI4CMD_SyncNXPG2XReadProtectAny = 0x0180,          // 0x0180
RRUI4CMD_SyncNXPG2XReadProtectSpecific,              // 0x0181
RRUI4CMD_SyncNXPG2XResetReadProtectAny,              // 0x0182
RRUI4CMD_SyncNXPG2XResetReadProtectSpecific,         // 0x0183
RRUI4CMD_SyncNXPG2XChangeEASAny,                     // 0x0184
RRUI4CMD_SyncNXPG2XChangeEASSpecific,                // 0x0185
RRUI4CMD_SyncNXPG2XEASAlarm,                         // 0x0186
RRUI4CMD_ASyncNXPG2XReadProtectAny,                  // 0x0187
RRUI4CMD_ASyncNXPG2XResetReadProtectAny,             // 0x0188
RRUI4CMD_ASyncNXPG2XChangeEASAny,                    // 0x0189
RRUI4CMD_ASyncNXPG2XEASAlarm,                        // 0x018A
RRUI4CMD_GetAntennaList = 0x0201,                    // 0x0201
RRUI4CMD_SetAntenna,                                 // 0x0202
RRUI4CMD_SetFrequency,                               // 0x0203
RRUI4CMD_GetNoiseValue,                              // 0x0204
RRUI4CMD_GetETSILBTReferenceNoiseValue,              // 0x0205
RRUI4CMD_CarrierOnOff,                               // 0x0206
RRUI4CMD_RandomModulationOnOff,                      // 0x0207
RRUI4CMD_MeasureTxPwr,                               // 0x0208
RRUI4CMD_MeasureRxPwr,                               // 0x0209
RRUI4CMD_MeasureVSWR,                                // 0x020A
RRUI4CMD_StartInventory,                             // 0x020B
RRUI4CMD_OpenNoninventoriedTag,                      // 0x020C
RRUI4CMD_OpenSpecificTag,                            // 0x020D
RRUI4CMD_TxGen2CmdSelect,                            // 0x020E
RRUI4CMD_TxGen2CmdACK,                               // 0x020F
RRUI4CMD_TxGen2CmdNAK,                               // 0x0210
RRUI4CMD_TxGen2CmdReqRN,                             // 0x0211
RRUI4CMD_TxGen2CmdAccess,                            // 0x0212
RRUI4CMD_TxGen2CmdRead,                              // 0x0213
RRUI4CMD_TxGen2CmdWrite,                             // 0x0214
RRUI4CMD_TxGen2CmdKill,                              // 0x0215
RRUI4CMD_TxGen2CmdLock,                              // 0x0216
RRUI4CMD_TxGen2CmdBlockWrite,                        // 0x0217
RRUI4CMD_TxGen2CmdBlockErase,                        // 0x0218
RRUI4CMD_TxGen2CmdQuery,                             // 0x0219
RRUI4CMD_TxGen2CmdQueryAdjust,                       // 0x021A
RRUI4CMD_TxGen2CmdQueryRep,                          // 0x021B
RRUI4CMD_TxGen2CmdCustomCmd,                         // 0x021C
RRUI4CMD_EN302208TestmodulationOnOff,                // 0x021D

// RRUI4 command responses
```

4. Command and response transmission

```
RRUI4RES_GetMode = 0x8001,                                  // 0x8001
RRUI4RES_SetMode,                                           // 0x8002
RRUI4RES_GetCommStandard,                                   // 0x8003
RRUI4RES_SetCommStandard,                                   // 0x8004
RRUI4RES_GetPortPower,                                      // 0x8005
RRUI4RES_SetPortPower,                                      // 0x8006
RRUI4RES_GetCarrierFollowUpTime,                            // 0x8007
RRUI4RES_SetCarrierFollowUpTime,                            // 0x8008
RRUI4RES_GetAntennaMode,                                    // 0x8009
RRUI4RES_SetAntennaMode,                                    // 0x800A
RRUI4RES_GetPortMultiplexSequenceAndExposureTime,           // 0x800B
RRUI4RES_SetPortMultiplexSequenceAndExposureTime,           // 0x800C
RRUI4RES_GetCableLossAndAntennaGain,                        // 0x800D
RRUI4RES_SetCableLossAndAntennaGain,                        // 0x800E
RRUI4RES_GetETSIPortChannelList,                            // 0x800F
RRUI4RES_SetETSIPortChannelList,                            // 0x8010
RRUI4RES_GetETSIPortChannelSwitchingMode,                   // 0x8011
RRUI4RES_SetETSIPortChannelSwitchingMode,                   // 0x8012
RRUI4RES_GetProfileList,                                    // 0x8013
RRUI4RES_GetProfile,                                        // 0x8014
RRUI4RES_SetProfile,                                        // 0x8015
RRUI4RES_GetModulationType,                                 // 0x8016
RRUI4RES_SetModulationType,                                 // 0x8017
RRUI4RES_GetExtResultFlag,                                  // 0x8018
RRUI4RES_SetExtResultFlag,                                  // 0x8019
RRUI4RES_GetErrorStatus,                                    // 0x801A
RRUI4RES_GetDefaultParamset,                                // 0x801B
RRUI4RES_SetDefaultParamset,                                // 0x801C
RRUI4RES_GetActiveParamset,                                 // 0x801D
RRUI4RES_SetActiveParamset,                                 // 0x801E
RRUI4RES_SaveActiveParamset,                                // 0x801F
RRUI4RES_GetParameterById,                                  // 0x8020
RRUI4RES_SetParameterById,                                  // 0x8021
RRUI4RES_GetIOCardHwConfig,                                 // 0x8022
RRUI4RES_SetIOCardHwConfig,                                 // 0x8023
RRUI4RES_GetIOCardPrtklConfig,                              // 0x8024
RRUI4RES_SetIOCardPrtklConfig,                              // 0x8025
RRUI4RES_GetTime,                                           // 0x8026
RRUI4RES_SetTime,                                           // 0x8027
RRUI4RES_GetTemp,                                           // 0x8028
RRUI4RES_GetSWVersion,                                      // 0x8029
RRUI4RES_GetHWVersion,                                      // 0x802A
RRUI4RES_GetLicenseKey,                                     // 0x802B
RRUI4RES_SetLicenseKey,                                     // 0x802C
RRUI4RES_LoadFactoryDefaults = 0x8040,                      // 0x8040
RRUI4RES_FwUpdPrepare,                                      // 0x8041
RRUI4RES_FwUpdPutData,                                      // 0x8042
RRUI4RES_FwUpdFlash,                                        // 0x8043
RRUI4RES_ResetSystem,                                       // 0x8044
RRUI4RES_ActivateBuzzer,                                    // 0x8045
RRUI4RES_GetDeviceSerialNmbr,                               // 0x8046
RRUI4RES_GetDeviceTypeNmbr,                                 // 0x8047
RRUI4RES_GetSelSessionAndTarget = 0x8080,                   // 0x8080
```

```
RRUI4RES_SetSelSessionAndTarget,                    // 0x8081
RRUI4RES_GetInitialQValue,                          // 0x8082
RRUI4RES_SetInitialQValue,                          // 0x8083
RRUI4RES_GetMaxAirCommErrors,                       // 0x8084
RRUI4RES_SetMaxAirCommErrors,                       // 0x8085
RRUI4RES_GetASyncObservedListParameters,            // 0x8086
RRUI4RES_SetASyncObservedListParameters,            // 0x8087
RRUI4RES_GetSelectFilterOnOff,                      // 0x8088
RRUI4RES_SetSelectFilterOnOff,                      // 0x8089
RRUI4RES_GetSelectFilterData,                       // 0x808A
RRUI4RES_SetSelectFilterData,                       // 0x808B
RRUI4RES_GPIOGetIOData,                             // 0x808C
RRUI4RES_GPIOSetOutput,                             // 0x808D
RRUI4RES_GPIOAddActionToActionlist,                 // 0x808E
RRUI4RES_GPIOClearActionlist,                       // 0x808F
RRUI4RES_GPIOAssignInputToActionlist,               // 0x8090
RRUI4RES_SyncGetEPCs = 0x8101,                      // 0x8101
RRUI4RES_SyncBulkGetEPCs,                           // 0x8102
RRUI4RES_SyncReadDataAny,                           // 0x8103
RRUI4RES_SyncReadDataSpecific,                      // 0x8104
RRUI4RES_SyncWriteDataAny,                          // 0x8105
RRUI4RES_SyncWriteDataSpecific,                     // 0x8106
RRUI4RES_SyncWriteMaskedDataAny,                    // 0x8107
RRUI4RES_SyncWriteMaskedDataSpecific,               // 0x8108
RRUI4RES_SyncBlockEraseAny,                         // 0x8109
RRUI4RES_SyncBlockEraseSpecific,                    // 0x810A
RRUI4RES_SyncLockAny,                               // 0x810B
RRUI4RES_SyncLockSpecific,                          // 0x810C
RRUI4RES_SyncKillAny,                               // 0x810D
RRUI4RES_SyncKillSpecific,                          // 0x810E
RRUI4RES_SyncWriteEPCToSingleTag,                   // 0x810F
RRUI4RES_ASyncGetRawEPCs,                           // 0x8110
RRUI4RES_ASyncGetEPCs,                              // 0x8111
RRUI4RES_ASyncReadDataAny,                          // 0x8112
RRUI4RES_ASyncWriteDataAny,                         // 0x8113
RRUI4RES_ASyncWriteMaskedDataAny,                   // 0x8114
RRUI4RES_ASyncBlockEraseAny,                        // 0x8115
RRUI4RES_ASyncLockAny,                              // 0x8116
RRUI4RES_ASyncKillAny,                              // 0x8117
RRUI4RES_ASyncStopCommand,                          // 0x8118
RRUI4RES_ASyncGetTagrate,                           // 0x8119
RRUI4RES_SyncWriteEPCSpecific,                      // 0x811A
RRUI4RES_SyncBlockPermalockAny = 0x8120,            // 0x8120
RRUI4RES_SyncBlockPermalockSpecific,                // 0x8121
RRUI4RES_ASyncBlockPermalockAny,                    // 0x8122
RRUI4RES_SyncNXPG2XReadProtectAny = 0x8180,         // 0x8180
RRUI4RES_SyncNXPG2XReadProtectSpecific,             // 0x8181
RRUI4RES_SyncNXPG2XResetReadProtectAny,             // 0x8182
RRUI4RES_SyncNXPG2XResetReadProtectSpecific,        // 0x8183
RRUI4RES_SyncNXPG2XChangeEASAny,                    // 0x8184
RRUI4RES_SyncNXPG2XChangeEASSpecific,               // 0x8185
RRUI4RES_SyncNXPG2XEASAlarm,                        // 0x8186
```

