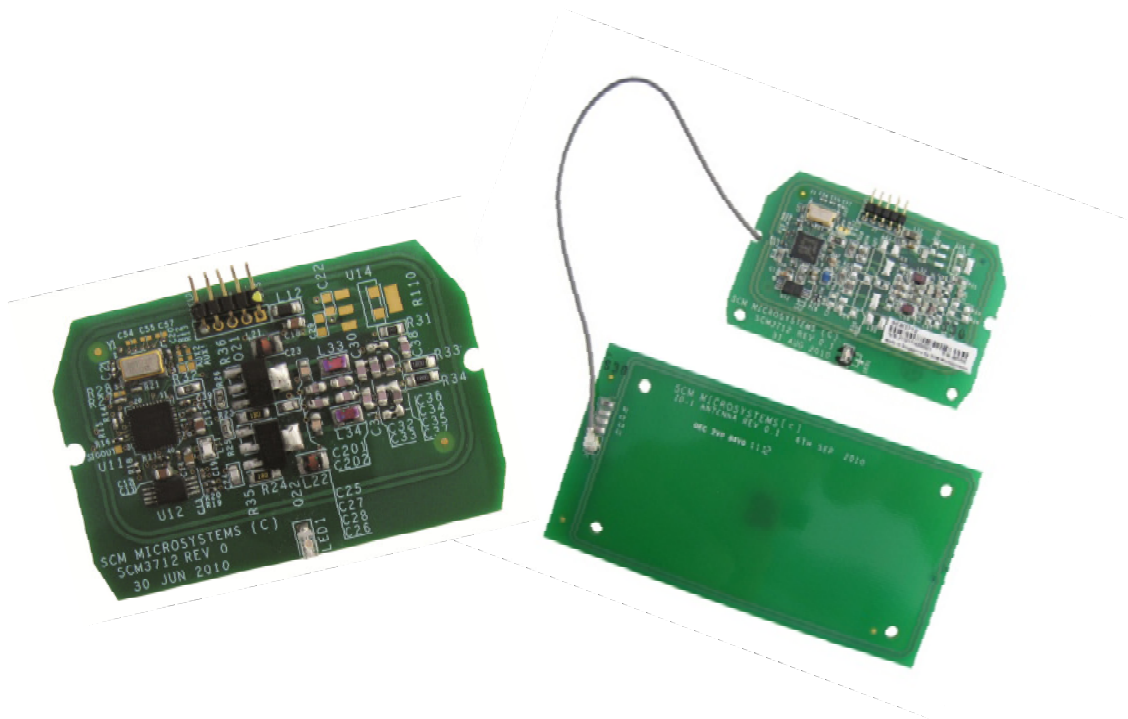


Identive

Reference Manual – version 1.0



# SCM3712

## NFC Reader Boards



# Reference manual

## SCM3712 NFC Reader Boards

---

© Identive GmbH  
Oskar-Messter-Straße, 13  
85737 Ismaning  
Germany

Phone +49 89 9595 5000 • Fax +49 89 9595 5555

---

## Document history

Date	Version	Description of change
01.07.2011	1.0	Initial Version

## Contact information

<http://www.identive-group.com/en/index.php>

For sales information, please email [sales@identive-group.com](mailto:sales@identive-group.com)

# Table of Contents

1. Legal information.....	6
1.1. Disclaimers.....	6
1.2. Licenses .....	6
1.3. Trademarks .....	6
2. Introduction to the manual.....	7
2.1. Objective of the manual .....	7
2.2. Target audience .....	7
2.3. Product version corresponding to the manual .....	7
2.4. Definition of various terms and acronyms.....	8
2.5. References.....	9
2.6. Conventions .....	10
3. General information about SCM3712 .....	11
3.1. SCM3712 key benefits.....	11
3.2. SCM3712 key features .....	11
3.3. SCM3712 product family ordering information .....	12
3.4. Contactless communication principles and SCM3712 usage recommendations.....	14
3.4.1. Power supply .....	16
3.4.2. Data exchange.....	16
3.4.3. Recommendations .....	17
3.5. Applications.....	18
3.5.1. General .....	18
3.5.2. Supporting applications provided by Identive .....	19
4. SCM3712 characteristics .....	20
4.1. SCM3712 high level architecture .....	20
4.1.1. Block diagrams .....	20
4.1.2. Software architecture .....	21
4.2. Quick reference data.....	22
4.2.1. SCM3712 mechanical dimensions .....	22
4.2.2. LED Status indication .....	23
4.2.3. SCM3712 Electrical Specification.....	24
5. Hardware.....	25
5.1. USB Connector Pin-Out.....	25
5.2. USB Connector Signal Min-/Max-Ratings.....	25
5.3. Antenna.....	26
6. Software modules.....	27
6.1. Installation .....	27
6.1.1. Command line parameters for installation .....	29
6.1.2. Command line parameters for de-installation.....	29
6.2. Utilities.....	29
6.3. Driver.....	30
6.3.1. SCM3712 listing.....	30
6.3.2. Supported operating systems.....	30
6.3.3. PC/SC 2.0 compliant ATR .....	31

7. Firmware .....	36
7.1.1. Transport protocol.....	36
7.1.2. Automatic PPS.....	36
8. Commands description .....	37
8.1. Generic APDUs.....	37
8.1.1. Get UID Command .....	37
8.1.2. Get DATA Command.....	38
8.1.3. T=CL user Command .....	39
8.1.4. PASS_THROUGH command .....	40
8.2. Set of APDU for contactless storage user tokens.....	41
8.2.1. STORAGE_CARD_CMDS_READ_BINARY.....	41
8.2.2. STORAGE_CARD_CMDS_WRITE_BINARY .....	43
8.2.3. STORAGE_CARD_CMDS_LOAD_KEYS.....	45
8.2.4. STORAGE_CARD_CMDS_AUTHENTICATE.....	46
8.2.5. STORAGE_CARD_CMDS_VALUE_BLOCK .....	48
8.3. Set of APDU for ISO/IEC 14443-4 user tokens .....	49
8.3.1. T=CL Command .....	49
8.4. MIFARE DESFire commands .....	50
8.5. Identive specific APDU set.....	50
8.5.1. Commands for communicating with NFC Forum Tags Type 1 .....	50
8.5.2. Commands for communicating with NFC Forum Tags Type 2 .....	56
8.5.3. Commands for communication with NFC Forum Tags Type 3 .....	56
8.5.4. Commands for communicating with NFC Forum Tags Type 4 .....	58
8.6. Escape IOCTL's supported in SCM3712.....	59
8.6.1. READER_CNTLESS_GET_ATS_ATQB.....	59
8.6.2. READER_GET_CARD_TYPE_POLLING .....	60
8.6.3. READER_CNTLESS_SET_TYPE.....	60
8.6.4. READER_CNTLESS_RF_SWITCH.....	61
8.6.5. READER_CNTLESS_DISABLE_PPS.....	61
8.6.6. READER_ENABLE_DISABLE_848 .....	62
8.6.7. READER_CNTLESS_BAUDRATE.....	62
8.6.8. READER_FORCE_BAUDRATE.....	63
8.6.9. READER_DISABLE_NAK_POLLING .....	64
8.6.10. FELICA_PASSTHROUGH .....	64
8.7. Vendor IOCTL used for P2P mode of operation.....	65
Generic IOCTLs .....	65
8.7.1. IOCTL_GET_CARD_TYPE .....	65
8.7.2. P2P Mode – Generic IOCTL.....	66
8.7.3. IOCTL_GET_OR_SET_RW_P2P_MODES .....	68
8.8. P2P Initiator Mode IOCTLs .....	69
8.8.1. IOCTL_INITIATOR_POLL .....	69
8.8.2. IOCTL_INITIATOR_CONNECT.....	70
8.8.3. IOCTL_INITIATOR_TRANSCEIVE .....	71
8.8.4. IOCTL_INITIATOR_DISCONNECT.....	71
P2P Target Mode IOCTLs .....	72
8.8.5. IOCTL_TARGET_RECEIVE.....	72
8.8.6. IOCTL_TARGET_SEND.....	74
9. Annexes .....	75
9.1. Annex A.....	75
9.1.1. Status words table .....	75
9.1.2. Further information about PC/SC .....	75
9.2. Annex B – Mechanical drawings.....	76
9.2.1. SCM3712 and SCM3712 NFC .....	76



## **IDENTIVE**

9.2.2.	SCM3712 EA and external antenna board .....	77
9.3.	Annex C – Installation Guidelines .....	79
9.3.1.	SCM3712 & SCM3712 NFC Mount on Metal Instruction .....	80
9.3.2.	SCM3712 ANT Mount on Metal Instruction .....	80
9.3.3.	SCM3712 & SCM3712 NFC Mount in Metal Environment Instruction .....	81
9.3.4.	SCM3712 ANT Mount in Metal Environment Instruction .....	81
9.3.5.	Recommended distance between readers .....	82
9.4.	Annex D – Certifications .....	83

## LIST OF FIGURES

Fig.3-1: SCM3712 in a Terminal Use Application .....	14
Fig.3-2: SCM3712 in a Kiosk Use Application.....	15
Fig.3-3: SCM3712 based system configuration in a Terminal Use Application .....	18
Fig.3-4: SCM3712 based system configuration in a Kiosk Use Application .....	19
Fig.4-1: SCM3712 Reader Board Block Schematic .....	20
Fig.4-2: SCM3712 NFC Reader Board Block Schematic.....	20
Fig.4-3: SCM3712 EA Reader Board Block Schematic .....	20
Fig.4-4: SCM3712 xxx software layer overview .....	21
Fig.9-1: SCM3712 & SCM3712 NFC Reader PCB top view.....	76
Fig.9-2: SCM3712 EA Reader PCB top view .....	77
Fig.9-3: External Antenna PCB top view .....	78
Fig.9-4: SCM3712 & SCM3712 NFC Mount on Metal Instruction.....	80
Fig.9-5: SCM3712 EA Mount on Metal Instruction.....	80
Fig.9-6: SCM3712 & SCM3712 NFC Mount in Metal Environment Instruction.....	81
Fig.9-7: SCM3712 EA Mount in Metal Environment Instruction.....	81
Fig.9-8: SCM3712 EA Mount in Metal Environment Instruction.....	82



## LIST OF TABLES

Table 4.1: SCM 3712 xxx mechanical data overview .....	22
Table 4.2: LED Status indication .....	23
Table 4.3: Electrical Specification .....	24
Table 5.1:USB Connector Pin-Out .....	25
Table 5.2: USB Connector Min-/Max Ratings, SCM3712 NFC & SCM3712 EA .....	25
Table 5.3: USB Connector Min-/Max Ratings, SCM3712 .....	25

# 1. Legal information

## 1.1. Disclaimers

The content published in this document is believed to be accurate. Identive does not, however, provide any representation or warranty regarding the accuracy or completeness of its content and regarding the consequences of the use of information contained herein. If this document has the status "Draft", its content is still under internal review and yet to be formally validated.

