TWN4

Simple Protocol

DocRev11, May 13, 2016

Elatec GmbH

Contents

1 Simple Protocol

This document describes the serial protocol of TWN4.

In order to operate this protocol, a firmware type TWN4_Cxvvv_PRSwww.bix is required, where vvv and www are the version numbers.

A firmware as mentioned above combines virtual USB (CDC) or true serial communication with an TWN4 app, which implements the simple protocol (PRS = PRotocol Simple).

This protocol is called simple because it is based on a communication with ASCII characters which can also be tested manually by using a terminal program. There is no additional overhead for things like packet repetition, address bytes...

The simple protocol is also available in binary mode. This means, that the data is not transmitted via ASCII characters but as single bytes.
Moreover it is possible to add a CRC at the end of every transmission. This lets you detect transmission errors.
The communication is based on a command/response structure: TWN4 will only send data to the host as a response of a command. Command and response are lines of bytes terminated by a carriage return. Carriage return is not shown explicitely anymore in the following documentation. A byte is always represented and transmitted by two hexadecimal ASCII characters.

1.1 Command

A command always starts with two bytes which reflect the API and function number to be executed.

1.2 Response

A response always starts with a byte, which reflects execution of the command on protocol level. Following possible error values:

ERR_NONE	0
ERR_UNKNOWN_FUNCTION	1
ERR_MISSING_PARAMETER	2
ERR_UNUSED_PARAMETERS	3
ERR_INVALID_FUNCTION	4
ERR_PARSER	5

1.3 Data Transmission

Data can be transmitted in two ways:
- by sending ASCII characters
- by sending binary values

1.3.1 ASCII

To transmit a value of e.g. 0x1F, it is necessary to split this into two ASCII characters '1' and 'F'. These characters has to be sent sequentially.

1.3.2 Binary

To transmit a value of e.g. 0x1F, it can be sent directly in binary format.

1.3.3 CRC

On both ASCII and binary format, a CRC can be added at the end of each transmission. The CRC is calculated as follows:

```
uint16_t UpdateCRC(uint16_t CRC,byte Byte)
{
        // Update CCITT CRC (reverse polynom 0x8408)
        Byte ^= (byte)CRC;
        Byte ^= (byte)(Byte << 4);
        return (uint16_t)(((Byte << 8) | (CRC >> 8)) ^ (Byte >> 4) ^ (Byte << 3));
}
```

The CRC calculation starts with CRC = 0xFFFF

1.3.4 Reference messages

The following table shows reference messages for function GetUSBType

Mode	CRC	Command (Host -> TWN4)	Response (TWN4 -> Host)
ASCII	Off	"0005\r"	"0001\r"
	On	"000515A7\r"	"000131E1\r"
Binary	Off	0x02 0x00 0x00 0x05	0x02 0x00 0x00 0x01
	On	0x04 0x00 0x00 0x05 0x15 0xA7	0x04 0x00 0x00 0x01 0x31 0xE1

1.4 Data Types

The description of the commands is using data types, which have to be built-up as follows:

Data Type	Description
[Byte]:	One single byte (sent as two hex digits)
[UInt16]:	Two bytes (LSB first)
[UInt32]:	Four bytes (LSB first)
[Bool]:	One single byte which can hold two values: 0 or 1
[Byte Array(n)]:	A sequence of bytes with known and fixed number of bytes. The number of bytes is not transferred explicitely, because both host and TWN4 do know this number.
[Byte Array(Var)]:	A sequence of bytes, where the first byte holds the number of following bytes
[Byte Array(Var), x LB]:	A sequence of bytes, where the first x bytes hold the number of following bytes
[ASCII string]:	A sequence of bytes which contain ASCII characters, except the first byte which holds the number of following bytes

In Simple Protocol, all numbers are sent with LSB first. For example, the number `0x1234` has to be sent as `3412`.

1.5 Commands

1.5.1 API SYS

1.5.1.1 Reset

Command:	[0001]
Response:	[00]
Example Command: Response:	0001

1.5.1.2 StartBootloader

Command:	[0002]
Response:	[00]
Example Command: Response:	0002

1.5.1.3 GetSysTicks

Command:	[0003]
Response:	[00][UInt32: *Ticks*]
Example	
Command:	0003
Response:	00D3480700
	(Ticks: 477395)

1.5.1.4 GetVersionString

Command:	[0004][Byte: *MaxLen*]
Response:	[00][ASCII string: *Version*]
Example	
Command:	0004FF
	(MaxLen: FF)
Response:	001D54574E342F42312E30332F434346312E35372F505253312E3033-2F5049
	(Version: TWN4/B1.03/CCF1.57/PRS1.03/PI)

1.5.1.5 GetUSBType

Command:	[0005]
Response:	[00][Byte: *Type*]
Example	
Command:	0005
Response:	0001
	(Type: 1)

1.5.1.6 GetDeviceType

Command:	[0006]
Response:	[00][Byte: *Type*]
Example	
Command:	0006
Response:	000B
	(Type: 11)

1.5.1.7 Sleep

Command:	[0007][UInt32: *Ticks*][UInt32: *Flags*]
Response:	[00][Byte: *Result*]
Example	
Command:	0007E803000001000000
	(Ticks: E8030000, Flags: 01000000)
Response:	0000
	(Result: 0)

1.5.1.8 GetDeviceUID

Command:	[0008]
Response:	[00][Byte Array(12): *UID*]
Example	
Command:	0008
Response:	002D002F000B47303531353233
	(UID: 2D002F000B47303531353233)

1.5.1.9 SetParameters

Command:	[0009][Byte Array(Var): *TLV*]
Response:	[00][Bool: *Result*]
Example	
Command:	00090707010103010200
	(TLV: 07010103010200)
Response:	0001
	(Result: true)

1.5.1.10 GetLastError

Command:	[000A]
Response:	[00][UInt32: *LastError*]
Example	
Command:	000A
Response:	00CB000000
	(LastError: 203)

1.5.2 API IO

1.5.2.1 WriteByte

Command:	[0100][Byte: *Channel*][Byte: *Byte*]
Response:	[00]
Example Command: Response:	 01000041 (Channel: 00, Byte: 41) 00

1.5.2.2 ReadByte

Command:	[0101][Byte: *Channel*]
Response:	[00][Byte: *Byte*]
Example Command: Response:	 010100 (Channel: 00) 0000 (Byte: 0)

1.5.2.3 TestEmpty

Command:	[0102][Byte: *Channel*][Byte: *Dir*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 01020001 (Channel: 00, Dir: 01) 0001 (Result: Yes)

1.5.2.4 TestFull

Command:	[0103][Byte: *Channel*][Byte: *Dir*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 01030001 (Channel: 00, Dir: 01) 0000 (Result: No)

1.5.2.5 GetBufferSize

Command:	[0104][Byte: *Channel*][Byte: *Dir*]
Response:	[00][UInt16: *BufferSize*]
Example Command: Response:	 01040001 (Channel: 00, Dir: 01) 000000 (BufferSize: 0)

1.5.2.6 GetByteCount

Command:	[0105][Byte: *Channel*][Byte: *Dir*]
Response:	[00][UInt16: *ByteCount*]
Example Command: Response:	 01050001 (Channel: 00, Dir: 01) 000000 (ByteCount: 0)

1.5.2.7 SetCOMParameters

Command:	[0109][Byte: *Channel*][UInt32: *Baudrate*][Byte: *WordLength*][Byte: *Parity*][Byte: *StopBits*][Byte: *FlowControl*]
Response:	[00][Bool: *Result*]
Example	
Command:	010902802500000008000100
	(Channel: 02, Baudrate: 80250000, WordLength: 08, Parity: 00, StopBits: 01, FlowControl: 00)
Response:	0001
	(Result: true)

1.5.2.8 GetUSBDeviceState

Command:	[010A]
Response:	[00][Byte: *State*]
Example	
Command:	010A
Response:	0003
	(State: USB_DEVICE_STATE_CONFIGURED)

1.5.2.9 GetHostChannel

Command:	[010B]
Response:	[00][Byte: *Channel*]
Example	
Command:	010B
Response:	0001
	(Channel: CHANNEL_USB)

1.5.3 API PERIPH

1.5.3.1 GPIOConfigureOutputs

Command:	[0400][Byte: *Bits*][Byte: *PullUpDown*][Byte: *OutputType*]
Response:	[00]
Example	
Command:	0400010000
	(Bits: 01, PullUpDown: 00, OutputType: 00)
Response:	00

1.5.3.2 GPIOConfigureInputs

Command:	[0401][Byte: *Bits*][Byte: *PullUpDown*]
Response:	[00]
Example Command:	 04010100 (Bits: 01, PullUpDown: 00)
Response:	00

1.5.3.3 GPIOSetBits

Command:	[0402][Byte: *Bits*]
Response:	[00]
Example Command:	 040201 (Bits: 01)
Response:	00

1.5.3.4 GPIOClearBits

Command:	[0403][Byte: *Bits*]
Response:	[00]
Example Command:	 040301 (Bits: 01)
Response:	00

1.5.3.5 GPIOToggleBits

Command:	[0404][Byte: *Bits*]
Response:	[00]
Example Command:	 040401 (Bits: 01)
Response:	00

1.5.3.6 GPIOBlinkBits

Command:	[0405][Byte: *Bits*][UInt16: *TimeHi*][UInt16: *TimeLo*]
Response:	[00]
Example Command: Response:	 04050164006400 (Bits: 01, TimeHi: 6400, TimeLo: 6400) 00

1.5.3.7 GPIOTestBit

Command:	[0406][Byte: *Bit*]
Response:	[00][Byte: *Result*]
Example Command: Response:	 040601 (Bit: 01) 0000 (Result: 0)

1.5.3.8 Beep

Command:	[0407][Byte: *Volume*][UInt16: *Frequency*][UInt16: *OnTime*][UInt16: *OffTime*]
Response:	[00]
Example Command: Response:	 0407646009F401F401 (Volume: 64, Frequency: 6009, OnTime: F401, OffTime: F401) 00

