



Software manual
Supplement

Communication protocol v3.00

DTE8xx
DTE9xx

UK

Content

1	Preliminary note	6
1.1	Symbols used	6
1.2	Safety instructions	6
1.3	Further documents	6
2	Transmission methods	7
2.1	Ethernet transmission 2. Reader generation	7
2.1.1	Frame set-up	7
2.1.2	Port	7
2.1.3	Example	7
2.1.4	Extended block structure	7
2.2	Ethernet transmission 3. Reader generation	8
2.2.1	Frame set-up	8
2.2.2	Port	8
2.2.3	Example	8
2.2.4	Extended block structure	8
3	Command and response transmission	9
3.1	Data types	9
3.1.1	Command / response IDs	9
3.1.2	Result flag	19
3.1.3	Reader function mode	20
3.1.4	Communication standards	20
3.1.5	Antenna modes	20
3.1.6	ETSI channel changeover modes	21
3.1.7	Modulation types	21
3.1.8	On-/Off flag	21
3.1.9	Coming/going flag	21
3.1.10	Start/Stop/Heartbeat flag	21
3.1.11	LED bar control flag	22
3.1.12	Message numbers	22
3.1.13	Extended result flag	23
3.1.14	Tag error codes	23
3.1.15	Tag memory bank	23
3.1.16	LED mode	24
3.1.17	I/O card types	24
3.1.18	I/O communication protocols	24
3.2	System and configuration commands	25
3.2.1	GetMode	25
3.2.2	SetMode	25
3.2.3	GetCommStandard	26
3.2.4	SetCommStandard	26
3.2.5	GetPortPower	27
3.2.6	SetPortPower	27
3.2.7	GetCarrierFollowUpTime	28
3.2.8	SetCarrierFollowUpTime	28
3.2.9	GetAntennaMode	29
3.2.10	SetAntennaMode	29
3.2.11	GetPortMultiplexSequenceAndExposureTime	30
3.2.12	SetPortMultiplexSequenceAndExposureTime	30
3.2.13	GetCableLossAndAntennaGain	31
3.2.14	SetCableLossAndAntennaGain	31
3.2.15	GetETSIPortChannelList	32
3.2.16	SetETSIPortChannelList	32
3.2.17	GetETSIPortChannelSwitchingMode	33
3.2.18	SetETSIPortChannelSwitchingMode	33
3.2.19	GetProfileList	34
3.2.20	GetProfile	34
3.2.21	SetProfile	34
3.2.22	GetModulationType	35
3.2.23	SetModulationType	35

3.2.24	GetExtResultFlag	36
3.2.25	SetExtResultFlag	36
3.2.26	GetErrorStatus	37
3.2.27	GetDefaultParamset	37
3.2.28	SetDefaultParamset	37
3.2.29	GetActiveParamset	38
3.2.30	SetActiveParamset	38
3.2.31	SaveActiveParamset	38
3.2.32	GetParameterByConfigId	39
3.2.33	SetParameterByConfigId	40
3.2.34	GetIOCardHwConfig	40
3.2.35	SetIOCardHwConfig	41
3.2.36	GetIOCardProtocolConfig	41
3.2.37	SetIOCardProtocolConfig	42
3.2.38	GetIOCardExtendedInfo	42
3.2.39	GetMillisecondsTimestamp	43
3.2.40	GetTime	43
3.2.41	SetTime	44
3.2.42	GetTemperature	44
3.2.43	GetSWVersion	45
3.2.44	GetHWVersion	45
3.2.45	GetLicenseKey	46
3.2.46	SetLicenseKey	46
3.2.47	GetMaxPowerLimits	47
3.2.48	SetConfigPassword	47
3.2.49	EnterConfigPassword	48
3.2.50	LoadFactoryDefaults	48
3.2.51	FirmwareUpdatePrepare	49
3.2.52	FirmwareUpdateTransmitData	49
3.2.53	FirmwareUpdateFlash	50
3.2.54	RebootSystem	50
3.2.55	GetLEDMode	51
3.2.56	SetLEDMode	51
3.2.57	ActivateBuzzer	51
3.2.58	GetDeviceSerialNumber	52
3.2.59	GetDeviceTypeNumber	52
3.3	Commands in "Normal Mode" and in "Direct Mode"	53
3.3.1	GetSelSessionAndTarget	53
3.3.2	SetSelSessionAndTarget	53
3.3.3	GetInitialQValue	54
3.3.4	SetInitialQValue	54
3.3.5	GetMaxAirCommErrors	54
3.3.6	SetMaxAirCommErrors	55
3.3.7	GetASyncObservedListParameters	55
3.3.8	SetASyncObservedListParameters	55
3.3.9	GetSelectFilterOnOff	56
3.3.10	SetSelectFilterOnOff	56
3.3.11	GetSelectFilterData	57
3.3.12	SetSelectFilterData	57
3.3.13	GPIOGetIOData	58
3.3.14	GPIOSetOutput	59
3.3.15	GPIOAddActionToActionlist	60
3.3.16	GPIOCLEARActionlist	60
3.3.17	GPIOAssignInputToActionlist	61
3.3.18	SetLED	61
3.4	Commands in "Normal Mode"	62
3.4.1	SyncGetEPCs	62
3.4.2	SyncBulkGetEPCs	63
3.4.3	SyncReadDataAny	64
3.4.4	SyncReadDataSpecific	65
3.4.5	SyncReadDataUntilEndOfBankAny	66
3.4.6	SyncReadDataUntilEndOfBankSpecific	67

3.4.7	SyncWriteDataAny	68
3.4.8	SyncWriteDataSpecific	69
3.4.9	SyncWriteMaskedDataAny	70
3.4.10	SyncWriteMaskedDataSpecific	71
3.4.11	SyncBlockEraseAny	72
3.4.12	SyncBlockEraseSpecific	73
3.4.13	SyncLockAny	74
3.4.14	SyncLockSpecific	75
3.4.15	SyncKillAny	76
3.4.16	SyncKillSpecific	77
3.4.17	SyncBlockPermalockAny	78
3.4.18	SyncBlockPermalockSpecific	79
3.4.19	SyncAuthenticateAny	80
3.4.20	SyncAuthenticateSpecific	81
3.4.21	SyncUntraceableAny	83
3.4.22	SyncUntraceableSpecific	84
3.4.23	SyncWriteEPCSpecific	85
3.4.24	SyncWriteEPCToSingleTag	86
3.4.25	SyncNXPG2XReadProtectAny	87
3.4.26	SyncNXPG2XReadProtectSpecific	88
3.4.27	SyncNXPG2XResetReadProtectAny	89
3.4.28	SyncNXPG2XResetReadProtectSpecific	89
3.4.29	SyncNXPG2XChangeEASAny	90
3.4.30	SyncNXPG2XChangeEASSpecific	91
3.4.31	SyncNXPG2XEASAlarm	92
3.4.32	SyncRSSILedBarSpecific	93
3.4.33	ASyncGetRawEPCs	94
3.4.34	ASyncGetEPCs	95
3.4.35	ASyncReadDataAny	95
3.4.36	ASyncReadDataUntilEndOfBankAny	96
3.4.37	ASyncWriteDataAny	98
3.4.38	ASyncWriteMaskedDataAny	99
3.4.39	ASyncBlockEraseAny	100
3.4.40	ASyncLockAny	101
3.4.41	ASyncKillAny	102
3.4.42	ASyncBlockPermalockAny	103
3.4.43	ASyncAuthenticateAny	104
3.4.44	ASyncUntraceableAny	105
3.4.45	ASyncNXPG2XReadProtectAny	106
3.4.46	ASyncNXPG2XResetReadProtectAny	107
3.4.47	ASyncNXPG2XChangeEASAny	108
3.4.48	ASyncNXPG2XEASAlarm	109
3.4.49	ASyncStopCommand	109
3.4.50	ASyncGetTagRate	110
3.5	Commands in "Direct Mode"	110
3.5.1	GetAntennaList	110
3.5.2	SetAntenna	111
3.5.3	SetFrequency	111
3.5.4	GetNoiseValue	112
3.5.5	GetETSILBTRreferenceNoiseValue	112
3.5.6	CarrierOnOff	113
3.5.7	RandomModulationOnOff	113
3.5.8	EN302208TestmodulationOnOFF	114
3.5.9	MeasureTxPwr	114
3.5.10	MeasureRxPwr	115
3.5.11	MeasureVSWR	115
3.5.12	StartInventory	115

3.5.13	OpenNoninventoriedTag	116
3.5.14	OpenSpecificTag	117
3.5.15	DirectBulkGetEPCs	118
3.5.16	TxGen2CmdSelect	119
3.5.17	TxGen2CmdACK	120
3.5.18	TxGen2CmdNAK	121
3.5.19	TxGen2CmdReqRN	121
3.5.20	TxGen2CmdAccess	122
3.5.21	TxGen2CmdRead	123
3.5.22	TxGen2CmdWrite	124
3.5.23	TxGen2CmdKill	125
3.5.24	TxGen2CmdLock	126
3.5.25	TxGen2CmdBlockWrite	127
3.5.26	TxGen2CmdBlockErase	128
3.5.27	TxGen2CmdQuery	129
3.5.28	TxGen2CmdQueryAdjust	129
3.5.29	TxGen2CmdQueryRep	130
3.5.30	TxGen2CmdCustomCmd	131
3.6	Extended result data	132
3.7	Asynchronous responses	133
3.7.1	Message transmission	133
3.7.2	Transmission of tag data	133
4	Change history	134

Copyright

Microsoft®, Windows®, Windows 7®, Windows 8®, Windows 8.1® and Windows 10® are registered trademarks of Microsoft Corporation.

Adobe® and Acrobat® are registered trademarks of Adobe Systems Inc.

All trademarks and company names used are subject to the copyright of the respective companies.

1 Preliminary note

With the communication protocol, the UHF RFID readers DTE8xx / DTE9xx are connected to a higher level. System, data and diagnostic commands as well as their responses can be transmitted.

1.1 Symbols used

- ▶ Instruction
- > Reaction, result
- [...] Designation of keys and buttons
- "..." Name of display text
- Cross-reference
-  Important note
Non-compliance may result in malfunction or interference.
-  Information
Supplementary note

1.2 Safety instructions

Please read the operating instructions prior to set-up of the device. Ensure that the product is suitable for your application without any restrictions.

If the operating instructions or the technical data are not adhered to, personal injury and/or damage to property can occur.

1.3 Further documents

- Operating instructions

 The documents can be downloaded at:
www.ifm.com

2 Transmission methods

2.1 Ethernet transmission 2. Reader generation

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.

2.1.1 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).

2.1.2 Port

The TCP communication port is the port 4007.

2.1.3 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

2.1.4 Extended block structure

If a data block to be transferred exceeds 16 kB, the data block must be divided into 16 kB blocks. The subdivided blocks are assigned a block number. The 1st block contains the total data length of the blocks. The receipt of a block must be confirmed with an answer.

