

SOFTWARE GUIDE

RFID WAP4 HF



Table of contents

1	INTRODUCTION	5
1.1	ABOUT THIS MANUAL	5
1.2	RFID READERS SUPPORTED.....	5
1.3	STANDARDS AND TAGS SUPPORTED	5
1.4	PLATFORMS SUPPORTED	5
1.5	RFID SDK STRUCTURE	6
1.6	RFID PROCESS.....	7
2	GETTING STARTED	8
2.1	RFID DRIVER INSTALLATION	8
2.2	PROGRAMMING LANGUAGES	9
2.2.1	<i>Dotnet</i>	9
2.2.1.1	Development Platforms.....	9
2.2.1.2	Create a new project	9
2.2.1.3	RFID Driver Library overview	10
2.2.1.3.1	Namespace	10
2.2.1.3.2	RFIDDriver class	10
2.2.1.3.3	Check RFID Driver installation.....	10
2.2.1.3.4	Enable driver.....	10
2.2.1.3.5	Get COM Port number.....	10
2.2.1.3.6	Disable driver	10
2.2.1.4	HF Reader Library overview.....	11
3	READER TO HOST COMMUNICATION.....	13
3.1	BINARY PROTOCOL	13
4	DEVICE CONFIGURATION	15
4.1	EEPROM MEMORY ORGANISATION.....	15
4.2	STATION ID REGISTER (04H)	16
4.3	PROTOCOL CONFIGURATION REGISTER 1 (05H).....	16
4.4	BAUD RATE CONTROL REGISTER (0CH).....	20
4.5	OPERATION MODE REGISTER (0EH).....	21
4.6	RESET OFF TIME REGISTER (14H)	21
4.7	RESET RECOVERY TIME (15H)	21
4.8	RX THRESHOLD REGISTER (1EH).....	22
4.9	HARDWARE CONFIGURATION REGISTER (20H)	22
4.10	MODULATION INDEX REGISTER 2 (21H)	23
4.11	RF LEVEL REGISTER (22H).....	23
4.12	WAKEUP TIME REGISTER (27H)	24
4.13	PROTOCOL CONFIGURATION REGISTER 2 (28H).....	24
4.14	PICC TIMEOUT (29H)	25
4.15	CW AMPLITUDE REGISTER (2AH).....	25
4.16	CWMAX(2BH).....	26
4.17	TX I LOAD REGISTER (2CH)	26

5	INSTRUCTION SET	27
5.1	ERROR CODES	27
5.2	SYSTEM COMMANDS.....	28
5.2.1	Get Serial Number	28
5.2.2	RF Field ON/OFF	29
5.2.3	Set/Get user ports	30
5.2.4	Read user ports	31
5.2.5	Write user ports.....	32
5.2.6	Get version	33
5.2.7	Get real version	34
5.2.8	Set version	34
5.2.9	Reset.....	35
5.2.10	Get hardware version	36
5.3	EEPROM COMMANDS	37
5.3.1	Read Reader EEPROM	37
5.3.2	Write Reader EEPROM	38
5.3.3	Set register dynamical.....	39
5.3.4	Save register settings	40
5.3.5	Read dynamical Settings	40
5.3.6	Reset to default	42
5.4	TAG COMMANDS.....	43
5.4.1	Select	44
5.4.2	Multitag list.....	45
5.4.3	Multitag select	46
5.4.4	Highspeed Select	47
5.4.5	Extended select commands.....	48
5.4.6	Extended Multilist command.....	49
5.4.7	Extended Highspeed Select.....	50
5.5	SEND 14443-4 APDU (T=CL).....	52
5.5.1	't' command	52
5.5.2	Extending 't' command.....	54
5.5.2.1	ISO14443B option bytes mapping	54
5.5.2.2	ISO15693 option bytes mapping	55
5.5.2.3	ICODE option bytes mapping.....	55
5.5.2.4	FELICA option bytes mapping	56
5.5.2.5	NFCIP1 option bytes	56
5.6	SEND SAM APDU (T=0, T=1).....	60
5.6.1	Activate/Deactivate level shifter for SAM	60
5.6.1.1	Turn ON- Set level shifter 1 & 2 to 1.8V	60
5.6.1.2	Turn ON- Set level shifter 1 & 2 to 3V	60
5.6.1.3	Turn ON- Set level shifter 1 & 2 to 5V	60
5.6.1.4	Turn OFF- Set level shifter 1 & 2 to 0V	60
5.6.2	Send SAM APDU (T=0, T=1)	61
5.7	MIFARE SPECIFIC COMMANDS	66
5.7.1	LOGIN	66
5.7.1.1	LOGIN WITH KEY DATA FROM EEPROM	68
5.7.1.2	USAGE OF KEY A OR B	68
5.7.2	Read Block.....	68
5.7.3	Read Multiple Block.....	69
5.7.4	Read Value Block.....	70
5.7.5	Write Block	71
5.7.6	Write Multiple Block.....	72
5.7.7	Write Value Block	73
5.7.8	Increment value Block	74
5.7.9	Decrement Value Block	75
5.7.10	Copy Value Block.....	76
5.7.11	Write Master Key	77

6	TAGS	78
6.1	OVERVIEW TO THE MIFARE FAMILY	78
6.2	MIFARE DESFIRE	79
6.2.1	<i>Desfire Memory Organisation</i>	<i>79</i>
6.2.2	<i>Desfire States</i>	<i>79</i>
6.2.3	<i>Command structure.....</i>	<i>81</i>
6.2.4	<i>Security related commands – Overview</i>	<i>81</i>
6.2.5	<i>PICC level commands – Overview.....</i>	<i>82</i>
6.2.6	<i>Application level commands – Overview.....</i>	<i>83</i>
6.2.7	<i>Data Manipulation Commands – Overview</i>	<i>84</i>
6.2.8	<i>Sample APDUs (Mifare DESFire EV1 8KByte).....</i>	<i>85</i>
6.2.9	<i>Desfire using ISO/IEC 7816-4.....</i>	<i>87</i>
6.2.9.1	<i>ISO/IEC 7816-4 – Basic inter-industry commands</i>	<i>87</i>
6.2.9.2	<i>Wrapping of native Desfire APDUs</i>	<i>88</i>
6.3	MIFARE PLUS.....	90
6.3.1	<i>Mifare Plus Memory Organisation</i>	<i>91</i>
6.3.2	<i>Instruction</i>	<i>91</i>
6.3.3	<i>Command overview Level 3 for Mifare Plus S</i>	<i>92</i>
7	OPTION FOR CONTACT INTERFACE (SAM)	94
7.1	SAM	94
7.1.1	<i>Pin Configuration.....</i>	<i>94</i>
7.1.2	<i>General Description.....</i>	<i>95</i>
7.2	SAMPLE APDUS (UP TO 4 SAM SUPPORTED)	96
8	APPENDIX A: TLV STRUCTURE	100
9	APPENDIX B: READER SPEED OPTIMIZATION	101
10	APPENDIX C: COMPLETE COMMAND LIST	101

1 Introduction

1.1 About this manual

This manual is a handbook for the design and development of applications using the RFID HF SDK. Applications developed with this SDK allow controlling RFID HF readers supplied by Zebra Technologies and run on Zebra Technologies Workabout Pro 4 hand-held computer.

1.2 RFID readers supported

Applications developed using the RFID HF SDK control the following RFID reader

WAP4 – HF – KR3 – 2S	
-----------------------------	--

1.3 Standards and tags supported

The standard supported are:

MIFARE® Classic, ISO14443 A; ISO14443 B; ISO15693; Felica; NFC IP-1 passive initiator mode;

106-848 kBaud air interface speed

The tags supported are:

MIFARE® Standard, MIFARE® Ultralight, MIFARE® Desfire, MIFARE® Plus, MIFARE® Plus S, MIFARE® Plus SL3, NXP SmartMx, GTML, 14443A Controller, 14443B Controller, ICODE SLI, ICODE SLI-S, ICODE SLI-L, ICODE UID, MyD 10p, MyD 02p, MyD 10S, MyD 01P, SLE66R35, SLE66R32P, SLE66CLX360, SLE66CLX800, Felica and more

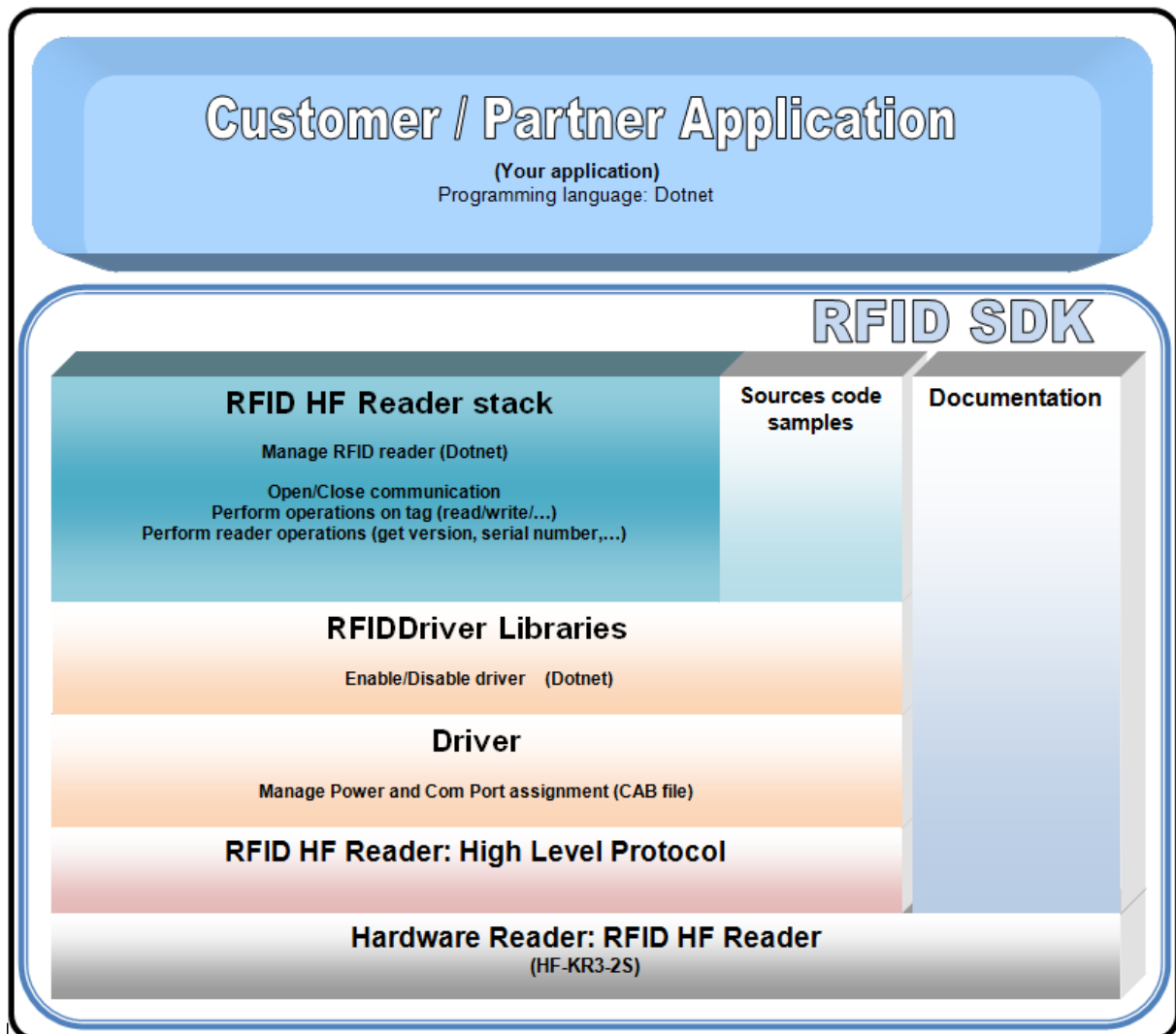
1.4 Platforms supported

This SDK is dedicated to Zebra Technologies Workabout Pro 4 handheld computer running under Windows Mobile 6.5 or Windows CE 6.