1.5.3.9 DiagLEDOn

Command:	[0408]
Response:	[00]
Example Command: Response:	 0408 00

1.5.3.10 DiagLEDOff

Command:	[0409]
Response:	[00]
Example	
Command:	0409
Response:	00

1.5.3.11 DiagLEDToggle

Command:	[040A]
Response:	[00]
Example	
Command:	040A
Response:	00

1.5.3.12 DiagLEDIsOn

Command:	[040B]
Response:	[00][Bool: *Result*]
Example	
Command:	040B
Response:	0000
	(Result: No)

1.5.3.13 SendWiegand

Command:	[040C][Byte: *GPIOData0*][Byte: *GPIOData1*][UInt16: *PulseTime*][UInt16: *IntervalTime*][Byte Array(Var): *Bits*][Byte: *BitCount*]
Response:	[00]
Example	
Command:	040C08106400E80301AA08
	(GPIOData0: 08, GPIOData1: 10, PulseTime: 6400, IntervalTime: E803, Bits: AA, BitCount: 08)
Response:	00

1.5.3.14 SendOmron

Command:	[040D][Byte: *GPIOClock*][Byte: *GPIOData*][UInt16: *T1*][UInt16: *T2*][UInt16: *T3*][Byte Array(Var): *Bits*][Byte: *BitCount*]
Response:	[00]
Example Command: Response:	 040D0810F401F401F40101AA08 (GPIOClock: 08, GPIOData: 10, T1: F401, T2: F401, T3: F401, Bits: AA, BitCount: 08) 00

1.5.4 API RF

1.5.4.1 SearchTag

Command:	[0500][Byte: *MaxIDBytes*]
Response:	[00][Bool: *Result*][Byte: *TagType*][Byte: *IDBitCount*][Byte Array(Var): *ID*]
Example Command: Response:	 050010 (MaxIDBytes: 10) 000180200466CF4DC2 (Result: true, TagType: ISO14443A/MIFARE, IDBitCount: 32, ID: 66CF4DC2)

1.5.4.2 SetRFOff

Command:	[0501]
Response:	[00]
Example Command: Response:	 0501 00

1.5.4.3 SetTagTypes

Command:	[0502][UInt32: *TagTypesLF*][UInt32: *TagTypesHF*]
Response:	[00]
Example Command: Response:	 0502FFFFFFFFFFFFFFFF (TagTypesLF: FFFFFFFF, TagTypesHF: FFFFFFFF) 00

1.5.4.4 GetTagTypes

Command:	[0503]
Response:	[00][UInt32: *LFTagTypes*][UInt32: *HFTagTypes*]
Example Command: Response:	 0503 002FFE0700F7000000 (LFTagTypes: 523823, HFTagTypes: 247)

1.5.4.5 GetSupportedTagTypes

Command:	[0504]
Response:	[00][UInt32: *LFTagTypes*][UInt32: *HFTagTypes*]
Example Command: Response:	 0504 002FFE0700F7000000 (LFTagTypes: 523823, HFTagTypes: 247)

1.5.5 API TILF

1.5.5.1 TILF_SearchTag

Command:	[0600][Byte: *MaxIDBytes*]
Response:	[00][Bool: *Result*][Byte: *IDBitCount*][Byte Array(Var): *ID*]
Example Command: Response:	 060010 (MaxIDBytes: 10) 0001400800000000042E8653 (Result: true, IDBitCount: 64, ID: 00000000042E8653)

1.5.5.2 TILF_ChargeOnlyRead

Command:	[0601]
Response:	[00][Bool: *Result*][Byte Array(8): *Data*]
Example Command: Response:	 0601 000100000000042E8653 (Result: true, Data: 00000000042E8653)

1.5.5.3 TILF_ChargeOnlyReadLo

Command:	[0602]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example	
Command:	0602
Response:	000100007F7E7EFFFDFFFFFFFFFFFFFFFFFD
	(Result: true, ReadData: 00007F7E7EFFFDFFFFFFFFFFFFFFFFFD)

1.5.5.4 TILF_SPProgramPage

Command:	[0603][Byte Array(8): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example	
Command:	06030001020304050607
	(WriteData: 0001020304050607)
Response:	000100007ECA61742000000000DADF7E0000
	(Result: true, ReadData: 00007ECA61742000000000DADF7E0000)

1.5.5.5 TILF_SPProgramPageLo

Command:	[0604][Byte Array(10): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example	
Command:	060400010203040506070809
	(WriteData: 00010203040506070809)
Response:	000100007ECA61742000000000DADF7E0000
	(Result: true, ReadData: 00007ECA61742000000000DADF7E0000)

1.5.5.6 TILF_MPGeneralReadPage

Command:	[0605][Byte: *Address*]
Response:	[00][Bool: *Result*][Byte Array(8): *ReadData*]
Example	
Command:	060500
	(Address: 00)
Response:	000100000000042E8653
	(Result: true, ReadData: 00000000042E8653)

1.5.5.7 TILF_MPSelectiveReadPage

Command:	[0606][Byte: *Address*][Byte Array(3): *SelectiveAddress*]
Response:	[00][Bool: *Result*][Byte Array(8): *ReadData*]
Example	
Command:	060600000102
	(Address: 00, SelectiveAddress: 000102)
Response:	000100000000042E8653
	(Result: true, ReadData: 00000000042E8653)

1.5.5.8 TILF_MPProgramPage

Command:	[0607][Byte: *Address*][Byte Array(8): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(8): *ReadData*]
Example	
Command:	0607004469726563746F72
	(Address: 00, WriteData: 4469726563746F72)
Response:	000100000000042E8653
	(Result: true, ReadData: 00000000042E8653)

1.5.5.9 TILF_MPSelectiveProgramPage

Command:	[0608][Byte: *Address*][Byte Array(3): *SelectiveAddress*][Byte Array(8): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(8): *ReadData*]
Example	
Command:	0608000001024469726563746F72
	(Address: 00, SelectiveAddress: 000102, WriteData: 4469726563746F72)
Response:	000100000000042E8653
	(Result: true, ReadData: 00000000042E8653)

1.5.5.10 TILF_MPLockPage

Command:	[0609][Byte: *Address*]
Response:	[00][Bool: *Result*][Byte Array(8): *ReadData*]
Example	
Command:	060900
	(Address: 00)
Response:	0000
	(Result: fail, ReadData:)

1.5.5.11 TILF_MPSelectiveLockPage

Command:	[060A][Byte: *Address*][Byte Array(3): *SelectiveAddress*]
Response:	[00][Bool: *Result*][Byte Array(8): *ReadData*]
Example Command: Response:	 060A00000102 (Address: 00, SelectiveAddress: 000102) 0000 (Result: fail, ReadData:)

1.5.5.12 TILF_MPGeneralReadPageLo

Command:	[060B][Byte: *Address*]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example Command: Response:	 060B00 (Address: 00) 000100007ECA61742000000000DADF7E0000 (Result: true, ReadData: 00007ECA61742000000000DADF7E0000)

1.5.5.13 TILF_MPSelectiveReadPageLo

Command:	[060C][Byte: *Address*][Byte Array(3): *SelectiveAddress*]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example Command: Response:	 060C00000102 (Address: 00, SelectiveAddress: 000102) 000100007ECA61742000000000DADF7E0000 (Result: true, ReadData: 00007ECA61742000000000DADF7E0000)

1.5.5.14 TILF_MPProgramPageLo

Command:	[060D][Byte: *Address*][Byte Array(10): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example Command: Response:	 060D00536F6D6520746578742E (Address: 00, WriteData: 536F6D6520746578742E) 000100007ECA61742000000000DADF7E0000 (Result: true, ReadData: 00007ECA61742000000000DADF7E0000)

1.5.5.15 TILF_MPSelectiveProgramPageLo

Command:	[060E][Byte: *Address*][Byte Array(3): *SelectiveAddress*][Byte Array(10): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example Command: Response:	 060E00000102536F6D6520746578742E (Address: 00, SelectiveAddress: 000102, WriteData: 536F6D6520746578742E) 000100007ECA61742000000000DADF7E0000 (Result: true, ReadData: 00007ECA61742000000000DADF7E0000)

1.5.5.16 TILF_MPLockPageLo

Command:	[060F][Byte: *Address*]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example Command: Response:	 060F00 (Address: 00) 0000 (Result: fail, ReadData:)

1.5.5.17 TILF_MPSelectiveLockPageLo

Command:	[0610][Byte: *Address*][Byte Array(3): *SelectiveAddress*]
Response:	[00][Bool: *Result*][Byte Array(16): *ReadData*]
Example Command: Response:	 061000000102 (Address: 00, SelectiveAddress: 000102) 000100007FEFFFFFFFBFF7FFFAFFFFFFFFF7 (Result: true, ReadData: 00007FEFFFFFFFBFF7FFFAFFFFFFFFF7)

1.5.5.18 TILF_MUGeneralReadPage

Command:	[0611][Byte: *Address*]
Response:	[00][Bool: *Result*][Byte Array(7): *Data*]
Example Command: Response:	 061100 (Address: 00) 0000 (Result: fail, Data:)

1.5.5.19 TILF_MUSelectiveReadPage

Command:	[0612][Byte: *Address*][Byte: *SelectiveAddress*]
Response:	[00][Bool: *Result*][Byte Array(7): *Data*]
Example Command: Response:	 06120000 (Address: 00, SelectiveAddress: 00) 0000 (Result: fail, Data:)

1.5.5.20 TILF_MUSpecialReadPage

Command:	[0613][Byte: *Address*][Byte Array(5): *SpecialAddress1*][Byte Array(3): *SpecialAddress2*]
Response:	[00][Bool: *Result*][Byte Array(7): *Data*]
Example Command: Response:	 06130000010203040000102 (Address: 00, SpecialAddress1: 0001020304, SpecialAddress2: 000102) 0000 (Result: fail, Data:)