1. block:

Start + block number is always 0 + 4 byte total data length + data block + end

All other blocks:

Start + block number + data block + end

Answer to confirm receipt:

Start + block number + end

The start consists of 0xAA 0xBB 0x0E 0x01. The end consists of 0xAA 0xCC. If the byte 0xAA occurs in the KBRP frame, this must be doubled (0xAA -> 0xAA 0xAA).

2.2 Ethernet transmission 3. Reader generation

In the protocol of the 3rd reader generation 3 frame types are used. If no data is received within one frame for 100 ms, the received part of the frame is discarded. The payload (D1 to Dn) is identical to the protocol of the 2nd reader generation.

2.2.1 Frame set-up

Frame: 5A 50 L1 L2 L3 L4 D1 ... Dn A5

5A: Start code

50 : Code for frame

L1-L4: Number of the data bytes (D1-Dn) in the frame (32 bit; LSB first)

D1-D4: Data bytes to be transmitted (Payload)

A5: End code

Ping: 5A 5F A5

5A: Start code

5F: Ping code

A5: End code

Reply to Ping: 5A 5F A5

5A: Start code

5F: Ping reply code

A5: End code

2.2.2 Port

The TCP communication port is the port 4007.

2.2.3 Example

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

0x5A 0x50 0x02 0x00 0x00 0x00 0x11 0x01 0xA5

2.2.4 Extended block structure

If a data block to be transferred exceeds 16 kB, the data block must be divided into 16 kB blocks. The subdivided blocks are assigned a block number. The 1st block contains the total data length of the blocks. The receipt of a block must be confirmed with an answer.

1. block:

Start + block number is always 0 + 4 byte total data length + data block + end

All other blocks:

Start + block number + data block + end

Answer to confirm receipt:

Start + block number + end

The start consists of 0xAA 0xBB 0x0E 0x01. The end consists of 0xAA 0xCC. If the byte 0xAA occurs in the KBRP frame, this must be doubled (0xAA -> 0xAA 0xAA).

3 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

3.1 Data types

3.1.1 Command / response IDs

typedef enum

```
{
    CMD_GetMode = 0x0001,           // 0x0001
    CMD_SetMode,                     // 0x0002
    CMD_GetCommunicationStandard,    // 0x0003
    CMD_SetCommunicationStandard,    // 0x0004
    CMD_GetPortPower,                // 0x0005
    CMD_SetPortPower,                // 0x0006
    CMD_GetCarrierFollowUpTime,       // 0x0007 (Generation 2 only)
    CMD_SetCarrierFollowUpTime,       // 0x0008 (Generation 2 only)
    CMD_GetAntennaMode,              // 0x0009 (Generation 2 only)
    CMD_SetAntennaMode,              // 0x000A (Generation 2 only)
    CMD_GetPortMultiplexSequenceAndExposureTime, // 0x000B
    CMD_SetPortMultiplexSequenceAndExposureTime, // 0x000C
    CMD_GetCableLossAndAntennaGain,  // 0x000D
    CMD_SetCableLossAndAntennaGain,  // 0x000E
    CMD_GetETSIPortChannelList,      // 0x000F
    CMD_SetETSIPortChannelList,      // 0x0010
    CMD_GetETSIPortChannelSwitchingMode, // 0x0011 (Generation 2 only)
    CMD_SetETSIPortChannelSwitchingMode, // 0x0012 (Generation 2 only)
    CMD_GetProfileList,              // 0x0013
    CMD_GetProfile,                  // 0x0014
    CMD_SetProfile,                  // 0x0015
    CMD_GetModulationType,            // 0x0016 (Generation 2 only)
    CMD_SetModulationType,            // 0x0017 (Generation 2 only)
    CMD_GetExtendedResultFlag,       // 0x0018
    CMD_SetExtendedResultFlag,       // 0x0019
    CMD_GetErrorStatus,               // 0x001A
    CMD_GetDefaultParameterSet,       // 0x001B
    CMD_SetDefaultParameterSet,       // 0x001C
    CMD_GetActiveParameterSet,        // 0x001D
}
```

CMD_SetActiveParameterSet,	// 0x001E
CMD_SaveActiveParameterSet,	// 0x001F
CMD_GetParameterByConfigId,	// 0x0020
CMD_SetParameterByConfigId,	// 0x0021
CMD_GetIOCardHwConfig,	// 0x0022 (Generation 2 only)
CMD_SetIOCardHwConfig,	// 0x0023 (Generation 2 only)
CMD_GetIOCardProtocolConfig,	// 0x0024 (Generation 2 only)
CMD_SetIOCardProtocolConfig,	// 0x0025 (Generation 2 only)
CMD_GetTime,	// 0x0026
CMD_SetTime,	// 0x0027
CMD_GetTemperature,	// 0x0028
CMD_GetSwVersion,	// 0x0029
CMD_GetHwVersion,	// 0x002A
CMD_GetLicenseKey,	// 0x002B
CMD_SetLicenseKey,	// 0x002C
CMD_GetIOCardExtendedInfo,	// 0x002D (Generation 2 only)
CMD_GetAntennaMapping,	// 0x002E (Generation 2 only)
CMD_SetAntennaMapping,	// 0x002F (Generation 2 only)
CMD_GetASyncPowerRampOffset,	// 0x0030 (Generation 2 only)
CMD_GetMaxPowerLimits,	// 0x0031
CMD_SetConfigPassword = 0x003E,	// 0x003E
CMD_EnterConfigPassword,	// 0x003F
CMD_LoadFactoryDefaults = 0x0040,	// 0x0040
CMD_FirmwareUpdatePrepare,	// 0x0041
CMD_FirmwareUpdateTransmitData,	// 0x0042
CMD_FirmwareUpdateFlash,	// 0x0043
CMD_RebootSystem,	// 0x0044
CMD_ActivateBuzzer,	// 0x0045
CMD_GetDeviceSerialNumber,	// 0x0046
CMD_GetDeviceTypeNumber,	// 0x0047
CMD_GetLEDMode,	// 0x0048
CMD_SetLEDMode,	// 0x0049
CMD_GetUptimeInMilliseconds,	// 0x004A
CMD_GetSelSessionAndTarget = 0x0080,	// 0x0080
CMD_SetSelSessionAndTarget,	// 0x0081
CMD_GetInitialQ,	// 0x0082
CMD_SetInitialQ,	// 0x0083
CMD_GetMaxAirCommErrors,	// 0x0084
CMD_SetMaxAirCommErrors,	// 0x0085
CMD_GetASyncObservedListParameters,	// 0x0086
CMD_SetASyncObservedListParameters,	// 0x0087
CMD_GetSelectFilterOnOff,	// 0x0088

CMD_SetSelectFilterOnOff,	// 0x0089
CMD_GetSelectFilterData,	// 0x008A
CMD_SetSelectFilterData,	// 0x008B
CMD_GPIOGetIOData,	// 0x008C
CMD_GPIOSetOutput,	// 0x008D
CMD_GPIOAddActionToActionlist,	// 0x008E
CMD_GPIOClearActionlist,	// 0x008F
CMD_GPIOAssignInputToActionlist,	// 0x0090
CMD_AddTagToBlacklist,	// 0x0091 (Generation 2 only)
CMD_DeleteBlacklist,	// 0x0092 (Generation 2 only)
CMD_GetBlacklistEntries,	// 0x0093 (Generation 2 only)
CMD_GetBlacklistEntriesExtended,	// 0x0094 (Generation 2 only)
CMD_SetEPCMemoryBankFilter = 0x0096,	// 0x0096 (Generation 2 only)
CMD_GPIOGetActionlist,	// 0x0097
CMD_GPIOSetActionlist,	// 0x0098
CMD_SetLED = 0x00A0,	// 0x00A0 (Generation 3 only)
CMD_KRAIInitializePort = 0x00D0,	// 0x00D0
CMD_KRAIGetPortInfo,	// 0x00D1
CMD_KRAISetLED,	// 0x00D2
CMD_KRAIGetSwVersion,	// 0x00D3 (Generation 3 only)
CMD_SyncGetEPCs = 0x0101,	// 0x0101
CMD_SyncBulkGetEPCs,	// 0x0102
CMD_SyncReadDataAny,	// 0x0103
CMD_SyncReadDataSpecific,	// 0x0104
CMD_SyncWriteDataAny,	// 0x0105
CMD_SyncWriteDataSpecific,	// 0x0106
CMD_SyncWriteMaskedDataAny,	// 0x0107
CMD_SyncWriteMaskedDataSpecific,	// 0x0108
CMD_SyncBlockEraseAny,	// 0x0109
CMD_SyncBlockEraseSpecific,	// 0x010A
CMD_SyncLockAny,	// 0x010B
CMD_SyncLockSpecific,	// 0x010C
CMD_SyncKillAny,	// 0x010D
CMD_SyncKillSpecific,	// 0x010E
CMD_SyncWriteEPCToSingleTag,	// 0x010F
CMD_ASyncGetRawEPCs,	// 0x0110
CMD_ASyncGetEPCs,	// 0x0111
CMD_ASyncReadDataAny,	// 0x0112
CMD_ASyncWriteDataAny,	// 0x0113
CMD_ASyncWriteMaskedDataAny,	// 0x0114
CMD_ASyncBlockEraseAny,	// 0x0115
CMD_ASyncLockAny,	// 0x0116