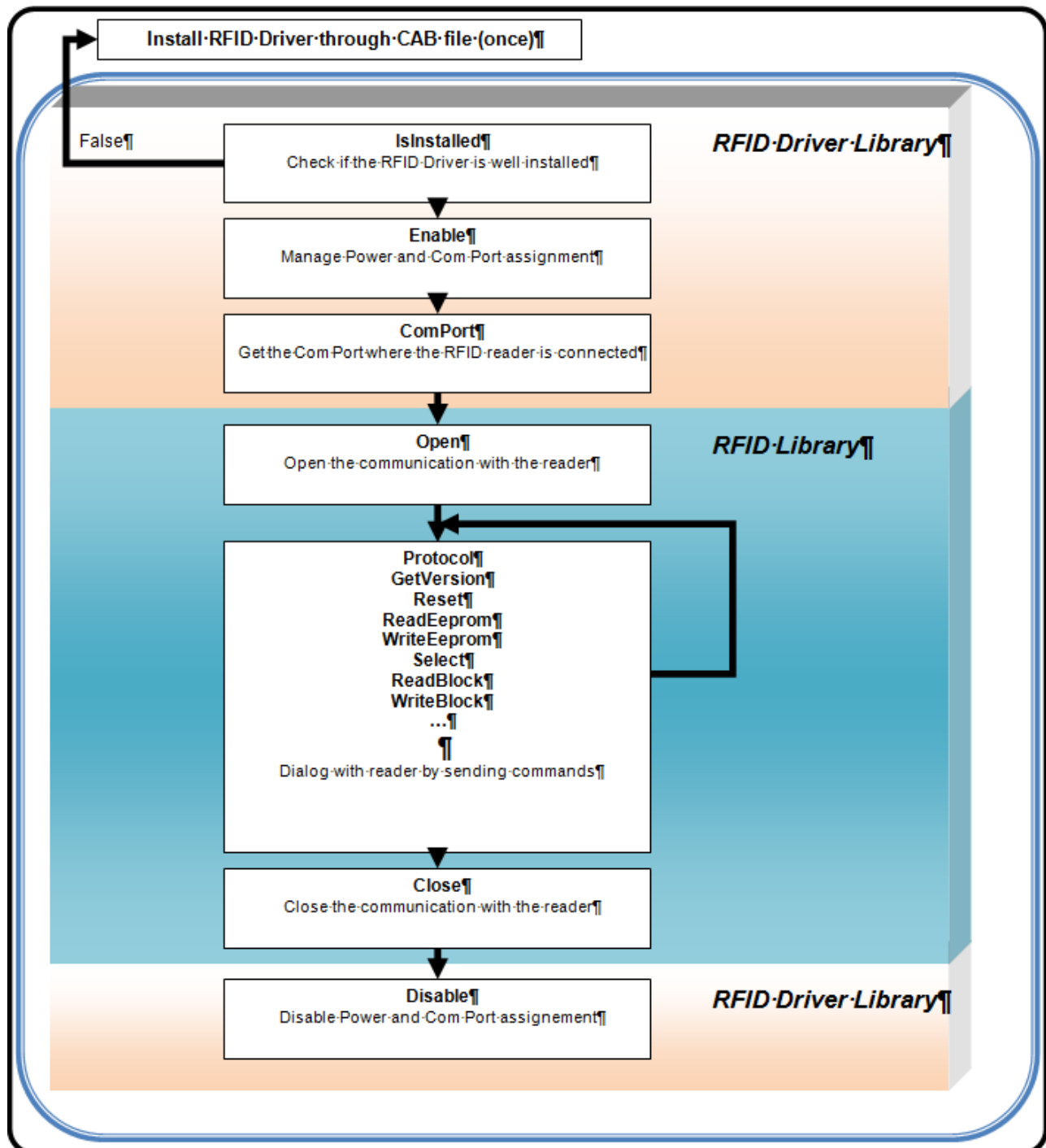
1.5 RFID SDK Structure

RFID SDK is based on different stacks.

- ➔ RFID Driver library stack allows turning on/off the power and assigning the com port assignment
- ➔ RFID HF Reader stack communicates with the reader and dialogs with tags



1.6 RFID Process



2 Getting Started

This handbook applies to the reader firmware version:

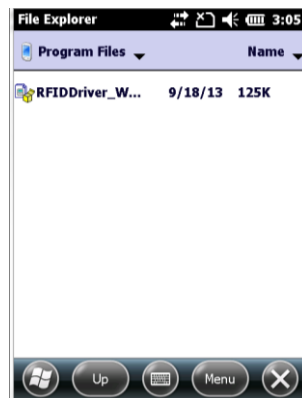
TA 1.09.10

2.1 RFID Driver installation

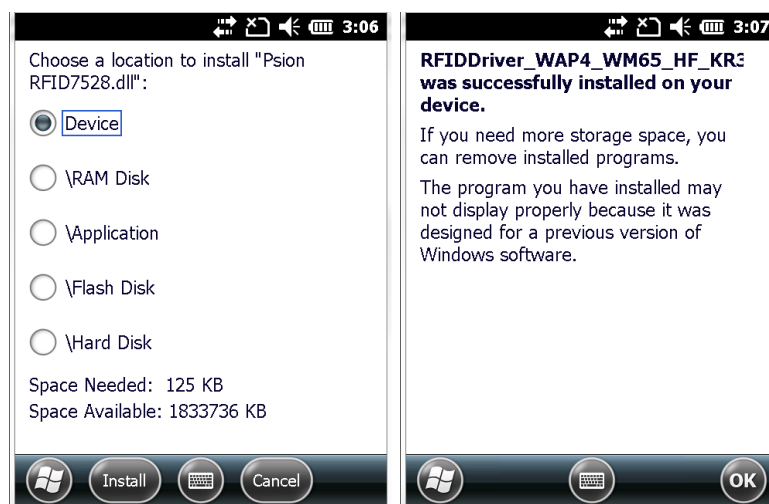
RFID Driver installation is mandatory. The driver is provided into the ZIP package as CAB file.

The first step is to select the right CAB file according to your operating system WinCE6 or WinWM6.5.

Then copy the RFID Driver CAB file to WAP4. (by activesync, usb stick, ..)



Then by double clicking on the CAB file, you should be able to install it.



Note: An unsuccessful installation occurs when the WAP4 can't dialog with the RFID reader.

The issue may result because the RFID reader is not mounted into the Workabout Pro 4 OR the RFID reader is badly connected.

2.2 Programming languages

This manual describes the reader protocol.

API libraries are available for the following programming languages:

- Dotnet (C# / VB.NET)

Note: Other programming languages can be supported by using the reader protocol through the RFID driver and serial interface.

2.2.1 Dotnet

The dotnet library is a high level API of FW protocol described on [section 3](#).

2.2.1.1 Development Platforms

The following development platforms should be used for developing .NET:

- Microsoft Visual Studio 2005
- Microsoft Visual Studio 2008

2.2.1.2 Create a new project

The following simple steps describe what is required to begin developing a .NET application by using the Zebra Technologies SDK in Visual Studio 2005 or 2008.

These are only the steps specific to incorporating the SDK; it is assumed the user is familiar with the Visual Studio environment.

In addition, sample .NET projects can be found in the **\Source code samples\Dotnet** subdirectory of your Zebra Technologies SDK installation.

Linking to the Zebra Technologies SDK Library

1. After creating a new project, open the Project menu and select **Add Reference...**
2. In the Browse tab, use the navigation tools to browse the **\Driver\Dotnet\CF2** or **\Driver\Dotnet\CF3.5** subdirectory of your **Zebra Technologies SDK** installation (according the usage of Compact Framework 2 or 3.5)
3. Select the **RFIDDriver.DLL** and click **OK**
4. In the Browse tab, use the navigation tools to browse the **\Dotnet\CF2** or **\Dotnet\CF35** subdirectory of your **Zebra Technologies SDK** installation (according the usage of Compact Framework 2 or 3.5)
5. Select the **ReaderTool.DLL** and click **OK**

2.2.1.3 RFID Driver Library overview

2.2.1.3.1 Namespace

```
using Psion.RFID;
```

2.2.1.3.2 RFIDDriver class

The RFID Driver class is the main object for piloting the RFID Driver.

```
RFIDDriver _rfidDriver= new RFIDDriver ();
```

2.2.1.3.3 Check RFID Driver installation

```
// check if the driver is installed  
if (!_rfidDriver.IsInstalled)  
    throw new Exception("RFID Driver is not installed");
```

2.2.1.3.4 Enable driver

Enable the RFID Driver, it manages the power (turn ON) and com port assignment.

```
// check and enable the driver  
// it means to power ON the module  
// assign com port number  
if (!_rfidDriver.IsEnabled) _rfidDriver.Enable();  
  
// necessary time to load the driver  
Thread.Sleep(250);
```

Throw an exception in case of error. Catch RFIDDriverException for more details

2.2.1.3.5 Get COM Port number

Once enabled the RFID driver returns the com port number from where the communication with the reader has to be established.

```
string comPort = string.Empty;  
if (_rfidDriver.ComPort < 10) comPort = String.Format("COM{0}",  
    _rfidDriver.ComPort);  
else comPort = String.Format("$device\\COM{0}", _rfidDriver.ComPort);
```

2.2.1.3.6 Disable driver

Disable the RFID Driver, it manages the power (turn OFF) and com port assignment (suppress all the assignments).

```
// disable rfid driver
// it means to destroy com port created
// and turn OFF the power on the module
if (_rfidDriver != null && _rfidDriver.IsInstalled && _rfidDriver.IsEnabled)
    _rfidDriver.Disable();
```

Throw an exception in case of error. Catch `RFIDDriverException` for more details

2.2.1.4 HF Reader Library overview

Namespace

```
using Psion.RFID.HF;
```

Main class

The reader class is the main object for dialoguing with the reader.

```
Reader _reader = new SerialReader ();
```

Throw an exception if RFID driver is not installed.

Open reader communication

```
string comPort = string.Empty;
if (_rfidDriver.ComPort < 10) comPort = String.Format("COM{0}",
    _rfidDriver.ComPort);
else comPort = String.Format("$device\\COM{0}", _rfidDriver.ComPort);

((SerialReader)_reader).Open(
    comPort,
    115200);
```

Throw an exception if open reader failed

Dialog with reader

The main function is `Protocol` which send instruction as described in [section 5](#).

This method compiles different arguments into bytes, and send data to reader in Binary mode and get reply. Valid arguments are char, string, byte, byte arrays, and int. If the last argument is int, then it is the timeout.

```
// get version
string command = "V";
byte [] data = _reader.Protocol(
    command,
    _reader.CommandMediumTimeout);
```

```
// read EEPROM
byte address = 0x0E; // operation mode register
byte[] data = _reader.Protocol("re", address, _reader.CommandShortTimeout);

// or

byte [] data = _reader.ReadEeprom(0x0E);
```

```
// read a tag
byte[] data = _reader.Protocol("s");

// or

Contactless _contactless = new Contactless(_reader);
byte [] data = _contactless.Select();
```

Close reader communication

```
_reader.Close();
```

3 Reader to Host Communication

3.1 Binary Protocol

This protocol was developed for industrial usage including synchronization and frame checking. A device driver is needed in order to use this protocol. Data is transmitted binary. The reader has no timeout for receiving data.