1.5.5.21 TILF_MUProgramPage

Command:	[0614][Byte: *Address*][Byte Array(5): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(7): *ReadData*]
Example Command: Response:	 06140048656C6C6F (Address: 00, WriteData: 48656C6C6F) 0000 (Result: fail, ReadData:)

1.5.5.22 TILF_MUSelectiveProgramPage

Command:	[0615][Byte: *Address*][Byte: *SelectiveAddress*][Byte Array(5): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(7): *ReadData*]
Example Command: Response:	 0615000048656C6C6F (Address: 00, SelectiveAddress: 00, WriteData: 48656C6C6F) 0000 (Result: fail, ReadData:)

1.5.5.23 TILF_MUSpecialProgramPage

Command:	[0616][Byte: *Address*][Byte Array(5): *SpecialAddress1*][Byte Array(3): *SpecialAddress2*][Byte Array(5): *WriteData*]
Response:	[00][Bool: *Result*][Byte Array(7): *ReadData*]
Example	
Command:	061600000102030400010248656C6C6F
	(Address: 00, SpecialAddress1: 0001020304, SpecialAddress2: 000102, WriteData: 48656C6C6F)
Response:	0000
	(Result: fail, ReadData:)

1.5.5.24 TILF_MULockPage

Command:	[0617][Byte: *Address*]
Response:	[00][Bool: *Result*][Byte Array(7): *ReadData*]
Example	
Command:	061700
	(Address: 00)
Response:	0000
	(Result: fail, ReadData:)

1.5.5.25 TILF_MUSelectiveLockPage

Command:	[0618][Byte: *Address*][Byte: *SelectiveAddress*]
Response:	[00][Bool: *Result*][Byte Array(7): *ReadData*]
Example	
Command:	06180000
	(Address: 00, SelectiveAddress: 00)
Response:	0000
	(Result: fail, ReadData:)

1.5.5.26 TILF_MUSpecialLockPage

Command:	[0619][Byte: *Address*][Byte Array(5): *SpecialAddress1*][Byte Array(3): *SpecialAddress2*]
Response:	[00][Bool: *Result*][Byte Array(7): *ReadData*]
Example	
Command:	0619000001020304000102
	(Address: 00, SpecialAddress1: 0001020304, SpecialAddress2: 000102)
Response:	0000
	(Result: fail, ReadData:)

1.5.6 API HITAG1S

1.5.6.1 Hitag1S_ReadPage

Command:	[0701][Byte: *PageAddress*]
Response:	[00][Bool: *Result*][Byte Array(4): *Data*]
Example	
Command:	070104
	(PageAddress: 04)
Response:	0001FF8CA64A
	(Result: true, Data: FF8CA64A)

1.5.6.2 Hitag1S_ReadBlock

Command:	[0702][Byte: *BlockAddress*]
Response:	[00][Bool: *Result*][Byte Array(Var): *Data*]
Example	
Command:	070204
	(BlockAddress: 04)
Response:	0001100001020398F8C802FFFFFFFFFFFFFFFF
	(Result: true, Data: 0001020398F8C802FFFFFFFFFFFFFFFF)

1.5.6.3 Hitag1S_WritePage

Command:	[0703][Byte: *PageAddress*][Byte Array(4): *Data*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 07030407040400 (PageAddress: 04, Data: 07040400) 0001 (Result: true)

1.5.6.4 Hitag1S_WriteBlock

Command:	[0704][Byte: *BlockAddress*][Byte Array(16): *Data*]
Response:	[00][Bool: *Result*][Byte: *BytesWritten*]
Example Command: Response:	 07040400000000000000000000000000000000 (BlockAddress: 04, Data: 00000000000000000000000000000000) 000110 (Result: true, BytesWritten: 16)

1.5.6.5 Hitag1S_Halt

Command:	[0705]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0705 0001 (Result: true)

1.5.7 API HITAG2

1.5.7.1 Hitag2_ReadPage

Command:	[0801][Byte: *PageAddress*]
Response:	[00][Bool: *Result*][Byte Array(4): *Data*]
Example	
Command:	080104
	(PageAddress: 04)
Response:	0001FF800000
	(Result: true, Data: FF800000)

1.5.7.2 Hitag2_WritePage

Command:	[0802][Byte: *PageAddress*][Byte Array(4): *Data*]
Response:	[00][Bool: *Result*]
Example	
Command:	080204FF800000
	(PageAddress: 04, Data: FF800000)
Response:	0001
	(Result: true)

1.5.7.3 Hitag2_Halt

Command:	[0803]
Response:	[00][Bool: *Result*]
Example	
Command:	0803
Response:	0001
	(Result: true)

1.5.7.4 Hitag2_SetPassword

Command:	[0804][Byte Array(4): *Password*]
Response:	[00]
Example	
Command:	080400010203
	(Password: 00010203)
Response:	00

1.5.8 API SM4X00

1.5.8.1 SM4X00_GenericRaw

Command:	[0900][Byte Array(Var): *TXData*][Byte: *MaxRXDataLength*][UInt16: *Timeout*]
Response:	[00][Bool: *Result*][Byte Array(Var): *RXData*]
Example Command: Response:	 090005040A00000040B80B (TXData: 040A000000, MaxRXDataLength: 40, Timeout: B80B) 00010D0A0000090105010018010301 00 (Result: true, RXData: 0A00000901050100180103 0100)

1.5.8.2 SM4X00_Generic

Command:	[0901][Byte Array(Var): *TXData*][Byte: *MaxRXDataLength*][UInt16: *Timeout*]
Response:	[00][Bool: *Result*][Byte Array(Var): *RXData*]
Example Command: Response:	 0901020A0040B80B (TXData: 0A00, MaxRXDataLength: 40, Timeout: B80B) 0001100F0A00000901050100180103 0100EB63 (Result: true, RXData: 0F0A000009010501001801030100EB63)

1.5.9 API I2C

1.5.9.1 I2CInit

Command:	[0A00][UInt16: *Mode*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0A000000 (Mode: 0000) 0001 (Result: true)

1.5.9.2 I2CDeInit

Command:	[0A01]
Response:	[00]
Example Command: Response:	 0A01 00

1.5.9.3 I2CMasterStart

Command:	[0A02]
Response:	[00]
Example Command: Response:	 0A02 00

1.5.9.4 I2CMasterStop

Command:	[0A03]
Response:	[00]
Example Command: Response:	 0A03 00

1.5.9.5 I2CMasterTransmitByte

Command:	[0A04][Byte: *Data*]
Response:	[00]
Example Command: Response:	 0A0400 (Data: 00) 00

1.5.9.6 I2CMasterReceiveByte

Command:	[0A05]
Response:	[00][Byte: *Data*]
Example Command: Response:	 0A05 0000 (Data: 0)

1.5.9.7 I2CMasterBeginWrite

Command:	[0A06][Byte: *Address*]
Response:	[00]
Example Command: Response:	 0A0630 (Address: 30) 00

1.5.9.8 I2CMasterBeginRead

Command:	[0A07][Byte: *Address*]
Response:	[00]
Example Command: Response:	 0A0730 (Address: 30) 00

1.5.9.9 I2CMasterSetAck

Command:	[0A08][Byte: *SetOn*]
Response:	[00]
Example Command: Response:	 0A0801 (SetOn: 01) 00

1.5.10 API MIFARECLASSIC

1.5.10.1 MifareClassic_Login

Command:	[0B00][Byte Array(6): *Key*][Byte: *KeyType*][Byte: *Sector*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0B00A0A1A2A3A4A50000 (Key: A0A1A2A3A4A5, KeyType: 00, Sector: 00) 0001 (Result: true)

1.5.10.2 MifareClassic_ReadBlock

Command:	[0B01][Byte: *Block*]
Response:	[00][Bool: *Result*][Byte Array(16): *Data*]
Example Command: Response:	 0B0102 (Block: 02) 000100000000000000000000000000000000 (Result: true, Data: 00000000000000000000000000000000)

1.5.10.3 MifareClassic_WriteBlock

Command:	[0B02][Byte: *Block*][Byte Array(16): *Data*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0B020200000000000000000000000000000000 (Block: 02, Data: 00000000000000000000000000000000) 0001 (Result: true)

1.5.10.4 MifareClassic_ReadValueBlock

Command:	[0B03][Byte: *Block*]
Response:	[00][Bool: *Result*][UInt32: *Value*]
Example Command: Response:	 0B0302 (Block: 02) 000101000000 (Result: true, Value: 1)

1.5.10.5 MifareClassic_WriteValueBlock

Command:	[0B04][Byte: *Block*][UInt32: *Value*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0B040201000000 (Block: 02, Value: 01000000) 0001 (Result: true)

1.5.10.6 MifareClassic_IncrementValueBlock

Command:	[0B05][Byte: *Block*][UInt32: *Value*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0B050201000000 (Block: 02, Value: 01000000) 0001 (Result: true)

1.5.10.7 MifareClassic_DecrementValueBlock

Command:	[0B06][Byte: *Block*][UInt32: *Value*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0B060201000000 (Block: 02, Value: 01000000) 0001 (Result: true)

1.5.10.8 MifareClassic_CopyValueBlock

Command:	[0B07][Byte: *SourceBlock*][Byte: *DestBlock*]
Response:	[00][Bool: *Result*]
Example	
Command:	0B07090A
	(SourceBlock: 09, DestBlock: 0A)
Response:	0001
	(Result: true)

1.5.11 API MIFAREULTRALIGHT

1.5.11.1 MifareUltralight_ReadPage

Command:	[0C00][Byte: *Page*]
Response:	[00][Bool: *Result*][Byte Array(16): *Data*]
Example	
Command:	0C0004
	(Page: 04)
Response:	000100010203147870672E636F6D3A636172
	(Result: true, Data: 00010203147870672E636F6D3A636172)

1.5.11.2 MifareUltralight_WritePage

Command:	[0C01][Byte: *Page*][Byte Array(4): *Data*]
Response:	[00][Bool: *Result*]
Example	
Command:	0C010400010203
	(Page: 04, Data: 00010203)
Response:	0001
	(Result: true)