CMD_ASyncKillAny,	// 0x0117
CMD_ASyncStopCommand,	// 0x0118
CMD_ASyncGetTagRate,	// 0x0119
CMD_SyncWriteEPCSpecific,	// 0x011A
CMD_SyncReadDataUntilEndOfBankAny,	// 0x011B
CMD_SyncReadDataUntilEndOfBankSpecific,	// 0x011C
CMD_ASyncReadDataUntilEndOfBankAny,	// 0x011D
CMD_SyncBlockPermalockAny = 0x0120,	// 0x0120
CMD_SyncBlockPermalockSpecific,	// 0x0121
CMD_ASyncBlockPermalockAny,	// 0x0122
CMD_SyncAuthenticateAny,	// 0x0123
CMD_SyncAuthenticateSpecific,	// 0x0124
CMD_ASyncAuthenticateAny,	// 0x0125
CMD_SyncUntraceableAny,	// 0x0126
CMD_SyncUntraceableSpecific,	// 0x0127
CMD_ASyncUntraceableAny,	// 0x0128
CMD_SyncSJ5511ActivateSecureModeAny = 0x0170,	// 0x0170 (Generation 3 only)
CMD_SyncSJ5511ActivateSecureModeSpecific,	// 0x0171 (Generation 3 only)
CMD_ASyncSJ5511ActivateSecureModeAny,	// 0x0172 (Generation 3 only)
CMD_SyncSJ5511AuthenticateOBUAny,	// 0x0173 (Generation 3 only)
CMD_SyncSJ5511AuthenticateOBUSpecific,	// 0x0174 (Generation 3 only)
CMD_ASyncSJ5511AuthenticateOBUAny,	// 0x0175 (Generation 3 only)
CMD_SyncNXPG2XReadProtectAny = 0x0180,	// 0x0180 (Generation 2 only)
CMD_SyncNXPG2XReadProtectSpecific,	// 0x0181 (Generation 2 only)
CMD_SyncNXPG2XResetReadProtectAny,	// 0x0182 (Generation 2 only)
CMD_SyncNXPG2XResetReadProtectSpecific,	// 0x0183 (Generation 2 only)
CMD_SyncNXPG2XChangeEASAny,	// 0x0184 (Generation 2 only)
CMD_SyncNXPG2XChangeEASSpecific,	// 0x0185 (Generation 2 only)
CMD_SyncNXPG2XEASAlarm,	// 0x0186 (Generation 2 only)
CMD_ASyncNXPG2XReadProtectAny,	// 0x0187 (Generation 2 only)
CMD_ASyncNXPG2XResetReadProtectAny,	// 0x0188 (Generation 2 only)
CMD_ASyncNXPG2XChangeEASAny,	// 0x0189 (Generation 2 only)
CMD_ASyncNXPG2XEASAlarm,	// 0x018A (Generation 2 only)
CMD_SyncRSSILedBarSpecific = 0x01A0,	// 0x01A0 (Generation 3 only)
CMD_GetAntennaList = 0x0201,	// 0x0201
CMD_SetAntenna,	// 0x0202
CMD_SetFrequency,	// 0x0203
CMD_GetNoiseValue,	// 0x0204
CMD_GetETSILBTReferenceNoiseValue,	// 0x0205 (Generation 2 only)
CMD_CarrierOnOff,	// 0x0206
CMD_RandomModulationOnOff,	// 0x0207 (Generation 2 only)
CMD_MeasureTxPower,	// 0x0208

```

CMD_MeasureRxPower,                // 0x0209
CMD_MeasureVSWR,                   // 0x020A
CMD_StartInventory,                 // 0x020B
CMD_OpenNoninventoriedTag,         // 0x020C
CMD_OpenSpecificTag,               // 0x020D
CMD_TxGen2CommandSelect,           // 0x020E
CMD_TxGen2CommandACK,              // 0x020F
CMD_TxGen2CommandNAK,              // 0x0210
CMD_TxGen2CommandReqRN,            // 0x0211
CMD_TxGen2CommandAccess,           // 0x0212
CMD_TxGen2CommandRead,             // 0x0213
CMD_TxGen2CommandWrite,            // 0x0214
CMD_TxGen2CommandKill,             // 0x0215
CMD_TxGen2CommandLock,             // 0x0216
CMD_TxGen2CommandBlockWrite,       // 0x0217
CMD_TxGen2CommandBlockErase,       // 0x0218
CMD_TxGen2CommandQuery,            // 0x0219
CMD_TxGen2CommandQueryAdjust,      // 0x021A
CMD_TxGen2CommandQueryRep,         // 0x021B
CMD_TxGen2CommandCustomCommand,    // 0x021C
CMD_EN302208TestmodulationOnOff,   // 0x021D (Generation 2 only)
CMD_DirectBulkGetEPCs,             // 0x021E
CMD_SelectFrequencyAndSwitchCarrierOn, // 0x021F
CMD_ExecuteAccessSequence,         // 0x0220
CMD_ExecuteKillSequence,           // 0x0221
CMD_ExecuteWriteSequence,          // 0x0222

// HSM
CMD_HSM_Encrypt = 0x1000,           // 0x1000 (Generation 3 only)
CMD_HSM_Decrypt,                   // 0x1001 (Generation 3 only)
CMD_HSM_GetRandomData,             // 0x1002 (Generation 3 only)
CMD_HSM_GetChallenge,              // 0x1003 (Generation 3 only)
CMD_HSM_Authenticate,              // 0x1004 (Generation 3 only)
CMD_HSM_SetPublicCertificate,       // 0x1005 (Generation 3 only)
CMD_HSM_SetWrappingKey,            // 0x1006 (Generation 3 only)
CMD_HSM_SetAESKey,                 // 0x1007 (Generation 3 only)

// Command for protocol tunnel interprocess communication
CMD_KBRPTInterProcComm = 0x3FFF,   // 0x3FFF

// Command responses
RES_GetMode = 0x8001,               // 0x8001

```

RES_SetMode,	// 0x8002
RES_GetCommunicationStandard,	// 0x8003
RES_SetCommunicationStandard,	// 0x8004
RES_GetPortPower,	// 0x8005
RES_SetPortPower,	// 0x8006
RES_GetCarrierFollowUpTime,	// 0x8007 (Generation 2 only)
RES_SetCarrierFollowUpTime,	// 0x8008 (Generation 2 only)
RES_GetAntennaMode,	// 0x8009 (Generation 2 only)
RES_SetAntennaMode,	// 0x800A (Generation 2 only)
RES_GetPortMultiplexSequenceAndExposureTime,	// 0x800B
RES_SetPortMultiplexSequenceAndExposureTime,	// 0x800C
RES_GetCableLossAndAntennaGain,	// 0x800D
RES_SetCableLossAndAntennaGain,	// 0x800E
RES_GetETSIPortChannelList,	// 0x800F
RES_SetETSIPortChannelList,	// 0x8010
RES_GetETSIPortChannelSwitchingMode,	// 0x8011 (Generation 2 only)
RES_SetETSIPortChannelSwitchingMode,	// 0x8012 (Generation 2 only)
RES_GetProfileList,	// 0x8013
RES_GetProfile,	// 0x8014
RES_SetProfile,	// 0x8015
RES_GetModulationType,	// 0x8016 (Generation 2 only)
RES_SetModulationType,	// 0x8017 (Generation 2 only)
RES_GetExtendedResultFlag,	// 0x8018
RES_SetExtendedResultFlag,	// 0x8019
RES_GetErrorStatus,	// 0x801A
RES_GetDefaultParameterSet,	// 0x801B
RES_SetDefaultParameterSet,	// 0x801C
RES_GetActiveParameterSet,	// 0x801D
RES_SetActiveParameterSet,	// 0x801E
RES_SaveActiveParameterSet,	// 0x801F
RES_GetParameterByConfigId,	// 0x8020
RES_SetParameterByConfigId,	// 0x8021
RES_GetIOCardHwConfig,	// 0x8022 (Generation 2 only)
RES_SetIOCardHwConfig,	// 0x8023 (Generation 2 only)
RES_GetIOCardProtocolConfig,	// 0x8024 (Generation 2 only)
RES_SetIOCardProtocolConfig,	// 0x8025 (Generation 2 only)
RES_GetTime,	// 0x8026
RES_SetTime,	// 0x8027
RES_GetTemperature,	// 0x8028
RES_GetSwVersion,	// 0x8029
RES_GetHwVersion,	// 0x802A
RES_GetLicenseKey,	// 0x802B

RES_SetLicenseKey,	// 0x802C
RES_GetIOCardExtendedInfo,	// 0x802D (Generation 2 only)
RES_GetAntennaMapping,	// 0x802E (Generation 2 only)
RES_SetAntennaMapping,	// 0x802F (Generation 2 only)
RES_GetASyncPowerRampOffset,	// 0x8030 (Generation 2 only)
RES_GetMaxPowerLimits,	// 0x8031
RES_SetConfigPassword = 0x803E,	// 0x803E
RES_EnterConfigPassword,	// 0x803F
RES_LoadFactoryDefaults = 0x8040,	// 0x8040
RES_FirmwareUpdatePrepare,	// 0x8041
RES_FirmwareUpdateTransmitData,	// 0x8042
RES_FirmwareUpdateFlash,	// 0x8043
RES_RebootSystem,	// 0x8044
RES_ActivateBuzzer,	// 0x8045
RES_GetDeviceSerialNumber,	// 0x8046
RES_GetDeviceTypeNumber,	// 0x8047
RES_GetLEDMode,	// 0x8048
RES_SetLEDMode,	// 0x8049
RES_GetUptimeInMilliseconds,	// 0x804A
RES_GetSelSessionAndTarget = 0x8080,	// 0x8080
RES_SetSelSessionAndTarget,	// 0x8081
RES_GetInitialQ,	// 0x8082
RES_SetInitialQ,	// 0x8083
RES_GetMaxAirCommErrors,	// 0x8084
RES_SetMaxAirCommErrors,	// 0x8085
RES_GetASyncObservedListParameters,	// 0x8086
RES_SetASyncObservedListParameters,	// 0x8087
RES_GetSelectFilterOnOff,	// 0x8088
RES_SetSelectFilterOnOff,	// 0x8089
RES_GetSelectFilterData,	// 0x808A
RES_SetSelectFilterData,	// 0x808B
RES_GPIOGetIOData,	// 0x808C
RES_GPIOSetOutput,	// 0x808D
RES_GPIOAddActionToActionlist,	// 0x808E
RES_GPIOClearActionlist,	// 0x808F
RES_GPIOAssignInputToActionlist,	// 0x8090
RES_AddTagToBlacklist,	// 0x8091 (Generation 2 only)
RES_DeleteBlacklist,	// 0x8092 (Generation 2 only)
RES_GetBlacklistEntries,	// 0x8093 (Generation 2 only)
RES_GetBlacklistEntriesExtended,	// 0x8094 (Generation 2 only)
RES_SetEPCMemoryBankFilter = 0x8096,	// 0x8096 (Generation 2 only)
RES_GPIOGetActionlist,	// 0x8097

RES_GPIOSetActionlist,	// 0x8098
RES_SetLED = 0x80A0,	// 0x80A0 (Generation 3 only)
RES_KRAIInitializePort = 0x80D0,	// 0x80D0
RES_KRAIGetPortInfo,	// 0x80D1
RES_KRAISetLED,	// 0x80D2
RES_KRAIGetSwVersion,	// 0x80D3 (Generation 3 only)
RES_SyncGetEPCs = 0x8101,	// 0x8101
RES_SyncBulkGetEPCs,	// 0x8102
RES_SyncReadDataAny,	// 0x8103
RES_SyncReadDataSpecific,	// 0x8104
RES_SyncWriteDataAny,	// 0x8105
RES_SyncWriteDataSpecific,	// 0x8106
RES_SyncWriteMaskedDataAny,	// 0x8107
RES_SyncWriteMaskedDataSpecific,	// 0x8108
RES_SyncBlockEraseAny,	// 0x8109
RES_SyncBlockEraseSpecific,	// 0x810A
RES_SyncLockAny,	// 0x810B
RES_SyncLockSpecific,	// 0x810C
RES_SyncKillAny,	// 0x810D
RES_SyncKillSpecific,	// 0x810E
RES_SyncWriteEPCToSingleTag,	// 0x810F
RES_ASyncGetRawEPCs,	// 0x8110
RES_ASyncGetEPCs,	// 0x8111
RES_ASyncReadDataAny,	// 0x8112
RES_ASyncWriteDataAny,	// 0x8113
RES_ASyncWriteMaskedDataAny,	// 0x8114
RES_ASyncBlockEraseAny,	// 0x8115
RES_ASyncLockAny,	// 0x8116
RES_ASyncKillAny,	// 0x8117
RES_ASyncStopCommand,	// 0x8118
RES_ASyncGetTagRate,	// 0x8119
RES_SyncWriteEPCSpecific,	// 0x811A
RES_SyncReadDataUntilEndOfBankAny,	// 0x811B
RES_SyncReadDataUntilEndOfBankSpecific,	// 0x811C
RES_ASyncReadDataUntilEndOfBankAny,	// 0x811D
RES_SyncBlockPermalockAny = 0x8120,	// 0x8120
RES_SyncBlockPermalockSpecific,	// 0x8121
RES_ASyncBlockPermalockAny,	// 0x8122
RES_SyncAuthenticateAny,	// 0x8123
RES_SyncAuthenticateSpecific,	// 0x8124
RES_ASyncAuthenticateAny,	// 0x8125
RES_SyncUntraceableAny,	// 0x8126