Protocol structure

STX	Station ID	Length	Data	BCC	ETX
1 Byte	1 Byte	1 Byte	various length	1 Byte	1 Byte

Example: Get Version

Request:

STX	Station ID	Length	Data	BCC	ETX
02h	FFh	01h	76h	88h	03h

Response:

STX	Station ID	Length	Data	BCC	ETX
02h	00h	0Ah	544120312E30392E3038 (HEX)	06h	03h

Data as string: **“TA 1.09.08”**

STX

Start of transmission (02h)

Station ID (Unique ID of the reader)

00h: Reserved for the bus master.
 FFh: Broadcast message. All devices will execute a command sent with broadcast.
 01h-FEh: Valid station ID's for the reader. The reader will only progress a command if the received station ID either matches the station ID of the reader or is FFh.

Length (Data Length Indicator)

Denotes the length of the Data block.

Data

This part contains the command and data. The command values are the same as in ASCII protocol mode like 's' for select or 'x' for reset, whereas data is transmitted binary.

The length of the command block depends on the instruction.

BCC

The BCC is used to detect transmission errors. To calculate the BCC value all bits excluding STX and ETX are XOR-ed.

$$\text{BCC} = \text{StationID} \text{ XOR } \text{Length} \text{ XOR } \text{Data}_0 \text{ XOR } \dots \text{ XOR } \text{Data}_N$$

Example: $\text{BCC} = 0\text{xFF} \text{ XOR } 0\text{x02} \text{ XOR } 0\text{x11} \text{ XOR } 0\text{x22} = 0\text{xCE}$

$$\text{BCC} = 0\text{xCE}$$

ETX

ETX shows the end of the command (03h)

Remarks

If the reader receive a wrong command or frame (i.e. BCC wrong) or the station ID does not match the internal ID of the reader, the command is not executed. The reader waits for the next valid frame.

The reader module answers in the same telegram format, with the ID-field set to 0.

4 Device Configuration

The reader devices have flags to configure their behaviour. The flags are stored in the EEPROM. Only during the start up phase the reader accept changes of the flags. After any changes in the EEPROM the reader device needs to be restarted.

4.1 EEPROM memory organisation

Register	Description
00h...03h	RFU
04h	Station ID Register
05h	Protocol Configuration
06h...0Bh	RFU
0Ch	Baud Rate Configuration Register
0Dh	RFU
0Eh	Operation Mode Register
0Fh...13	RFU
0x14	ResetOffTime
0x15	ResetRecoveryTime
1Eh	Rx Threshold Register
1Fh	RFU
20h	Hardware Configuration Register
21h	Modulation Index Register
22h	RF Level Register
23h...27h	RFU
28h	Protocol Configuration Register 2
29h	PiccTimeout
2Ah	CW Amplitude
2Bh	CW Max
2Ch	Tx iLoad
2Dh...30h	RFU

4.2 Station ID Register (04h)

This register contains the Station ID (Unique ID of reader).

Valid station ID's for the reader are 01h to FEh.

The reader will only progress a command if the received station ID either matches the station ID of the reader or is FFh.

The default value is 01h.

4.3 Protocol Configuration Register 1 (05h)

It specifies the general behaviour of the reader device.

The default value is DFh.

Protocol configuration register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFU	RFU	Fast Select	RFU	RFU	Extented ID	Protocol mode	Autostart

Autostart (Bit 0):

This bit is only applicable if the ASCII protocol is enabled. If this bit is set the reader starts automatically in the continuous read mode after start up. In this mode the command – response structure of the interface is broken up. The UID of a detected Tag is send autonomously to the host.

Value	Description
1	The reader will start continuous read after start up.
0	The reader won't poll for tags after start up.

Protocol mode (Bit 1):

If this bit is set the reader runs in binary mode, otherwise the reader runs in ASCII mode.

Note: It is highly recommended to use Binary Protocol.

Value	Description
1	Binary Protocol
0	ASCII Protocol

Extended ID (Bit 2):

If this bit is set, a unique information byte called Extended ID is transmitted with each tag-UID. This byte is different for most of the tag types.

The values for the Extended ID-prefix byte are given in the table below:

CL1: Cascade Level1 = Single UID (4 bytes)

CL2: Cascade Level2 = Double UID (7 bytes)

SL1-SL3: Security level 1-3 for Mifare® Plus tags only

Technology	Tag Type	Extended ID
		TA 01.09.08
ISO 14443 – A	Mifare Mini 0.3k (4 Byte UID)	01h
	Mifare Classic 1k (4 Byte UID)	02h
	Mifare Classic 1k (7 Byte UID)	0Bh
	Mifare Classic 4k (4 Byte UID)	03h
	Mifare Classic 4k (7 Byte UID)	0Ch
	Mifare UltraLight (7 Byte UID)	05h
	Mifare UltraLight C (7 Byte UID)	05h
	Mifare Desfire 4k (7 Byte UID)	06h
	Mifare Desfire EV1 2k (7 Byte UID)	06h
	Mifare Desfire EV1 4k (7 Byte UID)	06h
	Mifare Desfire EV1 8k (7 Byte UID)	06h

Mifare Plus 2k (4 Byte UID) SL0	09h
Mifare Plus 2k (4 Byte UID) SL1	02h
Mifare Plus 2k (4 Byte UID) SL2	RFU
Mifare Plus 2k (4 Byte UID) SL3	09h
Mifare Plus 2k (7 Byte UID) SL0	0Ah
Mifare Plus 2k (7 Byte UID) SL1	0Bh
Mifare Plus 2k (7 Byte UID) SL2	RFU
Mifare Plus 2k (7 Byte UID) SL3	0Ah
Mifare Plus 4k (4 Byte UID) SL0	0Dh
Mifare Plus 4k (4 Byte UID) SL1	03h
Mifare Plus 4k (4 Byte UID) SL2	07h
Mifare Plus 4k (4 Byte UID) SL3	09h
Mifare Plus 4k (7 Byte UID) SL0	0Ah
Mifare Plus 4k (7 Byte UID) SL1	0Ch
Mifare Plus 4k (7 Byte UID) SL2	08h
Mifare Plus 4k (7 Byte UID) SL3	0Ah
ISO/IEC 14443A Tag	0Dh
ISO/IEC 14443A Controller	15h / 16h
Infineon (SLE66CLxxx)	17h
Smart MX (4Byte UID)	18h
Infineon (SLE66R35 4Byte UID)	19h
MyD move (7 Byte UID)	05h
MyD-NFC (7 Byte UID)	05h
NTAG203 (7 Byte UID)	05h

ISO 14443 – B	ISO/IEC 14443B Controller	20h / 22h
	SRT 512	24h
FELICA	Felica 212kbs	51h
	Felica RCS-885 424kbs	51h
	Felica Lite	51h
	Felica 212kbs	51h
ISO 15693	ICODE SLI	31h
	ICODE SLI-S	32h
	ICODE SLI-L	33h
	MY-D 10P	40h
	MY-D 02P	41h
	MY-D 10S	42h
	MY-D 01P	43h
	BitCollision occurred	FEh
	unknown Transponder	FFh

Fast Select (Bit 5):

If this bit is set to '0' the selection polling wheel starts with the last detected Tag Type. Otherwise the polling wheel starts with ISO 14443 Type-A.

Value	Description
0	The selection series begins with the last known Tag Type
1	The selection series begins with the ISO 14443 Type-A

4.4 Baud Rate Control Register (0Ch)

This register defines the speed of the communication between the reader and the host.

The default value is 04h (115200 Baud).

Baud rate register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFU	RFU	RFU	RFU	RFU	BS2	BS1	BS0

This register defines the baud rate of the reader device.

BS2	BS1	BS0	Baudrate
0	0	0	9600
0	0	1	19200
0	1	0	38400
0	1	1	57600
1	0	0	115200
1	0	1	230400
1	1	0	460800
1	1	1	921600

Note: An error free communication is only guaranteed for baud rates of 460800baud or lower.

4.5 Operation Mode Register (0Eh)

The operation mode register defines which tag types the reader supports. This register enables fast tag recognition only defined tag types are requested.

If the bit is set, the specified tag type is supported.

Default value is 8Fh.

Operation Mode Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GTML	ICODEUID	RFU	ISO15693	Felica	SR716	14443B	14443A

4.6 Reset Off Time Register (14h)

The Reset Off Time Register defines how long the field is off by a fieldreset. The value is in milliseconds.

Default value is 03h.

4.7 Reset Recovery Time (15h)

The Reset Recovery Time Register defines the timeout between a fieldreset and the first frame which is send to the tag.

The value is in milliseconds.

Default value is 0Fh.

4.8 Rx Threshold Register (1Eh)

This register can be used to set thresholds for the Reader IC's bit decoder.

The default value is A5h.

Rx Threshold Register							
Min Level				Collision Level			
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Min3	Min2	Min1	Min0	Coll3	Coll2	Coll1	Coll0

Min Level

Defines the minimum signal strength at the decoder input that shall be accepted.

Collision Level

Defines the minimum signal strength at the decoder input that has to be reached by the weaker half-bit of the Manchester-coded signal to generate a bit collision relatively to the amplitude of the stronger half bit.

Remark

For high speed baudrate (424k or 848k), the value to use is 0x1E

4.9 Hardware Configuration Register (20h)

This register is used for customer specific hardware to operate with more than one SAM holder and a Level shifter. The content of has no effect in the regular firmware versions.

The default value is 00h.

Hardware Configuration Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
HWConf7	HWConf6	HWConf5	HWConf4	HWConf3	HWConf2	HWConf1	HWConf0

4.10 Modulation Index Register 2 (21h)

The modulation index register defines the Modulation Index of the Reader IC.

The modulation index is defined as the voltage ratio $(V_{max}-V_{min})/(V_{max}+V_{min})$.

The default value is 1Ah.

Modulation Index Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFU	RFU	ModReg5	ModReg4	ModReg3	ModReg2	ModReg1	ModReg0

4.11 RF Level Register (22h)

The default value is 76h.

RF Level Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFU	RxGain2	RxGain1	RxGain0	RFLevel3	RFLevel2	RFLevel1	RFLevel0

RFLevel

The RF Level Register configures the NFC RF level detector sensitivity.

RxGain

The three bits define the receivers signal voltage gain factor.

Bit 6	Bit 5	Bit 4	Description
0	0	0	18 db
0	0	1	23 db
0	1	0	18 db
0	1	1	23 db
1	0	0	33 db
1	0	1	38 db
1	1	0	43 db

1	1	1	48 db
---	---	---	-------

4.12 Wakeup Time Register (27h)

The default value is 26h.

Wakeup Time Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Wakeup time 7	Wakeup time 6	Wakeup time 5	Wakeup time 4	Wakeup time 3	Wakeup time 2	Wakeup time 1	Wakeup time 0

The Wakeup Time register is used in the “Low Power Polling” mode and in the “Low Power Tag Detection” mode. The value stored in this registers defines the time interval at which the reader device is waked up periodically.

The 8bit value is multiplied by 25msec.

The minimal value allowed is 10h. If the value stored is less than 10h then the default value of 26h will be used.

In the Low Power modes the internal timer of the micro Controller is inaccurate. The calculated wakeup time interval and the real wakeup time interval can differ up to 10%.

4.13 Protocol Configuration Register 2 (28h)

This register is reserved for future use.

The default value is FFh.

Protocol Configuration Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU

4.14 PICC timeout (29h)

PICC Timeout Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PICC Timeout7	PICC Timeout6	PICC Timeout5	PICC Timeout4	PICC Timeout3	PICC Timeout2	PICC Timeout1	PICC Timeout0

The PICC timeout defines the maximal time in msec to wait for an answer of the TAG. This timeout is only valid for an ISO14443A and FELICA tag. The value in the register will be multiplied with 14h.