1.5.11.3 MifareUltralightC_Authenticate

Command:	[0C02][Byte Array(16): *Key*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0C0249454D4B41455242214E4143554F5946 (Key: 49454D4B41455242214E4143554F5946) 0001 (Result: true)

1.5.11.4 MifareUltralightC_SAMAuthenticate

Command:	[0C03][Byte: *KeyNo*][Byte: *KeyVersion*][Byte Array(Var): *DIVInput*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0C03010000 (KeyNo: 01, KeyVersion: 00, DIVInput:) 0001 (Result: true)

1.5.11.5 MifareUltralightC_WriteKeyFromSAM

Command:	[0C04][Byte: *KeyNo*][Byte: *KeyVersion*][Byte Array(Var): *DIVInput*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0C04010000 (KeyNo: 01, KeyVersion: 00, DIVInput:) 0000 (Result: fail)

1.5.11.6 MifareUltralightEV1_FastRead

Command:	[0C05][Byte: *StartPage*][Byte: *NumberOfPages*]
Response:	[00][Bool: *Result*][Byte Array(Var): *Data*]
Example Command: Response:	 0C050401 (StartPage: 04, NumberOfPages: 01) 00010400000000 (Result: true, Data: 00000000)

1.5.11.7 MifareUltralightEV1_IncCounter

Command:	[0C06][Byte: *CounterAddr*][UInt32: *IncrValue*]
Response:	[00][Bool: *Result*]
Example	
Command:	0C060000000000
	(CounterAddr: 00, IncrValue: 00000000)
Response:	0001
	(Result: true)

1.5.11.8 MifareUltralightEV1_ReadCounter

Command:	[0C07][Byte: *CounterAddr*]
Response:	[00][Bool: *Result*][UInt32: *CounterValue*]
Example	
Command:	0C0700
	(CounterAddr: 00)
Response:	000102000000
	(Result: true, CounterValue: 2)

1.5.11.9 MifareUltralightEV1_ReadSig

Command:	[0C08]
Response:	[00][Bool: *Result*][Byte Array(32): *ECCSig*]
Example	
Command:	0C08
Response:	00013A4F2622AF2039E47F8AA1BF84C52EE949860DD07125BEF75EC4-17833B80C105
	(Result: true,
	ECCSig: 3A4F2622AF2039E47F8AA1BF84C52EE949860DD07125BEF75EC4-17833B80C105)

1.5.11.10 MifareUltralightEV1_GetVersion

Command:	[0C09]
Response:	[00][Bool: *Result*][Byte Array(8): *Version*]
Example Command: Response:	 0C09 000100040301010000E03 (Result: true, Version: 0004030101000E03)

1.5.11.11 MifareUltralightEV1_PwdAuth

Command:	[0C0A][Byte Array(4): *Password*][Byte Array(2): *PwdAck*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0C0AFFFFFFFF0000 (Password: FFFFFFFF, PwdAck: 0000) 0001 (Result: true)

1.5.11.12 MifareUltralightEV1_CheckTearingEvent

Command:	[0C0B][Byte: *CounterAddr*]
Response:	[00][Bool: *Result*][Byte: *ValidFlag*]
Example Command: Response:	 0C0B00 (CounterAddr: 00) 0001BD (Result: true, ValidFlag: 189)

1.5.12 API ISO15693

1.5.12.1 ISO15693_GenericCommand

Command:	[0D00][Byte: *Flags*][Byte: *Command*][Byte Array(Var): *Data*][Byte: *BufferSize*]
Response:	[00][Bool: *Result*][Byte Array(Var): *Data*]
Example	
Command:	0D001020010020
	(Flags: 10, Command: 20, Data: 00, BufferSize: 20)
Response:	00010400000000
	(Result: true, Data: 00000000)

1.5.12.2 ISO15693_GetSystemInformation

Command:	[0D01]
Response:	[00][Bool: *Result*][Byte Array(15): *SystemInfo*]
Example	
Command:	0D01
Response:	0001EF50781B06013C16E002000442000F
	(Result: true, SystemInfo: EF50781B06013C16E002000442000F)

1.5.12.3 ISO15693_GetSystemInformationExt

Command:	[0D02]
Response:	[00][Bool: *Result*][Byte Array(15): *SystemInfo*]
Example	
Command:	0D02
Response:	0001EF7D50C3ED084402E0000004000844
	(Result: true, SystemInfo: EF7D50C3ED084402E0000004000844)

1.5.12.4 ISO15693_GetTagTypeFromUID

Command:	[0D03][Byte Array(8): *UID*]
Response:	[00][Byte: *TagType*]
Example	
Command:	0D03E0163C01061B7850
	(UID: E0163C01061B7850)
Response:	00FF
	(TagType: 255)

1.5.12.5 ISO15693_GetTagTypeFromSystemInfo

Command:	[0D04][Byte Array(15): *SystemInfo*]
Response:	[00][Byte: *TagType*]
Example	
Command:	0D04EF7D50C3ED084402E0000004000844
	(SystemInfo: EF7D50C3ED084402E0000004000844)
Response:	0043
	(TagType: 67)

1.5.12.6 ISO15693_ReadSingleBlock

Command:	[0D05][UInt16: *BlockNumber*][Byte: *BufferSize*]
Response:	[00][Bool: *Result*][Byte Array(Var): *BlockData*]
Example	
Command:	0D050500FF
	(BlockNumber: 0500, BufferSize: FF)
Response:	00010400000000
	(Result: true, BlockData: 00000000)

1.5.12.7 ISO15693_ReadSingleBlockExt

Command:	[0D06][UInt16: *BlockNumber*][Byte: *BufferSize*]
Response:	[00][Bool: *Result*][Byte Array(Var): *BlockData*]
Example	
Command:	0D060000FF
	(BlockNumber: 0000, BufferSize: FF)
Response:	00010401020304
	(Result: true, BlockData: 01020304)

1.5.12.8 ISO15693_WriteSingleBlock

Command:	[0D07][UInt16: *BlockNumber*][Byte Array(Var): *BlockData*]
Response:	[00][Bool: *Result*]
Example	
Command:	0D0705000411223344
	(BlockNumber: 0500, BlockData: 11223344)
Response:	0001
	(Result: true)

1.5.12.9 ISO15693_WriteSingleBlockExt

Command:	[0D08][UInt16: *BlockNumber*][Byte Array(Var): *BlockData*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0D08000004426C612E (BlockNumber: 0000, BlockData: 426C612E) 0001 (Result: true)

1.5.13 API CRYPTO

1.5.13.1 Crypto_Init

Command:	[0E00][Byte: *CryptoEnv*][Byte: *CryptoMode*][Byte Array(Var): *Key*]
Response:	[00]
Example Command: Response:	 0E0000001000000000000000000000000000000000 (CryptoEnv: 00, CryptoMode: 00, Key: 00000000000000000000000000000000) 00

1.5.13.2 Encrypt

Command:	[0E01][Byte: *CryptoEnv*][Byte Array(Var): *PlainBlock*]
Response:	[00][Byte Array(Var): *CipheredBlock*]
Example Command: Response:	 0E010010800000000000000000000000000000000 (CryptoEnv: 00, PlainBlock: 80000000000000000000000000000000) 00103AD78E726C1EC02B7EBFE92B23D9EC34 (CipheredBlock: 3AD78E726C1EC02B7EBFE92B23D9EC34)

1.5.13.3 Decrypt

Command:	[0E02][Byte: *CryptoEnv*][Byte Array(Var): *CipheredBlock*]
Response:	[00][Byte Array(Var): *PlainBlock*]
Example Command: Response:	 0E0200103AD78E726C1EC02B7EBFE92B23D9EC34 (CryptoEnv: 00, CipheredBlock: 3AD78E726C1EC02B7EBFE92B23D9EC34) 001080000000000000000000000000000000 (PlainBlock: 80000000000000000000000000000000)

1.5.13.4 CBC_ResetInitVector

Command:	[0E03][Byte: *CryptoEnv*]
Response:	[00]
Example Command: Response:	 0E0300 (CryptoEnv: 00) 00

1.5.14 API DESFIRE

1.5.14.1 DESFire_GetApplicationIDs

Command:	[0F00][Byte: *CryptoEnv*][Byte: *MaxAIDCnt*]
Response:	[00][Bool: *Result*][variable number of UInt32: *AIDs*]
Example Command: Response:	 0F00001C (CryptoEnv: 00, MaxAIDCnt: 1C) 00010133221100 (Result: true, AIDs: 00112233)

1.5.14.2 DESFire_CreateApplication

Command:	[0F01][Byte: *CryptoEnv*][UInt32: *AID*][4 Bit: *ChangeKeyAccessRights*][1 Bit: *ConfigurationChangeable*][1 Bit: *FreeCreateDelete*][1 Bit: *FreeDirectoryList*][1 Bit: *AllowChangeMasterKey*][UInt32: *NumberOfKeys*][UInt32: *KeyType*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F0100907856000F0100000000000000
	(CryptoEnv: 00, AID: 90785600, ChangeKeyAccessRights: 15, ConfigurationChangeable: 1, FreeCreateDelete: 1, FreeDirectoryList: 1, AllowChangeMasterKey: 1, NumberOfKeys: 01000000, KeyType: 00000000)
Response:	0001
	(Result: true)

1.5.14.3 DESFire_DeleteApplication

Command:	[0F02][Byte: *CryptoEnv*][UInt32: *AID*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F020090785600
	(CryptoEnv: 00, AID: 90785600)
Response:	0001
	(Result: true)

1.5.14.4 DESFire_SelectApplication

Command:	[0F03][Byte: *CryptoEnv*][UInt32: *AID*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F030033221100
	(CryptoEnv: 00, AID: 33221100)
Response:	0001
	(Result: true)