Identive reserves the right to change the content of this document without prior notice. The content of this document supersedes the content of previous versions of the same document. The document may contain application descriptions and/or source code examples, which are for illustrative purposes only. Identive provides no representation or warranty that such descriptions or examples are suitable for the application that the reader may want to use them for.

Should you notice problems with the provided documentation, please provide your feedback to [support@identive-group.com](mailto:support@identive-group.com).

## 1.2. Licenses

If the document contains source code examples, they are provided for illustrative purposes only and subject to the following restrictions:

- You MAY at your own risk use or modify the source code provided in the document in applications you may develop. You MAY distribute those applications ONLY in form of compiled applications.
- You MAY NOT copy or distribute parts of or the entire source code without prior written consent from Identive.
- You MAY NOT combine or distribute the source code provided with Open Source Software or with software developed using Open Source Software in a manner that subjects the source code or any portion thereof to any license obligations of such Open Source Software.

If the document contains technical drawings related to Identive products, they are provided for documentation purposes only. Identive does not grant you any license to its designs.

## 1.3. Trademarks

MIFARE is a registered trademark of NXP Semiconductors BV.

FeliCa is a registered trademark of Sony Corporation.

Jewel and Topaz are trademarks of Innovision Research and Technology Plc.

Windows is a registered trademark of Microsoft Corporation

## 2. Introduction to the manual

### 2.1. Objective of the manual

This manual provides an overview of the hardware and software features of the SCM3712 NFC Reader Boards, hereafter referred to as “SCM3712”.

This manual describes in details interfaces and supported commands available for developers using SCM3712 in their applications.

### 2.2. Target audience

This document describes the technical implementation of SCM3712.

The manual targets software developers. It assumes knowledge about 13.56 MHz contactless technologies like ISO/IEC 14443 and commonly used engineering terms.

Should you have questions, you may send them to [support@identive-group.com](mailto:support@identive-group.com).

### 2.3. Product version corresponding to the manual

Item	Version
Hardware	0.2
Firmware	2.7.0
Driver	1.08
Installer	1.6

## 2.4. Definition of various terms and acronyms

Term	Expansion
APDU	Application Protocol Data Unit
ATR	Answer to Reset, defined in ISO7816
ATS	Answer to Select, defined in ISO14443
Byte	Group of 8 bits
CCID	Chip Card Interface Device
CID	Card Identifier
CL	Contactless
CLA	Class byte defined in ISO 7816
DFU	Device Firmware Upgrade
FeliCa™	Sony contactless technology standardized in ISO18092, technology underlying the NFC Forum tag type 3
INS	Instruction byte defined in ISO7816
Jewel/Topaz	Innovision contactless technology, technology underlying the NFC Forum tag type 1
LED	Light emitting diode
MIFARE	The ISO14443 Type A with extensions for security (NXP)
NA	Not applicable
NAD	Node Address
NDEF	NFC Data Exchange Format: data structure defined by the NFC Forum for NFC Forum tags.
NFC	Near Field Communication
Nibble	Group of 4 bits. 1 digit of the hexadecimal representation of a byte. <i>Example:</i> 0xA3 is represented in binary as (10100011)b. The least significant nibble is 0x3 or (0011)b and the most significant nibble is 0xA or (1010)b
P2P	Peer – to – Peer
PCB	Protocol Control Byte
PCD	Proximity Coupling Device
PC/SC	Personal Computer/Smart Card: software interface to communicate between a PC and a smart card
PICC	Proximity Integrated Chip Card
PID	Product ID
PPS	Protocol Parameter Selection
Proximity	Distance coverage till ~10 cm.
PUPI	Pseudo unique PICC identifier
RFU	Reserved for future use
RF	Radio Frequency
SNR	Serial Number
SW1 SW2	Status word defined in ISO7816
UID	Unique IDentifier
USB	Universal Serial Bus
VID	Vendor ID
(xyz)b	Binary notation of a number x, y, z $\in \{0,1\}$
0xYY	The byte value YY is represented in hexadecimal

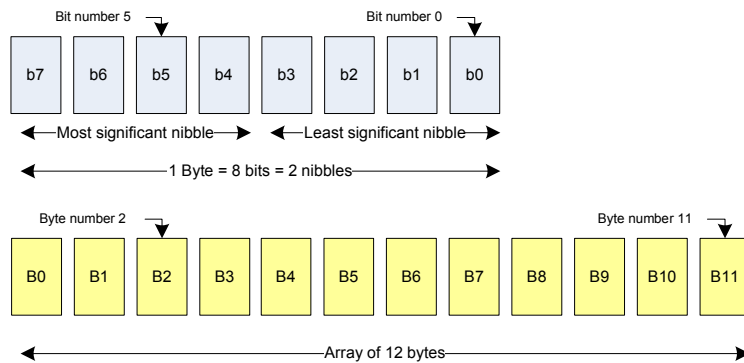
## 2.5. References

Doc ref in the manual	Description	Issuer
ISO/IEC 7816-4	Identification cards - Integrated circuit(s) cards with contacts Part 4: Inter Industry commands for interchange ISO/IEC 7816-4: 1995 (E)	ISO / IEC
ISO/IEC 14443-4	Identification cards — Contactless integrated circuit(s) cards — Proximity cards Part 4: Transmission protocol ISO/IEC 14443-4:2001(E)	ISO / IEC
ISO/IEC 18092	Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1) ISO/IEC 18092:2004(E)	ISO / IEC
NFC Forum tag type 1	NFCForum-TS-Type-1-Tag_1.0	NFC Forum
NFC Forum tag type 2	NFCForum-TS-Type-2-Tag_1.0	NFC Forum
NFC Forum tag type 3	NFCForum-TS-Type-3-Tag_1.0	NFC Forum
NFC Forum tag type 4	NFCForum-TS-Type-4-Tag_1.0	NFC Forum
PC/SC	Interoperability Specification for ICCs and Personal Computer Systems v2.01	PC/SC Workgroup
NFC wrapper	User manual of the NFC wrapper. This manual is part of IDENTIVE's Contactless SDK.	Identive
CCID	Specification for Integrated Circuit(s) Cards Interface Devices 1.1	USB-IF
USB	Universal Serial Bus Specification 2.0	USB-IF

## 2.6. Conventions

Bits are represented by lower case 'b' where followed by a numbering digit.

Bytes are represented by upper case 'B' where followed by a numbering digit.



### Example:

163 in decimal is represented

- in hexadecimal as 0xA3
- in binary as (10100011)b

The least significant nibble of 0xA3 is

- 0x3 in hexadecimal
- (0011)b in binary

The most significant nibble of =xA3 is

- 0xA in hexadecimal
- (1010)b in binary

## 3. General information about SCM3712

### 3.1. SCM3712 key benefits

The SCM3712 product family has been designed for an easy integration into various devices.

The SCM3712 form factors have been optimized for best RF performance and for minimizing the footprint for integration even into very compact devices.

The state of the art multi-protocol feature set of SCM3712 qualifies it to be used in a wide range of applications such as payment, loyalty and ID schemes, or to enable devices with NFC connectivity.

As a latest generation product, SCM3712 can be supported by IDENTIVE's middleware that resides above the PC/SC API and offers an optimized portability of applications and abstraction of smart card related details that need to be handled by applications developed on top of the PC/SC API.

### 3.2. SCM3712 key features

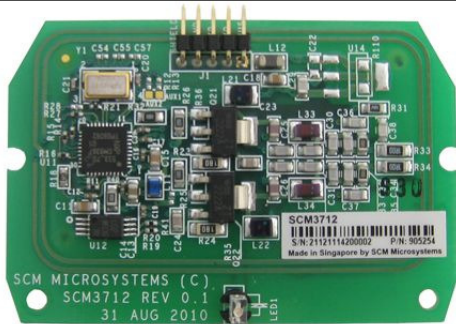
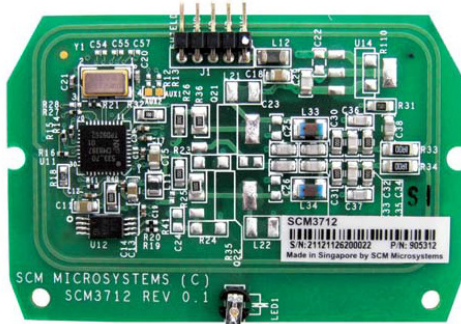
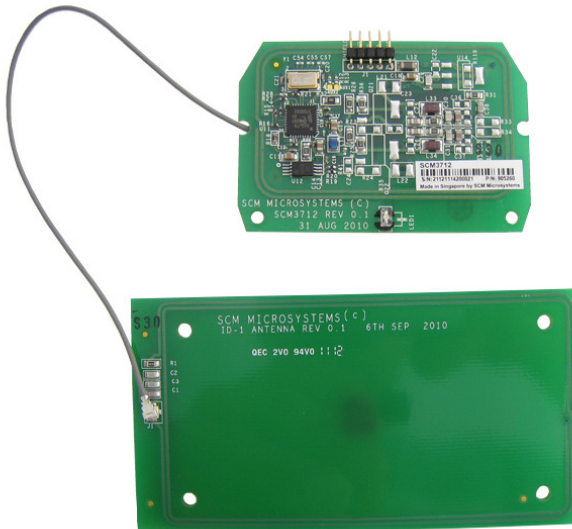
- Multi-protocol 13.56MHz contactless reader board family:
  - ISO14443 type A & B
  - MIFARE (Classic, DESFire, DESFire EV1, UL, UL-C, MIFARE PLUS S<sup>1</sup>)
  - FeliCa™ non-secure
  - NFC passive Peer-to-peer communication<sup>2</sup>
- USB 2.0 Full Speed Interface
- PC/SC v2.0 compliant

---


<sup>1</sup>Mifare Plus cards in security level 2, ISO14443-3 commands are not supported because the SAK byte of those user tokens doesn't indicate it is supported

<sup>2</sup> For SCM3712 NFC and SCM3712 EA only

### 3.3. SCM3712 product family ordering information

Item	Part number	Device
SCM3712	905254	
SCM3712 NFC	905312	
SCM3712 EA	905260	



Item	Part number	Device
NFC Solutions Development Kit	905319	 The image shows the packaging for the NFC Solutions Development Kit. The box is primarily black with a green circuit board graphic at the top. It features the text "NFC Solutions Development Kit" and the "SCM" logo. A circular inset on the front shows a close-up of an NFC tag being scanned by a device. The website "www.scmicro.com" is printed at the bottom of the box. The box is shown from a slightly elevated angle, revealing its reflection on the surface below.