Default value: 30h (960ms)

Example:

PICC Timeout Register value= 10h → Picc Timeout = 140h (320ms)

PICC Timeout Register value= 02h → Picc Timeout = 28h (40ms)

PICC Timeout Register value= 01h → Picc Timeout = 14h (20ms)

4.15 CW Amplitude Register (2Ah)

The default value is 00h.

CW Amplitude Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFU	RFU	RFU	RFU	RFU	RFU	cw_amp	cw_amp

00b: TVDD -100 mV

01b: TVDD -250 mV

10b: TVDD -500 mV

11b: TVDD -1000 mV

With the CW Amplitude Register the output power can be traded off against power supply rejection.

If CW max is set to 1 the voltage is pulled to the maximum possible. In this case the CW Amplitude Register has no influence.

4.16 CwMax(2Bh)

Cw Max Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFU	RFU	RFU	RFU	RFU	RFU	RFU	Cwmax

Bit 0: Set amplitude of continuous wave carrier to the maximum. If set to 1, CW Amplitude Register has no influence.

4.17 TX I Load Register (2Ch)

The default value is 0Ah.

TxILoad Register							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFU	RFU	RFU	RFU	Tx_iLoad	Tx_iLoad	Tx_iLoad	Tx_iLoad

This is the Factory trim value that sets the expected Tx load current.

Note: 0h is not allowed

5 Instruction Set

To give a better overview about all commands supported by the firmware they are combined into several command groups:

- System commands
- EEPROM Commands
- Tag Commands
- 14443-4 Commands
- SAM Commands
- Power Management Commands
- Mifare Specific Commands

Each command send by the host to the reader device expects a response by reader device. Exceptions are mentioned explicitly.

For the complete command list please refer to Appendix C: Complete Command List

5.1 Error codes

Error Code	Description
'E' (45h)	Invalid value format, specified block does not match the value format
'F' (46h)	General failure
'I' (49h)	Invalid key format
'N' (4Eh)	No tag in the field
'O' (4Fh)	Operation mode failure
'U' (55h)	Read after write failure
'X' (58h)	Authentication failed
'?' (3Fh)	Unknown command

5.2 System Commands

General System commands.

Supported commands:

Command	Description
'b'	Get serial number
'pon' / 'poff'	RF-Field on/off
'pp'	Set/Get user ports
'pr'	Reads user ports
'pw'	Writes user ports
'v'	Get version
'vr'	Get real version
'vs'	Set version
'vh'	Get hardware version
'x'	Reader software reset

5.2.1 Get Serial Number

This command returns the pre-programmed Serial Number of the reader.

Command:

Command	Data
'b'	none

Response:

Answer	Description
Serial Number (16 bytes)	Returns the Serial Number of the reader

5.2.2 RF Field ON/OFF

This command is used to switch the RF field on and off.

Command:

Command	Data	Description
'pon'	none	Turns the RF-Field on.
'poff'	none	Turns the RF-Field off.

Response:

Answer	Description
'P'	Positive Acknowledge

5.2.3 Set/Get user ports

This command is used to define the direction of the user ports 1-8. It also sets the level of the output ports and reads back the values from the input ports.

A bit set to '1' in the Input/Output Mask defines the specified user port as output, a bit set to '0' as input. The write value sets the output pin levels. The command always responds with the read state of the ports, even if they are set as outputs (output values are returned).

Command:

Command	Data
'pp'	Input/Output Mask (1 byte) Write Value (1 byte)

Response:

Answer	Description
Data	Read Value (1 byte)

The Input/Output Mask selects the bits addressed by the Write Value and the Read Value command. Each of the user ports 1-8 corresponds to a bit within the mask byte. The pin outlay is documented in the Hardware Guide.

Bit mapping:

IO Mask	none	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Port	RFU	User Port 8	User Port 7	User Port 6	User Port 5	User Port 4	User Port 3	User Port 2	User Port 1
OEM-PIN	RFU	10	9	8	7	24	23	22	21

Example:

Command	Description
Send: pp0602	Set User Port 2 & 3 as output; Rest input; Set User Port 2 high and User Port 3 low

5.2.4 Read user ports

This command is used to read the user ports 1-8. A mask is used to define the selected port pins. See 'pp' command for details about bit mapping.

Command:

Command	Data
'pr'	Input Mask (1 byte)

Response:

Answer	Description
Data	Read Value (1 byte)

The Input Mask selects the pins addressed by this command. Selected pins are set to input (high impedance). The input values of the selected pins are returned, none-selected bits are set to 0.

Example:

Command	Response
'pr'	01h

User port 1 is set to input (high impedance) and externally pulled high.

Example:

Command	Response
'pr' 03	01h

This command reads user ports 1 and 2 (set to input, high impedance), where user ports 1 is externally pulled high and user ports 2 is externally pulled low. All other user ports remain unchanged and return 00h.

5.2.5 Write user ports

This command is used to write the user ports 1-8. A mask is used to define the selected port pins. See 'pp' command for details about bit mapping.

Command:

Command	Data
'pw'	Output Mask (1 byte) (optional) Write Value (1 byte)

Response:

Answer	Description
Data	Write Value (1 byte)

The Output Mask selects the pins addressed by this command. The selected pins are set to the specified Write Value (low impedance). All other pins remain unchanged.

By only one byte data the addressed pins are set as output and high.

Example:

Command	Response
'pw' 0A	0Ah

User port 2 and 4 are set to output (low impedance) and high level.

Example:

Command	Response
'pw' 03 01	01h

This command sets user port 1 and 2 to output (low impedance). User port 1 is set high and user port 2 is set low. All other pins remain unchanged and return 0h.

5.2.6 Get version

This command returns the current version of the reader module.

Command:

Command	Data
'v'	none

Response:

Answer	Description
"Reader Version String"	The version string is returned, typically "TA 1.09.08"

Example Response:

Answer	Description
'TA 1.09.08	in ASCII Protocol mode
54 41 20 31 2E 30 39 2E 30 38	in Binary Protocol mode

5.2.7 Get real version

This command returns the real (hardcoded) firmware version of the reader module.

Command:

Command	Data
'vr'	none

Response:

Answer	Description
"Reader Version String"	The version string is returned, typically "TA 1.09.08"

Example Response:

Answer	Description
'TA 1.09.08'	in ASCII Protocol mode
54 41 20 31 2E 30 39 2E 30 38	in Binary Protocol mode

5.2.8 Set version

With this command the customer can set an arbitrary version string. The get version command will return this string.

The new version string will take effect after a reader reset.

The maximal length of the string is 20 byte

Command:

Command	Data
'vs'	n bytes (max. 20) in hexadecimal format

Response:

Answer	Description
Data	New version string of the reader

Example:

Command	Description
vs41424344	ABCD is the new Version string

Example:

Command	Description
vsff	Version string is reset to firmware version (TA 1.xx.xx)

If the first data byte is FFh the version string is reset to the hardcoded firmware version string, for example TA1.09.08.

5.2.9 Reset

This command invokes a reader software reset. All configuration settings will be reloaded. All tags in the antenna field are also reset.

The reader won't send a response to this command and will take 25ms to reboot.

Command:

Command	Data
'x'	none

Response:

Answer	Description
'TA 1.09.08	ASCII mode
none	Binary mode

5.2.10 Get hardware version

This command returns the hardware version (including reader chip and microcontroller) of the reader module.

Command:

Command	Data
'vh'	none

Response:

Answer	Description
Data	The hardware version of the reader module

The results appear in hexadecimal format.

Example:

Command	Response
'vh'	52433636332E53544D33324631303352432E4F454D (RC663.STM32F103RC.OEM)

5.3 EEPROM Commands

EEPROM register configuration.

Supported commands:

Command	Description
're'	Read EEPROM registers
'we'	Write EEPROM registers
'nr'	Set registers dynamical
'nv'	Save registers settings
'ne'	Read dynamical settings
'f'	Reset all EEPROM registers to default

5.3.1 Read Reader EEPROM

This command reads the internal reader EEPROM. The EEPROM contains all start up parameters.

For the EEPROM organization please have a look at chapter Device Configuration

Command:

Command	Data
're'	EEPROM address(1byte) (00h-2Fh)

Response:

Answer	Description
Data	EEPROM data (1byte)

Example:

Command	Description
re05	Read EEPROM Address 05h

5.3.2 Write Reader EEPROM

This command writes to the internal reader EEPROM. The EEPROM contains all start up parameters. When changing any of the start-up parameters, a reader reset must be executed to apply the new settings.

For the EEPROM organization please have a look at chapter Device Configuration

Command:

Command	Data
'we'	Address (1 byte), valid range 04h – 2Fh Data (1 byte)

Response:

Answer	Description
Data	EEPROM data (1byte)
'F'	Error: Read after write failure

Example:

Command	Description
we0501	Set EEPROM address 05 to 01h.

5.3.3 Set register dynamical

With the 'nr' command it is possible to change the register settings without a reset. The value take effect directly after the usage of the command.

All register can be changed dynamical except the following:

- addr: 05h ProtocolConfigurationRegister
- addr: 0Ch Baudrate
- addr: 0Ah StationID

Command	Data
'nr' + addr + value	'nr' command addr..... register address value..... register value

Response:

Answer	Description
value	returns the value which is set

Example:

Command	Description
'nr' 21h 10h	Sets the Modulation Index Register (21h) to 10h

Response	Description
10h	register value

5.3.4 Save register settings

With the 'nv' command it is possible to store the settings which are configured with the 'nr' command into the EEPROM.

Command	Data
'nv'	'nv' command

Response:

Answer	Description
value	Returns 00h

Example:

Command	Description
'nv'	saves the settings

Response	Description
00h	OK

5.3.5 Read dynamical Settings

With the 'ne' command it is possible to read the Settings which are not saved into the EEPROM.

Command	Data
'ne' + addr	'nr' command addr..... register address

Answer	Description
value	returns the value which is set

Example:

Command	Description
'ne' 21h	Reads the ModulationIndex

Response	Description
10h	Value of the ModulationIndex

5.3.6 Reset to default

This command resets one or all EEPROM register to their default values. To update the new register settings in the firmware a reset is required.

If a single register is reset its default value is returned; 00h otherwise.

Command	Data
'f'	None or Register Address[1 Byte]

Example:

Command	Description
'f'	reset all EEPROM register to their default values

Response	Description
00	All EEPROM register are reset to their default values

Command	Description
'f15'	resets the EEPROM register 15h (Reset Recovery Time) to the default value

Response	Description
0F	Reset to default value 0Fh

5.4 TAG Commands

Tag related commands.

Supported commands:

Command	Description
's'	Select
'm'	Multitag list
'm'	Multitag select
'h'	Highspeed select
'ns'	Extended select
'nm'	Extended multilist
'nh'	Extended highspeed select

5.4.1 Select

This command selects a single card in the RF field. If a card is in the field the reader returns the UID of the selected card. The reader detects the length of the card UID automatically.

If the Extended ID bit in the protocol configuration register (**Error! Reference source not found.**) is set the first byte of the returned value is the extended ID.

Command:

Command	Data
's'	none

Response:

Answer	Description
Data	Serial number of the tag
'N'	Error: No tag in field

Example:

Command	Description
's'	0B0493197AB32280 selected card UID: 0493197AB32280 selected tag type: 0B (Mifare Classic 1k CL2)

5.4.2 Multitag list

This command is used to detect multiple tags in the RF field.