1.5.14.5 DESFire_Authenticate

Command:	[0F04][Byte: *CryptoEnv*][Byte: *KeyNoTag*][Byte Array(Var): *Key*][Byte: *KeyType*][Byte: *Mode*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F04000010000000000000000000000000000000000000
	(CryptoEnv: 00, KeyNoTag: 00, Key: 00000000000000000000000000000000, KeyType: 00, Mode: 00)
Response:	0001
	(Result: true)

1.5.14.6 DESFire_GetKeySettings

Command:	[0F05][Byte: *CryptoEnv*]
Response:	[00][Bool: *Result*][Byte: *KeySettings*][UInt32: *NumberOfKeys*][UInt32: *KeyType*]
Example	
Command:	0F0500
	(CryptoEnv: 00)
Response:	00010F0100000000000000
	(Result: true, KeySettings: 15, NumberOfKeys: 1, KeyType: 0)

1.5.14.7 DESFire_GetFileIDs

Command:	[0F06][Byte: *CryptoEnv*][Byte: *MaxFileIDCount*]
Response:	[00][Bool: *Result*][variable number of Bytes: *FileIDList*]
Example	
Command:	0F0600FF
	(CryptoEnv: 00, MaxFileIDCount: FF)
Response:	00010400010203
	(Result: true, FileIDList: 00, 01, 02, 03)

1.5.14.8 DESFire_GetFileSettings

Command:	[0F07][Byte: *CryptoEnv*][Byte: *FileNo*]
Response:	[00][Bool: *Result*][Byte Array(20): *FileSettings*]
Example	
Command:	0F070000
	(CryptoEnv: 00, FileNo: 00)
Response:	00010000EEEE200000000000000000000000036322F50
	(Result: true, FileSettings: 0000EEEE200000000000000000000000036322F50)

1.5.14.9 DESFire_ReadData

Command:	[0F08][Byte: *CryptoEnv*][Byte: *FileNo*][UInt16: *Offset*][Byte: *Length*][Byte: *CommSet*]
Response:	[00][Bool: *Result*][Byte Array(Var): *Data*]
Example	
Command:	0F08000000000300
	(CryptoEnv: 00, FileNo: 00, Offset: 0000, Length: 03, CommSet: 00)
Response:	000103001122
	(Result: true, Data: 001122)

1.5.14.10 DESFire_WriteData

Command:	[0F09][Byte: *CryptoEnv*][Byte: *FileNo*][UInt16: *Offset*][Byte Array(Var): *Data*][Byte: *CommSet*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F09000000000300112200
	(CryptoEnv: 00, FileNo: 00, Offset: 0000, Data: 001122, CommSet: 00)
Response:	0001
	(Result: true)

1.5.14.11 DESFire_GetValue

Command:	[0F0A][Byte: *CryptoEnv*][Byte: *FileNo*][Byte: *CommSet*]
Response:	[00][Bool: *Result*][UInt32: *Value*]
Example Command: Response:	 0F0A000000 (CryptoEnv: 00, FileNo: 00, CommSet: 00) 000100000000 (Result: true, Value: 0)

1.5.14.12 DESFire_Credit

Command:	[0F0B][Byte: *CryptoEnv*][Byte: *FileNo*][UInt32: *Value*][Byte: *CommSet*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0F0B00040000000000 (CryptoEnv: 00, FileNo: 04, Value: 00000000, CommSet: 00) 0001 (Result: true)

1.5.14.13 DESFire_Debit

Command:	[0F0C][Byte: *CryptoEnv*][Byte: *FileNo*][UInt32: *Value*][Byte: *CommSet*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0F0C00040000000000 (CryptoEnv: 00, FileNo: 04, Value: 00000000, CommSet: 00) 0001 (Result: true)

1.5.14.14 DESFire_LimitedCredit

Command:	[0F0D][Byte: *CryptoEnv*][Byte: *FileNo*][UInt32: *Value*][Byte: *CommSet*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0F0D00040000000000 (CryptoEnv: 00, FileNo: 04, Value: 00000000, CommSet: 00) 0001 (Result: true)

1.5.14.15 DESFire_FreeMem

Command:	[0F0E][Byte: *CryptoEnv*]
Response:	[00][Bool: *Result*][UInt16: *FreeMemory*]
Example Command:	 0F0E00 (CryptoEnv: 00)
Response:	00016011 (Result: true, FreeMemory: 4448)

1.5.14.16 DESFire_FormatTag

Command:	[0F0F][Byte: *CryptoEnv*]
Response:	[00][Bool: *Result*]
Example Command:	 0F0F00 (CryptoEnv: 00)
Response:	0001 (Result: true)

1.5.14.17 DESFire_CreateDataFile

Command:	[0F10][Byte: *CryptoEnv*][Byte: *FileNo*][Byte: *FileType*][Byte: *CommSet*][UInt16: *AccessRights*][UInt32: *FileSize*][:]
Response:	[00][Bool: *Result*]
Example Command:	 0F1000050000EEEE0F000000000000000000000000000000 (CryptoEnv: 00, FileNo: 05, FileType: 00, CommSet: 00, AccessRights: EEEE, FileSize: 0F000000, 000000000000000000000000)
Response:	0001 (Result: true)

1.5.14.18 DESFire_CreateValueFile

Command:	[0F11][Byte: *CryptoEnv*][Byte: *FileNo*][Byte: *FileType*][Byte: *CommSet*][UInt16: *AccessRights*][UInt32: *LowerLimit*][UInt32: *UpperLimit*][UInt32: *LimitedCreditValue*][Bool: *LimitedCreditEnabled*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F1100040200EEEE000000000F0000000F00000001000000
	(CryptoEnv: 00, FileNo: 04, FileType: 02, CommSet: 00, AccessRights: EEEE, LowerLimit: 00000000, UpperLimit: 0F000000, LimitedCreditValue: 0F000000,)
Response:	0001
	(Result: true)

1.5.14.19 DESFire_GetVersion

Command:	[0F12][Byte: *CryptoEnv*]
Response:	[00][Bool: *Result*][Byte Array(34): *Version*]
Example	
Command:	0F1200
	(CryptoEnv: 00)
Response:	000104010101000010000005040101010300100000050000000000000 0BA14D0A7103110
	(Result: true,
	Version: 040101010000100000050401010103001000000500000000000000BA14D0A7103110)

1.5.14.20 DESFire_DeleteFile

Command:	[0F13][Byte: *CryptoEnv*][Byte: *FileNo*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F130005
	(CryptoEnv: 00, FileNo: 05)
Response:	0001
	(Result: true)

1.5.14.21 DESFire_CommitTransaction

Command:	[0F14][Byte: *CryptoEnv*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F1400
	(CryptoEnv: 00)
Response:	0001
	(Result: true)

1.5.14.22 DESFire_AbortTransaction

Command:	[0F15][Byte: *CryptoEnv*]
Response:	[00][Bool: *Result*]
Example	
Command:	0F1500
	(CryptoEnv: 00)
Response:	0001
	(Result: true)

1.5.14.23 DESFire_GetUID

Command:	[0F16][Byte: *CryptoEnv*][Byte: *BufferSize*]
Response:	[00][Bool: *Result*][Byte Array(Var): *UID*]
Example	
Command:	0F1600FF
	(CryptoEnv: 00, BufferSize: FF)
Response:	000107045243523D2480
	(Result: true, UID: 045243523D2480)

1.5.14.24 DESFire_GetKeyVersion

Command:	[0F17][Byte: *CryptoEnv*][Byte: *KeyNo*]
Response:	[00][Bool: *Result*][Byte Array(1): *KeyVersion*]
Example	
Command:	0F170000
	(CryptoEnv: 00, KeyNo: 00)
Response:	0001FF
	(Result: true, KeyVersion: FF)

1.5.14.25 DESFire_ChangeKeySettings

Command:	[0F18][Byte: *CryptoEnv*][4 Bit: *ChangeKeyAccessRights*][1 Bit: *ConfigurationChangeable*][1 Bit: *FreeCreateDelete*][1 Bit: *FreeDirectoryList*][1 Bit: *AllowChangeMasterKey*][UInt32: *NumberOfKeys*][UInt32: *KeyType*]
Response:	[00][Bool: *Result*]
Example Command:	 0F18000F0000000000000000 (CryptoEnv: 00, ChangeKeyAccessRights: 15, ConfigurationChangeable: 1, FreeCreateDelete: 1, FreeDirectoryList: 1, AllowChangeMasterKey: 1, NumberOfKeys: 00000000, KeyType: 00000000)
Response:	0001 (Result: true)

1.5.14.26 DESFire_ChangeKey

Command:	[0F19][Byte: *CryptoEnv*][Byte: *KeyNo*][Byte Array(Var): *OldKey*][Byte Array(Var): *NewKey*][Byte: *KeyVersion*][4 Bit: *ChangeKeyAccessRights*][1 Bit: *ConfigurationChangeable*][1 Bit: *FreeCreateDelete*][1 Bit: *FreeDirectoryList*][1 Bit: *AllowChangeMasterKey*][UInt32: *NumberOfKeys*][UInt32: *KeyType*]
Response:	[00][Bool: *Result*]
Example Command:	 0F190000100000000000000000000000000000000000100000000000000000000000000-0000000000F0100000000000000 (CryptoEnv: 00, KeyNo: 00, OldKey: 00000000000000000000000000000000, NewKey: 00000000000000000000000000000000, KeyVersion: 00, ChangeKeyAccessRights: 15, ConfigurationChangeable: 1, FreeCreateDelete: 1, FreeDirectoryList: 1, AllowChangeMasterKey: 1, NumberOfKeys: 01000000, KeyType: 00000000)
Response:	0001 (Result: true)

1.5.14.27 DESFire_ChangeFileSettings

Command:	[0F1A][Byte: *CryptoEnv*][Byte: *FileNo*][Byte: *NewCommSet*][UInt16: *OldAccessRights*][UInt16: *NewAccessRights*]
Response:	[00][Bool: *Result*]
Example Command:	 0F1A000000EEEEEEEE (CryptoEnv: 00, FileNo: 00, NewCommSet: 00, OldAccessRights: EEEE, NewAccessRights: EEEE)
Response:	0001 (Result: true)