RES_SyncUntraceableSpecific,	// 0x8127
RES_ASyncUntraceableAny,	// 0x8128
RES_SyncSJ5511ActivateSecureModeAny = 0x8170,	// 0x8170 (Generation 3 only)
RES_SyncSJ5511ActivateSecureModeSpecific,	// 0x8171 (Generation 3 only)
RES_ASyncSJ5511ActivateSecureModeAny,	// 0x8172 (Generation 3 only)
RES_SyncSJ5511AuthenticateOBUAny,	// 0x8173 (Generation 3 only)
RES_SyncSJ5511AuthenticateOBUSpecific,	// 0x8174 (Generation 3 only)
RES_ASyncSJ5511AuthenticateOBUAny,	// 0x8175 (Generation 3 only)
RES_SyncNXPG2XReadProtectAny = 0x8180,	// 0x8180 (Generation 2 only)
RES_SyncNXPG2XReadProtectSpecific,	// 0x8181 (Generation 2 only)
RES_SyncNXPG2XResetReadProtectAny,	// 0x8182 (Generation 2 only)
RES_SyncNXPG2XResetReadProtectSpecific,	// 0x8183 (Generation 2 only)
RES_SyncNXPG2XChangeEASAny,	// 0x8184 (Generation 2 only)
RES_SyncNXPG2XChangeEASSpecific,	// 0x8185 (Generation 2 only)
RES_SyncNXPG2XEASAlarm,	// 0x8186 (Generation 2 only)
RES_ASyncNXPG2XReadProtectAny,	// 0x8187 (Generation 2 only)
RES_ASyncNXPG2XResetReadProtectAny,	// 0x8188 (Generation 2 only)
RES_ASyncNXPG2XChangeEASAny,	// 0x8189 (Generation 2 only)
RES_ASyncNXPG2XEASAlarm,	// 0x818A (Generation 2 only)
RES_SyncRSSILedBarSpecific = 0x81A0,	// 0x81A0 (Generation 3 only)
RES_GetAntennaList = 0x8201,	// 0x8201
RES_SetAntenna,	// 0x8202
RES_SetFrequency,	// 0x8203
RES_GetNoiseValue,	// 0x8204
RES_GetETSILBTReferenceNoiseValue,	// 0x8205 (Generation 2 only)
RES_CarrierOnOff,	// 0x8206
RES_RandomModulationOnOff,	// 0x8207 (Generation 2 only)
RES_MeasureTxPower,	// 0x8208
RES_MeasureRxPower,	// 0x8209
RES_MeasureVSWR,	// 0x820A
RES_StartInventory,	// 0x820B
RES_OpenNoninventoriedTag,	// 0x820C
RES_OpenSpecificTag,	// 0x820D
RES_TxGen2CommandSelect,	// 0x820E
RES_TxGen2CommandACK,	// 0x820F
RES_TxGen2CommandNAK,	// 0x8210
RES_TxGen2CommandReqRN,	// 0x8211
RES_TxGen2CommandAccess,	// 0x8212
RES_TxGen2CommandRead,	// 0x8213
RES_TxGen2CommandWrite,	// 0x8214
RES_TxGen2CommandKill,	// 0x8215
RES_TxGen2CommandLock,	// 0x8216

RES_TxGen2CommandBlockWrite,	// 0x8217
RES_TxGen2CommandBlockErase,	// 0x8218
RES_TxGen2CommandQuery,	// 0x8219
RES_TxGen2CommandQueryAdjust,	// 0x821A
RES_TxGen2CommandQueryRep,	// 0x821B
RES_TxGen2CommandCustomCommand,	// 0x821C
RES_EN302208TestmodulationOnOff,	// 0x821D (Generation 2 only)
RES_DirectBulkGetEPCs,	// 0x821E
RES_SelectFrequencyAndSwitchCarrierOn,	// 0x821F
RES_ExecuteAccessSequence,	// 0x8220
RES_ExecuteKillSequence,	// 0x8221
RES_ExecuteWriteSequence,	// 0x8222
 // HSM	
RES_HSM_Encrypt = 0x9000,	// 0x9000 (Generation 3 only)
RES_HSM_Decrypt,	// 0x9001 (Generation 3 only)
RES_HSM_GetRandomData,	// 0x9002 (Generation 3 only)
RES_HSM_GetChallenge,	// 0x9003 (Generation 3 only)
RES_HSM_Authenticate,	// 0x9004 (Generation 3 only)
RES_HSM_SetPublicCertificate,	// 0x9005 (Generation 3 only)
RES_HSM_SetWrappingKey,	// 0x9006 (Generation 3 only)
RES_HSM_SetAESKey,	// 0x9007 (Generation 3 only)
 // Reply from protocol tunnel Interprocess communication	
RES_KBRPTInterProcComm = 0xBFFF,	// 0xBFFF
 // RRU4 asynchronous command responses	
REA_ASyncGetRawEPCs = 0xC110,	// 0xC110
REA_ASyncGetEPCs,	// 0xC111
REA_ASyncReadDataAny,	// 0xC112
REA_ASyncWriteDataAny,	// 0xC113
REA_ASyncWriteMaskedDataAny,	// 0xC114
REA_ASyncBlockEraseAny,	// 0xC115
REA_ASyncLockAny,	// 0xC116
REA_ASyncKillAny,	// 0xC117
REA_ASyncReadDataUntilEndOfBankAny = 0xC11D,	// 0xC11D
REA_ASyncBlockPermalockAny = 0xC122,	// 0xC122
REA_ASyncAuthenticateAny = 0xC125,	// 0xC125
REA_ASyncUntraceableAny = 0xC128,	// 0xC128
REA_ASyncSJ5511ActivateSecureModeAny = 0xC172,	// 0xC172 (Generation 3 only)
REA_ASyncSJ5511AuthenticateOBUAny = 0xC175,	// 0xC175 (Generation 3 only)
REA_ASyncNXPG2XReadProtectAny = 0xC187,	// 0xC187 (Generation 2 only)

```

    REA_ASyncNXPG2XResetReadProtectAny,           // 0xC188 (Generation 2 only)
    REA_ASyncNXPG2XChangeEASAny,                   // 0xC189 (Generation 2 only)
    REA_ASyncNXPG2XEASAlarm,                       // 0xC18A (Generation 2 only)

    // RRU4 asynchronous messages
    REA_ReaderMessage = 0xC301                      // 0xC301
} tHostCommCmds;

```

3.1.2 Result flag

```

typedef enum
{
    RF_NoError = 0,                                // 0x00
    RF_NoData,                                     // 0x01
    RF_CRCError,                                   // 0x02
    RF_NoLicense,                                  // 0x03
    RF_OutOfRange,                                 // 0x04
    RF_NoStandard,                                 // 0x05
    RF_NoAntenna,                                  // 0x06
    RF_NoFrequency,                               // 0x07
    RF_NoCarrier,                                  // 0x08
    RF_AntennaError,                              // 0x09
    RF_NoTag,                                      // 0x0A
    RF_MoreThanOneTagInField,                     // 0x0B
    RF_WrongLicenseKey,                           // 0x0C
    RF_FirmwareRejected,                          // 0x0D
    RF_WrongCommandFunctionMode,                  // 0x0E
    RF_NoHandle,                                   // 0x0F
    RF_NoProfile,                                  // 0x10
    RF_AccessDenied,                              // 0x11
    RF_TransmissionTimeExpired,                   // 0x12
    RF_DatabaseError,                             // 0x13
    RF_ParameterError,                            // 0x14
    RF_SignatureVerificationFailed,               // 0x15
    RF_OutOfMemory = 0x7F,                        // 0x7F
    RF_Nonspecific = 0x80                         // 0x80
} tResultFlag;

```

UK

3.1.3 Reader function mode

```
typedef enum
{
    CFM_NORMAL = 0,
    CFM_DIRECT,
    CFM_Last
} tCommandFunctionMode;
```

3.1.4 Communication standards

```
typedef enum
{
    CS_None = 0,
    CS_Autodetect,
    CS_ETSI_EN302208,
    CS_ETSI_EN302208_LBT,
    CS_FCC,
    CS_Special,
    CS_China,
    CS_Thailand,
    CS_Brazil,
    CS_SouthKorea,
    CS_Peru,
    CS_Singapore,
    CS_Australia,
    CS_Japan,
    CS_Malaysia,
    CS_Indonesia,
    CS_Last
} tCommStandard;
```

3.1.5 Antenna modes

```
typedef enum
{
    AM_All = 0,
    AM_Antenna1 = 0x10,
    AM_Antenna2,
    AM_Antenna3,
    AM_Antenna4,
    AM_Last
} tAntennaMode;
```

3.1.6 ETSI channel changeover modes

typedef enum

```
{
    EEPCSW_Mode0 = 0,
    EEPCSW_Mode1,
    EEPCSW_Last
} tETSIPortChannelSwitchingMode;
```

3.1.7 Modulation types

typedef enum

```
{
    MT_DSB = 0,           // double-sideband modulation
    MT_PRA                // PR-ASK Modulation
    MT_SSB,              // Single sideband modulation (currently not supported)
    MT_EXT,              // R1000 external double-sideband modulation (currently not supported)
    MT_Last
} tModulationTypes;
```

3.1.8 On-/Off flag

typedef enum

```
{
    OOF_Off = 0x00,
    OOF_On = 0xFF
} tOnOffFlag;
```

3.1.9 Coming/going flag

typedef enum

```
{
    CGF_Going = 0x00,
    CGF_DataChanging = 0x80,
    CGF_Coming = 0xFF
} tComingGoingFlag;
```

3.1.10 Start/Stop/Heartbeat flag

typedef enum

```
{
    SSHBF_Stop = 0x00,
    SSHBF_Heartbeat = 0x80,
    SSHBF_Start = 0xFF
} tStartStopHeartbeatFlag;
```

3.1.11 LED bar control flag

typedef enum

```
{
    RLBCF_Start = 0,
    RLBCF_Do,
    RLBCF_Stop,
    RLBCF_Last
} tRSSILedBarControlFlag;
```