```
        RRUI4RES_ASyncNXPG2XReadProtectAny,                 // 0x8187
        RRUI4RES_ASyncNXPG2XResetReadProtectAny,            // 0x8188
        RRUI4RES_ASyncNXPG2XChangeEASAny,                   // 0x8189
        RRUI4RES_ASyncNXPG2XEASAlarm,                       // 0x818A
        RRUI4RES_GetAntennaList = 0x8201,                   // 0x8201
        RRUI4RES_SetAntenna,                                // 0x8202
        RRUI4RES_SetFrequency,                              // 0x8203
        RRUI4RES_GetNoiseValue,                             // 0x8204
        RRUI4RES_GetETSILBTReferenceNoiseValue,             // 0x8205
        RRUI4RES_CarrierOnOff,                              // 0x8206
        RRUI4RES_RandomModulationOnOff,                     // 0x8207
        RRUI4RES_MeasureTxPwr,                              // 0x8208
        RRUI4RES_MeasureRxPwr,                              // 0x8209
        RRUI4RES_MeasureVSWR,                               // 0x820A
        RRUI4RES_StartInventory,                            // 0x820B
        RRUI4RES_OpenNoninventoriedTag,                     // 0x820C
        RRUI4RES_OpenSpecificTag,                           // 0x820D
        RRUI4RES_TxGen2CmdSelect,                           // 0x820E
        RRUI4RES_TxGen2CmdACK,                              // 0x820F
        RRUI4RES_TxGen2CmdNAK,                              // 0x8210
        RRUI4RES_TxGen2CmdReqRN,                            // 0x8211
        RRUI4RES_TxGen2CmdAccess,                           // 0x8212
        RRUI4RES_TxGen2CmdRead,                             // 0x8213
        RRUI4RES_TxGen2CmdWrite,                            // 0x8214
        RRUI4RES_TxGen2CmdKill,                             // 0x8215
        RRUI4RES_TxGen2CmdLock,                             // 0x8216
        RRUI4RES_TxGen2CmdBlockWrite,                       // 0x8217
        RRUI4RES_TxGen2CmdBlockErase,                       // 0x8218
        RRUI4RES_TxGen2CmdQuery,                            // 0x8219
        RRUI4RES_TxGen2CmdQueryAdjust,                      // 0x821A
        RRUI4RES_TxGen2CmdQueryRep,                         // 0x821B
        RRUI4RES_TxGen2CmdCustomCmd,                        // 0x821C
        RRUI4RES_EN302208TestmodulationOnOff,               // 0x821D

        // RRUI4 asynchronous command responses
        RRUI4REA_ASyncGetRawEPCs = 0xC110,                  // 0xC110
        RRUI4REA_ASyncGetEPCs,                              // 0xC111
        RRUI4REA_ASyncReadDataAny,                          // 0xC112
        RRUI4REA_ASyncWriteDataAny,                         // 0xC113
        RRUI4REA_ASyncWriteMaskedDataAny,                   // 0xC114
        RRUI4REA_ASyncBlockEraseAny,                        // 0xC115
        RRUI4REA_ASyncLockAny,                              // 0xC116
        RRUI4REA_ASyncKillAny,                              // 0xC117
        RRUI4REA_ASyncBlockPermalockAny = 0xC122,           // 0xC122
        RRUI4REA_ASyncNXPG2XReadProtectAny = 0xC187,        // 0xC187
        RRUI4REA_ASyncNXPG2XResetReadProtectAny,            // 0xC188
        RRUI4REA_ASyncNXPG2XChangeEASAny,                   // 0xC189
        RRUI4REA_ASyncNXPG2XEASAlarm,                       // 0xC18A

        // RRUI4 asynchronous messages
        RRUI4REA_ReaderMessage = 0xC301                     // 0xC301
} tRRUI4HostCommCmds;
```

4.1.2 Result flag

```
typedef enum
{
	RRUI4RESULTFLAG_NOERROR = 0,
	RRUI4RESULTFLAG_NODATA,
	RRUI4RESULTFLAG_CRCERROR,
	RRUI4RESULTFLAG_NOLICENSE,
	RRUI4RESULTFLAG_OUTOFRANGE,
	RRUI4RESULTFLAG_NOSTANDARD,
	RRUI4RESULTFLAG_NOANTENNA,
	RRUI4RESULTFLAG_NOFREQUENCY,
	RRUI4RESULTFLAG_NOCARRIER,
	RRUI4RESULTFLAG_ANTENNAERROR,
	RRUI4RESULTFLAG_NOTAG,
	RRUI4RESULTFLAG_MORETHANONETAGINFIELD,
	RRUI4RESULTFLAG_WRONGLICENSEKEY,
	RRUI4RESULTFLAG_FWREJECTED,
	RRUI4RESULTFLAG_WRONGCFM,
	RRUI4RESULTFLAG_NOHANDLE,
	RRUI4RESULTFLAG_NOPROFILE,
	RRUI4RESULTFLAG_NONSPECIFIED = 0x80
} tRRUI4ResultFlag;
```

4.1.3 Reader function mode

```
typedef enum
{
	RRUI4CFM_NORMAL = 0,
	RRUI4CFM_DIRECT
} tRRUI4CommandFunctionMode;
```

4.1.4 Communication standards

```
typedef enum
{
	RRUI4COMMSTANDARD_NONE = 0,
	RRUI4COMMSTANDARD_AUTODETECT,
	RRUI4COMMSTANDARD_ETSI_EN302208,
	RRUI4COMMSTANDARD_ETSI_EN302208_LBT,
	RRUI4COMMSTANDARD_FCC,
	RRUI4COMMSTANDARD_SPECIAL,
	RRUI4COMMSTANDARD_CHINA,
	RRUI4COMMSTANDARD_THAILAND
} tRRUI4CommStandard;
```

4.1.5 Antenna modes

```
typedef enum
{
	RRUI4ANTMODE_ALL = 0,
	RRUI4ANTMODE_ANT1 = 0x10,
	RRUI4ANTMODE_ANT2,
	RRUI4ANTMODE_ANT3,
	RRUI4ANTMODE_ANT4
} tRRUI4AntennaMode;
```

4. Command and response transmission

4.1.6 Channel changeover modes

```
typedef enum
{
        RRUI4PCSW_MODE0 = 0,
        RRUI4PCSW_MODE1
} tRRUI4ETSIPortChannelSwitchingMode;
```

4.1.7 Modulation types

```
typedef enum
{
        RRUI4MT_DSB = 0,        // double-sideband modulation
        RRUI4MT_PRA             // PR-ASK Modulation
} tRRUI4ModulationTypes;
```

4.1.8 On/off flag

```
typedef enum
{
        RRUI4OOF_Off = 0x00,
        RRUI4OOF_On = 0xFF
} tRRUI4OnOffFlag;
```

4.1.9 Coming/going flag

```
typedef enum
{
        RRUI4CG_GOING = 0x00,
        RRUI4CG_DATACHANGING = 0x80,
        RRUI4CG_COMING = 0xFF
} tRRUI4ComingGoingFlag;
```

4.1.10 Message numbers