### 3.4. Contactless communication principles and SCM3712 usage recommendations

SCM3712 is a contactless reader<sup>3</sup> designed to communicate with user tokens or via the NFC Peer-To-Peer modes (Reader-To-Tag Mode & Peer-To-Peer Mode) with NFC enabled Mobile phones also with other SCM3712/SCL3711 devices.

In any case the SCM3712 Reader Boards are being embedded into terminals, vending machines, kiosks or any other final device.

The below pictures show a typical system configuration for a terminal and a kiosk application.

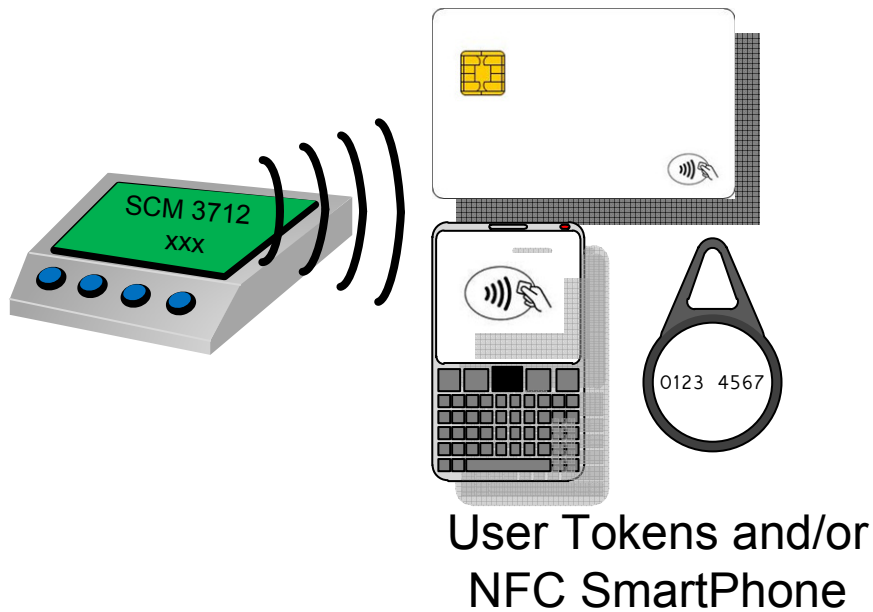


Fig.3-1: SCM3712 in a Terminal Use Application

<sup>3</sup> In the ISO/IEC 14443 standard, the reader is called the Proximity Coupling Device (PCD)

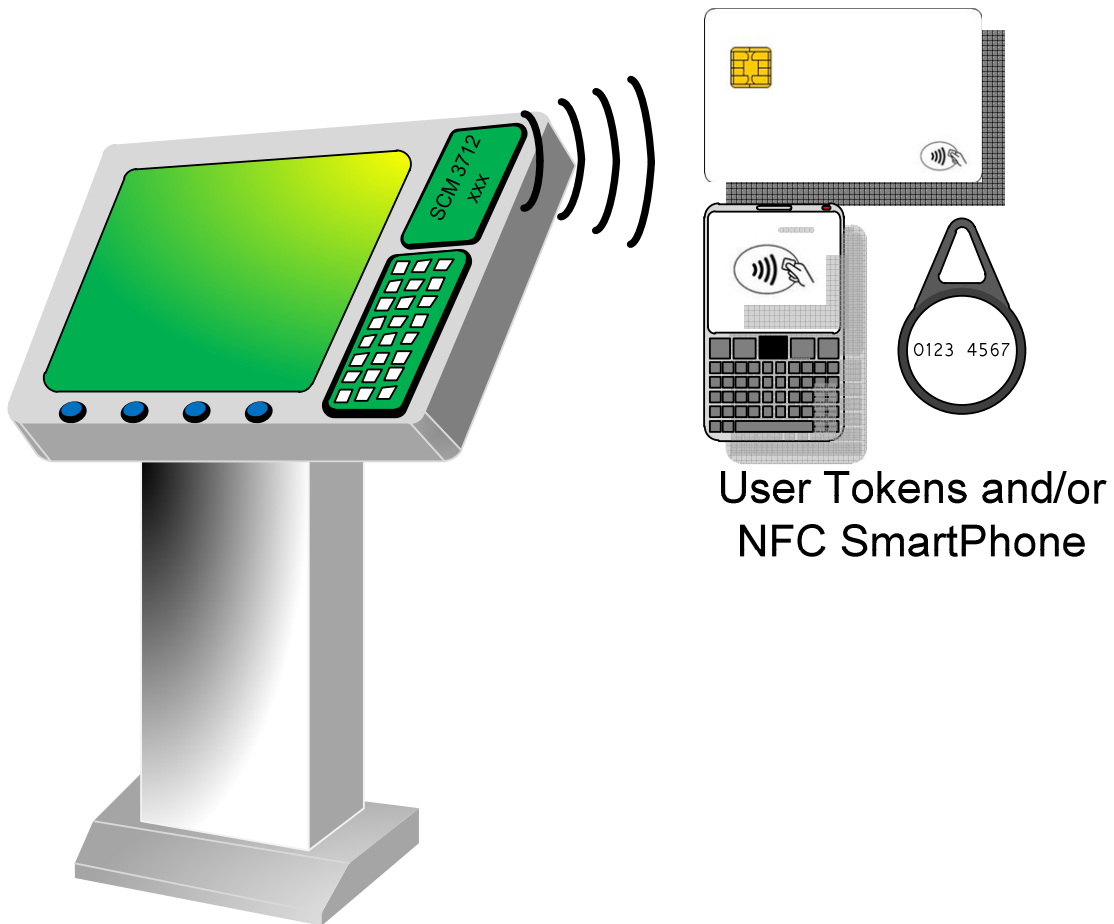


Fig.3-2: SCM3712 in a Kiosk Use Application

User tokens<sup>4</sup> are made of a contactless integrated circuit card connected to an antenna.

User tokens can take several form factors:

- Credit card sized smart card
- Key fob
- NFC mobile phone
- Mobile Device Stickers etc...

The communication between SCM3712 and user tokens uses magnetic field inductive coupling.

The SCM3712 generates a magnetic field with a carrier frequency of 13.56MHz.

---

<sup>4</sup>In the ISO/IEC 14443 standard, the user token is called Proximity Integrated Chip Card (PICC)

### 3.4.1. Power supply

When the user token is put in the magnetic field of the reader, its antenna couples with the reader and an induction current appears in the antenna thus providing power to the integrated circuit. The generated current is proportional to the magnetic flux going through the antenna of the user token.

### 3.4.2. Data exchange

The carrier frequency of the magnetic field is used as a fundamental clock signal for the communication between the reader and the card. It is also used as a fundamental clock input for the integrated circuit microprocessor to function.

To send data to the user token the reader modulates the amplitude of the field. There are several amplitude modulation and data encoding rules defined in ISO/IEC 14443 and ISO/IEC 18092. The reader of this document should refer to those standards for further details.

To answer to the reader, the integrated circuit card of the user token modulates its way of loading (impedance) the field generated by the reader. Here also further details can be found in ISO/IEC 14443 and ISO/IEC 18092.

### 3.4.3. Recommendations

The communication between the reader and the user token is sensitive to the presence of material or objects interfering with the magnetic field generated by the reader.

The presence of conductive materials like metal in the vicinity of the reader and the user token can severely degrade the communication and even make it impossible. The magnetic field of the reader generates Eddy or Foucault's currents in the conductive materials; the field is literally absorbed by that kind of material.



It is recommended for proper communication to avoid putting SCM3712 in close proximity of conductive materials.

The presence of multiple user tokens in the field also interferes with the communication. When several user tokens are in the field of the reader, load of the field increases which implies that less energy is available for each of them and that the system is detuned. For this reason, Identive has implemented in its driver the support for 1 slot only.



It is recommended to present only one user credential at a time in front of SCM3712.

The communication between the reader and the user token is sensitive to the geometry of the system {reader, user token}. Parameters like the geometry and especially the relative size, position and orientation of the reader and user token antennas directly influence the inductive coupling and therefore the communication.

SCM3712 was primarily designed and optimized to function with user credentials of various technologies having the size of a credit card.



It may happen that SCM3712 is not capable of communicating with extremely large or extremely small antennas.



In order to optimize the coupling between the reader and the user token, it is recommended to put both antennas as parallel as possible



In order to optimize transaction speed between the reader and the card it is recommended to place the user token as close as possible to the reader. This will increase the amount of energy supplied to the user credential which will then be able to use its microprocessor at higher speeds

## 3.5. Applications

### 3.5.1. General

SCM3712 is a transparent reader designed to interface a personal computer host supporting PC/SC interface with 13.56MHz user credentials used in public transport, loyalty programs, vending machine and payment applications.

Those user tokens can have several form factors like:

- Various card types
- tokens and coins of different form factors and sizes
- smart poster labels of different form factors and sizes
- key fobs
- mobile phone stickers
- NFC enabled mobile phones
- USB dongles like SCT3511 or @MAXX Lite offered by Identive