Command:

Command	Data
'm'	none

Response: Multitag list:

Answer	Description
I_1 + UID ₁ + I_2 + UID ₂ + ... + No. tags	length of 1 st UID (1 byte) + UID ₁ (length bytes) + length of 2 nd UID (1 byte) + UID ₂ (length bytes) + total number of tags (1 byte)
'N'	Error: No tag in field

Example:

Response	Description
04 DD 02 EA 02 07 04 0D 4E 61 80 1D 80 02	First tag with 4 byte UID Second tag with 7 byte UID 2 tags detected

In this example Extended ID is disabled in the Protocol Configuration Register.

5.4.3 Multitag select

This command is used to select one out of multiple tags in the field. All following operations are performed on this tag.

Command:

Command	Data
'm'	UID ¹ (n bytes)

¹ Important note: This parameter contains the UID of the tag without the optional Extended ID. It is recommended to disable the usage of Extended ID's in the Protocol Configuration Register to avoid misunderstandings.

Response:

Answer	Description
UID	Returns the UID if the tag was selected successfully.
'N'	Error: No tag in field

Example:

Command	Description
'm' DD 02 EA 02	Selects the tag with the specified UID

Response	Description
DD 02 EA 02	The tag was successfully selected

In this example the Extended ID is disabled in the Protocol Configuration Register.

5.4.4 Highspeed Select

This command selects a card and prepares it for a high baudrate communication. The command performs automatically a RATS and PPS command. It returns the

UID+ATS (optional) + selected baudrate.

The command supports only synchronous mode.

The answer contains the ATS if the Extended ID bit is set.(default)

Command:

Command	Data
'h'	Baudrate

Response:

Answer	Description
UID(nbytes) + ATS (nbytes)+ max. frame size and baudrate (1 byte)	UID..... 4 or 7 byte Serialnumber ATS.....Answer to select max. frame size.....defines the max length of a frame baudratedefines the selected airbaudrate
'N'	Error: No tag in field

The higher nibble of the last bytes contains the max frame size of the air interface.

The lower nibble contains the baudrate

Frame Size	Description
0x	16Bvtes
1x	24Bvtes
2x	32Bvtes
3x	40Bvtes
4x	48Bvtes
5x	64Bvtes
6x	96Bvtes
7x	128Bvtes
8x	256Bvtes

Baud Rate	Description
x0	106kBaud
x2	212kBaud
x4	424kBaud
x8	848kBaud

Remark

For a better reading distance performance at high speed baudrate (424k or 848k), the EEPROM value of RX Threshold (0x1E) can be modified to 0x1E instead of default 0xA5.

Additional select commands

5.4.5 Extended select commands

This command works like the select command but it returns more data in a TLV structure. For details of the structure see Appendix A.

Command:

Command	Data
'ns'	None

Response:

Response	Description
data	response data (select TLV structure)
'N'	no Tag in the field
'E'	Error

Example:

Command	Response (Mifare Desfire EV1)
'ns'	AC1D07048940D2DF2580AA4403BA20E106C102D10A

Type	Data	Name
AC		Starting Byte of TLV structure
1D	07	UID Length
	04 89 40 D2 DF 25 80	UID
AA	44 03	ATQA
BA	20	SAK
E1	06	Extended ID
C1	02	Cascade level
D1	0A	Protocol

5.4.6 Extended Multilist command

This command works like the 'm' command but it returns more data in the form of a select TLV.

This command is slower like the 'm' command but it returns also the ATQA of all PICCs in the radio field in a correct form.

For details of the structure see Appendix A.

Command:

Command	Data
'nm'	None

Response:

Response	Description
data	response data (select TLV structure)
'N'	no Tag in the field
'E'	Error

Example:

Command	Data
'nm'	nm command

Response	Description
0xAC 0xD1 0x0A 0xC1 0x02 0xE1 0x06 0xBA 0x24 0xAA 0x44 0x03 0x1D 0x07 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0xA5 0x08 0x06 0x75 0x77 0x81 0x02 0x00 0xC0 0xC1 0xAC 0xD1 0x0B 0x0B 0x50 0x12 0x23 0x45 0x56 0x12 0x53 0x54 0x4E 0x33 0x81 0xC3 0x1D 0x04 0x12 0x23 0x45 0x56 0xE1 0x20	Received data (TLV structure)

translated data:

PICC 1

Protocol = ISO type A

Cascade level = 2

Extended ID = 06h
 SAK = 24h
 ATQA = 44 03
 UID = 01 02 03 04 05 06 07
 ATS = 06 75 77 81 02 00 C0 C1

PICC 2

Protocol = ISO type B
 ATQB = 50 12 23 45 56 12 53 54 4E 33 81 C3
 UID = 12 23 45 56
 Extended ID = 20h

5.4.7 Extended Highspeed Select

This command works like the 'h' command but it returns more data in the form of a select TLV. For details of the structure see Appendix A.

Command:

Command	Data
'nh'	baudrate

Value	Baudrate
00	106kbaud
02	212kbaud
04	424kbaud
08	848kbaud

Response:

Response	Description
Data	response data (select TLV structure)
'N'	no Tag in the field
'E'	Error

Example:

Command	Description
---------	-------------

'nh' + 04	nh = command 04 = 424kbaud
-----------	-------------------------------

Response	Description
AC 1D 07 04 62 29 C9 92 26 80 AA 44 03 BA 20 E1 06 C1 02 D1 0A A5 06 06 75 77 81 02 80 D5 04	Received data (TLV structure)

translated data:

UID = 04 62 29 C9 92 26 80

ATQA = 04 00

SAK = 20

Extended ID = 06 = Mifare Desfire

Cascade level = 02

Protocol = ISO type A

selected baudrate = 04 = 424kbaud

5.5 Send 14443-4 APDU (T=CL)

This so-called “transfer command” sends any command in ISO/IEC 14443-4 format to a tag.

According to ISO/IEC 14443-4 a select and a RATS command must be performed to activate a compatible tag into 14443-4 layer mode.

5.5.1 ‘t’ command

The actual information is stored in the INF-field.

Please refer to ISO/IEC 14443-4:2008 for detailed information.

Note: With this command it also can be send an APDU to other tag types, like Felica or 14443B. To make this work the according tag has to be selected first.

The data to send has to be written in the data block.

14443-4 Command:

Command	Downlink length	Option byte	Data			
			PCB (optional)	CID (optional)	NAD (optional)	INF
‘t’	1 byte	1 byte	1 byte	1 byte	1 byte	Various length

Command ‘t’

This command sends a custom data block to a contactless tag.

Downlink length

The Downlink length contains the length of the Data.

Option byte

This byte contains the transfer options and must be 0Fh. All other values are RFU and will lead to an error. The message will not be send.

Data

Data contains the request block sent to the tag. Data should be formatted according to ISO/IEC 14443-4 without EDC-field!

PCB holds the protocol control information and usually toggles between 02h and 03h (0Ah and 0Bh when CID is used)

CID is optional and holds the card ID when more than one tag is in the field.

NAD is optional and can be used to set up logical connections.

INF contains the actual command and data

14443-4 Response:

Length	Data		
	PCB	CID (optional)	INF
1 byte	1 byte	1 byte	Various length

Response length

The response length contains the length of the Data.

Data

Data contains the response block from the tag. Data consists of:

PCB and **CID** as described above

INF contains the response data from the tag. Usually the first byte is the status byte.

Implementation Notes

Please take care of the following notes when implementing this command.

- The Epilogue Field containing a checksum is automatically calculated and appended by the reader.
- Protocol regulating S-Blocks are handled by the reader
- R-Blocks, PCB and chaining must be handled by the host

5.5.2 Extending 't' command

The extending 't' command has more options than the standard 't' command.

With the extended 't' command it is possible to modify the transmission protocol.

Command	Downlink length	Protocol	Option byte(s)	Data
't'	1 byte	1 byte	1-2 byte	n bytes

Downlink length:

Defines the length of the data bytes. Protocol and option bytes are not included.

Protocol:

Value	Protocol	Number of Option bytes
00	ISO 14443A	1
01	ISO 14443B	2
02	ISO 15693	1
03	ICODE	1
04	SR716	1
05	FELICA	1
06	NFCIP1 Initiator	2
07	NFCIP1 Target	2

All other values are RFU and will lead to an error.

5.5.2.1 ISO14443B option bytes mapping

Option byte 0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RxCRC	TxCRC	RFU	RFU	RFU	NoRxSOF	RFU	NoRxEOF

RxCRC:

If this bit is set the CRC checking is enabled and the CRC byte is cut off from the frame.

TxCRC:

If this bit is set a CRC is appended to the transmitted frame.

NoRxSOF:

If this bit is set a missing SOF at the received frame will be ignored.

NoRxEOF:

If this bit is set a missing EOF at the received frame will be ignored.

Option byte 1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NoTxSOF	NoTxEOF	RFU	RFU	RFU	RFU	RFU	RFU

NoTxSOF:

If this bit is set no SOF will be transmitted

NoTxEOF:

If this bit is set no EOF will be transmitted

5.5.2.2 ISO15693 option bytes mapping**Option byte 0**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RxCRC	TxCRC	RFU	RFU	RFU	RFU	RFU	RFU

RxCRC:

If this bit is set the CRC checking is enabled and the CRC byte is cut off from the frame.

TxCRC:

If this bit is set a CRC is appended to the transmitted frame.

5.5.2.3 ICODE option bytes mapping**Option byte 0**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RxCRC	TxCRC	RFU	RFU	RFU	RFU	RFU	RFU

RxCRC:

If this bit is set the CRC checking is enabled and the CRC byte is cut off from the frame.

TxCRC:

If this bit is set a CRC is appended to the transmitted frame.

5.5.2.4 FELICA option bytes mapping

Option byte 0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RxCRC	TxCRC	RFU	RFU	RFU	RFU	RFU	RFU

RxCRC:

If this bit is set the CRC checking is enabled and the CRC byte is cut off from the frame.

TxCRC:

If this bit is set a CRC is appended to the transmitted frame.

5.5.2.5 NFCIP1 option bytes

Option byte 0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Initialization	RFU	Air Baudrate		Parameter	RFU	Mode	RFU

Initialization:

If this bit is set the reader will be initialised with the chosen options.

Air Baudrate:

Bit 5	Bit 4	Description
1	1	424kbaud
1	0	212kbaud
0	1	106kbaud
0	0	Only valid if a baudrate has been chosen before. This must happen at Initializing the reader or at least at the first frame (for example at sending a ATR_REQ)

Parameter:

With this bit it is possible to change the parameter bytes (according to ISO/IEC 18092) of the Initiator or Target at the command ATR (Option byte 1: 02h).

If this bit is set as Initiator the first four bytes of the data will be taken as parameter bytes in order: DIDi, BSi, BRi and PPi.

As Target the first five bytes will be taken as parameter bytes, in order: DIDt, BSt, BRt, TO and PPt.

If it isn't set all data bytes are handled as general bytes.

Mode:

Only valid at initializing the reader. It indicates if the reader will be initialized in active or passive mode.

Bit 1	Description
1	Active Mode
0	Passive Mode

Option byte 1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU	RLS	DSL	DEP	PSL	WUP	ATR	RFU

Bit 1 – ATR:**Initiator:**

Will perform an Attribute Request. If Parameter at Option byte 0 is set. The first four bytes of the data will be used as parameter bytes. The rest of the data is attached to the ATR_REQ as general bytes.

If the bit Parameter isn't set. All data will be attached as general bytes.

At maximum 48 general bytes are allowed!

Target:

The ATR_RES will be handled automatically. With this command you can set the parameter bytes and store general bytes which will be added at the ATR_RES.

If the bit Parameter (Option byte 0: 08h) is set the first five bytes of the data will be used to set the parameter bytes.