```
typedef enum
{
        RRUI4MSGID_ERROR_NOFREECHANNEL = 0,
        RRUI4MSGID_ERROR_ANTENNA1,
        RRUI4MSGID_ERROR_ANTENNA2,
        RRUI4MSGID_ERROR_ANTENNA3,
        RRUI4MSGID_ERROR_ANTENNA4,
        RRUI4MSGID_ERROR_ANTENNA5,
        RRUI4MSGID_ERROR_ANTENNA6,
        RRUI4MSGID_ERROR_ANTENNA7,
        RRUI4MSGID_ERROR_ANTENNA8,
        RRUI4MSGID_ERROR_ETSIPWRCHECK1,
        RRUI4MSGID_ERROR_ETSIPWRCHECK2,
        RRUI4MSGID_ERROR_ETSIPWRCHECK3,
        RRUI4MSGID_ERROR_ETSIPWRCHECK4,
        RRUI4MSGID_ERROR_ETSIPWRCHECK5,
        RRUI4MSGID_ERROR_ETSIPWRCHECK6,
        RRUI4MSGID_ERROR_ETSIPWRCHECK7,
        RRUI4MSGID_ERROR_ETSIPWRCHECK8,
        RRUI4MSGID_ERROR_LAST
} tRRUI4ErrorMsgID;
```

4.1.11 Result flags

```
// Explanation of the codes
// --------------------
// A - Antenna port
// R - RSSI value
// T - Time stamp
// PC - Tag protocol control word including where necessary XPC_W1 and XPC_W2
typedef enum
{
        RRUI4ERF_0000 = 0,              // no transmission of PC, T, R or A
        RRUI4ERF_000A,                  // A is transmitted
        RRUI4ERF_00R0,                  // R is transmitted
        RRUI4ERF_00RA,                  // R and A are transmitted
        RRUI4ERF_0T00,                  // T is transmitted
        RRUI4ERF_0T0A,                  // T and A are transmitted
        RRUI4ERF_0TR0,                  // T and R are transmitted
        RRUI4ERF_0TRA,                  // T, R and A are transmitted
        RRUI4ERF_P000,                  // PC is transmitted
        RRUI4ERF_P00A,                  // PC and A are transmitted
        RRUI4ERF_P0R0,                  // PC and R are transmitted
        RRUI4ERF_P0RA,                  // PC, R and A are transmitted
        RRUI4ERF_PT00,                  // PC and T are transmitted
        RRUI4ERF_PT0A,                  // PC, T and A are transmitted
        RRUI4ERF_PTR0,                  // PC, T and R are transmitted
        RRUI4ERF_PTRA,                  // PC, T, R and A are transmitted
        RRUI4ERF_LAST
} tRRUI4ExtendedResultFlag;
```

4.1.12 Tag error codes

```
typedef enum
{
        RRUI4TEC_NOERROR = 0x00,
        RRUI4TEC_NOANSWER,
        RRUI4TEC_ACCESSDENIED,
        RRUI4TEC_VERIFYFAILED,
        RRUI4TEC_UNSPECIFIED,
        RRUI4TEC_OTHER = 0x80,
        RRUI4TEC_MEMORYOVERRUN = 0x83,
        RRUI4TEC_MEMORYLOCKED = 0x84,
        RRUI4TEC_INSUFFICIENTPOWER = 0x8B,
        RRUI4TEC_NONSPECIFIC = 0x8F
} tRRUI4TagErrorCodes;
```

4.1.13 I/O card types

```
typedef enum
{
        IOCT_AUTODETECT = 0,
        IOCT_CARDNOTUSED,
        IOCT_RS232,
        IOCT_RS485,
        IOCT_ETHERNET,
        IOCT_CAN,
        IOCT_GPIO,
        IOCT_LAST
} tIOCardType;
```

4.1.14 I/O communications protocols