1.5.14.28 DESFire_DisableFormatCard

Command:	[0F1B][Byte: *CryptoEnv*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0F1B00 (CryptoEnv: 00) 0001 (Result: true)

1.5.14.29 DESFire_EnableRandomID

Command:	[0F1C][Byte: *CryptoEnv*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0F1C00 (CryptoEnv: 00) 0001 (Result: true)

1.5.14.30 DESFire_SetDefaultKey

Command:	[0F1D][Byte: *CryptoEnv*][Byte Array(Var): *Key*][Byte: *KeyVersion*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0F1D001000000000000000000000000000000000FF (CryptoEnv: 00, Key: 00000000000000000000000000000000, KeyVersion: FF) 0001 (Result: true)

1.5.14.31 DESFire_SetATS

Command:	[0F1E][Byte: *CryptoEnv*][Byte Array(Var): *ATS*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 0F1E0008087577810280CAFE (CryptoEnv: 00, ATS: 087577810280CAFE) 0001 (Result: true)

1.5.15 API ISO7816

1.5.15.1 ISO7816_GetSlotStatus

Command:	[1000][Byte: *Channel*]
Response:	[00][Bool: *Result*][Byte Array(3): *SlotStatus*]
Example Command: Response:	 100020 (Channel: 20) 0001000000 (Result: true, SlotStatus: 000000)

1.5.15.2 ISO7816_IccPowerOn

Command:	[1001][Byte: *Channel*][Byte: *MaxATRByteCnt*][Byte: *bPowerSelect*]
Response:	[00][Bool: *Result*][Byte Array(Var): *ATR*][Byte: *bStatus*][Byte: *bError*]
Example Command: Response:	 100120FF00 (Channel: 20, MaxATRByteCnt: FF, bPowerSelect: 00) 00010F3B959680B1FE551FC747726163651300000 (Result: true, ATR: 3B959680B1FE551FC7477261636513, bStatus: 0, bError: 0)

1.5.15.3 ISO7816_IccPowerOff

Command:	[1002][Byte: *Channel*]
Response:	[00][Bool: *Result*][Byte Array(3): *SlotStatus*]
Example Command: Response:	 100220 (Channel: 20) 0001010000 (Result: true, SlotStatus: 010000)

1.5.15.4 ISO7816_SetCommSettings

Command:	[1003][Byte: *Channel*][Byte Array(13): *CommSettings*]
Response:	[00][Bool: *Result*]
Example	
Command:	1003200100740101000000FF5500FE00
	(Channel: 20, CommSettings: 0100740101000000FF5500FE00)
Response:	0001
	(Result: true)

1.5.15.5 ISO7816_Transceive

Command:	[1004][Byte: *Channel*][Byte Array(Var), 2 LB: *TX*][Byte: *MaxRXByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var), 2 LB: *RX*]
Example	
Command:	100420050000C10120E0FF
	(Channel: 20, TX: 00C10120E0, MaxRXByteCnt: FF)
Response:	000102006E00
	(Result: true, RX: 6E00)

1.5.15.6 ISO7816_ExchangeAPDU

Command:	[1005][Byte: *Channel*][Byte Array(9): *Header*][Byte Array(Var), 2 LB: *TXData*][UInt16: *MaxRXByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var), 2 LB: *RXData*][UInt16: *StatusWord*]
Example	
Command:	10052000A40004020000000102003F008000
	(Channel: 20, Header: 00A400040200000001, TXData: 3F00, MaxRXByteCnt: 8000)
Response:	00010000006E
	(Result: true, RXData: , StatusWord: 28160)

1.5.15.7 ISO7816_T0_TPDU

Command:	[1006][Byte: *Channel*][Byte Array(5): *Header*][Byte Array(Var), 2 LB: *TXData*][UInt16: *MaxRXByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var), 2 LB: *RXData*][UInt16: *StatusWord*]
Example	
Command:	10062000A400040202003F008000
	(Channel: 20, Header: 00A4000402, TXData: 3F00, MaxRXByteCnt: 8000)
Response:	00010000006E
	(Result: true, RXData: , StatusWord: 28160)

1.5.16 API ICLASS

1.5.16.1 ICLASS_GetPACBits

Command:	[1100][Byte: *MaxPACBytes*]
Response:	[00][Bool: *Result*][Byte: *PACBitCnt*][Byte Array(Var): *PAC*]
Example	
Command:	1100FF
	(MaxPACBytes: FF)
Response:	00011A0405000980
	(Result: true, PACBitCnt: 26, PAC: 00140026)

1.5.17 API ISO14443

1.5.17.1 ISO14443A_GetATS

Command:	[1200][Byte: *MaxATSByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var): *ATS*]
Example	
Command:	120020
	(MaxATSByteCnt: 20)
Response:	000106067577810280
	(Result: true, ATS: 067577810280)

1.5.17.2 ISO14443B_GetATQB

Command:	[1201][Byte: *MaxATQBByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var): *ATQB*]
Example Command: Response:	 1201FF (MaxATQBByteCnt: FF) 00010C5077FB135400000000B37171 (Result: true, ATQB: 5077FB135400000000B37171)

1.5.17.3 ISO14443_4_CheckPresence

Command:	[1202]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1202 0001 (Result: true)

1.5.17.4 ISO14443_4_TDX

Command:	[1203][Byte Array(Var): *TX*][Byte: *MaxRXByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var): *RX*]
Example Command: Response:	 1203016020 (TX: 60, MaxRXByteCnt: 20) 0001026F00 (Result: true, RX: 6F00)

1.5.17.5 ISO14443A_GetATQA

Command:	[1204]
Response:	[00][Bool: *Result*][Byte Array(2): *ATQA*]
Example Command: Response:	 1204 00010403 (Result: true, ATQA: 0403)

1.5.17.6 ISO14443A_GetSAK

Command:	[1205]
Response:	[00][Bool: *Result*][Byte Array(1): *SAK*]
Example	
Command:	1205
Response:	000120
	(Result: true, SAK: 20)

1.5.17.7 ISO14443B_GetAnswerToATTRIB

Command:	[1206][Byte: *MaxAnswerToATTRIBByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var): *AnswerToATTRIB*]
Example	
Command:	1206FF
	(MaxAnswerToATTRIBByteCnt: FF)
Response:	00010100
	(Result: true, AnswerToATTRIB: 00)

1.5.17.8 ISO14443_3_TDX

Command:	[1207][Byte Array(Var): *TX*][Byte: *MaxRXByteCnt*][UInt16: *Timeout*]
Response:	[00][Bool: *Result*][Byte Array(Var): *RX*]
Example	
Command:	1207041A004176FFFF00
	(TX: 1A004176, MaxRXByteCnt: FF, Timeout: FF00)
Response:	00010104
	(Result: true, RX: 04)

1.5.17.9 ISO14443A_SearchMultiTag

Command:	[1208][Byte: *MaxUIDListByteCnt*]
Response:	[00][Bool: *Result*][Byte: *UIDCnt*][variable number of Bytes: *UIDList*]
Example	
Command:	1208FF
	(MaxUIDListByteCnt: FF)
Response:	000103180704D7A79A97378007042DA79A973780070450A79A973780
	(Result: true, UIDCnt: 3, UIDList: 04D7A79A973780, 042DA79A973780, 0450A79A973780)

1.5.17.10 ISO14443A_SelectTag

Command:	[1209][Byte Array(Var): *UID*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 12090704D7A79A973780 (UID: 04D7A79A973780) 0001 (Result: true)

1.5.18 API AT55

1.5.18.1 AT55_Begin

Command:	[1500]
Response:	[00]
Example Command: Response:	 1500 00

1.5.18.2 AT55_ReadBlock

Command:	[1501][Byte: *Address*]
Response:	[00][Bool: *Result*][Byte Array(4): *Data*]
Example Command: Response:	 150100 (Address: 00) 0001F0148040 (Result: true, Data: F0148040)

1.5.18.3 AT55_ReadBlockProtected

Command:	[1502][Byte: *Address*][Byte Array(4): *Password*]
Response:	[00][Bool: *Result*][Byte Array(4): *Data*]
Example Command: Response:	 15020000000000 (Address: 00, Password: 00000000) 0001B8A31C02 (Result: true, Data: B8A31C02)

1.5.18.4 AT55_WriteBlock

Command:	[1503][Byte: *Address*][Byte Array(4): *Data*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 15030000010203 (Address: 00, Data: 00010203) 0001 (Result: true)

1.5.18.5 AT55_WriteBlockProtected

Command:	[1504][Byte: *Address*][Byte Array(4): *Data*][Byte Array(4): *Password*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1504000001020300000000 (Address: 00, Data: 00010203, Password: 00000000) 0001 (Result: true)

1.5.18.6 AT55_WriteBlockAndLock

Command:	[1505][Byte: *Address*][Byte Array(4): *Data*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 15050000010203 (Address: 00, Data: 00010203) 0001 (Result: true)

1.5.18.7 AT55_WriteBlockProtectedAndLock

Command:	[1506][Byte: *Address*][Byte Array(4): *Data*][Byte Array(4): *Password*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1506000001020300000000 (Address: 00, Data: 00010203, Password: 00000000) 0001 (Result: true)

1.5.19 API NFCSNEP

1.5.19.1 SNEP_Init

Command:	[1800]
Response:	[00][Bool: *Result*]
Example	
Command:	1800
Response:	0001
	(Result: true)

1.5.19.2 SNEP_GetConnectionState

Command:	[1801]
Response:	[00][Byte: *ConnectionState*]
Example	
Command:	1801
Response:	0002
	(ConnectionState: 2)

1.5.19.3 SNEP_GetFragmentByteCount

Command:	[1802][Byte: *Direction*]
Response:	[00][UInt16: *ByteCount*]
Example	
Command:	180201
	(Direction: 01)
Response:	000000
	(ByteCount: 0)