3.1.12 Message numbers

typedef enum

```
{
    EMID_Error_NoFreeChannel = 0,
    EMID_Error_Antenna1,
    EMID_Error_Antenna2,
    EMID_Error_Antenna3,
    EMID_Error_Antenna4,
    EMID_Error_Antenna5,
    EMID_Error_Antenna6,
    EMID_Error_Antenna7,
    EMID_Error_Antenna8,
    EMID_Error_ETSIPowerCheck1,
    EMID_Error_ETSIPowerCheck2,
    EMID_Error_ETSIPowerCheck3,
    EMID_Error_ETSIPowerCheck4,
    EMID_Error_ETSIPowerCheck5,
    EMID_Error_ETSIPowerCheck6,
    EMID_Error_ETSIPowerCheck7,
    EMID_Error_ETSIPowerCheck8,
    EMID_Error_KRAIOverCurrent1,
    EMID_Error_KRAIOverCurrent,
    EMID_Error_KRAIOverCurrent3,
    EMID_Error_KRAIOverCurrent4,
    EMID_Error_KRAIOverCurrent5,
    EMID_Error_KRAIOverCurrent6,
    EMID_Error_KRAIOverCurrent7,
    EMID_Error_KRAIOverCurrent8,
    EMID_Error_Last
} tErrorMessageId;
```

3.1.13 Extended result flag

The extended result flag determines which information is transferred within the extended result data. If a bit is set, the corresponding information is transmitted.

```
// Macros for accessing the extended result flag
#define ERF_MASK_A 0x01           // Antenna
#define ERF_MASK_R 0x02           // RSSI value
#define ERF_MASK_T 0x04           // Timestamp
#define ERF_MASK_PC 0x08          // Protocol control words
#define ERF_MASK_FRTF 0x10        // Frequency and tag phase
#define ERF_MASK_RDBM 0x20        // RSSI value in dBm
#define ERF_MASK_ALL 0x3F
```

3.1.14 Tag error codes

```
typedef enum
{
    TEC_NoError = 0x00,
    TEC_NoAnswer,
    TEC_AccessDenied,
    TEC_VerifyFailed,
    TEC_Unspecified,
    TEC_Other = 0x80,
    TEC_NotSupported = 0x81,
    TEC_InsufficientPrivileges = 0x82,
    TEC_MemoryLocked = 0x84,
    TEC_CryptoSuiteError = 0x85,
    TEC_CommandNotEncapsulated = 0x86,
    TEC_ResponseBufferOverflow = 0x87,
    TEC_SecurityTimeout = 0x88,
    TEC_InsufficientPower = 0x8B,
    TEC_NonSpecific = 0x8F
} tTagErrorCode;
```

3.1.15 Tag memory bank

```
typedef enum
{
    TMB_RFU = 0,
    TMB_EPC,
    TMB_TID,
    TMB_User
} tTagMemoryBank;
```

3.1.16 LED mode

```
typedef enum
{
    LM_Auto = 0,
    LM_Off,
    LM_Green,
    LM_Red,
    LM_Orange,
    LM_Last
} tLEDMode;
```

3.1.17 I/O card types

```
typedef enum
{
    IOCT_Autodetect = 0,
    IOCT_CardNotUsed,
    IOCT_RS232,
    IOCT_RS485,
    IOCT_Ethernet,
    IOCT_CAN,
    IOCT_GPIO,
    IOCT_GPIORRUMOD,
    IOCT_LED,
    IOCT_FTDI,
} tIOCardType;
```

3.1.18 I/O communication protocols

```
typedef enum
{
    IOCS_NotSpecified = 0,
    IOCS_KBP, // *** out of use ***
    IOCS_STDPRTCL, // *** out of use ***
    IOCS_KBRP, // Reader protocol (KBRP)
    IOCS_KBRPNHS, // Reader protocol (KBRP) without handshake
    IOCS_KBRPETH, // Reader protocol (KBRP) for Ethernet connections
    IOCS_Last
} tIOCardService;
```

3.2 System and configuration commands

3.2.1 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:

CFM_NORMAL

CFM_DIRECT

3.2.2 SetMode

Command block structure:

I1 I2 FM

I1: Command ID low byte

I2: Command ID high byte

FM: Function mode

Possible function modes:

CFM_NORMAL

CFM_DIRECT

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_OutOfRange

3.2.3 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

Communication standards: (→ „3.1.4 Communication standards“)

3.2.4 SetCommStandard

Command block structure:

I1 I2 CS

I1: Command ID low byte

I2: Command ID high byte

CS: Communication standard

Communication standards: (→ „3.1.4 Communication standards“)

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_NoLicense

RF_OutOfRange

RF_AccessDenied

3.2.5 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:

RF_NoError

RF_NoLicense

RF_OutOfRange

The transmit power of the antenna port is only returned if the Result flag of RF_NoError is set. The command returns the antenna power set for the antenna port in 1/4 dBm (erp).

3.2.6 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 0x44 (17 dBm(erp)) and 0x84 (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:

RF_NoError

RF_NoLicense

RF_OutOfRange

RF_AccessDenied

3.2.7 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.

3.2.8 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:

RF_NoError

RF_AccessDenied

RF_NonSpecified

3.2.9 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:

ANTMODE_ALL

ANTMODE_ANT1

ANTMODE_ANT2

ANTMODE_ANT3

ANTMODE_ANT4

3.2.10 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:

ANTMODE_ALL

ANTMODE_ANT1

ANTMODE_ANT2

ANTMODE_ANT3

ANTMODE_ANT4

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_OutOfRange

3.2.11 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 in milliseconds.



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

3.2.12 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 returned in milliseconds. 1 to 8 ports and their exposure times can be given.

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoLicense

RF_AccessDenied



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

3.2.13 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 1/4 dB

AG: Antenna gain in 1/4 dBic

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoLicense

3.2.14 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 1/4 dB (in the range from 0 dB to 63,75 dB)

AG: Antenna gain in 1/4 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:

RF_NoError

RF_OutOfRange

RF_NoLicense

RF_AccessDenied

3.2.15 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: ETSI channel in the range from 1 to 15

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoLicense

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

3.2.16 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 returns the global channel list – 1 to 4 return the list for the related antenna port)

CH: 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:

RF_NoError
 RF_OutOfRange
 RF_NoLicense
 RF_AccessDenied

UK

3.2.17 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:

EPCSW_MODE0
 EPCSW_MODE1

3.2.18 SetETSIPortChannelSwitchingMode

Command block structure:

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

Possible switching modes:

EPCSW_MODE0
 EPCSW_MODE1

Response block structure:

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

Possible result flags:

RF_NoError

RF_OutOfRange

RF_AccessDenied

3.2.19 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

3.2.20 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

3.2.21 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:

RF_NoError

RF_OutOfRange

RF_AccessDenied

3.2.22 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:

MT_DSB

MT_PRA

3.2.23 SetModulationType

Command block structure:

I1 I2 MT

I1: Command ID low byte

I2: Command ID high byte

MT: Modulation type

Possible modulation types:

MT_DSB

MT_PRA

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_OutOfRange

RF_AccessDenied

3.2.24 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

Extended result flags: (→ „3.1.13 Extended result flag“)

3.2.25 SetExtResultFlag

Command block structure:

II1 I2 EF

I1: Command ID low byte

I2: Command ID high byte

EF: Extended result flag

Extended result flags: (→ „3.1.13 Extended result flag“)

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_OutOfRange

RF_AccessDenied

3.2.26 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 2. byte

E3: Error flag register 3. byte

E4: Error flag register MSB

The bits in the error flag register have the following significance: (→ „3.1.12 Message numbers“)

3.2.27 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

3.2.28 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:

RF_NoError

RF_AccessDenied

3.2.29 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

3.2.30 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:

RF_NoError

RF_OutOfRange

RF_AccessDenied

3.2.31 SaveActiveParamset

Saves the parameter of the commands "SetCommStandard", "SetPortPower", "SetCarrierFollowUpTime", "SetPortMultiplexSequenceAndExposureTime", "SetCableLossAndAntennaGain", "SetETSIIPortChannelList", "SetETSIIPortChannelSwitchingMode", "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:

RF_NoError

3.2.32 GetParameterByConfigId

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 2. byte

P3: Parameter ID 3. 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 2. byte

P3: Parameter ID 3. byte

P4: Parameter ID MSB

V1: Parameter value LSB

V2: Parameter value 2. byte

V3: Parameter value 3. byte

V4: Parameter value MSB

Possible result flags:

RF_NoError

RF_OutOfRange

3.2.33 SetParameterByConfigId

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 2. byte

P3: Parameter ID 3. byte

P4: Parameter ID MSB

V1: Parameter value LSB

V2: Parameter value 2. byte

V3: Parameter value 3. 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 2. byte

P3: Parameter ID 3. byte

P4: Parameter ID MSB

Possible result flags:

RF_NoError

RF_OutOfRange

RF_AccessDenied

3.2.34 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:

RF_NoError
 RF_NoData
 RF_OutOfRange

UK

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

3.2.35 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:

RF_NoError
 RF_OutOfRange
 RF_AccessDenied
 RF_NonSpecified

3.2.36 GetIOCardProtocolConfig

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:

RF_NoError

RF_OutOfRange

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

3.2.37 SetIOCardProtocolConfig

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:

RF_NoError

RF_OutOfRange

RF_AccessDenied

RF_NonSpecified

3.2.38 GetIOCardExtendedInfo

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 [EI...EI]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

CN: I/O card number

EI: Extended module information

Possible result flags:

RF_NoError

RF_OutOfRange

RF_AccessDenied

RF_NonSpecified

3.2.39 GetMillisecondsTimestamp

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 timestamp LSB

T2: UTC timestamp Byte 2

T3: UTC timestamp Byte 3

T4: UTC timestamp MSB

The time stamp indicates the milliseconds since the device was last switched on.

3.2.40 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 timestamp LSB

T2: UTC timestamp Byte 2

T3: UTC timestamp Byte 3

T4: UTC timestamp MSB

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

3.2.41 SetTime

Command block structure:

I1 I2 T1 T2 T3 T4

I1: Command ID low byte

I2: Command ID high byte

T1: UTC timestamp LSB

T2: UTC timestamp Byte 2

T3: UTC timestamp Byte 3

T4: UTC timestamp MSB

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

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_AccessDenied

3.2.42 GetTemperature

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.