```
typedef enum
{
        IOCS_NOTSPECIFIED = 0,
        IOCS_KBP,                       // *** obsolete ***
        IOCS_STDPRTKL,                  // *** obsolete ***
        IOCS_KBRP,                      // Reader Protocol (KBRP)
        IOCS_MODBUS,                    // *** reserved ***
        IOCS_LAST
} tIOCardService;
```

4. Command and response transmission

4.2 System and configuration commands

4.2.1 GetMode

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 FM

I1: Response ID low byte
I2: Response ID high byte
FM: Function mode

Possible function modes:

RRUI4CFM_NORMAL
RRUI4CFM_DIRECT

4.2.2 SetMode

Command block structure:

I1 I2 FM

I1: Command ID low byte
I2: Command ID high byte
FM: Function mode

Possible function modes:

RRUI4CFM_NORMAL
RRUI4CFM_DIRECT

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.3 GetCommStandard

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 CS

I1: Response ID low byte
I2: Response ID high byte
CS: Communication standard

Possible communication standards:

RRUI4COMMSTANDARD_NONE
RRUI4COMMSTANDARD_AUTODETECT
RRUI4COMMSTANDARD_ETSI_EN302208
RRUI4COMMSTANDARD_ETSI_EN302208_LBT
RRUI4COMMSTANDARD_FCC
RRUI4COMMSTANDARD_SPECIAL
RRUI4COMMSTANDARD_CHINA
RRUI4COMMSTANDARD_THAILAND

4.2.4 SetCommStandard

Command block structure:

I1 I2 CS

I1: Command ID low byte
I2: Command ID high byte
CS: Communication standard

Possible communication standards:

RRUI4COMMSTANDARD_NONE
RRUI4COMMSTANDARD_AUTODETECT
RRUI4COMMSTANDARD_ETSI_EN302208
RRUI4COMMSTANDARD_ETSI_EN302208_LBT
RRUI4COMMSTANDARD_FCC
RRUI4COMMSTANDARD_SPECIAL
RRUI4COMMSTANDARD_CHINA
RRUI4COMMSTANDARD_THAILAND

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOLICENSE
RRUI4RESULTFLAG_OUTOFRANGE

4.2.5 GetPortPower

Command block structure:

I1 I2 PN

I1: Command ID low byte
I2: Command ID high byte
PN: Antenna port number (in the range from 1 to 4)

Response block structure:

I1 I2 RF PN [PP]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
PN: Antenna port number (in the range from 1 to 4)
PP: Port power

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOLICENSE
RRUI4RESULTFLAG_OUTOFRANGE

Port power:

The power radiated by the antenna is returned if the result flag is RRUI4RESULTFLAG_NOERROR. The command indicates the power radiation setting for the antenna port in accordance with the cable attenuation and antenna gain.

4.2.6 SetPortPower

Command block structure:

I1 I2 PN PP

I1: Command ID low byte
I2: Command ID high byte
PN: Antenna port number (in the range from 1 to 4)
PP: Port power

The transmission power of an antenna port is specified in 1/4 dBm(erp) (antenna radiated power).
If the value is zero, the antenna port is cut off. The setting range is between 0 x 44 (17 dBm(erp)) and 0 x 84 (33dBm (erp)).

Response block structure:

I1 I2 RF PN

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
PN: Antenna port number (in the range from 1 to 4)

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOLICENSE
RRUI4RESULTFLAG_OUTOFRANGE

4.2.7 GetCarrierFollowUpTime

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 T1 T2
I1: Response ID low byte
I2: Response ID high byte
T1: Carrier follow-up time low byte
T2: Carrier follow-up time high byte

The carrier follow-up time is given in seconds.

4.2.8 SetCarrierFollowUpTime

Command block structure:

I1 I2 T1 T2

I1: Command ID low byte
I2: Command ID high byte
T1: Carrier follow-up time low byte
T2: Carrier follow-up time high byte

The carrier follow-up time is given in seconds.

Response block structure:

I1 I2 RF
I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR

4.2.9 GetAntennaMode

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 AS AA

I1: Response ID low byte
I2: Response ID high byte
AS: Antenna mode for synchronous commands
AA: Antenna mode for asynchronous commands

The following antenna modes are possible:

RRUI4ANTMODE_ALL
RRUI4ANTMODE_ANT1
RRUI4ANTMODE_ANT2
RRUI4ANTMODE_ANT3
RRUI4ANTMODE_ANT4

4.2.10 SetAntennaMode

Command block structure:

I1 I2 AS AA

I1: Command ID low byte
I2: Command ID high byte
AS: Antenna mode for synchronous commands
AA: Antenna mode for asynchronous commands

The following antenna modes are possible:

RRUI4ANTMODE_ALL
RRUI4ANTMODE_ANT1
RRUI4ANTMODE_ANT2

RRUI4ANTMODE_ANT3
RRUI4ANTMODE_ANT4

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.11 GetPortMultiplexSequenceAndExposureTime

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 PN ET1 ET2 [[PN ET1 ET2] ... [PN ET1 ET2]]

I1: Response ID low byte
I2: Response ID high byte
PN: Antenna port number (in the range from 1 to 4)
ET1: Exposure time low byte
ET2: Exposure time high byte

The exposure time is returned by the reader in milliseconds.

Note

Exposure times are only taken into account with asynchronous reader commands.

4.2.12 SetPortMultiplexSequenceAndExposureTime

Command block structure:

I1 I2 PN ET1 ET2 [[PN ET1 ET2] ... [PN ET1 ET2]]

I1: Command ID low byte
I2: Command ID high byte
PN: Antenna port number (in the range from 1 to 4)
ET1: Exposure time low byte
ET2: Exposure time high byte

The exposure time is given in milliseconds.
1 to 8 ports and their exposure times can be given.

Note

Exposure times are only taken into account with asynchronous reader commands.

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOLICENSE

4.2.13 GetCableLossAndAntennaGain

Command block structure:

I1 I2 PN

I1: Command ID low byte
I2: Command ID high byte
PN: Antenna port number (in the range from 1 to 4)

Response block structure:

I1 I2 RF PN [CL AG]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
PN: Antenna port number (in the range from 1 to 4)
CL: Cable loss in ¼ dB
AG: Antenna gain in ¼ dBic

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOLICENSE

4.2.14 SetCableLossAndAntennaGain

Command block structure:

I1 I2 PN CL AG

I1: Command ID low byte
I2: Command ID high byte
PN: Antenna port number (in the range from 1 to 4)
CL: Cable loss in ¼ dB (in the range from 0 dB to 63.75 dB)
AG: Antenna gain in ¼ dBic (in the range from -32 dBic to 31.75 dBic)

Response block structure:

I1 I2 RF PN

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
PN: Antenna port number (in the range from 1 to 4)

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOLICENSE

4.2.15 GetETSIPortChannelList

Command block structure:

I1 I2 PN

I1: Command ID low byte
I2: Command ID high byte
PN: Antenna port number (in the range from 0 to 4; 0 returns the global channel list – 1 to 4 return the list for the related antenna port)

Response block structure:

I1 I2 RF PN [CH ... CH]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
PN: Antenna port number (in the range from 0 to 4)
CH: .10. ETSI channel in the range from 1 to 15

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOLICENSE

A channel list is returned only if the result flag is RRUI4RESULTFLAG_NOERROR and also only if the list for the respective port is stored in the reader.

4.2.16 SetETSIPortChannelList

Command block structure:

I1 I2 PN [CH ... CH]

I1: Command ID low byte
I2: Command ID high byte
PN: Antenna port number (in the range from 0 to 4; 0 for the global channel list – 1 to 4 for the related antenna port)
CH: .10. ETSI channel in the range from 1 to 15

1 to 16 channels can be passed.

Response block structure:

I1 I2 RF PN

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
PN: Antenna port number (in the range from 0 to 4)

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOLICENSE

4.2.17 GetETSIPortChannelSwitchingMode

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 SM

I1: Response ID low byte
I2: Response ID high byte
SM: Switching mode

Possible switching modes:

RRUI4PCSW_MODE0
RRUI4PCSW_MODE1

4.2.18 GetETSIPortChannelSwitchingMode

Command block structure:

I1 I2 SM

I1: Command ID low byte
I2: Command ID high byte
SM: Switching mode

Possible switching modes:

RRUI4PCSW_MODE0
RRUI4PCSW_MODE1

Response block structure:

I1 I2 RF
I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.19 GetProfileList

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 [PL ... PL]

I1: Response ID low byte
I2: Response ID high byte
PL: Data bytes for profile list

The profile lists comprises 0 to n profile list entries that are sent one after the other. A profile list entry has the following structure:

PN PT ... PT 00

PN: Profile number
PT: ASCII-coded profile description text
00: Null byte as terminating character

4.2.20 GetProfile

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 PN

I1: Response ID low byte
I2: Response ID high byte
PN: Profile number

4.2.21 SetProfile

Command block structure:

I1 I2 PN

I1: Command ID low byte
I2: Command ID high byte
PN: Profile number

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.22 GetModulationType

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 MT

I1: Response ID low byte
I2: Response ID high byte
MT: Modulation type

Possible modulation types:

RRUI4MT_DSB
RRUI4MT_PRA

4.2.23 SetModulationType

Command block structure:

I1 I2 MT

I1: Command ID low byte
I2: Command ID high byte
MT: Modulation type

Possible modulation types:

RRUI4MT_DSB
RRUI4MT_PRA

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.24 GetExtResultFlag

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 EF

I1: Response ID low byte
I2: Response ID high byte
EF: Extended result flag

Possible extended result flags:

RRUI4ERF_0000
RRUI4ERF_000A
RRUI4ERF_00R0
RRUI4ERF_00RA
RRUI4ERF_0T00
RRUI4ERF_0T0A
RRUI4ERF_0TR0
RRUI4ERF_0TRA
RRUI4ERF_P000
RRUI4ERF_P00A
RRUI4ERF_P0R0
RRUI4ERF_P0RA
RRUI4ERF_PT00
RRUI4ERF_PT0A
RRUI4ERF_PTR0
RRUI4ERF_PTRA

4.2.25 SetExtResultFlag

Command block structure:

II1 I2 EF

I1: Command ID low byte
I2: Command ID high byte
EF: Extended result flag

Possible extended result flags:

RRUI4ERF_0000
RRUI4ERF_000A
RRUI4ERF_00R0
RRUI4ERF_00RA
RRUI4ERF_0T00
RRUI4ERF_0T0A
RRUI4ERF_0TR0
RRUI4ERF_0TRA
RRUI4ERF_P000
RRUI4ERF_P00A
RRUI4ERF_P0R0
RRUI4ERF_P0RA
RRUI4ERF_PT00
RRUI4ERF_PT0A
RRUI4ERF_PTR0
RRUI4ERF_PTRA

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.26 GetErrorStatus

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 E1 E2 E3 E4

I1: Response ID low byte
I2: Response ID high byte
E1: Error flag register LSB
E2: Error flag register 2nd byte
E3: Error flag register 3rd byte
E4: Error flag register MSB

The bits in the error flag register have the following significance:

D0: ETSI: No free transmit channel
D1: Antenna error antenna 1
D2: Antenna error antenna 2
D3: Antenna error antenna 3

D4: Antenna error antenna 4
D5: Antenna error antenna 5 (not used)
D6: Antenna error antenna 6 (not used)
D7: Antenna error antenna 7 (not used)
D8: Antenna error antenna 8 (not used)
D9: ETSI: Configuration power check error antenna 1
D10: ETSI: Configuration power check error antenna 2
D11: ETSI: Configuration power check error antenna 3
D12: ETSI: Configuration power check error antenna 4