Else all data will be stored and attached to the ATR_RES.

Bit 2 – WUP:**Initiator:**

This command will send a Wakeup Request.

Target:

Must be handled from host.

Bit 3 – PSL:**Initiator:**

This command will send a Parameter selection Request.

Target:

Must be handled from host.

Bit 4 – DEP:

Initiator:

This command will send a Data Exchange Protocol Request.

All data will be attached to it. At minimum one byte has to send and at maximum 250 bytes are allowed.

If no data is send, this command is used to receive data.

Target:

This command will send a Data Exchange Protocol Response.

All data will be attached to it. At minimum one byte has to send and at maximum 250 bytes are allowed.

If no data is send, this command is used to receive data.

Bit 5 – DSL:**Initiator:**

This command send a Deselect Request.

Target:

Must be handled from host.

Bit 6 – RSL:**Initiator:**

This command will send a Release Request.

Target:

Must be handled from host.

Example: Initialize as Initiator:

Command	Length	Protocol	Option byte 0	Option byte 1	Data
't'	0x07	0x06	0x98	0x01	00,07,07,32,0A,0B,0C

This will initialize the Initiator in Active-mode and will directly after perform a ATR_REQ at 106kbaud with the parameter bytes DIdi = 0x00, BSi = 0x07, BRi = 0x07, PPi = 0x32 and the general bytes attached to the ATR_REQ will be: 0x0A, 0x0B, 0x0C.

Example: ATR_REQ

Command	Length	Protocol	Option byte 0	Option byte 1	Data
't'	0x07	0x06	0x30	0x01	00,07,07,32,0A,0B,0C

This will perform a ATR_REQ at 424kbaud with the general bytes (cause Parameter is not set) of: 0x00, 0x07, 0x07, 0x32, 0x0A, 0x0B, 0x0C

Example: Initialize a Target:

Command	Length	Protocol	Option byte 0	Option byte 1	Data
't'	0x00	0x07	0x90	0x00	None

This will initialize the Target in Passive-Mode.

Example: DEP_REQ

Command	Length	Protocol	Option byte 0	Option byte 1	Data
't'	0x07	0x06	0x00	0x04	00,07,07,32,0A,0B,0C

To send data according to the ISO/IEC 18092 (NFCIP1 data exchange protocol).

This will send a DEP_REQ at the baudrate which has been chosen at Initializing, sending an ATR_REQ or selecting a Target, with general bytes of: 0x00, 0x07, 0x07, 0x32, 0x0A, 0x0B, 0x0C

Example: Receiving data at Card-Emulation

Command	Length	Protocol	Option byte 0	Option byte 1	Data
't'	0x00	0x06	0x40	0x01	none

Incoming data will be directly passed to the host without modifying.

5.6 Send SAM APDU (T=0, T=1)

5.6.1 Activate/Deactivate level shifter for SAM

Before to use the SAM, it is necessary to activate the SAM with the required power (1.8, 3 & 5 V)

5.6.1.1 Turn ON- Set level shifter 1 & 2 to 1.8V

Command sequence
'pp' 0F0F 'pp' 0F01

5.6.1.2 Turn ON- Set level shifter 1 & 2 to 3V

Command sequence
'pp' 0F0F 'pp' 0F02

5.6.1.3 Turn ON- Set level shifter 1 & 2 to 5V

Command sequence
'pp' 0F0F 'pp' 0F03

5.6.1.4 Turn OFF- Set level shifter 1 & 2 to 0V

Command sequence
'pp' 0F0F

5.6.2 Send SAM APDU (T=0, T=1)

This command sends a custom data block to a SAM.

Option for Contact Interface (SAM).

Command structure

Command	Downlink length	Option byte	Time-out	Transmission factor byte	Return length	SAM holder	Data
'e'	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	n bytes

Command

'e' → This command sends a custom data block to a SAM.

Downlink length

The Downlink length contains the length of the data.

Option byte

This byte contains the transfer options.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0h ... default timeout 1h ... manual timeout	0h...T=0 1h...T=1 2h...Raw mode 3h...RFU		0h ... Transaction 1h ... Activation 2h ... Deactivation 3h ... Warm reset 7h ... Transmit to buffer 8h ... Receive from buffer all other values are RFU and should not be used			

7h Transmit to buffer

This command sends up to 11 bytes into the internal buffer and should be used if the send data will exceeds the host frame limits of 255bytes. Use this command just before using the transaction command, because the transaction command automatically concatenates the buffer data with the data from the transaction command and sends them together to the SAM.

Timeout, transmission factor and return length are not used.

8h Receive from buffer

This command receives the remaining data from the internal buffer.

Note that the internal buffer will be cleared after execution.

Timeout, transmission factor and return length are not used.

Time-out

The time-out byte is used as the communication time-out. It's necessary to set Bit 6 in the Option byte to enable manual timeouts; otherwise the default time-out is used!

One time slice is around 10ms. The longest time-out value allowed is 2.5 seconds (FFh).

Note: If the timeout byte is set to 00h, the default time-out of 100ms is used!

Transmission factor byte

This byte contains the clock rate conversion factor and the baud rate adjustment factor according ISO 7816-3. The default value on start-up is 11h (F/D = 1).

Return length

The value of return length should be set to the expected data length. 00h means receive all data from PICC.

SAM holder

This byte contains the number of the SAM socket to be used; if only 1 SAM is supported by hardware, this byte is not sent.

Data

Data contains the request data sent to the SAM.

Response structure

Status	Length (optional)	Data (optional)
00h	1 byte	Various length

Status

The Status byte contains 00h if the command was successful.

Note that a Deactivation command only returns the Status byte 00h without Length and Data.

Length

The Length contains the length of the data.

Data

The Data contains the received data bytes from the SAM.

Response Errors

Error code	Possible reason
01 00	SAM didn't respond
'E'	No SAM present or SAM not activated yet Invalid SAM APDU format SAM holder byte required/missing
'F'	General failure
0xAF	More data available in receiveBuffer

Example Command - Activation:

Command	Description
'e' 00 81 00 11 00	Activation-Command

Response	Description
00 1B 3B DE 18 FF 81 F1 FE 43 00 3F 07 83 44 45 53 46 69 72 65 38 20 53 41 4D 2D 58 17	ATR response

Example Command - PPS:

Command	Description
'e' 04 A0 00 11 04 FF 11 11 FF	Sends a PPS command

Response	Description
00 04 FF 11 11 FF	PPS response

For more information and examples see [Option for Contact interface \(SAM\)](#).

Important Notes

- ✓ A Transaction or Warm Reset may only be performed on an activated SAM.
- ✓ A SAM should be deactivated before removing from the holder.

5.7 Mifare Specific Commands

5.7.1 Login

Performs an authentication to access one sector of a Mifare® card. Only one sector can be accessed at once.

To authenticate the host can either transmit the authentication key or reference an authentication key stored in the reader's EEPROM.

To store keys in the EEPROM the write master key command is used. It is possible to store up to 32 keys in the EEPROM of the reader. In order to login the sector has to be selected.

Command:

Command	Data		
	1 byte	2 byte	3 – 8 byte (optional)
'I'	Sector	Key type	Key data

Sector (1byte):

The sector you want to authenticate.

Valid range: 00h – 3Fh

Key type (2byte):

Key Type	Description
AA	Authenticate with key type A
FF	Authenticate with key type A, transport key FFFFFFFFFFFFFh
BB	Authenticate with key type B
10h – 2Fh	Authenticate with key type A using stored key 00h – 1Fh
30h – 4Fh	Authenticate with key type B using stored key 00h – 1Fh

Key data (3-8byte)(optional):

The key data is optional. If you want login with a specific not stored key you can add the key you want to login with. The key length is 6 bytes.

A more detailed information is given in the example below.

Response:

Answer	Description
Data	Login Status (1 Byte)
'L'	Login success
'E'	Error: Invalid key format
'F'	Error: General failure
'N'	Error: No tag

Example:

Command	Description
I01AAFFFFFFFFFFFFFFF	Authenticate for sector 1, using key FFFFFFFFFFFFFFFh, key type A
I0114	Authenticate for sector 1, using EEPROM key 4, key type A
I0130	Authenticate for sector 1, using EEPROM key 0, key type B
I0932	Authenticate for sector 9, using EEPROM key 2, key type B
I0110	Authenticate for sector 1, using EEPROM key 0, key type A
I0ABBFF12FFFFFFF35	Authenticate for sector 10, using key FF12FFFFFFF35h, key type B

5.7.1.1 Login with key data from EEPROM

Each key stored in the reader EEPROM can be used as key type A or key type B. If you want to use a key as type A you have to add 10h to the key index. If you want the key as type B you have to add 30h to the key index.

5.7.1.2 Usage of key A or B

Mifare® cards support different crypto keys for every sector. Each key is 32 bit long and is stored in the end of the sector on a card. You can set difference access settings for the keys.

5.7.2 Read Block

This command reads a data block from a memory card.

Command:

Command	Data
'r'	Block address (00h-40h)(1Byte)
'rb'	Block address (1Byte)

Response:

Answer	Description
Data	block data (depends on tag type)
'F'	Error: read failure
'N'	Error: No tag in field
'O'	Error: Operation mode failure

Example:

Command	Description
rb05	Read Block 05h

5.7.3 Read Multiple Block

This command reads multiple data blocks from a sector. This command needs a successful login.

Command:

Command	Data
'rd'	Block address (1Byte), Number of blocks (1Byte)

Response:

Answer	Description
Data	block data (depends on tag type)
'F'	Error: read failure
'N'	Error: No tag in field
'O'	Error: Operation mode failure

Example:

Command	Description
rd0503	Read starting from Block address 05h, 3 blocks
Rd1510	Read starting from block address 15h, 16 (10h) blocks

5.7.4 Read Value Block

Reads a value block from a Mifare® card. The command checks if data is in value block format. This command needs a successful login.

Command:

Command	Data
'rv'	Value Block(1byte)

Response:

Answer	Description
Data	Read value (4 bytes)
'F'	Error: General failure
'I'	Error: value block failure
'N'	Error: No tag in field

Example:

Command	Description
rv04	Read value Block with the address 04

5.7.5 Write Block

This command writes data to a block of a memory card.

Command:

Command	Data
'w'	Block address (1 byte), valid range 00h 40h Data (n bytes)
'wb'	Block address (1 byte) Data (n bytes)

Response:

Answer	Description
Data	Block data (depends on tag type)
'F'	Error: General failure
'N'	Error: No tag in field
'O'	Error: Operation mode failure

Example:

Command	Description
wb0412345678	Writes data 12345678 on block 04.

5.7.6 Write Multiple Block

This command writes sequentially data into the blocks of a sector. A successful login is needed.

Command:

Command	Data
'wd'	Start of Block address (00h-40h)(1Byte) Number of Block to Write(00h-10h)(1Byte) n Datatbytes

Response:

Answer	Description
00h	command was successful
'F'	Error: read failure
'N'	Error: No tag in field
'O'	Error: Operation mode failure

Example:

Command	Description
rd0403ABCDEF1234...	starting Block address 04 , writes 03 blocks Data ABCDEF1234...

5.7.7 Write Value Block

This command formats a block on a Mifare® card as a value block containing a 32-bit value. Value blocks need a complete 16-byte block due to redundant storage. The write value block command needs a successful login.

Command:

Command	Data
'wv'	Value block (1 byte) Value (4 bytes)

Response:

Answer	Description
Data	Written value (4 bytes)
'I'	Error: value block failure
'F'	Error: increment failure
'N'	Error: No tag in field
'U'	Error: Read after write failure

Example:

Command	Description
ww0422552255	Read value Block with the address 04Writes value 22552255h to block 4.