3.2.43 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 2. byte

B3: Build number 3. 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)

3.2.44 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

3.2.45 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

3.2.46 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:

RF_NoError

RF_WRONGLICENSEKEY

3.2.47 GetMaxPowerLimits

Command block structure:

I1 I2 PN

I1: Command ID low byte

I2: Command ID high byte

PN: Antenna port number (range 1 to 4)

Response block structure:

I1 I2 RF PN [P1 P2 P3]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

PN: Antenna port number (range 1 to 4)

P1: Maximum antenna power output (in 1/4 dBm (erp))

P2: Maximum power at antenna port (in 1/4 dBm)

P3: Maximum power at reader antenna connector (in 1/4 dBm)

Possible result flags:

RF_NoError

RF_NoLicense

RF_OutOfRange

RF_NonSpecified

3.2.48 SetConfigPassword

Command block structure:

I1 I2 PO .. PO 00 PN .. PN 00

I1: Command ID low byte

I2: Command ID high byte

PO: Old ConfigPassword

00: Password terminating character

PN: New ConfigPassword

00: Password terminating character

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_OutOfRange

RF_AccessDenied

RF_NonSpecified

3.2.49 EnterConfigPassword

Command block structure:

I1 I2 CP .. CP 00

I1: Command ID low byte

I2: Command ID high byte

CP: ConfigPasseword

00: Password terminating character

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_OutOfRange

RF_AccessDenied

RF_NonSpecified

3.2.50 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:

RF_NoError

RF_AccessDenied

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

3.2.51 FirmwareUpdatePrepare

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:

RF_NoError

RF_NoLicense

RF_AccessDenied

RF_NonSpecified

3.2.52 FirmwareUpdateTransmitData

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:

RF_NoError

RF_FIRMWAREREJECTED

RF_NoLicense

3.2.53 FirmwareUpdateFlash

Command block structure:

I1 I2

I1: Command ID low byte

I2: Command ID high byte

[5A A5:] Identifier for loading the standard configuration after the restart

If the two bytes 0x5A and 0xA5 are appended to the command ID, the reader reads the standard configuration after a restart.

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_FIRMWAREREJECTED

RF_NoLicense

3.2.54 RebootSystem

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:

RF_NoError

RF_OutOfRange

3.2.55 GetLEDMode

Command block structure:

I1 I2

I1: Command ID low byte

I2: Command ID high byte

Response block structure:

I1 I2 RF LM

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

LM: LED mode (→ „3.1.16 LED-Modus“)

Possible result flags:

RF_NoError

RF_OutOfRange

3.2.56 SetLEDMode

Command block structure:

I1 I2 LM

I1: Command ID low byte

I2: Command ID high byte

LM: LED mode (→ „3.1.16 LED-Modus“)

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NonSpecified

3.2.57 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:

RF_NoError

RF_OutOfRange

3.2.58 GetDeviceSerialNumber

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:

RF_NoError

RF_NonSpecified

3.2.59 GetDeviceTypeNumber

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:

RF_NoError

RF_NonSpecified

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

3.3.1 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 V2.0.0 page 76)

SN: Query-Session (0 to 3; see EPCglobal specification V2.0.0 page 76)

TA: Query target (0: A; 1: B; see EPCglobal specification V2.0.0 page 76)

3.3.2 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 V2.0.0 page 76)

SN: Query-Session (0 to 3; see EPCglobal specification V2.0.0 page 76)

TA: Query target (0: A; 1: B; see EPCglobal specification V2.0.0 page 76)



In normal mode the parameter “Query-Sel” affects only the commands “...GetEPCs”, “...Any” and “SyncWriteEPCToSingleTag” 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:

RF_NoError

RF_OutOfRange

RF_AccessDenied

3.3.3 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

3.3.4 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:

RF_NoError

RF_OutOfRange

RF_AccessDenied

3.3.5 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

3.3.6 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:

RF_NoError

RF_AccessDenied

3.3.7 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

3.3.8 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:

RF_NoError

RF_AccessDenied

3.3.9 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:

RF_NoError

RF_OutOfRange

3.3.10 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:

RF_NoError

RF_OutOfRange

3.3.11 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: Query target (see EPCglobal specification V2.0.0 page 73)

AC: Action (see EPCglobal specification V 2.0.0 page 73)

MB: Memory bank (see EPCglobal specification V 2.0.0 page 44)

P1: Bit pointer LSB

P2: Bit pointer 2. byte

P3: Bit pointer 3. byte

P4: Bit pointer MSB

ML: Masked length in bits

MD: Masked data

Possible result flags:

RF_NoError

RF_OutOfRange

3.3.12 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: Query target (see EPCglobal specification V2.0.0 page 73)

AC: Action (see EPCglobal specification V 2.0.0 page 73)

MB: Memory bank (see EPCglobal specification V 2.0.0 page 44)

P1: Bit pointer LSB

P2: Bit pointer 2. byte
P3: Bit pointer 3. byte
P4: Bit pointer MSB
ML: Masked length in bits
MD: Masked data



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:

RF_NoError

RF_OutOfRange

3.3.13 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:

RF_NoError

RF_OutOfRange



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.

UK

3.3.14 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



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:

RF_NoError

RF_OutOfRange

3.3.15 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



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:

RF_NoError

RF_OutOfRange

3.3.16 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:

RF_NoError

RF_OutOfRange

3.3.17 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:

RF_NoError

RF_OutOfRange

3.3.18 SetLED

Command block structure:

I1 I2 00 LN T1 T2

I1: Command ID low byte

I2: Command ID high byte

00: Reserved (immer 0x00)

LN: LED number (from 1 to 12)

T1: LSB activation time in ms

T2: MSB activation time in ms

If the activation time is set to 0xFFFFFFFF ms, the LED is permanently activated. To switch off an LED before the activation time has elapsed: Send a command with the activation time "0".

Response block structure:

I1 I2 RF 00 LN

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

00: Reserved

LN: LED number

Possible result flags:

RF_NoError

RF_OutOfRange

3.4 Commands in "Normal Mode"

3.4.1 SyncGetEPCs

Command block structure:

I1 I2

I1: Command ID low byte

I2: Command ID high byte

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En]]

...

[[ERD] EW [E1 ... En]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.2 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]

[[ERD] EW [E1 ... En]]

...

[[ERD] EW [E1 ... En]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.3 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 2. byte

W3: Access password 3. 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]

[[ERD] EW [E1 ... En] EC [WC D1 ... Dx]]

...

[[ERD] EW [E1 ... En] EC [WC D1 ... Dx]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

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:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.4 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 range of 0 to 31)

E1: EPC LSB

...

En: EPC MSB

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2nd byte

P3: Word pointer 3rd byte

P4: Word pointer MSB

WC: Number of words read

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC [WC D1 ... Dx]]

...

[[ERD] EW [E1 ... En] EC [WC D1 ... Dx]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

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:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.5 SyncReadDataUntilEndOfBankAny

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 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. byte

P4: Word pointer MSB

WC: Number of words read

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC [WC D1 ... Dx]]

...

[[ERD] EW [E1 ... En] EC [WC D1 ... Dx]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

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:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.6 SyncReadDataUntilEndOfBankSpecific

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

E1: EPC LSB

...

En: EPC MSB

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. byte

P4: Word pointer MSB

WC: Number of words read

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC [WC D1 ... Dx]]

...

[[ERD] EW [E1 ... En] EC [WC D1 ... Dx]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

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:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.7 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. 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]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.8 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 range of 0 to 31)

E1: EPC LSB

...

En: EPC MSB

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. 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]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.9 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. 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]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag
 ERD: Extended result data (→ „3.6 Extended result data“)
 EW: Number of EPC words
 E1: EPC LSB
 ...
 En: EPC MSB
 EC: Tag error code

Possible result flags:

RF_NoError
 RF_OutOfRange
 RF_NoStandard
 RF_NoTag
 RF_WrongCommandFunctionMode

3.4.10 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 range of 0 to 31)
 E1: EPC LSB
 ...
 En: EPC MSB
 W1: Access password LSB
 W2: Access password 2. byte
 W3: Access password 3. byte
 W4: Access password MSB
 MB: Memory bank (see EPCglobal specification V2.0.0 page 44)
 P1: Word pointer LSB
 P2: Word pointer 2. byte
 P3: Word pointer 3. 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]
 [[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.11 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. byte

P4: Word pointer MSB

WC: Number of words to erase

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte
 RF: Result flag
 EF: Extended result flag
 ERD: Extended result data (→ „3.6 Extended result data“)
 EW: Number of EPC words
 E1: EPC LSB
 ...
 En: EPC MSB
 EC: Tag error code

Possible result flags:

RF_NoError
 RF_OutOfRange
 RF_NoStandard
 RF_NoTag
 RF_WrongCommandFunctionMode

3.4.12 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 range of 0 to 31)
 E1: EPC LSB
 ...
 En: EPC MSB
 W1: Access password LSB
 W2: Access password 2. byte
 W3: Access password 3. byte
 W4: Access password MSB
 MB: Memory bank (see EPCglobal specification V2.0.0 page 44)
 P1: Word pointer LSB
 P2: Word pointer 2. byte
 P3: Word pointer 3. byte
 P4: Word pointer MSB
 WC: Number of words to erase

Response block structure:

I1 I2 RF [EF]
 [[ERD] EW [E1 ... En] EC]
 ...
 [[ERD] EW [E1 ... En] EC]

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
ERD: Extended result data (→ „3.6 Extended result data“)
EW: Number of EPC words
E1: EPC LSB
...
En: EPC MSB
EC: Tag error code

Possible result flags:

RF_NoError
RF_OutOfRange
RF_NoStandard
RF_NoTag
RF_WrongCommandFunctionMode

3.4.13 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 2. byte
W3: Access password 3. byte
W4: Access password MSB
L1: Payload LSB
L2: Payload 2. byte
L3: Payload MSB

Response block structure:

I1 I2 RF [EF]
[[ERD] EW [E1 ... En] EC]
...
[[ERD] EW [E1 ... En] EC]
I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
EF: Extended result flag
ERD: Extended result data (→ „3.6 Extended result data“)
EW: Number of EPC words

E1: EPC LSB
 ...
 En: EPC MSB
 EC: Tag error code

Possible result flags:

RF_NoError
 RF_NoStandard
 RF_NoTag
 RF_WrongCommandFunctionMode

3.4.14 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 range of 0 to 31)
 E1: EPC LSB
 ...
 En: EPC MSB
 W1: Access password LSB
 W2: Access password 2. byte
 W3: Access password 3. byte
 W4: Access password MSB
 L1: Payload LSB
 L2: Payload 2. byte
 L3: Payload MSB

Response block structure:

I1 I2 RF [EF]
 [[ERD] EW [E1 ... En] EC]
 ...
 [[ERD] EW [E1 ... En] EC]
 I1: Response ID low byte
 I2: Response ID high byte
 RF: Result flag
 EF: Extended result flag
 ERD: Extended result data (→ „3.6 Extended result data“)
 EW: Number of EPC words
 E1: EPC LSB
 ...
 En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.15 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 2. byte

W3: Kill password 3. byte

W4: Kill password MSB

RC: Recommissioning bits (see EPCglobal specification V1.2.0 page 68)

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.16 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 range of 0 to 31)

E1: EPC LSB

...