D13: ETSI: Configuration power check error antenna 5 (not used)
D14: ETSI: Configuration power check error antenna 6 (not used)
D15: ETSI: Configuration power check error antenna 7 (not used)
D16: ETSI: Configuration power check error antenna 8 (not used)

4.2.27 GetDefaultParamset

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 PS

I1: Response ID low byte
I2: Response ID high byte
PS: parameter set number

4.2.28 SetDefaultParamset

Command block structure:

I1 I2 PS

I1: Command ID low byte
I2: Command ID high byte
PS: Parameter set number (in the range from 0 to 7; > 7 sets the active parameter set as the default value)

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR

4.2.29 GetActiveParamset

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 PS

I1: Response ID low byte
I2: Response ID high byte
PS: parameter set number

4.2.30 SetActiveParamset

Command block structure:

I1 I2 PS

I1: Command ID low byte
I2: Command ID high byte
PS: Parameter set number (in the range from 0 to 7)

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.31 SaveActiveParamset

Saves the parameter of the commands “SetCommStandard”, “SetPortPower”, “SetCarrierFollowUpTime”, “SetPortMultiplexSequenceAndExposureTime”, “SetCableLossAndAntennaGain”, “SetETSIPortChannelList”, “SetETSIPortChannelSwitchingMode”, “SetProfile”, “SetModulationType”, “SetExtResultFlag”, “SetSelSessionAndTarget”, “SetInitialQValue”, “SetMaxAirCommErrors” and “SetASyncObservedListParameters” in the internal EEPROM. The parameters saved are used again automatically after a re-start.

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR

4.2.32 GetParameterById

Command block structure:

I1 I2 P1 P2 P3 P4

I1: Command ID low byte
I2: Command ID high byte
P1: Parameter ID LSB
P2: Parameter ID 2nd byte

P3: Parameter ID 3rd byte
P4: Parameter ID MSB

Response block structure:

I1 I2 RF [P1 P2 P3 P4 [V1 [... [V4]]]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
P1: Parameter ID LSB
P2: Parameter ID 2nd byte
P3: Parameter ID 3rd byte
P4: Parameter ID MSB
V1: Parameter value LSB
V2: Parameter value 2nd byte
V3: Parameter value 3rd byte
V4: Parameter value MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.33 SetParameterById

Command block structure:

I1 I2 P1 P2 P3 P4 V1 [V2 [V3 [V4]]]

I1: Command ID low byte
I2: Command ID high byte
P1: Parameter ID LSB
P2: Parameter ID 2nd byte
P3: Parameter ID 3rd byte
P4: Parameter ID MSB
V1: Parameter value LSB
V2: Parameter value 2nd byte
V3: Parameter value 3rd byte
V4: Parameter value MSB

Response block structure:

I1 I2 RF [P1 P2 P3 P4]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
P1: Parameter ID LSB
P2: Parameter ID 2nd byte
P3: Parameter ID 3rd byte
P4: Parameter ID MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.34 GetIOCardHwConfig

Command block structure:

I1 I2 CN

I1: Command ID low byte
I2: Command ID high byte
CN: I/O card number (in the range from 0 to 15)

Response block structure:

I1 I2 RF [CN [CT CS CD ... CD]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
CN: I/O card number
CT: I/O card type
CS: I/O communications protocol
CD: Configuration data

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_OUTOFRANGE

Configuration data are only returned if the result flag is RRUI4RESULTFLAG_NOERROR. The explanation of the structure of the configuration data is not part of this document.

4.2.35 SetIOCardHwConfig

Command block structure:

I1 I2 CN CT CS CD ... CD

I1: Command ID low byte
I2: Command ID high byte
CN: I/O card number (in the range from 0 to 15)
CT: I/O card type
CS: I/O communications protocol
CD: Configuration data

Response block structure:

I1 I2 RF [CN]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
CN: I/O card number

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.36 GetIOCardPrtklConfig

Command block structure:

I1 I2 CN

I1: Command ID low byte
I2: Command ID high byte

CN: I/O card number (in the range from 0 to 15)

Response block structure:

I1 I2 RF [CN [CD ... CD]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
CN: I/O card number
CD: Configuration data

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
Configuration data are only returned if the result flag is RRUI4RESULTFLAG_NOERROR. The explanation of the structure of the configuration data is not part of this document.

4.2.37 SetIOCardPrtkIConfig

Command block structure:

I1 I2 CN CD ... CD

I1: Command ID low byte
I2: Command ID high byte
CN: I/O card number (in the range from 0 to 15)
CD: Configuration data

Response block structure:

I1 I2 RF [CN]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
CN: I/O card number

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.38 GetTime

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 T1 T2 T3 T4

I1: Response ID low byte
I2: Response ID high byte
T1: UTC time stamp LSB
T2: UTC time stamp byte 2
T3: UTC time stamp byte 3
T4: UTC time stamp MSB

The time stamp gives the UTC time in seconds since 1.1.1970 0:00 hours.

4.2.39 SetTime

Command block structure:

I1 I2 T1 T2 T3 T4

I1: Command ID low byte
I2: Command ID high byte
T1: UTC time stamp LSB
T2: UTC time stamp byte 2
T3: UTC time stamp byte 3
T4: UTC time stamp MSB

The time stamp gives the UTC time in seconds since 1.1.1970 0:00 hours.

Response block structure:1

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR

4.2.40 GetTemp

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 T1 T2

I1: Response ID low byte
I2: Response ID high byte
T1: Reader internal temperature low byte
T2: Reader internal temperature high byte

The internal temperature in the reader is transmitted as a signed 16-bit integer with a resolution of 1/100°C.

4.2.41 GetSWVersion

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 H1 H2 N1 N2 U1 U2 B1 B2 B3 B4

I1: Response ID low byte
I2: Response ID high byte
H1: Major version number low byte
H2: Major version number high byte
N1: Minor version number low byte

N2: Minor version number high byte
U1: Revision number low byte
U2: Revision number high byte
B1: Build number LSB
B2: Build number 2nd byte
B3: Build number 3rd byte
B4: Build number MSB

Version structure:

Software version: major version no. point minor version no. point revision no. (Build: build number)

The minor version number and the revision number have two digits as a minimum.
For example:
Software version: 1.22.00 (Build: 1914)

4.2.42 GetHWVersion

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 CV .. CV 00 PV .. PV 00 [[C0 .. C0 00] .. [CF .. CF 00]]

I1: Response ID low byte
I2: Response ID high byte
CV: Identifier text (ASCII-coded) for the CPU module
PV: Identifier text (ASCII-coded) for the PA module
00: Separating and terminating character
C0 to CF: Identifier text (ASCII-coded) for the I/O modules 0 to 15 (only as far as available and as far as the modules support this function)

Version structure:

CPU module: identifier text
PA module: identifier text

For example:

CPU module: 136B211B
PA module: 136B224B

4.2.43 GetLicenseKey

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 LK ... LK 00

I1: Response ID low byte
I2: Response ID high byte
LK: License key (ASCII-coded)
00: Terminating character

A license key has the format:

xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx

4.2.44 SetLicenseKey

Command block structure:

I1 I2 LK ... LK 00

I1: Command ID low byte
I2: Command ID high byte
LK: License key (ASCII-coded)
00: Terminating character

A license key has the format:

xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx-xxxxx

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_WRONGLICENSEKEY

4.2.45 LoadFactoryDefaults

Command block structure:

I1 I2 [PS]

I1: Command ID low byte
I2: Command ID high byte
PS: Set of parameters number (for values between 0 and 7:>7 the factory defaults are loaded in the active set of parameters). If no number is given for the set of parameters, all the parameters are set back to factory default.

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
If no number is given for the set of parameters, all the parameters are overwritten with the factory defaults.

4.2.46 FwUpdPrepare

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOLICENSE

4.2.47 FwUpdPutData

Command block structure:

I1 I2 FD [FD ... FD]

I1: Command ID low byte
I2: Command ID high byte
FD: Firmware data

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_FWREJECTED
RRUI4RESULTFLAG_NOLICENSE

4.2.48 FwUpdFlash

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_FWREJECTED
RRUI4RESULTFLAG_NOLICENSE

4.2.49 ResetSystem

Command block structure:

I1 I2 5A A5

I1: Command ID low byte
I2: Command ID high byte
5A: Constant
A5: Constant

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.50 ActivateBuzzer

Command block structure:

I1 I2 T1 T2

I1: Command ID low byte
I2: Command ID high byte
T1: Buzzer time LSB
T2: Buzzer time MSB

The buzzer time sets the length of time in milliseconds for which the buzzer is activated. A value of 0 switches the buzzer off, a value of 0xFFFF switches the buzzer on continuously.

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.2.51 GetDeviceSerialNmbr

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF [[ST ... ST] 00 [HT ... HT] 00]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
ST: Text serial number
HT: Text hardware version

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NONSPECIFIED

4.2.52 GetDeviceTypeNmbr

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF [[DT ... DT] 00]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
DT: Text device type number

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NONSPECIFIED

4. Command and response transmission

4.3 Commands in “Normal Mode” and in “Direct Mode”

4.3.1 GetSelSessionAndTarget

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 SL SN TA

I1: Response ID low byte
I2: Response ID high byte
SL: Query-Sel (0 to 3; see EPCglobal specification V1.2.0 page 57)
SN: Query session (0 to 3; see EPCglobal specification V1.2.0 page 57)
TA: Query-Target (0: A; 1: B; see EPCglobal specification V1.2.0 page 57)

4.3.2 SetSelSessionAndTarget

Command block structure:

I1 I2 SL SN TA

I1: Command ID low byte
I2: Command ID high byte
SL: Query-Sel (0 to 3; see EPCglobal specification V1.2.0 page 57)
SN: Query session (0 to 3; see EPCglobal specification V1.2.0 page 57)
TA: Query-Target (0: A; 1: B; see EPCglobal specification V1.2.0 page 57)

Note

In normal mode the parameter “Query-Sel” affects only the commands “...GetEPCs”, “...Any” and “SyncWriteEPC-ToSingleTag” and then only if at least one select filter is active. The parameter “Query-Target” does not affect the “...Specific” commands.

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.3 GetInitialQValue

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 QV

I1: Response ID low byte
I2: Response ID high byte
QV: Q value

4.3.4 SetInitialQValue

Command block structure:

I1 I2 QV