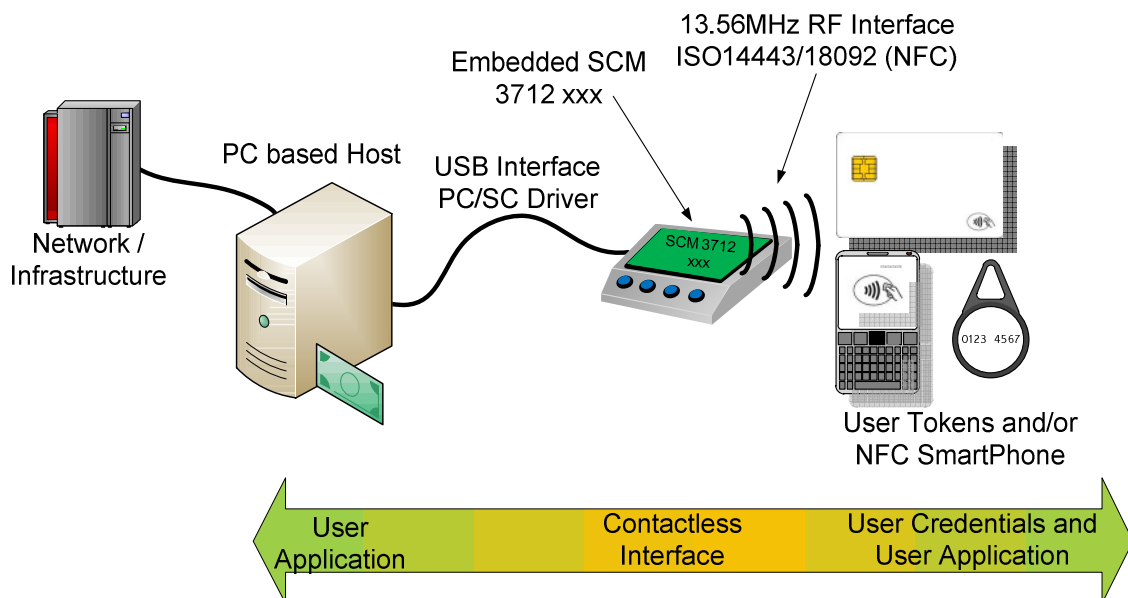


Fig.3-3: SCM3712 based system configuration in a Terminal Use Application

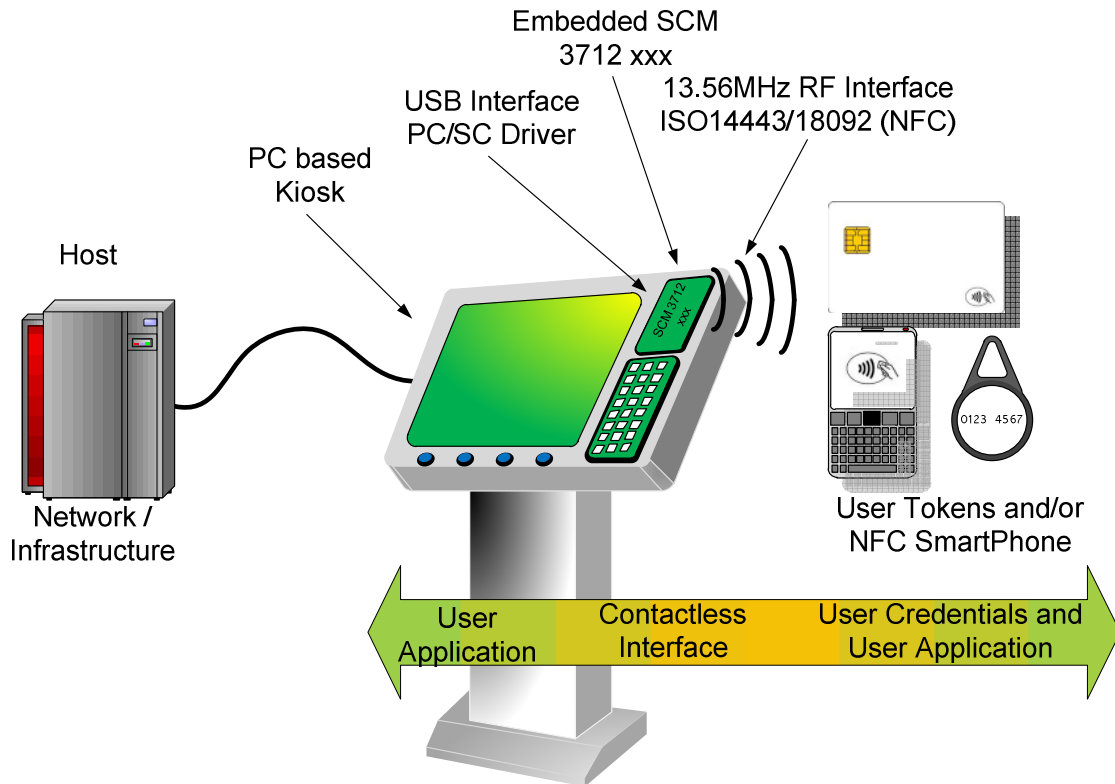


Fig.3-4: SCM3712 based system configuration in a Kiosk Use Application

SCM3712 itself handles the communication protocol but not the application related to the token. The application-specific logic has to be implemented by software developers on the host.

### 3.5.2. Supporting applications provided by Identive

Identive does not provide payment or transport applications.

Identive provides a few applications for development and evaluation purposes that can work with SCM3712. They are available within the software development kit. There are multiple tools provided but the two main ones are:

- The NFC forum tag reader/writer is a standalone application that enables the user to read and write NFC forum compliant records into NFC forum compatible tags. It is an easy to use tool to configure rapidly NFC forum tag demonstrations.
- Smart card commander version 1.1 provides a module which for NFC forum tags parses and presents the content of the tags in XML format. Smart card commander also contains powerful scripting functionality which can be very useful for developers to develop and debug their applications.

## 4. SCM3712 characteristics

### 4.1. SCM3712 high level architecture

#### 4.1.1. Block diagrams

The link between SCM3712 and the host to which it is connected is the USB interface providing both the power and the communication channel.

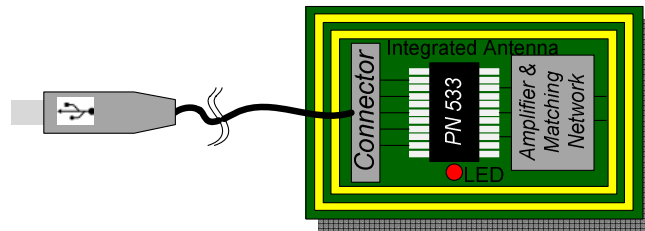


Fig.4-1: SCM3712 Reader Board Block Schematic

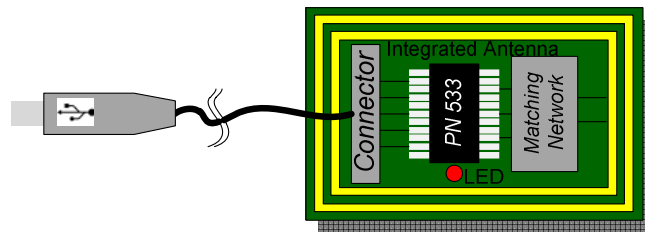


Fig.4-2: SCM3712 NFC Reader Board Block Schematic

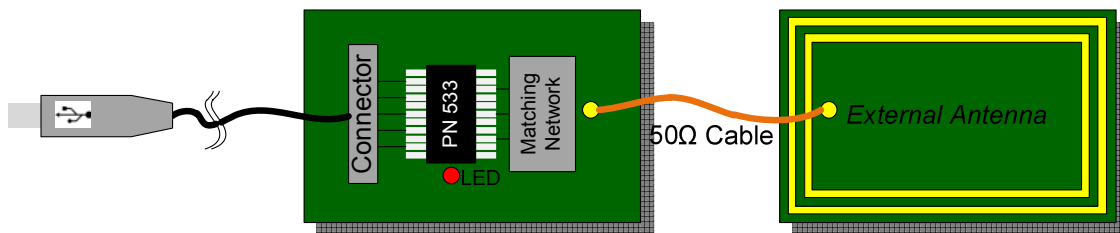


Fig.4-3: SCM3712 EA Reader Board Block Schematic

SCM3712 is basically designed around an NFC controller which handles the USB communication to the host and the RF communication. This controller ensures the coding/decoding/framing modulation/demodulation required for the RF communication.

The matching circuitry provides the transmission and receiver paths adaptation for the antenna to function properly.



#### 4.1.2. Software architecture

Applications can interface either with the driver directly through the PC/SC interface or through the IDENTIVE proprietary interface to the NFC wrapper.

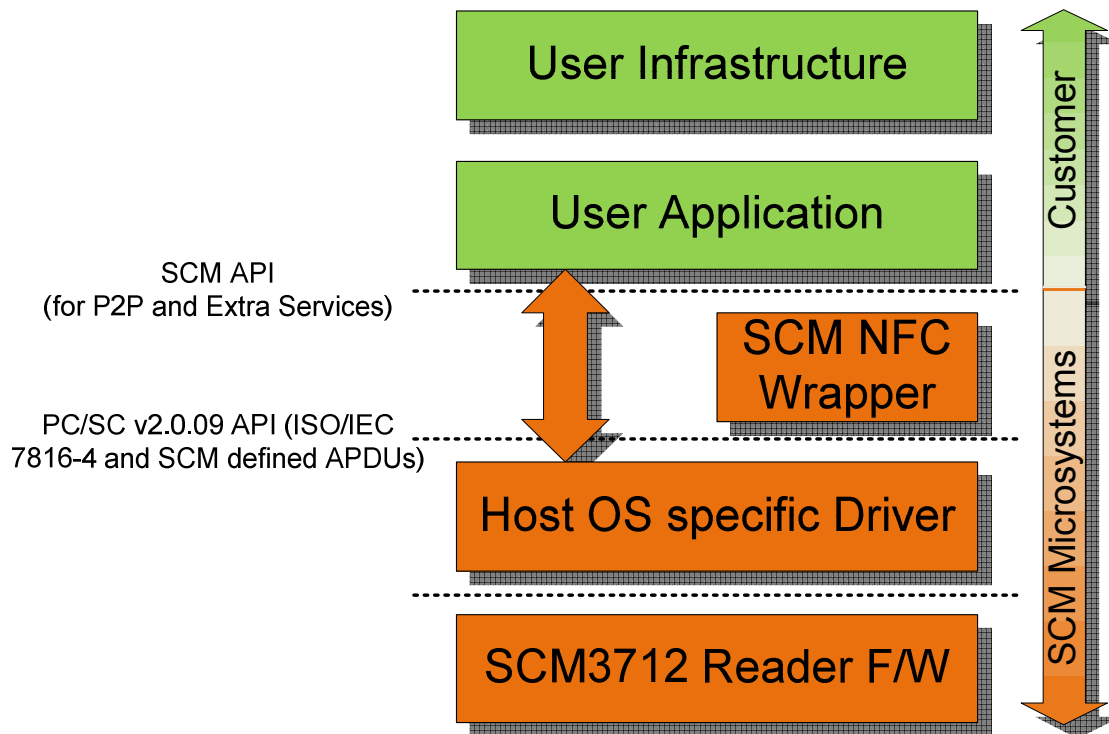


Fig.4-4: SCM3712 xxx software layer overview

The NFC wrapper simplifies the usage of the different NFC Forum tags with the SCM3712 and other IDENTIVE contactless readers. It provides a unique API to application developers, which enables them to read and modify NDEF records without further knowledge of the underlying hardware and protocols. Detailed information about the NFC wrapper can be found in IDENTIVE's Contactless SDK.

The SCM3712 driver implements PC/SC v2.0 API towards upper layers. The SCM3712 driver for Windows platforms is based on the Windows Driver Framework (WDF) version 1.09.