1.5.19.4 SNEP_BeginMessage

Command:	[1803][UInt32: *MsgByteCnt*]
Response:	[00][Bool: *Result*]
Example	
Command:	1803FF000000
	(MsgByteCnt: FF000000)
Response:	0001
	(Result: true)

1.5.19.5 SNEP_SendMessageFragment

Command:	[1804][Byte Array(Var), 2 LB: *MsgFrag*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 18041500D101115501656C617465632D726669642E636F6D2F (MsgFrag: D101115501656C617465632D726669642E636F6D2F) 0001 (Result: true)

1.5.19.6 SNEP_TestMessage

Command:	[1805]
Response:	[00][Bool: *Result*][UInt32: *MsgByteCnt*]
Example Command: Response:	 1805 0000 (Result: fail, MsgByteCnt:)

1.5.19.7 SNEP_ReceiveMessageFragment

Command:	[1806][UInt16: *FragByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var), 2 LB: *MsgFrag*]
Example Command: Response:	 1806FF00 (FragByteCnt: FF00) 0000 (Result: fail, MsgFrag:)

1.5.19.8 SNEP_RequestMessage

Command:	[1807][UInt32: *MsgByteCnt*][UInt32: *AcceptableLength*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1807FF000000FF000000 (MsgByteCnt: FF000000, AcceptableLength: FF000000) 0001 (Result: true)

1.5.20 API EM4150

1.5.20.1 EM4150_Login

Command:	[1900][Byte Array(4): *Password*]
Response:	[00][Bool: *Result*]
Example	
Command:	190000000000
	(Password: 00000000)
Response:	0001
	(Result: true)

1.5.20.2 EM4150_ReadWord

Command:	[1901][Byte: *Address*]
Response:	[00][Bool: *Result*][Byte Array(4): *Word*]
Example	
Command:	190101
	(Address: 01)
Response:	000100010203
	(Result: true, Word: 00010203)

1.5.20.3 EM4150_WriteWord

Command:	[1902][Byte: *Address*][Byte Array(4): *Word*]
Response:	[00][Bool: *Result*]
Example	
Command:	19020100010203
	(Address: 01, Word: 00010203)
Response:	0001
	(Result: true)

1.5.20.4 EM4150_WritePassword

Command:	[1903][Byte Array(4): *ActualPassword*][Byte Array(4): *NewPassword*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 19030000000001010101 (ActualPassword: 00000000, NewPassword: 01010101) 0001 (Result: true)

1.5.20.5 EM4150_GetTagInfo

Command:	[1904]
Response:	[00][UInt32: *TagInfo*]
Example Command: Response:	 1904 0001000000 (TagInfo: 1)

1.5.21 API FILESYS

1.5.21.1 FSMount

Command:	[1A00][Byte: *StorageID*][UInt32: *Mode*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1A000102000000 (StorageID: 01, Mode: 02000000) 0001 (Result: true)

1.5.21.2 FSFormat

Command:	[1A01][Byte: *StorageID*][UInt32: *MagicValue*]
Response:	[00][Bool: *Result*]
Example	
Command:	1A0101446F4974
	(StorageID: 01, MagicValue: 446F4974)
Response:	0001
	(Result: true)

1.5.21.3 FSOpen

Command:	[1A02][Byte: *FileEnv*][Byte: *StorageID*][UInt32: *FileID*][Byte: *Mode*]
Response:	[00][Bool: *Result*]
Example	
Command:	1A0200013322110000
	(FileEnv: 00, StorageID: 01, FileID: 33221100, Mode: 00)
Response:	0001
	(Result: true)

1.5.21.4 FSClose

Command:	[1A03][Byte: *FileEnv*]
Response:	[00][Bool: *Result*]
Example	
Command:	1A0300
	(FileEnv: 00)
Response:	0001
	(Result: true)

1.5.21.5 FSCloseAll

Command:	[1A04]
Response:	[00]
Example	
Command:	1A04
Response:	00

1.5.21.6 FSSeek

Command:	[1A05][Byte: *FileEnv*][Byte: *Origin*][UInt32: *Pos*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1A05000001000000 (FileEnv: 00, Origin: 00, Pos: 01000000) 0001 (Result: true)

1.5.21.7 FSTell

Command:	[1A06][Byte: *FileEnv*][Byte: *Origin*]
Response:	[00][Bool: *Result*][UInt32: *Pos*]
Example Command: Response:	 1A060000 (FileEnv: 00, Origin: 00) 000101000000 (Result: true, Pos: 1)

1.5.21.8 FSReadBytes

Command:	[1A07][Byte: *FileEnv*][UInt16: *ByteCount*]
Response:	[00][Bool: *Result*][Byte Array(Var), 2 LB: *Data*]
Example Command: Response:	 1A07001E00 (FileEnv: 00, ByteCount: 1E00) 000107004D792064617461 (Result: true, Data: 4D792064617461)

1.5.21.9 FSWriteBytes

Command:	[1A08][Byte: *FileEnv*][Byte Array(Var), 2 LB: *Data*]
Response:	[00][Bool: *Result*][UInt16: *BytesWritten*]
Example Command: Response:	 1A080007004D792064617461 (FileEnv: 00, Data: 4D792064617461) 00010700 (Result: true, BytesWritten: 7)

1.5.21.10 FSFindFirst

Command:	[1A09][Byte: *StorageID*]
Response:	[00][Bool: *Result*][Byte Array(8): *FileInfo*]
Example Command: Response:	 1A0901 (StorageID: 01) 00013322110002000000 (Result: true, FileInfo: 3322110002000000)

1.5.21.11 FSFindNext

Command:	[1A0A]
Response:	[00][Bool: *Result*][Byte Array(8): *FileInfo*]
Example Command: Response:	 1A0A 00013422110002000000 (Result: true, FileInfo: 3422110002000000)

1.5.21.12 FSDelete

Command:	[1A0B][Byte: *StorageID*][UInt32: *FileID*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1A0B0133221100 (StorageID: 01, FileID: 33221100) 0001 (Result: true)

1.5.21.13 FSRename

Command:	[1A0C][Byte: *StorageID*][UInt32: *OldFileID*][UInt32: *NewFileID*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1A0C017766554433221100 (StorageID: 01, OldFileID: 77665544, NewFileID: 33221100) 0001 (Result: true)

1.5.21.14 FSGetStorageInfo

Command:	[1A0D][Byte: *StorageID*]
Response:	[00][Bool: *Result*][Byte Array(9): *StorageInfo*]
Example Command: Response:	 1A0D01 (StorageID: 01) 000101204B0000004B0000 (Result: true, StorageInfo: 01204B0000004B0000)

1.5.22 API MIFAREPLUS

1.5.22.1 MFP_WritePerso

Command:	[1B00][UInt16: *BlockNr*][Byte Array(16): *Data*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1B00004000000000000000000000000000000000 (BlockNr: 0040, Data: 00000000000000000000000000000000) 0001 (Result: true)

1.5.22.2 MFP_CommitPerso

Command:	[1B01]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1B01 0001 (Result: true)

1.5.22.3 MFP_Authenticate

Command:	[1B02][Byte: *CryptoEnv*][UInt16: *KeyBNr*][Byte Array(16): *Key*]
Response:	[00][Bool: *Result*]
Example	
Command:	1B0200004000000000000000000000000000000000
	(CryptoEnv: 00, KeyBNr: 0040, Key: 00000000000000000000000000000000)
Response:	0001
	(Result: true)

1.5.22.4 MFP_ReadBlock

Command:	[1B03][Byte: *CryptoEnv*][UInt16: *Block*]
Response:	[00][Bool: *Result*][Byte Array(16): *Data*]
Example	
Command:	1B03000400
	(CryptoEnv: 00, Block: 0400)
Response:	000101020304050607080900010203040506
	(Result: true, Data: 01020304050607080900010203040506)

1.5.22.5 MFP_WriteBlock

Command:	[1B04][Byte: *CryptoEnv*][UInt16: *Block*][Byte Array(16): *Data*]
Response:	[00][Bool: *Result*]
Example	
Command:	1B0400040001020304050607080900010203040506
	(CryptoEnv: 00, Block: 0400, Data: 01020304050607080900010203040506)
Response:	0001
	(Result: true)

1.5.22.6 MFP_ReadValueBlock

Command:	[1B05][Byte: *CryptoEnv*][UInt16: *Block*]
Response:	[00][Bool: *Result*][UInt32: *Value*]
Example	
Command:	1B05000400
	(CryptoEnv: 00, Block: 0400)
Response:	000100000000
	(Result: true, Value: 0)

1.5.22.7 MFP_WriteValueBlock

Command:	[1B06][Byte: *CryptoEnv*][UInt16: *Block*][UInt32: *Value*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1B0600040000000000 (CryptoEnv: 00, Block: 0400, Value: 00000000) 0001 (Result: true)

1.5.22.8 MFP_IncrementValueBlock

Command:	[1B07][Byte: *CryptoEnv*][UInt16: *Block*][UInt32: *Value*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1B0700040001000000 (CryptoEnv: 00, Block: 0400, Value: 01000000) 0001 (Result: true)

1.5.22.9 MFP_DecrementValueBlock

Command:	[1B08][Byte: *CryptoEnv*][UInt16: *Block*][UInt32: *Value*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1B0800040001000000 (CryptoEnv: 00, Block: 0400, Value: 01000000) 0001 (Result: true)

1.5.22.10 MFP_CopyValueBlock

Command:	[1B09][Byte: *CryptoEnv*][UInt16: *SourceBlock*][UInt16: *DestBlock*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1B090004000500 (CryptoEnv: 00, SourceBlock: 0400, DestBlock: 0500) 0001 (Result: true)

1.5.23 API ADC

1.5.23.1 ADCInitChannel

Command:	[1C00][Byte: *ADCChannel*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1C0001 (ADCChannel: 01) 0001 (Result: true)