I1: Command ID low byte
I2: Command ID high byte
QV: Q value

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.5 GetMaxAirCommErrors

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 AE

I1: Response ID low byte
I2: Response ID high byte
AE: MaxAirCommErrors value

4.3.6 SetMaxAirCommErrors

Command block structure:

I1 I2 AE

I1: Command ID low byte
I2: Command ID high byte
AE: MaxAirCommErrors value

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR

4.3.7 GetASyncObservedListParameters

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 P1 P2 P3

I1: Response ID low byte
I2: Response ID high byte
P1: Glimpsed timeout count
P2: Observed threshold count
P3: Observed timeout count

4.3.8 SetASyncObservedListParameters

Command block structure:

I1 I2 P1 P2 P3

I1: Command ID low byte
I2: Command ID high byte
P1: Glimpsed timeout count (in the range from 0 to 255)
P2: Observed threshold count (in the range from 0 to 255)
P3: Observed timeout count (in the range from 0 to 255)

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR

4.3.9 GetSelectFilterOnOff

Command block structure:

I1 I2 FN

I1: Command ID low byte
I2: Command ID high byte
FN: Filter number (in the range from 0 to 31)

Response block structure:

I1 I2 RF FN [OO]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
FN: Filter number
OO: On/off flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.10 SetSelectFilterOnOff

Command block structure:

I1 I2 FN OO

I1: Command ID low byte
I2: Command ID high byte
FN: Filter number (in the range from 0 to 31)
OO: On/off flag

Response block structure:

I1 I2 RF FN

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
FN: Filter number

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.11 GetSelectFilterData

Command block structure:

I1 I2 FN

I1: Command ID low byte
I2: Command ID high byte
FN: Filter number (in the range from 0 to 31)

Response block structure:

I1 I2 RF FN [OO TA AC MB P1 P2 P3 P4 ML [MD ... MD]]

I1: Response ID low byte
I2: Response ID high byte

RF: Result flag
FN: Filter number
OO: On/off flag
TA: Target (see EPCglobal specification V1.2.0 page 56)
AC: Action (see EPCglobal specification V1.2.0 page 56)
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Bit pointer LSB
P2: Bit pointer 2nd byte
P3: Bit pointer 3rd byte
P4: Bit pointer MSB
ML: Masked length in bits
MD: Masked data

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.12 SetSelectFilterData

Command block structure:

I1 I2 FN OO TA AC MB P1 P2 P3 P4 ML [MD ... MD]

I1: Command ID low byte
I2: Command ID high byte
FN: Filter number (in the range from 0 to 31)
OO: On/off flag
TA: Target (see EPCglobal specification V1.2.0 page 56)
AC: Action (see EPCglobal specification V1.2.0 page 56)
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Bit pointer LSB
P2: Bit pointer 2nd byte
P3: Bit pointer 3rd byte
P4: Bit pointer MSB
ML: Masked length in bits
MD: Masked data

Note

The bytes of the masked data are transmitted over the air interface in the sequence of their issue from the MSB to the LSB. If the masked data end within a byte with a masked length that is not a multiple of eight, the last bits of the lower range of the last masked data byte are used. Thus if for instance five bits remain to be transmitted, bits B4 to B0 of the last masked byte are transmitted over the air interface in this sequence.

Response block structure:

I1 I2 RF FN

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
FN: Filter number

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.13 GPIOGetIOData

Command block structure:

I1 I2 CN

I1: Command ID low byte
I2: Command ID high byte
CN: GPIO card number (in the range from 0 to 15)

Response block structure:

I1 I2 RF [CN [IP1 IP2 IL1 IL2 ID1 ID2 OL1 OL2 OP1 OP2]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
CN: GPIO card number
IP1: LSB physical inputs
IP2: MSB physical inputs
IL1: LSB logical inputs
IL2: MSB logical inputs
ID1: Debounced LSB logical inputs
ID2: Debounced MSB logical inputs
OL1: Debounced LSB logical outputs
OL2: MSB logical outputs
OP1: LSB physical outputs
OP2: MSB physical outputs

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

Note

Since setting the GPIO outputs and reading the GPIO inputs have a higher priority than executing this command, it may happen during the course of transmission that inconsistencies arise between the physical and logical inputs and outputs.

4.3.14 GPIOSetOutput

Command block structure:

I1 I2 CN ON T1 T2

I1: Command ID low byte
I2: Command ID high byte
CN: GPIO card number (in the range from 0 to 15)
ON: Output number (starting at 0)
T1: Activation time of LSB, in milliseconds
T2: Activation time of MSB, in milliseconds

Note

If the activation time is set to 65535 milliseconds, the output is continuously activated. To switch an output off before expiry of the activation time, transmit a command with the activation time “0”.

Response block structure:

I1 I2 RF [CN ON]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
CN: GPIO card number
ON: Output number

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.15 GPIOAddActionToActionlist

Command block structure:

I1 I2 AL AD ... AD

I1: Command ID low byte
I2: Command ID high byte
AL: Action list (in the range from 0 to 127)
AD: Action data

Note

The “Action data” range consists of a command ID and its associated parameters. At least 2 bytes of “Action data” must be transmitted.

Response block structure:

I1 I2 RF [AL]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
AL: Action list

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.16 GPIOClearActionlist

Command block structure:

I1 I2 AL

I1: Command ID low byte
I2: Command ID high byte
AL: Action list (in the range from 0 to 127)

Response block structure:

I1 I2 RF [AL]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
AL: Action list

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4.3.17 GPIOAssignInputToActionlist

Command block structure:

I1 I2 CN IN EG AL

I1: Command ID low byte
I2: Command ID high byte
CN: GPIO card number (in the range from 0 to 15)
IN: Input number (starting at 0)
EG: Flank (0: negative flank; <>0: positive flank)
AL: Action list (in the range 0 to 127; from a value of 128 the flank is not assigned to an action list)

Response block structure:

I1 I2 RF [CN IN EG]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
CN: GPIO card number
IN: Input number
EG: Flank

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE

4. Command and response transmission

4.4 Commands in “Normal Mode”

4.4.1 SyncGetEPCs

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En]]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.2 SyncBulkGetEPCs

Command block structure:

I1 I2 T1 T2

I1: Command ID low byte
I2: Command ID high byte
T1: Bulk read time low byte
T2: Bulk read time high byte

The bulk read time is given in milliseconds.

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En]]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.3 SyncReadDataAny

Command block structure:

I1 I2 W1 W2 W3 W4 MB P1 P2 P3 P4 WC

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to read

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC [WC D1 ... Dx]]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC [WC D1 ... Dx]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte

P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code
WC: Number of words read
D1: Response data LSB
...
Dx: Response data MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.4 SyncReadDataSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 MB P1 P2 P3 P4 WC

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to read

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC [WC D1 ... Dx]]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC [WC D1 ... Dx]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte

P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code
WC: Number of words read
D1: Response data LSB
...
Dx: Response data MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.5 SyncWriteDataAny

Command block structure:

I1 I2 W1 W2 W3 W4 MB P1 P2 P3 P4 WC DL DH [DL DH ... DL DH]

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to write
DL: Word to write low byte
DH: Word to write high byte

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte

X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Mögliche Result-Flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.6 SyncWriteDataSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 MB P1 P2 P3 P4 WC DL DH [DL DH ... DL DH]

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to write
DL: Word to write low byte
DH: Word to write high byte

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte

EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.7 SyncWriteMaskedDataAny

Command block structure:

I1 I2 W1 W2 W3 W4 MB P1 P2 P3 P4 WC DL DH [DL DH ... DL DH] ML MH [ML MH ... ML MH]

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to write
DL: Word to write low byte
DH: Word to write high byte
ML: Write mask low byte
MH: Write mask high byte

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.8 SyncWriteMaskedDataSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 MB P1 P2 P3 P4 WC DL DH [DL DH ... DL DH] ML MH [ML MH ... ML MH]

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to write
DL: Word to write low byte
DH: Word to write high byte
ML: Write mask low byte
MH: Write mask high byte

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.9 SyncBlockEraseAny

Command block structure:

I1 I2 W1 W2 W3 W4 MB P1 P2 P3 P4 WC

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to erase

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.10 SyncBlockEraseSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 MB P1 P2 P3 P4 WC

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to erase

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.11 SyncLockAny

Command block structure:

I1 I2 W1 W2 W3 W4 L1 L2 L3

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
L1: Payload LSB
L2: Payload 2nd byte
L3: Payload MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.12 SyncLockSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 L1 L2 L3

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
L1: Payload LSB
L2: Payload 2nd byte
L3: Payload MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.13 SyncKillAny

Command block structure:

I1 I2 W1 W2 W3 W4 RC

I1: Command ID low byte
I2: Command ID high byte
W1: Kill password LSB

W2: Kill password 2nd byte
W3: Kill password 3rd byte
W4: Kill password MSB
RC: Recommissioning bits (see EPCglobal specification V1.2.0 page 68)

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.14 SyncKillSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 RC

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Kill password LSB
W2: Kill password 2nd byte
W3: Kill password 3rd byte