## 4.2. Quick reference data

### 4.2.1. SCM3712 mechanical dimensions

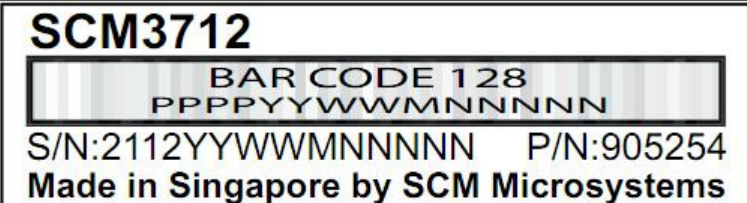
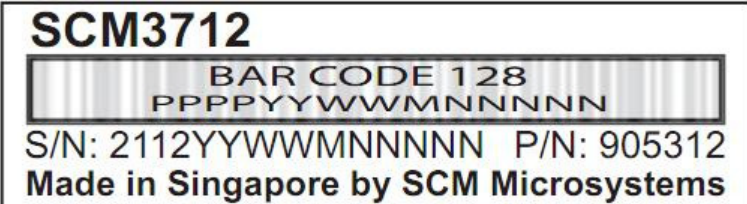

Characteristic	Value		
	SCM3712	SCM3712 NFC	SCM3712 EA
Weight	5,5 g $\pm$ 5% / 0,012 lb $\pm$ 5%	5,5 g $\pm$ 5% / 0,012 lb $\pm$ 5%	19,5 g $\pm$ 5% / 0,043 lb $\pm$ 5% <sup>5</sup>
Reader PCB Size (LxWxH)	60 x 40,3 x 10 mm $\pm$ 0.13mm 2,36 x 1,59 x 0,39 Inch $\pm$ 0,005 Inch	60 x 40,3 x 10 mm $\pm$ 0.13mm 2,36 x 1,59 x 0,39 Inch $\pm$ 0,005 Inch	60 x 40,3 x 10 mm $\pm$ 0.13mm 2,36 x 1,59 x 0,39 Inch $\pm$ 0,005 Inch
Antenna PCB Size (LxWxH)	integrated	integrated	92,7 x 49,5 x 4 mm 3,65 x 1,95 x 0,16 Inch
Antenna Cable Length	NA	NA	200mm / 7,87 Inch
Antenna Connector	NA	NA	ULTRA MINI COAX
Default Barcode labels	 <p><b>SCM3712</b> BAR CODE 128 PPPPYYWWMNNNNN S/N: 2112YYWWMNNNNN P/N: 905254 Made in Singapore by SCM Microsystems</p>		
	 <p><b>SCM3712</b> BAR CODE 128 PPPPYYWWMNNNNN S/N: 2112YYWWMNNNNN P/N: 905312 Made in Singapore by SCM Microsystems</p>		
	 <p><b>SCM3712</b> BAR CODE 128 PPPPYYWWMNNNNN S/N: 2112YYWWMNNNNN P/N: 905260 Made in Singapore by SCM Microsystems</p>		

Table 4.1: SCM 3712 xxx mechanical data overview

Detailed drawings with dimensions of the SCM3712 reader boards and accessories can be found in annex. B, from page 76 onwards.

<sup>5</sup> SCM3712 EA Reader Board + Cable + Antenna

#### 4.2.2. LED Status indication

The LED behavior of the SCM3712 is given below.

SCM3712states	LED Indication (GREEN)
After plug-in (no driver loaded)	OFF
Driver successfully loaded	ON
User token arriving in the field	One blink
User token removed from the field	ON, no specific visual indication
Suspend/hibernate/shutdown state	OFF
SCM3712disabled	OFF
P2P mode	Three blinks

Table 4.2: LED Status indication

## 4.2.3. SCM3712 Electrical Specification

Parameter	Value/Description		
	SCM3712	SCM3712 NFC	SCM 3712 EA
Power supply	5VDC, via USB Interface		
Power consumption	High bus powered <200mA 260µA at power down mode	Low bus powered <100mA 260µA at power down mode	Low bus powered <100mA 260µA at power down mode
Device controller clock	Max 27.12MHz		
RF carrier frequency	13.56 MHz ± 50 ppm		
Modulation index	As defined in ISO/IEC 14443		
USB specification	USB 2.0 Full Speed Device		
USBspeed	Full Speed Device (12Mbit/s)		
Device class	Vendor		
PID	0x5592	0x5993	0x5994
VID	0x04E6		
API	PC/SC 2.0		
Supported credential types	NFC forum tag type 1 through IDENTIVE-specific APDU NFC forum tag type 2 through PC/SC-defined APDUs NFC forum tag type 3 through IDENTIVE-specific APDU NFC forum tag type 4 through PC/SC APDUs ISO/IEC 14443-4 PICC type A and type B MIFARE (Classic, Ultralight, Ultralight C, MIFARE PLUS <sup>6</sup> , DESFire, DESFire EV1 2/4/8k), Non-Secure FeliCa™, Innovision Topaz		
Supported NFC Modes	Reader To Tag	Reader To Tag Peer To Peer	Reader To Tag Peer To Peer
Supported RF baud rates	106 / 212 / 424 Kbps		
Multiple PICC in field	Not supported		
Operating temperature range	0° to +60°C / +32 to +140 F		
Storage temperature range	-20 to +60°C / -4 to +140 F		
Operating humidity range	Up to 95%RH non condensing		
Certifications / Compliances	WHQL, RoHS		

Table 4.3: Electrical Specification

<sup>6</sup>MIFARE PLUS cards in security level 2, ISO14443-3 commands are not supported because the SAK byte of those user tokens doesn't indicate it is supported

## 5. Hardware

The SCM3712 reader board comes with an integrated USB interface, which also is used for the power supply of the unit. Therefore only the USB interface needs to be connected. The drawing and the table below are showing the exact Pin-Out of the USB connector.

### 5.1. USB Connector Pin-Out

Pin Number	Pin Name	Description
1	USB +5V	Supply Voltage
2	Data -	Data Line
3	Data +	Data Line
4	GND	Signal Ground
5	Shield	Connector for optional cable shielding

Table 5.1:USB Connector Pin-Out

### 5.2. USB Connector Signal Min-/Max-Ratings

SCM3712 NFC, SCM3712 EA

Pin #	Min	Avg.	Max.	Comment
1	4.75V	5.0V	5.25V	USB DC Power
	80 mA	90 mA	100 mA	
2	3.0V	3.3 V	3.6 V	According to USB 2.0 FS
3	3.0V	3.3 V	3.6 V	According to USB 2.0 FS
4	-	-	-	Signal Ground
5	-	-	-	Signal Ground

Table 5.2: USB Connector Min-/Max Ratings, SCM3712 NFC & SCM3712 EA

SCM3712

Pin #	Min	Avg.	Max.	Comment
1	4.75V	5.0V	5.25V	USB DC Power
	170 mA	185 mA	200 mA	
2	3.0V	3.3 V	3.6 V	According to USB 2.0 FS
3	3.0V	3.3 V	3.6 V	According to USB 2.0 FS
4	-	-	-	Signal Ground
5	-	-	-	Signal Ground

Table 5.3: USB Connector Min-/Max Ratings, SCM3712

Please make sure the reader always is correctly connected. A wrong connection may prevent the reader from working as expected or may even damage the reader. Defects caused by connection errors are not covered by warranty regulations.

### 5.3. Antenna

The SCM3712 and the SCM3712NFC reader come with an integrated antenna, while the SCM3712 ANT comes with an external antenna. Any of those ready to use antennas do not require any tuning as the antennas already are being tuned for optimized performance during the production process. Any changes on the tuning circuit will result in a negatively influenced reader performance.

## 6. Software modules

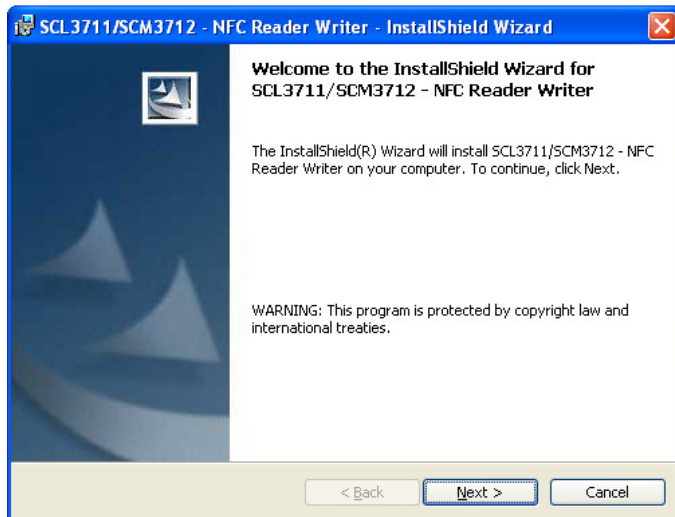
SCM3712 is provided with an installer.

### 6.1. Installation

Make sure the SCM3712 is not plugged in your PC before you start.

Start the installer by double clicking on setup.exe  and then follow the wizard instructions

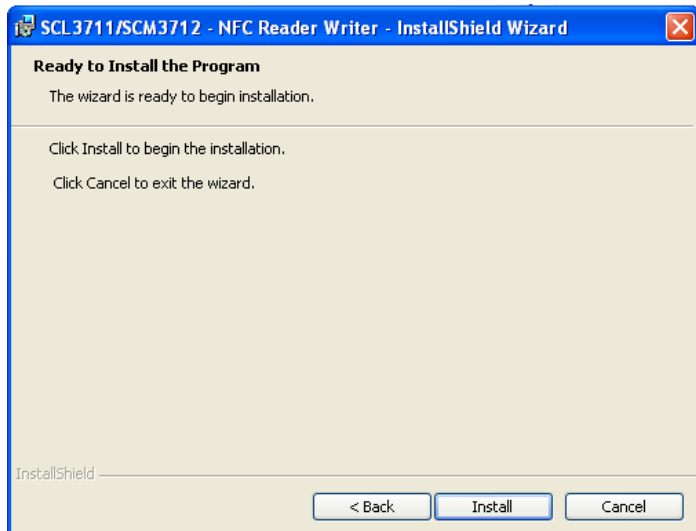
Click Next on the welcome page of the installer



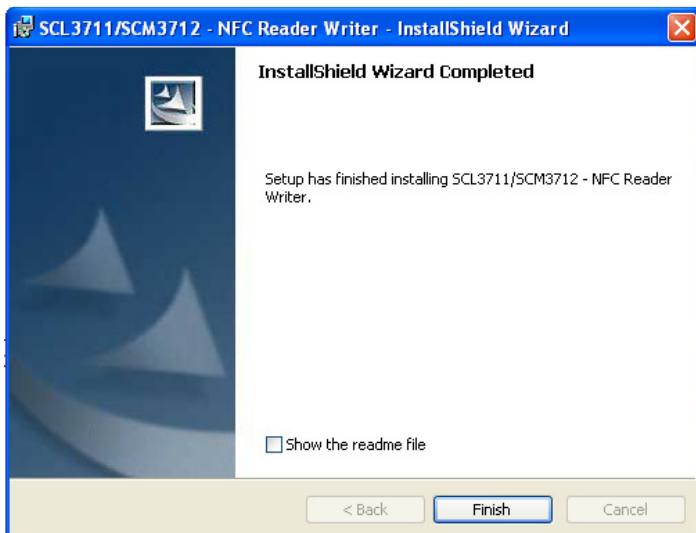
Carefully read and accept the license agreement in order to install the driver.



Then install



After a few minutes, you are notified the installation happened correctly



You are ready to use your SCM3712.

In some very rare cases, you may be asked to reboot your PC. Please do so if this is the case.