En: EPC MSB

W1: Kill password LSB

W2: Kill password 2. byte

W3: Kill password 3. byte

W4: Kill password MSB

RC: Recommissioning bits (see EPCglobal specification V1.2.0 page 68)

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.17 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 2. byte
- W3: Access password 3. byte
- W4: Access password MSB
- 00: Reserved (always 0x00)
- RL: Read / Lock (see EPCglobal specification V2.0.0 page 94)
- MB: Memory bank (see EPCglobal specification V2.0.0 page 44)
- P1: Block word pointer LSB
- P2: Block word pointer 2. byte
- P3: Block word pointer 3. byte
- P4: Block word pointer MSB
- BR: Block range (see EPCglobal specification V2.0.0 page 95)
- LL: Lock data LSB
- LH: Lock data MSB

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC [BR L1 ... Lx]]

...

[[ERD] EW [E1 ... En] EC [BR L1 ... Lx]]

- I1: Response ID low byte
- I2: Response ID high byte
- RF: Result flag
- EF: Extended result flag
- ERD: Extended result data (→ „3.6 Extended result data“)
- 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:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.18 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 range of 0 to 31)

E1: EPC LSB

...

En: EPC MSB

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

00: Reserved (always 0x00)

RL: Read / Lock (see EPCglobal specification V2.0.0 page 94)

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Block word pointer LSB

P2: Block word pointer 2. byte

P3: Block word pointer 3. byte

P4: Block word pointer MSB

BR: Block range (see EPCglobal specification V2.0.0 page 95)

LL: Lock data LSB

LH: Lock data MSB

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC [BR L1 ... Lx]]

...

[[ERD] EW [E1 ... En] EC [BR L1 ... Lx]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

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:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.19 SyncAuthenticateAny

Command block structure:

I1 I2 W1 W2 W3 W4 CS L1 L2 [D1 ... Dn]

I1: Command ID low byte

I2: Command ID high byte

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

CS: Encryption package

L1: Number of bytes in the message (LSB)

L2: Number of bytes in the message (MSB)

D1: 1. byte of the message

...

Dn: Last byte of the message

Transmission sequence: The first bit to be transmitted is located in the MSB of the first byte (D1) of the message data.

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC [L1 L2 [D1 ... Dn]]]

...

[[ERD] EW [E1 ... En] EC [L1 L2 [D1 ... Dn]]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

L1: Number of bytes in the message (LSB)

L2: Number of bytes in the message (MSB)

D1: 1. byte of the reply message

...

Dn: Last byte of the reply message

Receive sequence: The first bit to be received is located in the MSB of the first byte (D1) of the message data.

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

RF_NonSpecified

3.4.20 SyncAuthenticateSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 CS L1 L2 [D1 ... Dn]

I1: Command ID low byte

I2: Command ID high byte

EW: Number of EPC words (in range of 0 to 31)

E1: EPC LSB

...

En: EPC MSB

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

CS: Encryption package

L1: Number of bytes in the message (LSB)

L2: Number of bytes in the message (MSB)

D1: 1. byte of the message

...

Dn: Last byte of the message

Transmission sequence: The first bit to be transmitted is located in the MSB of the first byte (D1) of the message data.

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC [L1 L2 [D1 ... Dn]]]

...

[[ERD] EW [E1 ... En] EC [L1 L2 [D1 ... Dn]]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

L1: Number of bytes in the message (LSB)

L2: Number of bytes in the message (MSB)

D1: 1. byte of the reply message

...

Dn: Last byte of the reply message

Receive sequence: The first bit to be received is located in the MSB of the first byte (D1) of the message data.

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

RF_NonSpecified

3.4.21 SyncUntraceableAny

Command block structure:

I1 I2 W1 W2 W3 W4 UB EM TM UM RG

I1: Command ID low byte

I2: Command ID high byte

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

UB: U-bit in XPC_W (see EPCglobal specification V2.0.0 page 107)

EM: Hide EPC (see EPCglobal specification V2.0.0 page 107)

TM: Hide TID (see EPCglobal specification V2.0.0 page 107)

UM: Hide user memory (see EPCglobal specification V2.0.0 page 107)

RG: Operating range (see EPCglobal specification V2.0.0 page 107)

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.22 SyncUntraceableSpecific

Command block structure:

I1 I2 EW [E1 ... En] W1 W2 W3 W4 UB EM TM UM RG

I1: Command ID low byte

I2: Command ID high byte

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

UB: U-bit in XPC_W (see EPCglobal specification V2.0.0 page 107)

EM: Hide EPC (see EPCglobal specification V2.0.0 page 107)

TM: Hide TID (see EPCglobal specification V2.0.0 page 107)

UM: Hide user memory (see EPCglobal specification V2.0.0 page 107)

RG: Operating range (see EPCglobal specification V2.0.0 page 107)

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.23 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 range of 0 to 31)
 E1: EPC LSB
 ...
 En: EPC MSB
 W1: Access password LSB
 W2: Access password 2. byte
 W3: Access password 3. byte
 W4: Access password MSB
 NW: Number of EPC words new EPC (in range of 0 to 31)
 N1: new EPC LSB
 ...
 Nn: new EPC MSB

Response block structure:

I1 I2 RF [EF]
 [[ERD] EW [E1 ... En] EC]
 ...
 [[ERD] EW [E1 ... En] EC]
 I1: Response ID low byte
 I2: Response ID high byte
 RF: Result flag
 EF: Extended result flag
 ERD: Extended result data (→ „3.6 Extended result data“)
 EW: Number of EPC words
 E1: EPC LSB
 ...
 En: EPC MSB
 EC: Tag error code

Possible result flags:

RF_NoError
 RF_OutOfRange
 RF_NoStandard
 RF_NoTag
 RF_WrongCommandFunctionMode

3.4.24 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

NW: Number of EPC words new EPC (in range of 1 to 31)

N1: new EPC LSB

...

Nn: new EPC MSB

Response block structure:

I1 I2 RF [EF]

[[ERD] EW E1 ... En EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoStandard

RF_MoreThanOneTagInField

RF_NoTag

RF_WrongCommandFunctionMode

3.4.25 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.26 SyncNXP G2XReadProtectSpecific

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 range of 0 to 31)

E1: EPC LSB

...

En: EPC MSB

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.27 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.28 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 range of 0 to 31)

E1: EPC LSB

...

En: EPC MSB

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.29 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

AB: EAS alarm bit

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En] EC]

...

[[ERD] EW [E1 ... En] EC]

I1: Response ID low byte
 I2: Response ID high byte
 RF: Result flag
 EF: Extended result flag
 ERD: Extended result data (→ „3.6 Extended result data“)
 EW: Number of EPC words
 E1: EPC LSB
 ...
 En: EPC MSB
 EC: Tag error code

Possible result flags:

RF_NoError
 RF_NoStandard
 RF_NoTag
 RF_WrongCommandFunctionMode

3.4.30 SyncNXP2XChangeEASSpecific

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 range of 0 to 31)
 E1: EPC LSB
 ...
 En: EPC MSB
 W1: Access password LSB
 W2: Access password 2. byte
 W3: Access password 3. byte
 W4: Access password MSB
 AB: EAS alarm bit

Response block structure:

I1 I2 RF [EF]
 [[ERD] EW [E1 ... En] EC]
 ...
 [[ERD] EW [E1 ... En] EC]
 I1: Response ID low byte
 I2: Response ID high byte
 RF: Result flag
 EF: Extended result flag
 ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.31 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: Timestamp LSB

T2: Timestamp 2. byte

T3: Timestamp 3. byte

T4: Timestamp MSB

00: Reserved (always 0x00)

Possible result flags:

RF_NoError (Alarmcode empfangen)

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.32 SyncRSSILedBarSpecific

Command block structure:

I1 I2 EW [E1 ... En] CF [PN]

I1: Command ID low byte

I2: Command ID high byte

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

CF: Control flag (0: Start; 1: Execute; 2: Stop)

PN: Antenna port number

Response block structure:

I1 I2 RF [EF]

[[ERD] EW [E1 ... En]]

...

[[ERD] EW [E1 ... En]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

Possible result flags:

RF_NoError

RF_NoStandard

RF_NoTag

RF_WrongCommandFunctionMode

3.4.33 ASyncGetRawEPCs

Command block structure:

I1 I2

I1: Command ID low byte

I2: Command ID high byte

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_WrongCommandFunctionMode

Block structure of the asynchronous response:

I1 I2 EF [ERD] EW [E1 ... En]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

3.4.34 ASyncGetEPCs

Command block structure:

I1 I2

I1: Command ID low byte

I2: Command ID high byte

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_WrongCommandFunctionMode

Block structure of the asynchronous response:

I1 I2 CG EF [ERD] EW [E1 ... En]

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

3.4.35 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. byte
P4: Word pointer MSB
WC: Number of words read

Response block structure:

I1 I2 RF FL

I1: Response ID low byte
I2: Response ID high byte
RF: Result flag
FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError
RF_NoStandard
RF_OutOfRange
RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC [WC D1 ... Dx]

I1: Response ID low byte
I2: Response ID high byte
CG: Come/Go flag (Come message: 0xFF)
EF: Extended result flag
ERD: Extended result data (→ „3.6 Extended result data“)
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

3.4.36 ASyncReadDataUntilEndOfBankAny

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 2. byte
W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V1.2.0 page 37)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. byte

P4: Word pointer MSB

WC: Number of words to write

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_OutOfRange

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC [WC D1 ... Dx]

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

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

3.4.37 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V1.2.0 page 37)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. 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 FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_OutOfRange

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.38 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. 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 FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_OutOfRange

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.39 ASyncBlockEraseAny

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 2. byte

W3: Access password 3. byte

W4: Access password MSB

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. byte

P4: Word pointer MSB

WC: Number of words to erase

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_OutOfRange

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.40 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

L1: Payload LSB

L2: Payload 2. byte

L3: Payload MSB

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.41 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 2. byte

W3: Kill password 3. byte

W4: Kill password MSB

RC: Recommissioning bits (see EPCglobal specification V1.2.0 page 68)

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.42 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

00: Reserved (always 0x00)

RL: Read / Lock (see EPCglobal specification V2.0.0 page 94)

MB: Memory bank (see EPCglobal specification V2.0.0 page 44)

P1: Block word pointer LSB

P2: Block word pointer 2. byte

P3: Block word pointer 3. byte

P4: Block word pointer MSB

BR: Block range (see EPCglobal specification V2.0.0 page 95)

LL: Lock data LSB

LH: Lock data MSB

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_OutOfRange

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC [BR L1 ... Lx]

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

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

3.4.43 ASyncAuthenticateAny

Command block structure:

I1 I2 W1 W2 W3 W4 CS L1 L2 [D1 ... Dn]

I1: Command ID low byte

I2: Command ID high byte

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

CS: Encryption package

L1: Number of bytes in the message (LSB)

L2: Number of bytes in the message (MSB)

D1: 1. byte of the message

...

Dn: Last byte of the message

Transmission sequence: The first bit to be transmitted is located in the MSB of the first byte (D1) of the message data.

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_OutOfRange

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC [L1 L2 [D1 ... Dn]]

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

L1: Number of bytes in the message (LSB)

L2: Number of bytes in the message (MSB)

D1: 1. byte of the reply message

...