W4: Kill password MSB
RC: Recommissioning bits (see EPCglobal specification V1.2.0 page 68)

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.15 SyncBlockPermalockAny

Command block structure:

I1 I2 W1 W2 W3 W4 00 RL MB P1 P2 P3 P4 BR [LL LH ... [LL LH]]

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
00: reserved always 0x00
RL: Read / Lock (see EPCglobal specification V1.2.0 page 76)
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Block word pointer LSB
P2: Block word pointer 2nd byte
P3: Block word pointer 3rd byte
P4: Block word pointer MSB
BR: Block range (see EPCglobal specification V1.2.0 page 77)
LL: Lock data LSB
LH: Lock data MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC [BR L1 ... Lx]]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC [BR L1 ... Lx]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag

AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code
BR: Block range
L1: Lock data LSB
...
Lx: Lock data MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.16 SyncBlockPermalockSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 00 RL MB P1 P2 P3 P4 BR [LLLH ... [LL LH]]

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
00: reserved always 0x00
RL: Read / Lock (see EPCglobal specification V1.2.0 page 76)
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Block word pointer LSB
P2: Block word pointer 2nd byte
P3: Block word pointer 3rd byte
P4: Block word pointer MSB
BR: Block range (see EPCglobal specification V1.2.0 page 77)
LL: Lock data LSB
LH: Lock data MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC [BR L1 ... Lx]]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC [BR L1 ... Lx]]

I1: Response ID low byte
I2: Response ID high byte

RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code
BR: Block range
L1: Lock data LSB
...
Lx: Lock data MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.17 SyncWriteEPCSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 NW [N1 … Nn]

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
NW: Number of EPC words in the new EPC (in the range from 0 to 31)
N1: new EPC LSB
...
Nn: new EPC MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port

RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.18 SyncWriteEPCToSingleTag

Command block structure:

I1 I2 W1 W2 W3 W4 EW E1 ... En

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
EW: Number of EPC words (in the range from 1 to 31)
E1: EPC LSB
...
En: EPC MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW E1 ... En EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_MORETHANONETAGINFIELD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.19 SyncNXPG2XReadProtectAny

Command block structure:

I1 I2 W1 W2 W3 W4

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.20 SyncNXPG2XReadProtectSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.21 SyncNXPG2XResetReadProtectAny

Command block structure:

I1 I2 W1 W2 W3 W4

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.22 SyncNXPG2XResetReadProtectSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.23 SyncNXPG2XChangeEASAny

Command block structure:

I1 I2 W1 W2 W3 W4 AB

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
AB: EAS alarm bit

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

II1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte

T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.24 SyncNXPG2XChangeEASSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 AB

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (in the range from 0 to 31)
E1: EPC LSB
...
En: EPC MSB
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
AB: EAS alarm bit

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1... En] EC]
...
[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.25 SyncNXPG2XEASAlarm

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF [EF]
[[AP] [RS] [T1 T2 T3 T4] 00]
...
[[AP] [RS] [T1 T2 T3 T4] 00]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
00: (0x00: wildcard)

Possible result flags:
RRUI4RESULTFLAG_NOERROR (receive alarm code)
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM

4.4.26 ASyncGetRawEPCs

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_WRONGCFM

Asynchronous response block structure:

I1 I2 EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En]

I1: Response ID low byte
I2: Response ID high byte
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB

4.4.27 ASyncGetEPCs

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_WRONGCFM

Asynchronous response block structure:

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2LX2H]]] EW [E1 ... En]

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte

EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB

4.4.28 ASyncReadDataAny

Command block structure:

I1 I2 W1 W2 W3 W4 MB P1 P2 P3 P4 WC

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to read

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC [WC D1 ... Dx]

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code
WC: Number of words read
D1: Response data LSB
...
Dx: Response data MSB

4.4.29 ASyncWriteDataAny

Command block structure:

I1 I2 W1 W2 W3 W4 MB P1 P2 P3 P4 WC DL DH [DL DH ... DL DH]

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to write
DL: Word to write low byte
DH: Word to write high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

4.4.30 ASyncWriteMaskedDataAny

Command block structure:

I1 I2 W1 W2 W3 W4 MB P1 P2 P3 P4 WC DL DH [DL DH ... DL DH] ML MH [ML MH ... ML MH]

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to write
DL: Word to write low byte
DH: Word to write high byte
ML: Write mask low byte
MH: Write mask high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

4.4.31 ASyncBlockEraseAny

Command block structure:

II1 I2 W1 W2 W3 W4 MB P1 P2 P3 P4 WC

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to erase

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

4.4.32 ASyncLockAny

Command block structure:

I1 I2 W1 W2 W3 W4 L1 L2 L3

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
L1: Payload LSB
L2: Payload 2nd byte
L3: Payload MSB

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

4.4.33 ASyncKillAny

Command block structure:

I1 I2 W1 W2 W3 W4 RC

I1: Command ID low byte
I2: Command ID high byte
W1: Kill password LSB
W2: Kill password 2nd byte
W3: Kill password 3rd byte
W4: Kill password MSB
RC: Recommissioning bits (see EPCglobal specification V1.2.0 page 68)

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

4.4.34 ASyncBlockPermalockAny

Command block structure:

I1 I2 W1 W2 W3 W4 00 RL MB P1 P2 P3 P4 BR [LL LH ... [LL LH]]

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
00: reserved → always 0x00
RL: Read / Lock (see EPCglobal specification V1.2.0 page 76)
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Block word pointer LSB
P2: Block word pointer 2nd byte
P3: Block word pointer 3rd byte
P4: Block word pointer MSB
BR: Block range (see EPCglobal specification V1.2.0 page 77)
LL: Lock data LSB
LH: Lock data MSB

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC [BR L1 ... Lx]

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code
BR: Block range
L1: Lock data LSB
...
Lx: Lock data MSB

4.4.35 ASyncNXPG2XReadProtectAny

Command block structure:

I1 I2 W1 W2 W3 W4

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

4.4.36 ASyncNXPG2XResetReadProtectAny

Command block structure:

I1 I2 W1 W2 W3 W4

I1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

4.4.37 ASyncNXPG2XChangeEASAny

Command block structure:

I1 I2 W1 W2 W3 W4 AB

II1: Command ID low byte
I2: Command ID high byte
W1: Access password LSB
W2: Access password 2nd byte
W3: Access password 3rd byte
W4: Access password MSB
AB: EAS alarm bit

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] EC

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag XPC_W1 low byte
X1H: Tag XPC_W1 high byte
X2L: Tag XPC_W2 low byte
X2H: Tag XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

4.4.38 ASyncNXPG2XEASAlarm

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOSTANDARD
RRUI4RESULTFLAG_WRONGCFM

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] 00

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (coming message: 0xFF)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
00: (0x00: wildcard)

4.4.39 ASyncStopCommand

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_WRONGCFM

4.4.40 ASyncGetTagrate

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF [R1 R2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
R1: Tag rate low byte
R2: Tag rate high byte

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_WRONGCFM

4. Command and response transmission

4.5 Commands in “direct mode”

4.5.1 GetAntennaList

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 PM

I1: Response ID low byte
I2: Response ID high byte
PM: Antenna port mask (each 1 bit stands for a configured antenna)

4.5.2 SetAntenna

Command block structure:

I1 I2 PN

I1: Command ID low byte
I2: Command ID high byte
PN: Antenna port number (in the range from 1 to 4)

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_WRONGCFM

4.5.3 SetFrequency

Command block structure:

I1 I2 F1 F2 F3

I1: Command ID low byte
I2: Command ID high byte
F1: Frequency LSB
F2: Frequency 2nd byte
F3: Frequency MSB

Command block structure:

I1 I2 F1 F2 F3

I1: Command ID low byte
I2: Command ID high byte
F1: Frequency LSB
F2: Frequency 2nd byte
F3: Frequency MSB

The frequency is given as a multiple of 1 kHz.
Example for 865.7MHz: I1 I2 A4 35 0D

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_WRONGCFM

4.5.4 GetNoiseValue

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF [NV]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
NV: Noise value

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOANTENNA
RRUI4RESULTFLAG_NOFREQUENCY
RRUI4RESULTFLAG_WRONGCFM

4.5.5 GetETSILBTReferenceNoiseValue

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RV

I1: Response ID low byte
I2: Response ID high byte
RV: LBT reference noise value

4.5.6 CarrierOnOff

Command block structure:

I1 I2 OO

I1: Command ID low byte
I2: Command ID high byte
OO: On/off flag

Response block structure:

1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOANTENNA
RRUI4RESULTFLAG_NOFREQUENCY
RRUI4RESULTFLAG_NOPROFILE
RRUI4RESULTFLAG_ANTENNAERROR
RRUI4RESULTFLAG_WRONGCFM

4.5.7 RandomModulationOnOff

Command block structure:

I1 I2 OO

I1: Command ID low byte
I2: Command ID high byte
OO: On/off flag

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.8 EN302208TestmodulationOnOFF

This command generates a test signal to chapter 6.1.1 and chapter 8.4.2 of the “EN 302 208-1 V1.3.1 of December 2009“.

Command block structure:

I1 I2 OO

I1: Command ID low byte
I2: Command ID high byte
OO: On-/Off flag

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:
RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.9 MeasureTxPwr