### 6.1.1. Command line parameters for installation

A few parameters of the installer can be configured when launching the installer from the command line

Silent mode of installation	Setup.exe /s /v"/qn"
Installation with no dialogs	Setup.exe /v"LIMITUI=1"
No reboot dialog	Setup.exe /v"REBOOTREQD=0"

### 6.1.2. Command line parameters for de-installation.

A few parameters of the installer can be configured when launching the installer from the command line

Silent mode of de-installation	<system folder>\Msiexec.exe /x<path to msi file>\<msi file name> /qn
De-installation with no dialogs	<system folder>\Msiexec.exe /x<path to msi file>\<msi file name> LIMITUI=1
No reboot dialog	<system folder>\Msiexec.exe /x<path to msi file>\<msi file name> REBOOTREQD=0

## 6.2. Utilities

NA

## 6.3. Driver

The driver for Windows platforms is based on Microsoft WDF architecture 1.09.

The driver package contains INF, SYS, CAT and the co-installer DLL required for the WDF architecture.

### 6.3.1. SCM3712 listing

SCM3712 enumerates as *SCM3712-NFC&RW*

After the driver is installed, the different SCM3712 reader boards do appear in the Microsoft Windows device manager as in the table below:

Product Type	TLPN#	PID
SCM3712	905254	SCM3712 Reader r
SCM3712 NFC	905312	SCM3712 NFC Reader
SCM3712 EA	905260	SCM3712 EA Reader

Depending on the product variant in PC/SC applications the SCM3712 is listed as:

- *Identive* SCM3712 Reader device N
- *Identive* SCM3712 NFC Reader device N
- *Identive* SCM3712 EA Reader device N

Where N=0 if only one SCM3712 reader is connected but is incremented in case several SCM3712 are being connected to the host.

### 6.3.2. Supported operating systems

Operating systems supported by the driver:

- Windows 2000 SP4
- Windows 2003 Server (32 & 64 bit)
- Windows XP (32 & 64 bit)
- Windows Vista (32 & 64 bit)
- Windows 7 (32 & 64 bit)
- Windows Server 2008 (32 & 64 bit)
- Linux (32 & 64 bit)
- MACOSX
- WinCE 5.0 and 6.0



### 6.3.3. PC/SC 2.0 compliant ATR

#### 6.3.3.1. Determining the technology of the user credential

The ScardControl method of PC/SC(see [http://msdn.microsoft.com/en-us/library/aa379474\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa379474(VS.85).aspx)) should be used to send the 0x90 IOCTL to SCM3712 in order to determine what type of technology is the user token based on. The output buffer is a BYTE with the following meaning:

Technology	Value
MIFARE1K	0x01
MIFARE4K	0x02
MIFARE Ultralight and Ultralight C	0x03
ISO14443-4A DESFire	0x04
FeliCa	0x05
Topaz	0x06
ISO14443-4B	0x07

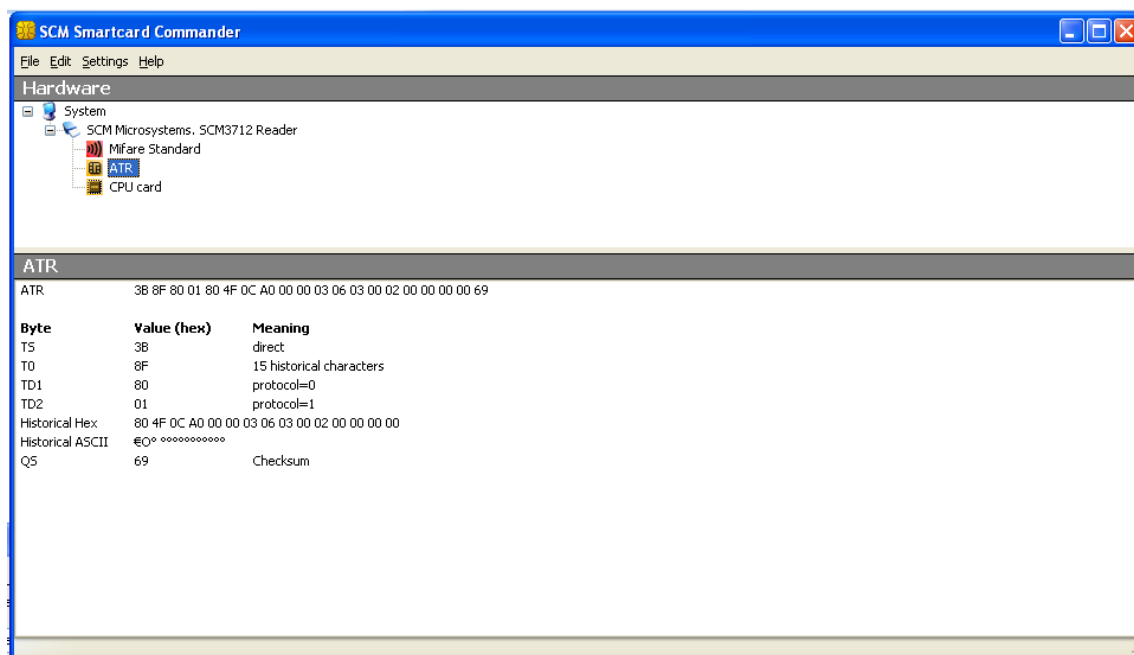
Once a user credential is selected, the driver constructs an ATR from the fixed elements identifying the token. Depending on the user technology this ATR can be analyzed as described hereunder.

### 6.3.3.2. ATR for type A memory user tokens

The ATR of the user token is composed as described in the table below. In order to allow the application to identify the storage card properly, it's Standard and Card name describing bytes must be interpreted according to the Part 3 Supplemental Document, maintained by PC/SC.

Byte#	Value	Designation	Description
0	0x3B	Initial header	
1	0x8n	T0	n indicates the number of historical bytes in following ATR
2	0x80	TD1	Nibble8 indicates no TA2, TB2, TC2 Nibble0 means T=0
3	0x01	TD2	Nibble8 indicates no TA3, TB3, TC3 Nibble1 means T=1
4...3+n	0x80		A status indicator may be present in an optional TLV data object
	0x4F	Optional TLV data object	Tag: Application identifier
	Length		1 byte
	RID		Registered identifier on 5bytes
	PIX		Proprietary identifier extension on 3bytes
	0x00 0x000x000x00		4 RFU bytes
4+n	0x91	TCK	XOR of all previous bytes

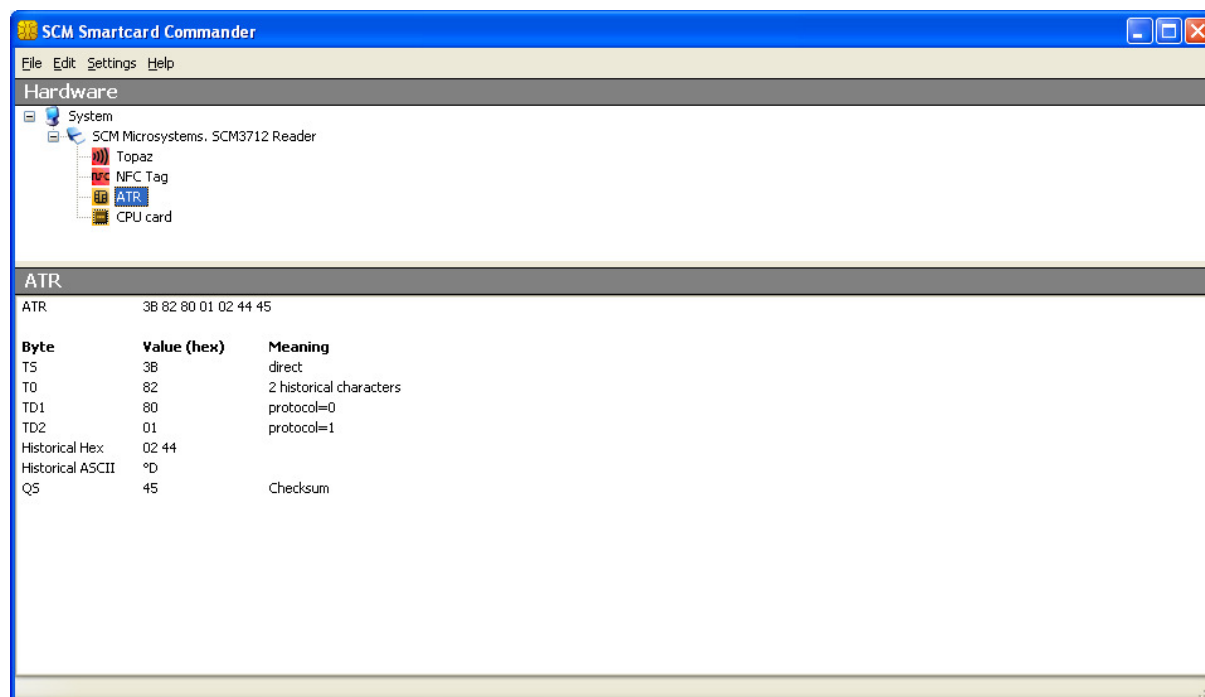
Example of the ATR built for a MIFARE Classic 4K card:



### 6.3.3.3. ATR for an NFC Forum tag type 1 user token (Topaz)

Byte#	Value	Designation	Description
0	0x3B	Initial header	
1	0x82	T0	TD1 present. 2 historical bytes in following ATR
2	0x80	TD1	Nibble8 indicates no TA2, TB2, TC2 and TD2 present Nibble0 means T=0
3	0x01	TD2	Nibble8 indicates no TA3, TB3, TC3 Nibble1 means T=1
4	0x02	Card Mode	NFC TAG operating at Passive 106 baud rate
5	0x44	Card Type	Card type is Topaz
6	0xXX	TCK	XOR of all previous bytes