5.7.8 Increment value Block

Increments a value block on a Mifare® card with a defined value. After a write command a read command is executed to check the written data. The command fails if the source block is not in value block format. This command needs a successful login.

Command:

Command	Data
'+'	Block (1 byte) Value (4 bytes)

Response:

Answer	Description
Data	Value (4 bytes)
'I'	Error: General failure
'F'	Error: value block failure
'N'	Error: No tag in field
'X'	Error: Unable to read after write

Example:

Command	Description
+0500000001	adds 1 to value block 5

5.7.9 Decrement Value Block

Decrements a value block on a Mifare® card with a defined value. After a write command a read command is executed to check the written data. The command fails if the source block is not in value block format. The command needs a successful login.

Command:

Command	Data
'D'	Block (1 byte) Value (4 bytes)

Response:

Answer	Description
Data	Value (4 bytes)
'I'	Error: General failure
'F'	Error: value block failure
'N'	Error: No tag in field
'X'	Error: Unable to read after write

Example:

Command	Description
-0500000001	subtract 1 to value block 5

5.7.10 Copy Value Block

Copies one value block on a Mifare® card to another block of the same sector. Used for backup and error recovery. The command needs a successful login.

Command:

Command	Data
'='	Source block (1 byte) Target block (1 byte)

Response:

Answer	Description
Data	New value of target block (4 bytes).
'I'	Error: value block failure
'F'	Error: increment failure
'N'	Error: No tag in field
'X'	Error: Unable to read after write

Example:

Command	Description
=0405	copy value block 4 to block 5 .
=0506	copy value block 5 to block 6

5.7.11 Write Master Key

This command stores a MIFARE Standard key into the master key memory of the reader device. The reader can store up to 32 keys.

Command:

Command	Data
'wm'	Key number (1 byte) 00h 1Fh Key (6 bytes)

Response:

Answer	Description
Data	Written key (6Byte)
'F'	Error: Write failure

Example:

Command	Description
wm02123456789ABC	Store key 123456789ABCh in EEPROM (key number 2).

The memory area reserved for the master keys is read protected. Keys can only be written but not read back. Nevertheless the reader returns correct error messages if the writing process fails.

A verification of the master key can only be done using an appropriate card and a correct login.

It is possible to use every stored key for key A and key B for authentication.

Each key is 6 bytes long.

6 Tags

6.1 Overview to the Mifare Family

Type	Concepts and Features	Properties
Mifare Ultralight	Memory No security 64 byte EEPROM	Low cost Plain data storage Usages: ID Card, one time ticket
Mifare 1k Mifare 4k	Memory Crypto1 stream cipher 1kb/4kb EEPROM	Low cost Medium security Basic value handling Limited multi application Usages: ID card, medium cost ticket, data storage
Mifare Plus	Memory Crypto1, AES 2kb/4kb EEPROM	Field upgrade from Mifare 1k, 4k mode High security option that can be upgraded in the field Usage: like Mifare 1k, 4k
Mifare Desfire	Smartcard 2kb/4kb/8kb EEPROM (SAM Version: 72kb) 3DES	Very high security Flexible multi applications Flexible file system for data, transaction logging, secure access Usage: secure ID, secure ticketing

6.2 Mifare Desfire

The following is a brief overview only. Please refer to the confidential NXP manuals for details on Desfire.

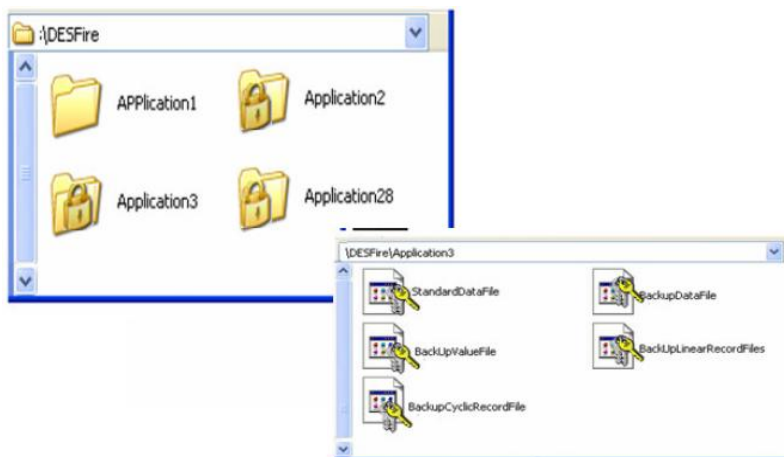
Desfire cards use special commands called APDUs. The reader supports sending and receiving APDUs.

Desfire supports different security algorithms like AES, DES or TDES. Also MACing is supported.

6.2.1 Desfire Memory Organisation

A Mifare® Desfire is a contactless smartcard with a built in operating system. It can store data in files and directories similar to a PC's file system. State of the art security is provided for protecting from unauthorized access. The memory can hold 28 different applications. Every application can store up to 16 files. Every application can be configured to use up to 14 keys with different access rights.

Please refer to NXP documents for details and latest information.



6.2.2 Desfire States

- **Activate Card**
- **Select application**
- **Login application**
- **Create/Select file**
- **Read/Write file**
- **Commit / Abort transaction**

Activate Card:

To activate a Desfire card it has to be selected. The select returns the card's 7 byte UID. A RATS command must be performed according to ISO 14443-4 to communicate with the card.

Select application:

After the card activation the main application (AID 00 00 00) is selected automatically. An application is comparable with a directory. To change to another application simply select it.

Login application:

Applications can be protected using different security mechanisms like TDES, DES, AES or MACing. A protected application needs an authentication to access files, etc.

Create/Select file:

Within one application up to 16 files can be created. Depending on the application's security settings an authentication can be required to create files.

Select one file to read and write its contents.

Read/ Write file:

Mifare® Desfire cards support different file structures for different purposes. Read and Write operations may need an authentication, depending on the security settings.

6.2.3 Command structure

To use a Desfire card you can send 7816 APDUs to the tag as well as Desfire native APDUs. After a select - RATS sequence (see ISO/IEC 14443-4) the card is in the 14443-4 layer mode and ready to operate.

6.2.4 Security related commands – Overview

Command	Description
0x0A	Authenticate (3)DES
0x1A	Authenticate DES in ISO CBC send mode
0xAA	Authenticate AES
0x54	Change Key Settings
0x45	Get Key Settings
0xC4	Change Key
0x64	Get Key Version

Example:

Command:

Command	Description
't' 030F026400	Get Key Version for Key 00

Response:

Answer	Description
03020000	Response of correct command

6.2.5 PICC level commands – Overview

Command	Description
0xCA	Create Application
0xDA	Delete Application
0x6A	Get Applications IDs
0x6E	Free Memory
0x6D	Get DF Names
0x45	Get Key Settings
0x5A	Select Application
0xFC	Format PICC
0x60	Get Version
0x51	Get Card UID

6.2.6 Application level commands – Overview

Command	Description
0x6F	Get File IDs
0xF5	Get File Settings
0x5F	Change File Settings
0xCD	Create Std Data File
0xCB	Create Backup Data File
0xCC	Create Value File
0xC1	Create Linear Record File
0xC0	Create Cyclic Record File
0xDF	Delete File
0x61	Get ISO File IDs

6.2.7 Data Manipulation Commands – Overview

Command	Description
0xBD	Read Data
0x3D	Write Data
0x6C	Get Value
0x0C	Credit
0xDC	Debit
0x1C	Limited Credit
0x3B	Write Record
0xBB	Read Records
0xEB	Clear Record File
0xC7	Commit Transaction
0xA7	Abort Transaction

6.2.8 Sample APDUs (Mifare DESFire EV1 8KByte)

Example: Select Tag

Command:

Command	Description
'S'	Select DESFire EV1TAG

Response:

Answer	Description
06 04 5E 6E C9 92 26 80	1 byte - Extended ID 7 byte - UID of TAG

Example: Send RATS (FSD=256, CID=0x00)

Command:

Command	Description
't' 02 0F E0 80	Send RATS to DESFire EV1 TAG

Response:

Answer	Description
06 06 75 77 81 02 80	1 byte - length of data 6 byte - ATS of DESFire EV1 TAG

Example: Create Application (PCB = 0x02 (CID is not following), Command = 0xCA, AID = 12 34 56, Key Sett. 1 = 0x00, Key Sett. 2 = 0x01)

Command:

Command	Description
't' 07 0F 02 CA 12 34 56 00 01	Create Application with AID 123456

Response:

Answer	Description
02 02 00	1 byte - length of data 1 byte – PCB 1 byte - Status

Example: Get Application ID's (PCB = 0x03 (CID is not following), Command = 0x6A)

Command:

Command	Description
't' 02 0F 03 6A	Get all Application ID's on TAG

Response:

Answer	Description
05 02 00 12 34 56	1 byte – length of data 1 byte – PCB 1 byte – Status N bytes – AID's on TAG

6.2.9 Desfire using ISO/IEC 7816-4

Desfire tags support several ISO 7816-4 commands as well as the wrapping of native Desfire commands according to ISO7816-4.

Note that ISO 7816-4 does not support encoded communication. READ BINARY for example can only read unencrypted files.

The first APDU after entering ISO 14443-4 decides whether ISO 7816-4 or native framing is used. If this first command uses ISO 7816-4 framing all following commands must use the same framing. The same applies for native framing. To switch the framing a new Select-RATS sequence must be performed.

See ISO/IEC 7816-4 for more details.

6.2.9.1 ISO/IEC 7816-4 – Basic inter-industry commands

Command	Description
0xA4	SELECT FILE
0xB0	READ BINARY
0xD6	UPDATE BINARY
0xB2	READ RECORDS
0xD2	UPDATE RECORD
0xE2	APPEND RECORD
0x84	GET CHALLENGE
0x88	INTERNAL AUTHENTICATE
0x82	EXTERNAL AUTHENTICATE

Example:**Command:**

Command	Description
⚠ 0A 0F 02 00 A4 04 00 03 0A 0B 0C 00	SELECT FILE with the DF name 0A 0B 0C

Response:

Answer	Description
03 02 90 00	Response of correct command

6.2.9.2 Wrapping of native Desfire APDUs

The wrapping of native Desfire APDUs is done according to ISO 7816-4 “case4”.

Command:

CLA	INS	P1	P2	Lc	Data	Le
0x90	Desfire cmd code	0x00	0x00	Length of Data	Desfire command parameters	0x00

Response:

Data (optional)	SW1	SW2
Desfire response data	0x91	Status byte of Desfire command

Example:**Command:**

Command	Description
Ⓣ 0A 0F 02 90 5A 00 00 03 00 00 01 00	Select the Application with the AID 00 00 01

Response:

Answer	Description
03 02 91 00	Response of correct command

For more information about the Desfire card refer to Desfire Documentation, Philips,
<http://www.semiconductors.philips.com>

6.3 Mifare Plus

The following is a brief overview only. Please refer to the confidential NXP manuals for details on MifarePlus.

Mifare Plus supports different operating modes called levels. This enables to field upgrade contactless applications smoothly.

Mifare Plus on Level 1 and 2 emulates a Mifare 4k, Mifare1k and Mifare Mini. This mode is intended to support existing infrastructures. Notice that the encryption of Mifare Classic is outdated. For new designs additional security measures or level 3 are recommended.

Level 1 supports short (4 byte) UIDs. Unfortunately duplicate numbering can occur since most available numbers had been used up already. This level will run on most of the existing infrastructures.

Level 2 supports long UIDs (7 Byte) and an optional higher security. Duplicate numbers are no issue here. Notice that not all devices on the market support long UIDs as provided optionally in this level, so you might run into compatibility problems within existing infrastructures. Zebra Technologies readers do support all options.