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 V1 V2 V3 V4

I1: Response ID low byte
I2: Response ID high byte
V1: Measured value LSB
V2: Measured value 2nd byte
V3: Measured value 3rd byte
V4: Measured value MSB

The measured value is a 32-bit floating point number with the units dBm.

4.5.10 MeasureRxPwr

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 V1 V2 V3 V4

I1: Response ID low byte
I2: Response ID high byte
V1: Measured value LSB
V2: Measured value 2nd byte
V3: Measured value 3rd byte
V4: Measured value MSB

The measured value is a 32-bit floating point number with the units dBm.

4.5.11 MeasureVSWR

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 V1 V2 V3 V4

I1: Response ID low byte
I2: Response ID high byte
V1: Measured value LSB
V2: Measured value 2nd byte
V3: Measured value 3rd byte
V4: Measured value MSB

The measured value is a 32-bit floating point number.

4.5.12 Startinventory

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.13 OpenNoninventoriedTag

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF [EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] [H1 H2]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag-PC Low-Byte
P2: Tag-PC High-Byte
X1L Tag-XPC_W1 low byte
X1H Tag-XPC_W1 high byte
X2L: Tag-XPC_W2 low byte
X2H: Tag-XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM
RRUI4RESULTFLAG_NOHANDLE

4.5.14 OpenSpecificTag

Command block structure:

I1 I2 EW [E1 ... En]

I1: Command ID low byte
I2: Command ID high byte
EW: Number of EPC words (between 0 and 31)
E1: EPC LSB
...
En: EPC MSB

Response block structure:

I1 I2 RF [EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] [H1 H2]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag-PC low byte
P2: Tag-PC high byte
X1L: Tag-XPC_W1 low byte
X1H: Tag-XPC_W1 high byte
X2L: Tag-XPC_W2 low byte
X2H: Tag-XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
E12: EPC MSB
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_NOTAG
RRUI4RESULTFLAG_WRONGCFM
RRUI4RESULTFLAG_NOHANDLE

4.5.15 TxGen2CmdSelect

Command block structure:

I1 I2 TA AC MB P1 P2 P3 P4 ML [MD ... MD]

I1: Command ID low byte
I2: Command ID high byte
TA: Target (see EPCglobal specification V1.2.0 page 56)
AC: Action (see EPCglobal specification V1.2.0 page 56)
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Bit pointer LSB
P2: Bit pointer 2nd byte
P3: Bit pointer 3rd byte
P4: Bit pointer MSB
ML: Masked length in bits
MD: Masked data

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.16 TxGen2CmdACK

Command block structure:

I1 I2 H1 H2

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte

Response block structure:

I1 I2 RF [P1 P2 [X1L X1H [X2L X2H]] [E1 ... En]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
P1: Tag PC low byte
P2: Tag PC high byte
X1L: Tag-XPC_W1 low byte
X1H: Tag-XPC_W1 high byte
X2L: Tag-XPC_W2 low byte
X2H: Tag-XPC_W2 high byte
E1: EPC LSB
...
En: EPC MSB

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.17 TxGen2CmdNAK

Command block structure:

I1 I2

I1: Command ID low byte
I2: Command ID high byte

Response block structure:

I1 I2 RF

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.18 TxGen2CmdReqRN

Command block structure:

I1 I2 H1 H2

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte

Response block structure:

I1 I2 RF [R1 R2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
R1: Tag random number low byte
R2: Tag random number high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.19 TxGen2CmdAccess

Command block structure:

I1 I2 H1 H2 W1 W2

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte
W1: Half encrypted access password low byte
W2: Half encrypted access password high byte

Response block structure:

I1 I2 RF [H1 H2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.20 TxGen2CmdRead

Command block structure:

I1 I2 H1 H2 MB P1 P2 P3 P4 WC

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to read

Response block structure:

I1 I2 RF [EC [D1 ... Dx] H1 H2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EC: tag error code
D1: Response data LSB
...
Dx: Response data MSB
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.21 TxGen2CmdWrite

Command block structure:

I1 I2 H1 H2 MB P1 P2 P3 P4 DL DH

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
DL: Encrypted word to write low byte
DH: Encrypted word to write high byte

Response block structure:

I1 I2 RF [EC H1 H2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EC: tag error code
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.22 TxGen2CmdKill

Command block structure:

I1 I2 H1 H2 W1 W2 RC SC

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte
W1: Half encrypted kill password low byte
W2: Half encrypted kill password high byte
RC: Recommissioning bits (see EPCglobal specification V1.2.0 page 68)
SC: “Second kill” flag (the response from the tag varies between the first and second kill)

Response block structure:

I1 I2 RF [EC H1 H2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EC: tag error code
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.23 TxGen2CmdLock

Command block structure:

I1 I2 H1 H2 L1 L2 L3

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte
L1: Payload LSB
L2: Payload 2nd byte
L3: Payload MSB

Response block structure:

I1 I2 RF [EC H1 H2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EC: tag error code
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.24 TxGen2CmdBlockWrite

Command block structure:

I1 I2 H1 H2 MB P1 P2 P3 P4 WC DL DH [DL DH ... DL DH]

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte
MB: Memory bank (see EPCglobal specification V1.2.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to write
DL: Word to write low byte
DH: Word to write high byte

Response block structure:

I1 I2 RF [EC H1 H2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EC: tag error code
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.25 TxGen2CmdBlockErase

Command block structure:

I1 I2 H1 H2 MB P1 P2 P3 P4 WC

I1: Command ID low byte
I2: Command ID high byte
H1: Handle low byte
H2: Handle high byte
MB: Memory bank (see EPCglobal specification V1.1.0 page 37)
P1: Word pointer LSB
P2: Word pointer 2nd byte
P3: Word pointer 3rd byte
P4: Word pointer MSB
WC: Number of words to erase

Response block structure:

I1 I2 RF [EC H1 H2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EC: tag error code
H1: Handle low byte
H2: Handle high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_CRCERROR
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.26 TxGen2CmdQuery

Command block structure:

I1 I2 DR MC TE SL SE TA QV

I1: Command ID low byte
I2: Command ID high byte
DR: Divide ratio (0: DR = 8; otherwise DR = 64/3)
MC: Miller coefficient (in the range from 0 to 3; see EPCglobal specification V1.2.0 page 57)
TE: TRext (0: No audio tone; see EPCglobal specification V1.2.0 page 57)
SL: Sel (in the range from 0 to 3; see EPCglobal specification V1.2.0 page 57)
SE: Session (in the range from 0 to 3; see EPCglobal specification V1.2.0 page 57)
TA: Target (0: A; otherwise B; see EPCglobal specification V1.2.0 page 57)
QV: Q value (see EPCglobal specification V1.2.0 page 57)

Response block structure:

I1 I2 RF [R1 R2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
R1: Tag random number low byte
R2: Tag random number high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.27 TxGen2CmdQueryAdjust

Command block structure:

I1 I2 SE AV

I1: Command ID low byte
I2: Command ID high byte
SE: Session (in the range from 0 to 3; see EPCglobal specification V1.2.0 page 58)
AV: Adjust value (0, 3 or 6; see EPCglobal specification V1.2.0 page 58)

Response block structure:

I1 I2 RF [R1 R2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
R1: Tag random number low byte
R2: Tag random number high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.28 TxGen2CmdQueryRep

Command block structure:

I1 I2 SE

I1: Command ID low byte
I2: Command ID high byte
SE: Session (in the range from 0 to 3; see EPCglobal specification V1.2.0 page 59)

Response block structure:

I1 I2 RF [R1 R2]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
R1: Tag random number low byte
R2: Tag random number high byte

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_OUTOFRANGE
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4.5.29 TxGen2CmdCustomCmd

Command block structure:

I1 I2 BC CD1 [... CDn] RC FR

I1: Command ID low byte
I2: Command ID high byte
BC: Number of command bits
CD: Command data (sent from LSB to MSB)
RC: Number of response bits expected
FR: Flag register

Transmit sequence: The first bit to be transmitted is in the MSB for the first byte (CD1) of the command data.

Structure of flag register:

FR D0: Tx preamble flag (0: FrameSync is sent; 1: preamble is sent)
FR D1: Rx timeout flag (0: Normal timeout; 1: Extended timeout [for write commands])
FR D2 – D7: -- not used –

Response block structure:

I1 I2 RF [RD1 [... RDn]]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
RD: response data

Receive sequence: The first bit received is in the MSB for the first byte (RD1) of the response data.

Possible result flags:

RRUI4RESULTFLAG_NOERROR
RRUI4RESULTFLAG_NODATA
RRUI4RESULTFLAG_NOCARRIER
RRUI4RESULTFLAG_WRONGCFM

4. Command and response transmission

4.6 Asynchronous responses

4.6.1 Message transmission

Response block structure:

I1 I2 T1 T2 T3 T4 MT MN CG

I1: Response ID low byte
I2: Response ID high byte
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
MT: Message type (reserved: always 0)
MN: Message number
CG: Coming / going flag

4.6.2 Transmission of tag data

The transmission of the tag data (coming message) as a consequence of an asynchronous command is described in the individual asynchronous commands. In the case of going messages, along with the extended result flag only the EPC and a tag counter value are transmitted. This tag counter value records how often a tag has been detected since its appearance in the field up until the going message. The tag counter value cannot exceed the value 65535.

Block structure of the asynchronous response for a tag going message:

I1 I2 CG EF [AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] EW [E1 ... En] C1 C2

I1: Response ID low byte
I2: Response ID high byte
CG: Coming / going flag (going message: 0x00)
EF: Extended result flag
AP: Antenna port
RS: RSSI value
T1: Time stamp LSB
T2: Time stamp 2nd byte
T3: Time stamp 3rd byte
T4: Time stamp MSB
P1: Tag-PC low byte
P2: Tag-PC high byte
X1L: Tag-XPC_W1 low byte
X1H: Tag-XPC_W1 high byte
X2L: Tag-XPC_W2 low byte
X2H: Tag-XPC_W2 high byte
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
C1: Number of times the tag has been read low byte
C2: Number of times the tag has been read high byte

706004 / 00 10 / 2012

ifm electronic gmbh - Friedrichstraße 1 - 45128 Essen - Germany
Telefon: 0800 16 16 16 4 - Internet: www.ifm.com