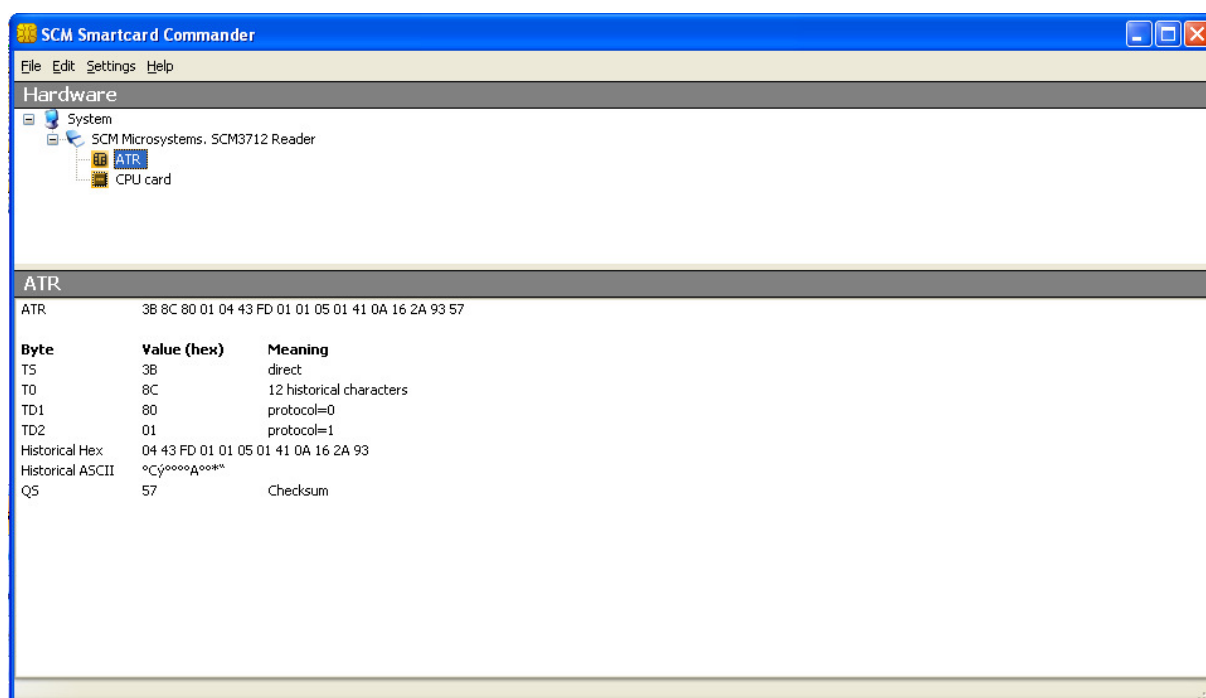
Example of the ATR built for a Topaz tag:



#### 6.3.3.4. ATR for a NFC Forum tag type 3 user token (FeliCa)

Byte#	Value	Designation	Description
0	0x3B	Initial header	
1	0x8C	T0	TD1 present. 12 historical bytes in following ATR
2	0x80	TD1	Nibble8 indicates no TA2, TB2, TC2 and TD2 present Nibble0 means T=0
3	0x01	TD2	Nibble8 indicates no TA3, TB3, TC3 Nibble1 means T=1
4	0x04	Card Mode	NFC TAG operating at Passive 212 baud rate
5	0x43	Card Type	Card type is FeliCa
6	0xFD	IFS	Maximum frame size of FeliCa card
7-14	-	ID	FeliCa card Identifier – 8 bytes
15	0xFF	Timeout	Write Timeout indicated by card
16	0xFF	TCK	XOR of all previous bytes

Example of the ATR built for a FeliCa user token:



### 6.3.3.5. ATR for ISO/IEC 14443-4 user tokens

The user token exposes its ATS or application information which is mapped to an ATR. The table describes how this mapping is done.

Byte#	Value	Designation	Description
0	0x3B	Initial header	
1	0x8n	T0	n indicates the number of historical bytes in following ATR
2	0x80	TD1	Nibble8 indicates no TA2, TB2, TC2 Nibble0 means T=0
3	0x01	TD2	Nibble8 indicates no TA3, TB3, TC3 Nibble1 means T=1
4...3+n		Historical bytes or application information	Type A: the historical bytes from the ATS (up to 15 bytes) Type B (8 bytes): <ul style="list-style-type: none"> <li>Byte 0 through 3: application data from ATQB,</li> <li>Byte 4 through 6: protocol info byte from ATQB,</li> <li>Byte 7: highest nibble is the MBLI (maximum buffer length index) from ATTRIB, lowest nibble is 0x0</li> </ul>
4+n		TCK	XOR of all previous bytes

Example of the ATR built for an ISO14443-4 user tokens:

Type A

The screenshot shows the SCM Smartcard Commander interface. Under the 'Hardware' tab, the 'ATR' section displays the following data:

Byte	Value (hex)	Meaning
TS	3B	direct
T0	84	4 historical characters
TD1	80	protocol=0
TD2	01	protocol=1
Historical Hex	00 00 90 00	
Historical ASCII	""□°	
Q5	95	Checksum

Type B

The screenshot shows the SCM Smartcard Commander interface. Under the 'Hardware' tab, the 'ATR' section displays the following data:

Byte	Value (hex)	Meaning
TS	3B	direct
T0	88	8 historical characters
TD1	80	protocol=0
TD2	01	protocol=1
Historical Hex	05 00 01 03 73 81 C1 00	
Historical ASCII	""°s□Á°	
Q5	3D	Checksum

## 7. Firmware

### 7.1.1. Transport protocol

SCM3712 implements a transport protocol which is proprietary to NXP Semiconductors.

### 7.1.2. Automatic PPS

Automatic PPS implemented is implemented. SCM3712 will automatically switch the highest baud rate commonly supported by the SCM3712 and the user token

The maximum speed supported by SCM3712 is 424Kbps by default.



## 8. Commands description

### 8.1. Generic APDUs

#### 8.1.1. Get UID Command

##### 8.1.1.1. Description

This command will retrieve the UID or SNR or PUPI of the user token. This command can be used for all supported contactless technologies.

##### 8.1.1.2. Format

CLA	INS	P1	P2	Lc	Data in	Le
0xFF	0xCA	0x00	0x00	-	-	XX

Setting Le = 0x00 can be used to request the full UID or PUPI is sent back

- For ISO14443A possible lengths are 4, 7 or 10
- For ISO14443B possible length is 4 bytes PUPI
- For FeliCa™ or NFC Forum type 3 tags possible length is 12 bytes of NFCID
- For NFC Forum type 1 tags possible length is 7 bytes of UID

##### 8.1.1.3. Response

Data Out
Data + SW1 + SW2

##### 8.1.1.4. Status Words

SW1	SW2	Description
0x90	0x00	NO ERROR
0x62	0x82	WARNING: specified Le is greater than data to be retrieved
0x6C	0XX	ERROR: Wrong Length. 0XX is the exact value for Le

Further error codes can be found in annex

## 8.1.2. Get DATA Command

### 8.1.2.1. Description

This command can be used to retrieve the ATS of an ISO/IEC14443-4A user token only.

### 8.1.2.2. Format

CLA	INS	P1	P2	Lc
0xFF	0xCA	0x01	0x00	0x00

### 8.1.2.3. Response

Data Out
ATS + SW1 + SW2

### 8.1.2.4. Status Words

SW1	SW2	Description
0x90	0x00	NO ERROR
0x6A	0x81	Command not supported

### 8.1.3. T=CL user Command

#### 8.1.3.1. Description

This command can be used to send raw data to the user token. SCM3712 will add T=CL protocol data to the raw data you send.

#### 8.1.3.2. Format

CLA	INS	P1	P2	P3	Data
0xFF	0xFE	0x00	0x00	Lraw_data	Raw_data

#### 8.1.3.3. Response

Data Out
PICC response data+ SW1 + SW2

#### 8.1.3.4. Status Words

SW1	SW2	Description

User should refer to the status words defined by the PICC manufacturer for a description of the status words

#### 8.1.3.5. Example

Let's consider the Select command defined in ISO7816-4. This command being ISO can be sent to the user token in 2 different way:

- Using the T=CL command
- Using the T=CL user command

Here are the 2 answers for the select command:

```
ATR length: 14
ATR: 3B 89 80 01 4D 54 43 4F 53 73 01 01 01 3C
APDU: 00 A4 00 00
SW12: 9000 (OK)
```

```
APDU: FF FE 00 00 04 00 A4 00 00
SW12: 9000 (OK)
```

Nevertheless, the T=CL command is more useful for sending commands which are not defined in ISO7816.

### 8.1.4. PASS\_THROUGH command

#### 8.1.4.1. Description

This command can be used to send raw data to the user token. SCM3712 will not add transport protocol data to the raw data – e.g. PCB, NAD, CID etc.

#### 8.1.4.2. Format

CLA	INS	P1	P2	P3	Data
0xFF	0xEF	0x00	0x00	Lraw_data	Raw_data

#### 8.1.4.3. Response

Output buffer
PICC response data

#### 8.1.4.4. Status Words

NA

#### 8.1.4.5. Example

This command can be used to send commands to a MIFARE Ultralight C

The command for generating an 8-byte random number on MIFARE Ultralight C is 0x1A 0x00:

Sending the APDU 0xFF 0xEF 0x00 0x00 0x02 0x1A 0x00

Will return 0xAF followed by 8 byte random number

## 8.2. Set of APDU for contactless storage user tokens

Command specific return codes are given under each command. Please refer section 7.1.1 (Status words table) for common return codes.

### 8.2.1. STORAGE\_CARD\_CMDS\_READ\_BINARY

#### 8.2.1.1. Description

Using this APDU, application can read a memory block on user tokens based on technologies like MIFARE Classic 1K or 4K (block size 0x10 bytes) or MIFARE UltraLight (block size 0x04 bytes).

#### 8.2.1.2. Format

CLA	INS	P1	P2	Le
0xFF	0xB0	0x00	Block #	0xFF

Where:

- P2 indicates the block number from where to read
- Le can be a short (maximum value 255) or extended (maximum value 65535). If Le=0x00, then all the bytes until the end of the file are read within the limit of 256 for a short Le field and 65536 for an extended Le field.