1.5.23.2 ADCGetConversionValue

Command:	[1C01][Byte: *ADCChannel*]
Response:	[00][UInt16: *Value*]
Example Command: Response:	 1C0101 (ADCChannel: 01) 003700 (Value: 55)

1.5.24 API FELICA

1.5.24.1 FeliCa_TDX

Command:	[1D00][Byte Array(Var): *TX*][Byte: *MaxRXByteCnt*][Byte: *MaximumResponseTime*][Byte: *NumberOfBlocks*]
Response:	[00][Bool: *Result*][Byte Array(Var): *RX*]
Example Command: Response:	 1D00060600FFFF0000FFFF04 (TX: 0600FFFF0000, MaxRXByteCnt: FF, MaximumResponseTime: FF, NumberOfBlocks: 04) 000112120101010701450F16000120220427674EFF (Result: true, RX: 120101010701450F16000120220427674EFF)

1.5.24.2 FeliCa_ReadWithoutEncryption

Command:	[1D01][variable number of UInt16: *ServiceCodeList*][variable number of UInt16: *BlockList*]
Response:	[00][Bool: *Result*][Byte Array(Var), 2 LB: *Data*]
Example Command: Response:	 1D01010B10010000 (ServiceCodeList: 100B, BlockList: 0000) 0001100000000000000000000000000000000000 (Result: true, Data: 00000000000000000000000000000000)

1.5.24.3 FeliCa_WriteWithoutEncryption

Command:	[1D02][variable number of UInt16: *ServiceCodeList*][variable number of UInt16: *BlockList*][Byte Array(Var), 2 LB: *Data*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 1D02010910010000100000000000000000000000000000000000 (ServiceCodeList: 1009, BlockList: 0000, Data: 00000000000000000000000000000000) 0001 (Result: true)

1.5.24.4 FeliCa_RequestSystemCode

Command:	[1D03][Byte: *MaxNumberOfSystemCodes*]
Response:	[00][Bool: *Result*][variable number of UInt16: *SystemCodeList*]
Example Command: Response:	 1D0308 (MaxNumberOfSystemCodes: 08) 000103030000FEA786 (Result: true, SystemCodeList: 0003, FE00, 86A7)

1.5.24.5 FeliCa_Poll

Command:	[1D04][UInt16: *SystemCode*]
Response:	[00][Bool: *Result*][Byte Array(8): *IDm*][Byte Array(8): *PMm*]
Example	
Command:	1D04FFFF
	(SystemCode: FFFF)
Response:	0001011603002D0CA50B03014B024F4993FF
	(Result: true, IDm: 011603002D0CA50B, PMm: 03014B024F4993FF)

1.5.24.6 FeliCa_RequestService

Command:	[1D05][variable number of UInt16: *ServiceCodeList*]
Response:	[00][Bool: *Result*][variable number of UInt16: *KeyVersionList*]
Example	
Command:	1D05010000
	(ServiceCodeList: 0000)
Response:	0001010100
	(Result: true, KeyVersionList: 0001)

1.5.25 API SLE44XX

1.5.25.1 SLE_GetATR

Command:	[1F00][Byte: *Channel*]
Response:	[00][Bool: *Result*][Byte Array(4): *ATR*]
Example	
Command:	1F0028
	(Channel: 28)
Response:	0001FFFFFFFF
	(Result: true, ATR: FFFFFFFF)

1.5.25.2 SLE_ReadMainMemory

Command:	[1F01][Byte: *Channel*][UInt16: *Address*][UInt16: *ByteCnt*]
Response:	[00][Bool: *Result*][Byte Array(Var), 2 LB: *Data*]
Example Command:	 1F012800000100 (Channel: 28, Address: 0000, ByteCnt: 0100)
Response:	00010100FF (Result: true, Data: FF)

1.5.25.3 SLE_UpdateMainMemory

Command:	[1F02][Byte: *Channel*][UInt16: *Address*][Byte: *Value*]
Response:	[00][Bool: *Result*]
Example Command:	 1F0228000000 (Channel: 28, Address: 0000, Value: 00)
Response:	0001 (Result: true)

1.5.25.4 SLE_ReadSecurityMemory

Command:	[1F03][Byte: *Channel*]
Response:	[00][Bool: *Result*][Byte Array(4): *SecMemData*]
Example Command:	 1F0328 (Channel: 28)
Response:	0001FFFFFFFF (Result: true, SecMemData: FFFFFFFF)

1.5.25.5 SLE_UpdateSecurityMemory

Command:	[1F04][Byte: *Channel*][Byte: *Address*][Byte: *SecMemData*]
Response:	[00][Bool: *Result*]
Example Command:	 1F042800FF (Channel: 28, Address: 00, SecMemData: FF)
Response:	0001 (Result: true)

1.5.25.6 SLE_ReadProtectionMemory

Command:	[1F05][Byte: *Channel*]
Response:	[00][Bool: *Result*][Byte Array(4): *ProtMemData*]
Example	
Command:	1F0528
	(Channel: 28)
Response:	0001FFFFFFFF
	(Result: true, ProtMemData: FFFFFFFF)

1.5.25.7 SLE_WriteProtectionMemory

Command:	[1F06][Byte: *Channel*][Byte: *Address*][Byte: *ProtMemData*]
Response:	[00][Bool: *Result*]
Example	
Command:	1F062800FF
	(Channel: 28, Address: 00, ProtMemData: FF)
Response:	0001
	(Result: true)

1.5.25.8 SLE_CompareVerificationData

Command:	[1F07][Byte: *Channel*][Byte: *Address*][Byte: *VerificationData*]
Response:	[00][Bool: *Result*]
Example	
Command:	1F072800FF
	(Channel: 28, Address: 00, VerificationData: FF)
Response:	0001
	(Result: true)

1.5.26 API NTAG

1.5.26.1 NTAG_Read

Command:	[2000][Byte: *Page*]
Response:	[00][Bool: *Result*][Byte Array(16): *Page*]
Example Command: Response:	 200004 (Page: 04) 000103B691028C537091016855016E78702E (Result: true, Page: 03B691028C537091016855016E78702E)

1.5.26.2 NTAG_Write

Command:	[2001][Byte: *Page*][Byte Array(4): *Data*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 20010400000000 (Page: 04, Data: 00000000) 0001 (Result: true)

1.5.26.3 NTAG_FastRead

Command:	[2002][Byte: *StartPage*][Byte: *NumberOfPages*]
Response:	[00][Bool: *Result*][Byte Array(Var): *Data*]
Example Command: Response:	 20020401 (StartPage: 04, NumberOfPages: 01) 00010403B69102 (Result: true, Data: 03B69102)

1.5.26.4 NTAG_ReadCounter

Command:	[2003]
Response:	[00][Bool: *Result*][UInt32: *CounterValue*]
Example	
Command:	2003
Response:	000101000000
	(Result: true, CounterValue: 1)

1.5.26.5 NTAG_ReadSig

Command:	[2004]
Response:	[00][Bool: *Result*][Byte Array(32): *ECCSig*]
Example	
Command:	2004
Response:	0001A9AC15AFB52080BA26A45B1DA442F363E31B41271AB12B3E6F67-864615B05321
	(Result: true,
	ECCSig: A9AC15AFB52080BA26A45B1DA442F363E31B41271AB12B3E6F67-864615B05321)

1.5.26.6 NTAG_GetVersion

Command:	[2005]
Response:	[00][Bool: *Result*][Byte Array(8): *Version*]
Example	
Command:	2005
Response:	00010004040502011503
	(Result: true, Version: 0004040502011503)

1.5.26.7 NTAG_PwdAuth

Command:	[2006][Byte Array(4): *Password*][Byte Array(2): *PwdAck*]
Response:	[00][Bool: *Result*]
Example	
Command:	2006FFFFFFFF0000
	(Password: FFFFFFFF, PwdAck: 0000)
Response:	0001
	(Result: true)

1.5.26.8 NTAG_SectorSelect

Command:	[2007][Byte: *Sector*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 200700 (Sector: 00) 0001 (Result: true)

1.5.27 API SRX

1.5.27.1 SRX_ReadBlock

Command:	[2100][Byte: *Block*]
Response:	[00][Bool: *Result*][Byte Array(4): *Data*]
Example Command: Response:	 210000 (Block: 00) 000100000000 (Result: true, Data: 00000000)

1.5.27.2 SRX_WriteBlock

Command:	[2101][Byte: *Block*][Byte Array(4): *Data*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 21010000000000 (Block: 00, Data: 00000000) 0001 (Result: true)

1.5.28 API SAMAVX

1.5.28.1 SAMAVx_AuthenticateHost

Command:	[2200][Byte: *CryptoEnv*][Byte: *KeyNo*][Byte Array(Var): *Key*][Byte: *KeyType*]
Response:	[00][Bool: *Result*]
Example Command: Response:	 22000000100000000000000000000000000000000000 (CryptoEnv: 00, KeyNo: 00, Key: 00000000000000000000000000000000, KeyType: 00) 0001 (Result: true)

1.5.28.2 SAMAVx_GetKeyEntry

Command:	[2201][Byte: *KeyNo*]
Response:	[00][Bool: *Result*][Byte Array(13): *TSAMAVxKeyEntryData*]
Example Command: Response:	 220101 (KeyNo: 01) 000100010200000000000000FF0C00 (Result: true, TSAMAVxKeyEntryData: 00010200000000000000FF0C00)

1.5.29 API EM4102

1.5.29.1 EM4102_GetTagInfo

Command:	[2300]
Response:	[00][UInt32: *TagInfo*]
Example Command: Response:	 2300 0001000000 (TagInfo: 1)

2 Disclaimer

Elatec reserves the right to change any information or data in this document without prior notice. The distribution and the update of this document is not controlled. Elatec declines all responsibility for the use of product with any other specifications but the ones mentioned above. Any additional requirement for a specific custom application has to be validated by the customer himself at his own responsibility. Where application information is given, it is only advisory and does not form part of the specification.

All referenced brands, product names, service names and trademarks mentioned in this document are the property of their respective owners.