Dn: Last byte of the reply message

3.4.44 ASyncUntraceableAny

Command block structure:

I1 I2 W1 W2 W3 W4 UB EM TM UM RG

I1: Command ID low byte

I2: Command ID high byte

W1: Access password LSB

W2: Access password 2. byte

W3: Access password 3. byte

W4: Access password MSB

UB: U-bit in XPC_W (see EPCglobal specification V2.0.0 page 107)

EM: Hide EPC (see EPCglobal specification V2.0.0 page 107)

TM: Hide TID (see EPCglobal specification V2.0.0 page 107)

UM: Hide user memory (see EPCglobal specification V2.0.0 page 107)

RG: Operating range (see EPCglobal specification V2.0.0 page 107)

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_OutOfRange

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.45 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

Response block structure:

I1 I2 RF FL

I1 : Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.46 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 2. byte

W3: Access password 3. byte

W4: Access password MSB

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.47 ASyncNXPG2XChangeEASAny

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 2. byte

W3: Access password 3. byte

W4: Access password MSB

AB: EAS alarm bit

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_WrongCommandFunctionMode

Block structure of the asynchronous response (coming message):

I1 I2 CG EF [ERD] EW [E1 ... En] EC

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

EC: Tag error code

3.4.48 ASyncNXPG2XEASAlarm

Command block structure:

I1 I2

I1: Command ID low byte

I2: Command ID high byte

Response block structure:

I1 I2 RF FL

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

FL: Start/Stop/Heartbeat flag

Possible result flags:

RF_NoError

RF_NoStandard

RF_WrongCommandFunctionMode

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: Come/Go flag (Come message: 0xFF)

EF: Extended result flag

AP: Antenna port

RS: RSSI value

T1: Timestamp LSB

T2: Timestamp 2. byte

T3: Timestamp 3. byte

T4: Timestamp MSB

00: Reserved (always 0x00)

3.4.49 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:

RF_NoError

RF_WrongCommandFunctionMode

3.4.50 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:

RF_NoError

RF_WrongCommandFunctionMode

3.5 Commands in "Direct Mode"

3.5.1 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 represents a configured antenna)

3.5.2 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:

RF_NoError

RF_OutOfRange

RF_WrongCommandFunctionMode

3.5.3 SetFrequency

Command block structure:

I1 I2 F1 F2 F3

I1: Command ID low byte

I2: Command ID high byte

F1: Frequency LSB

F2: Frequency 2. byte

F3: Frequency MSB

The frequency is given in multiples of 1 kHz.

Example for 865.7 MHz: 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:

RF_NoError

RF_OutOfRange

RF_WrongCommandFunctionMode

3.5.4 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

N1: Noise value in 1/100 dBm LSB (3rd generation readers only)

N2: Noise value in 1/100 MSB (3rd generation readers only)

Possible result flags:

RF_NoError

RF_NoAntenna

RF_NoFrequency

RF_WrongCommandFunctionMode

3.5.5 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

3.5.6 CarrierOnOff

Command block structure:

I1 I2 OO

I1: Command ID low byte

I2: Command ID high byte

OO: On/Off flag

Response block structure:

I1 I2 RF

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

Possible result flags:

RF_NoError

RF_NoAntenna

RF_NoFrequency

RF_NoProfile

RF_AntennaError

RF_WrongCommandFunctionMode

3.5.7 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:

RF_NoError

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.8 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:

RF_NoError

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.9 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 2. byte

V3: Measured value 3. byte

V4: Measured value MSB

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

3.5.10 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 2. byte

V3: Measured value 3. byte

V4: Measured value MSB

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

3.5.11 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 2. byte

V3: Measured value 3. byte

V4: Measured value MSB

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

3.5.12 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:

RF_NoError

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.13 OpenNoninventoriedTag

Command block structure:

I1 I2

I1: Command ID low byte

I2: Command ID high byte

Response block structure:

I1 I2 RF [EF [ERD] EW [E1 ... En] [H1 H2]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

H1: Handle low byte

H2: Handle high byte

Possible result flags:

RF_NoError

RF_NoCarrier

RF_NoTag

RF_WrongCommandFunctionMode

RF_NoHandle

3.5.14 OpenSpecificTag

Command block structure:

I1 I2 EW [E1 ... En]

I1: Command ID low byte

I2: Command ID high byte

EW: Number of EPC words (im Bereich von 0bis 31)

E1: EPC LSB

...

En: EPC MSB

Response block structure:

I1 I2 RF [EF [ERD] EW [E1 ... En] [H1 H2]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

H1: Handle low byte

H2: Handle high byte

Possible result flags:

RF_NoError

RF_NoCarrier

RF_NoTag

RF_WrongCommandFunctionMode

RF_NoHandle

3.5.15 DirectBulkGetEPCs

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]

[[ERD] EW [E1 ... En]]

...

[[ERD] EW [E1 ... En]]

I1: Response ID low byte

I2: Response ID high byte

RF: Result flag

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

EW: Number of EPC words

E1: EPC LSB

...

En: EPC MSB

Possible result flags:

RF_NoError

RF_OutOfRange

RF_NoCarrier

RF_NoTag

RF_WrongCommandFunctionMode

3.5.16 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 V2.0.0 page 73)

AC: Action (see EPCglobal specification V2.0.0 page 73)

MB: Memory bank (see EPCglobal specification V2.0.0 page 73)

P1: Bit pointer LSB

P2: Bit pointer 2. byte

P3: Bit pointer 3. 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:

RF_NoError

RF_OutOfRange

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.17 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]] [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:

RF_NoError

RF_NoData

RF_CRCError

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.18 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:

RF_NoError

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.19 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:

RF_NoError

RF_NoData

RF_CRCErrors

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.20 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:

RF_NoError

RF_NoData

RF_CRCError

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.21 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 2. byte

P3: Word pointer 3. byte

P4: Word pointer MSB

WC: Number of words 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:

RF_NoError

RF_NoData

RF_CRCErr

RF_OutOfRange

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.22 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 2. byte

P3: Word pointer 3. 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:

RF_NoError

RF_NoData

RF_CRCError

RF_OutOfRange

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.23 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:

RF_NoError

RF_NoData

RF_CRCErrors

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.24 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 2. 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:

- RF_NoError
- RF_NoData
- RF_CRCError
- RF_NoCarrier
- RF_WrongCommandFunctionMode

3.5.25 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 2. byte

P3: Word pointer 3. 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:

RF_NoError

RF_NoData

RF_CRCErr

RF_OutOfRange

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.26 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 V2.0.0 page 44)

P1: Word pointer LSB

P2: Word pointer 2. byte

P3: Word pointer 3. 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:

RF_NoError

RF_NoData

RF_CRCError

RF_OutOfRange

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.27 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:

RF_NoError

RF_NoData

RF_OutOfRange

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.28 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 V2.0.0 page 77)

AV: Adjust value (0, 3 or 6; see EPCglobal specification V2.0.0 page 77)

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:

RF_NoError

RF_NoData

RF_OutOfRange

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.29 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 V2.0.0 page 77)

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:

RF_NoError

RF_NoData

RF_OutOfRange

RF_NoCarrier

RF_WrongCommandFunctionMode

3.5.30 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:

RF_NoError

RF_NoData

RF_NoCarrier

RF_WrongCommandFunctionMode

3.6 Extended result data

Tag operations contain the following extended result data:

- the antenna connection,
- the RSSI value,
- the timestamp of tag access,
- the Protocol Control Word (s) of the tags,
- the frequency,
- the phase angle φ the tag response.

Which information is transmitted depends on the Extended result flags (EF) (→ „3.1.13 Extended result flag“).

Structure:

[[AP] [RS] [T1 T2 T3 T4] [P1 P2 [X1L X1H [X2L X2H]]] [F1 F2 F3 PL PH] [R1 R2]]

AP: Antenna connection

RS: RSSI value

T1: Timestamp LSB

T2: Timestamp 2. byte

T3: Timestamp 3. byte

T4: Timestamp 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

F1: Frequency LSB

F2: Frequency 2. byte

F3: Frequency MSB

PL: Tag phase LSB

PH: Tag phase MSB

R1: RSSI value in dBm low byte

R2: RSSI value in dBm high byte

3.7 Asynchronous responses

3.7.1 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: Timestamp LSB

T2: Timestamp 2. byte

T3: Timestamp 3. byte

T4: Timestamp MSB

MT: Message type (reserved: always 0x00)

MN: Message number

CG: Come/Go flag

3.7.2 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 [ERD] EW [E1 ... En] C1 C2

I1: Response ID low byte

I2: Response ID high byte

CG: Come/Go flag (go message: 0x00)

EF: Extended result flag

ERD: Extended result data (→ „3.6 Extended result data“)

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

4 Change history

Version	Change	Date
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	- TagErrorCodes-Enum extended - Commands "GetActiveParamset", "SetActiveParamset", "GetASyncObservedListParameters" and "SetASyncObservedListParameters" added - Command "SaveConfiguration" renamed as "SetActiveParamset" - lower power figure in chapter 4.2.6 corrected - Commands "GetParameterByld" and "SetParameterByld2" added - mode "PCSW_MODE2" removed from the Enum "ETSIPortChannelSwitchingMode" (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 "COMMSTANDARD_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 "TagErrorCodes" expanded by the value "TEC_VERIFYFAILED" - Naming error in the command IDs rectified ("CMD_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 "SetActiveParamset" (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 "ResultFlag" expanded by the entry "RESULTFLAG_WRONGCFM" and this return value added for the relevant functions	2008-12-10

Version	Change	Date
1.12	- Structure "ResultFlag" expanded by the entry "RESULTFLAG_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 "GetParameterByld" (chapter 4.2.32) now also returns the transferred ID in the response data - Function "SetParameterByld" (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 "HostCommCmds" (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 "RESULTFLAG_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
1.17	Specification version valid from reader firmware v2.00.00 - Number of select filters increased to 32 - Chapters were converted to variable EPC length.	2009-12-08
1.18	- IDs for the additional commands to support NXP-UCODE commands added in the chapter "Command / response IDs" - Chapters were added.	2009-12-16

Version	Change	Date
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 "RESULTFLAG_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" - Chapters were extended by XPC functionality. - IDs for the BlockPermalock commands extended in the chapter "Command / response IDs" - Chapters "SyncBlockPermalockAny", "SyncBlockPermalockSpecific" and "ASyncBlockPermalockAny" added	2010-01-13
1.20	- Structure of return values added. - Missing asynchronous response IDs extended in the chapter "Command / response IDs"	2010-02-02
1.21	- Value "CG_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 "RESULTFLAG_OUTOFRANGE" extended for the command "SyncWriteEPCToSingleTag"	2010-07-14
1.23	- Command "GetDeviceSerialNmbr" added	2010-08-25
1.24	- Chapter "LoadFactoryDefaults" is extended by the transfer of a parameter set number. - Command "GetDeviceTypeNmbr" added - Chapter "Ethernet transmission" added	2010-12-17

Version	Change	Date
3.00	Updates for the 3rd generation of readers	2019-09-03