Level 3 supports strong encryption and authentication using AES. The reader supports sending and receiving ISO/IEC 14443-4 APDUs.

Mifare Plus is available in two versions, Mifare S and Mifare X.

6.3.1 Mifare Plus Memory Organisation

At level 1 a fixed memory structure is provided. Available with 2kb or 4kb EEPROM

The structure is identical with the Mifare 4k structure.

6.3.2 Instruction

You can use a Mifare Plus card in four security Levels.

Level 0: Product delivery. The card accepts both Mifare Classic commands and ISO 14443-4 APDUs.

Level 1: In this level the card is compatible with Mifare Classics cards. The card accepts Mifare Classic commands. Authentication can be done with CRYPTO1 as well as with an AES key called SL1 Card Authentication Key. An ISO 14443-4 activation is provided for level switches only.

Level 2: This level is only available for Mifare Plus X tags. It supports both ISO 14443-4 activation for a limited command set in ISO 14443-4 mode and the backwards compatibility mode for using any Mifare Classic commands. Authentication is primarily done using AES encryption.

Level 3: In this level you can use ISO14443-4 APDUs only. The backwards compatibility protocol is not available in this level. Authentication with AES-keys is required in order to access data. CRYPTO1 authentication is not possible in Level 3.

Before you can use a Mifare Plus card in a higher level than level 0 you have to personalize it. Therefore you have to write the Card Configuration Key, the Card Master Key, the Level Switch Keys (if available) and the sector AES keys (recommended) using the WritePerso command. The CommitPerso command finalizes the personalization.

In higher levels a level switch is done by authentication with the proper Level Switch Key. Note that a level switch to a lower level is not possible.

6.3.3 Command overview Level 3 for Mifare Plus S

The following commands are available in security level 3 for Mifare Plus S tags. Mifare Plus X tags additionally offer a rich set of read and write commands, value operations and support proximity checks and virtual cards.

Command code	Command	Description
0x70	First Authenticate (part1)	First authentication
0x72	Authenticate (part2)	2 nd step within the authentication
0x76	Authenticate	Performs a following (re-) authentication
0x78	ResetAuth	Reset the authentication
0x33	Read Plain MACed	Reading in plain, MAC on command and MAC on response
0xA1	Write encrypted MACed	Write encrypted, MAC on command and MAC on response
0xA3	Write Plain MACed	Writing in plain, MAC on command and MAC on response
0x4B	Virtual Card Support Last	Select the virtual card and retrieve the UID
0x48	Deselect Virtual Card	Deselect the virtual card

Example: First Authenticate (part1) in level 3

(Select and RATS-Command without using CID/NAD were performed)

Command:

Command	Description
't' 05 0F 02 70 01 40 00	First Authenticate with the AES-key 4001 (LittleEndian mode)

Response:

Answer	Description
12 02 90 BC F3 FC C6 B8 5E 91 9F CC 5C 94 F0 88 0F 2A E8	Status code from the PICC followed by an encrypted 16 byte random number.

An exemplary transaction with first and following authentication, write and read operations is given in the Mifare Plus datasheet.

For more information about the Mifare Plus card refer to Mifare Plus Objective data sheet Documentation, Philips, <http://www.semiconductors.philips.com>

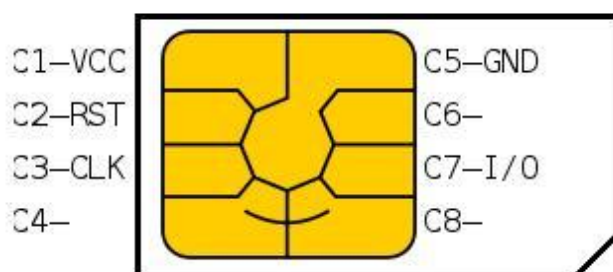
7 Option for Contact Interface (SAM)

Support for contactless smartcards (SAM) is available as an option.

Contact smartcards are used for fraud safe applications. Typically a SAM is used to digitally sign transactions, encrypt sensitive information or to establish secure communication channels over insecure networks.

7.1 SAM

7.1.1 Pin Configuration



Pin Definitions and Functions

Card Contact	Symbol	Function
C1	VCC	Supply Voltage
C2	RST	Control input (Reset signal)
C3	CLK	Clock input
C4	N.C.	Not connected
C5	GND	Ground
C6	N.C.	Not connected
C7	I/O	Bi-directional data line
C8	N.C.	Not connected

7.1.2 General Description

Communication

The communication interface is implemented according to ISO 7816-3 [2]. Both protocols T=0 and T=1 are supported.

The clock frequency of the interface is 4Mhz.

Supported Baudrates:

FI DI	372 Baudrate[kb/s]	558 Baudrate[kb/s]	128 (non ISO) Baudrate[kb/s]
1	10752,68817	7168,458781	1000000
2	21505,37634	14336,91756	
4	43010,75269	28673,83513	
8	86021,50538	57347,67025	
16	172043,0108	114695,3405	
32	344086,0215	229390,681	
RFU			
12	129032,2581	86021,50538	
20	215053,7634	143369,1756	

7.2 Sample APDUs (up to 4 SAM supported)

The following commands had been tested with **Infineon EasySAM SLF 9620**, Sicrypt in SAM holder 1. It's necessary to add the number of the SAM holder to the command. Note that APDUs may vary on other card operating systems.

Example: Activation (Version 2, holder 1)

Command:

Command	Description
'e' 00 81 00 00 00 01	Activation

Response:

Answer	Description
00 18 3B DF 18 00 81 31 FE 65 45 41 53 01 42 53 FF FF 01 00 53 C9 7E 33 00 7D	Response of ATR

Example: PPS (Version 2, holder 1)

Command:

Command	Description
'e' 04 A0 00 11 04 01 FF 11 11 FF	PPS

Response:

Answer	Description
00 04 FF 11 11 FF	Response of PPS

Example: Warm reset (Version 2, holder 1)

Command:

Command	Description
'e' 00 83 00 00 00 01	Warm reset

Response:

Answer	Description
00 18 3B DF 18 00 81 31 FE 65 45 41 53 01 42 53 FF FF 01 00 53 C9 7E 33 00 7D	Response of ATR

Note that some SAMs have different ATRs after a cold reset (Activation) and a warm reset.

Example: Deactivation (Version 2, holder 1)

Command:

Command	Description
'e' 00 82 00 00 00 01	Deactivation

Response:

Answer	Description
00	SAM is deactivated now

Example: Select Application on EasySAM SLF 9620 in holder 1

Command:

Command	Description
'e' 0C A0 00 11 00 01 00 00 08 00 A4 08 00 02 A7 00 12 13	Select Application

Response:

Answer	Description
00 18 00 00 14 85 10 00 00 A7 00 02 00 FF FF FF 01 05 08 00 10 00 00 90 00 57	response

Example: Determine Administrative Keys on EasySAM SLF 9620 in holder 1

Command:

Command	Description
'e' 0E A0 00 11 00 01 00 40 0A 00 A4 08 00 04 A7 00 00 2B 0F 61	SelectEF by FID

Response:

Answer	Description
00 15 00 40 11 85 0F 00 14 00 2B 04 00 3F FF FF 00 04 00 00 90 00 4B	response

Example: shut down EasySAM SLF 9620 in holder 1**Command:**

Command	Description
'e' 00 82 00 00 00 01	Shut down SAM1

Response:

Answer	Description
00	response

8 Appendix A: TLV structure

Type				
0xAC	Subtype	Name	Data size	Data
	0xAA	ATQA	2 bytes	ATQA
	0xAB	ATQB	12 bytes	ATQB
	0xC1	Cascade level	1 bytes	0x01=CL1 0x02=CL2 0x03=CL3
	0xE1	Extended ID	1 bytes	Refer Extended ID
	0xD1	Protocol	1 bytes	0x01=ISO15963 0x02=SR176 0x03=Felica 0x0A=ISO type A 0x0B= ISO type B
	0xBA	SAK	1 bytes	SAK
	0x1D	UID	1+n bytes	UID length UID
	0xA5	ATS	1+n bytes	ATS length ATS
	0xD5	Baudrate of airinterface	1Byte	Baudrates: 0x00 ->106kbaud 0x02 ->212kbaud 0x04 -> 424kbaud 0x08 -> 848kbaud

9 Appendix B: Reader speed optimization

to increase the performance you have following options:

- change the ResetOffTimeRegister(addr: 0x14) to a lower value
- change the ResetRecoveryTimeout(addr: 0x15) to a lower value
- activate only the tags which are you using.

Example:

If you are using only Mifare tags you can change following Register to a lower value:

Name	Address	Value
ResetOffTimeRegister	0x14	0x03
ResetRecoveryTimeout	0x15	0x04/0x05
OperationmodeRegister	0x0E	0x01

10 Appendix C: Complete Command List

Command Group	Command	Description	FW Version	Chapter
			TA 1.09.10	
System	'b'	Get Serial Number	□	5.2.1
	'poff '/pon'	RF Field off/on	□	5.2.2
	'pp'	Set/Get User ports	□	5.2.3
	'pr'/'pw'	Read/Write User ports	□	5.2.4 / 5.2.5
	'v'	Get Version	□	5.2.6
	'vr'	Get Real version	□	5.2.7
	'vs'	Set Version	□	5.2.8
	'x'	Reset	□	5.2.9
	'vh'	Get HW version	□	5.2.10
EEPROM	're'	Read Reader EEPROM	□	5.3.1

	'we'	Write Reader EEPROM	<input type="checkbox"/>	5.3.2
	'nr'	Set Resister Dynamical	<input type="checkbox"/>	5.3.3
	'nv'	Save Resister Settings	<input type="checkbox"/>	5.3.4
	'ne'	Read Dynamical Settings	<input type="checkbox"/>	5.3.5
	'f'	Reset to Default	<input type="checkbox"/>	5.3.6
TAG	's'	Select	<input type="checkbox"/>	5.4.1
	'm'	Multitag List / Multitag Select	<input type="checkbox"/>	5.4.2 / 5.4.3
	'h'	Highspeed Select	<input type="checkbox"/>	5.4.4
	'ns'	Extended Select	<input type="checkbox"/>	5.4.5
	'nm'	Extended Multilist	<input type="checkbox"/>	5.4.6
	'nh'	Extended Highspeed	<input type="checkbox"/>	5.4.7
ISO 14443-4	't'	Send 14443-4 APDU	<input type="checkbox"/>	5.5
	't'	Extending 't'	<input type="checkbox"/>	5.5.1
SAM	'e'	Send SAM APDU	<input type="checkbox"/>	5.6.2

To be continued on next page...

Command Group	Command	Description	XXL++ FW Version	Chapter
			TA 1.09.08	
Mifare Specific	'l'	Login	<input type="checkbox"/>	5.7.1
	'r' / 'rb'	Read Block	<input type="checkbox"/>	5.7.2
	'rd'	Read Multiple Block	<input type="checkbox"/>	5.7.3
	'rv'	Read Value Block	<input type="checkbox"/>	5.7.4
	'w' / 'wb'	Write Block	<input type="checkbox"/>	5.7.5
	'wd'	Write Multiple Block	<input type="checkbox"/>	5.7.6
	'wv'	Write Value Block	<input type="checkbox"/>	5.7.7
	'+'	Increment Value Block	<input type="checkbox"/>	5.7.8
	'-'	Decrement Value Block	<input type="checkbox"/>	5.7.9
	'='	Copy Value Block	<input type="checkbox"/>	5.7.10
	wm	Write Master Key	<input type="checkbox"/>	5.7.11