#### 8.2.1.3. Response

Data Out
Data + SW1 + SW2

#### 8.2.1.4. Status Words

SW1	SW2	Description
0x90	0x00	NO ERROR
0x69	0x82	Security status not satisfied
0x64	0x00	State of nonvolatile memory unchanged

### 8.2.1.5. Example

For a MIFARE Classic 1K card which has the following memory content:

SCIM SmartCard Commander

File Settings Help

Hardware

- System
  - SQL3711
    - Mifare Standard
    - NFC Tag
    - ATR
    - CPU card

Card type: **Mifare Standard**  
 Memory size: **1024 Bytes**  
 Unique ID: **FA 92 6C D6**

Sector Hex	Block Read	Block Write	Block Inc	Block Dec	Key A Read	Key A Write	Key B Read	Key B Write	AC Read	AC Write
0	FA92	6CD6	D288	0400	1649	8652	4510	1008		
0F00	03E1	03E1	03E1	03E1	03E1	03E1	03E1	03E1		
03E1	03E1	03E1	03E1	03E1	03E1	03E1	03E1	03E1		
0A01	A2A3	A4A5	7077	00C1	????	????	????	????		
1	032D	D102	2853	7091	0108	5402	656E	4E46		
4320	4465	6D6F	5101	1555	0373	636B	5D69			
6372	4F2E	636F	6D6F	7363	6C33	3731	30FE			
D3F7	D3F7	D3F7	7E07	8840	????	????	????	????		
2	0000	0000	0000	0000	0000	0000	0000	0000		
0000	0000	0000	0000	0000	0000	0000	0000	0000		
0000	0000	0000	0000	0000	0000	0000	0000	0000		
D3F7	D3F7	D3F7	7E07	8840	????	????	????	????		
3	0000	0000	0000	0000	0000	0000	0000	0000		
0000	0000	0000	0000	0000	0000	0000	0000	0000		
0000	0000	0000	0000	0000	0000	0000	0000	0000		
D3F7	D3F7	D3F7	7E07	8840	????	????	????	????		
4	0000	0000	0000	0000	0000	0000	0000	0000		

USB Bytes    Key A    Access Bits    Data Bytes  
 Internal Bytes    Key B    General Purpose Bytes    Read Only Bytes

To read the seventh block, you have to issue the following command and get the following response:

```
APDU: FF B0 00 06 10
SW12: 9000 (OK)
DataOut: 63 72 6F 2E 63 6F 6D 2F 73 63 6C 33 37 31 30 FE (16 byte(s))
```

## 8.2.2. STORAGE\_CARD\_CMDS\_WRITE\_BINARY

### 8.2.2.1. Description

This APDU writes data pattern in to a memory address

### 8.2.2.2. Format

CLA	INS	P1	P2	Lc	Data in
0xFF	0xD6	0x00	Block #	0XX	Data

Where:

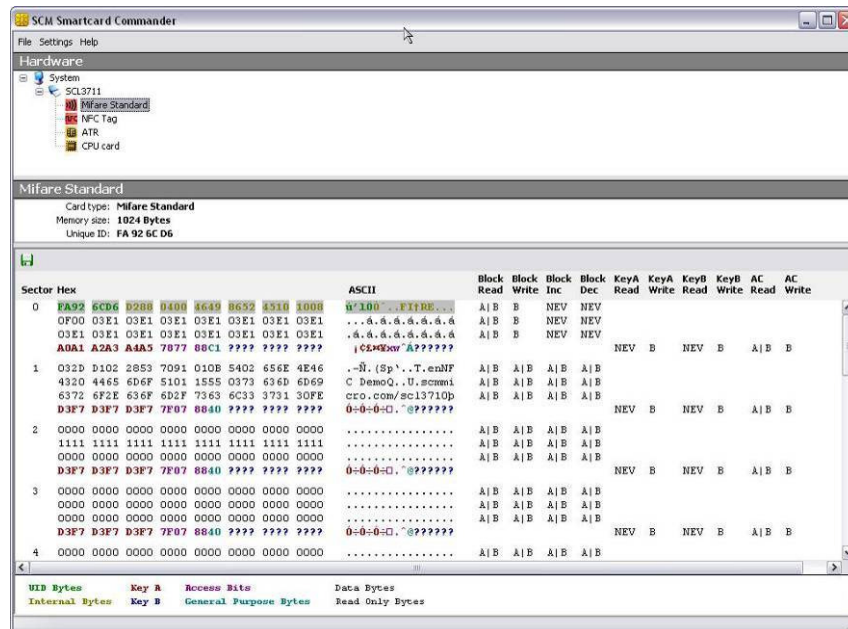
- P2 indicate the memory block number where data should be written
- Lc=0x10 for MIFARE Classic 1K/4K. Lc=0x04 for MIFARE Ultralight

### 8.2.2.3. Response

Data Out
SW1 + SW2

### 8.2.2.4. Status Words

SW1	SW2	Description
0x90	0x00	NO ERROR
0x69	0x82	Security status not satisfied
0x64	0x00	State of non volatile memory unchanged





### 8.2.3. STORAGE\_CARD\_CMDS\_LOAD\_KEYS

#### 8.2.3.1. Description

Some type of user tokens like MIFARE Classic may require that an authentication happens before any data can be read or written. To encrypt perform this authentication, the required keys need to be loaded in the reader's memory using this command.

#### 8.2.3.2. Format

CLA	INS	P1	P2	Lc	Data in
0xFF	0x82	0x00	Key Type	Key Length	Key value

Where P2 can have the following values (please refer to MIFARE documentation from NXP for further details on what is key A and Key B):

- 0x60 to use the Key A
- 0x61 to use the Key B

#### 8.2.3.3. Response

Data Out
SW1 + SW2

#### 8.2.3.4. Status Words

SW1	SW2	Description
0x90	0x00	NO ERROR
0x69	0x83	Reader key not supported
	0x85	Secured transmission not supported
	0x87	Non volatile memory not available
	0x88	Key number not valid
	0x89	Key length not correct

## 8.2.4. STORAGE\_CARD\_CMDS\_AUTHENTICATE

### 8.2.4.1. Description

This command enables to perform authentication for user tokens based on MIFARE Classic 1K or 4K. Before this command can be successfully executed, the STORAGE\_CARD\_CMDS\_LOAD\_KEY command must have been executed.

### 8.2.4.2. Format

CLA	INS	P1	P2	Lc	Data in
0xFF	0x86	0x00	0x00	0x05	Data

Where the data field is structured as follow

Byte #	Value	Description
B0	0x01	Version
B1		Address MSB
B2		Address LSB
B3	0x60	Key A
	0x61	Key B
B4		Number of the key to be used for authentication

Information about memory structure of MIFARE Classic must be requested from NXP Semiconductors.

### 8.2.4.3. Response

Data Out
SW1 + SW2

### 8.2.4.4. Status Words

SW1	SW2	Description
0x90	0x00	NO ERROR
0x63	0x00	WARNING no further info
0x69	0x82	Security status not satisfied
	0x84	Referenced key not usable
	0x86	Key type not known



## 8.2.5. STORAGE\_CARD\_CMDS\_VALUE\_BLOCK

### 8.2.5.1. Description

This APDU is used to interact with MIFARE Classic e-purse applications. Please refer to MIFARE Classic documentation available from NXP Semiconductors for further details on MIFARE classic memory mapping and commands.

### 8.2.5.2. Format

CLA	INS	P1	P2	Lc	Data in
0xFF	0xF0	0x00	Block #	Lc	Increment/Decrement, Block number, Value

Where P1, P2 code the address of the block number addressed

Where the data field is structured as follow

Byte #	Value	Description
B0	0xC0	Increment
	0xC1	Decrement
B1		Block number
B2-B5		Value (LSB first)

### 8.2.5.3. Response

Data Out
SW1 + SW2

### 8.2.5.4. Status Words

SW1	SW2	Description
0x90	0x00	NO ERROR
0x69	0x82	Security status not satisfied

### 8.2.5.5. Example

CLA	INS	P1	P2	Lc	Data in
0xFF	0xF0	0x00	0x1E	0x06	0xC0 0x1E 0x01 0x00 0x000x00

Will increment block number 0x1E of a MIFARE Classic-based user token by a value of 0x01.

[illegible]

## 8.4. MIFARE DESFire commands

MIFARE DESFire native commands can be mapped onto case 4 APDU as described hereunder:

CLA	INS	P1	P2	P3	Data	Le
0x90	DESFirecmd code	0x00	0x00	Length of data field	DESFire command parameters	0x00

The response from a DESFire user token will be mapped as follow

Data	SW1 SW2
User token answer	0x91 0xYY

0xYY is the DESFire native status byte as described in NXP documentation.

Note: In the past Identive had its own proprietary APDU for handling DESFire cards that was implemented on SCL010 and SDI010 products. For backwards compatibility reasons it is still supported but IDENTIVE recommends using the above mapping method for any new integration development.

## 8.5. Identive specific APDU set

### 8.5.1. Commands for communicating with NFC Forum Tags Type 1

Commands for Static and Dynamic Memory Models

- Read Identification (RID)
- Read All Blocks 0 – Eh (RALL)
- Read Byte (READ)

Commands for Dynamic Memory Model

- Read Segment (RSEG)
- Read 8 Bytes (READ8)
- Write-No-Erase 8 Bytes (WRITE-NE8)

#### 8.5.1.1. Read Identification (RID)

Description

This command is used to retrieve the tag's unique identifier.

Format

CLA	INS	P1	P2	P3	Data
0xFF	0x50	0x00	0x00	0x00	-

Response

Data	SW1 SW2
HR0 HR1 UID0 UID1 UID2 UID3	0x90 0x00

### 8.5.1.2. Read All Blocks (RALL)

#### Description

The RALL command reads-out the two header ROM bytes and the whole of the static memory blocks 0x0-0xE.

#### Format

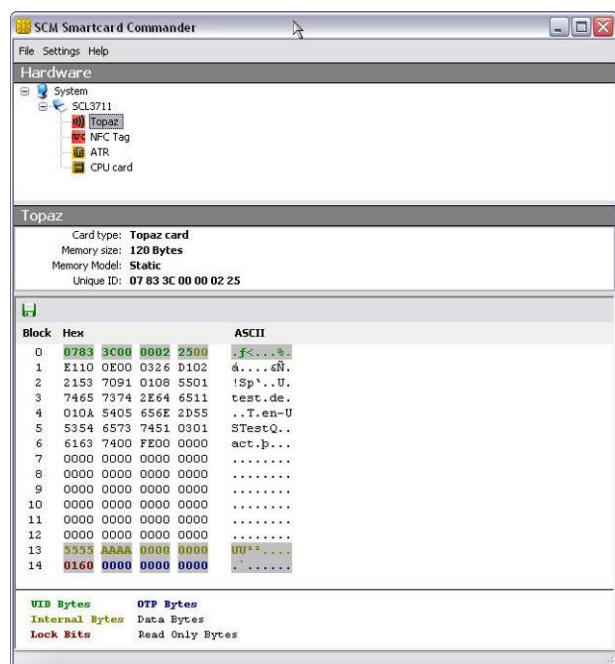
CLA	INS	P1	P2	P3	Data
0xFF	0x52	0x00	0x00	0x00	-

#### Response

Data	SW1 SW2
HR0 HR1 120 bytes (Blocks 0x0 – 0xE)	0x900x00

#### Example

For a Topaz-based user token that has the following memory content



The following APDU sequence can be used to retrieve the identifier and read all the blocks

```

ATR length: 7
ATR: 3B 82 80 01 02 44 45
APDU: FF 50 00 00 00
SW12: 9000 (OK)
DataOut: 11 48 07 83 3C 00 (6 byte(s))

APDU: FF 52 00 00 00
SW12: 9000 (OK)
DataOut: 11 48 07 83 3C 00 00 02 25 00 E1 10 0E 00 03 26 D1 02 21 53 70 91 01 08 55 01 74 65 73 74 2E 64 65 11 01 0A 54 05 65 6E 2D 55 53 54 65 73 74 51 03 01 61 63 74 00 FE 00 00 00 00 00

```

### 8.5.1.3. Read Byte (READ)

#### Description

This command reads a single EEPROM memory byte within the static memory model area of blocks 0x0-0xE.

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x54	0x00	Byte Address	0x00	-

Where P2 is coded as follow

Bit #	Value	Description
b0 – b2		Byte number to be addressed(value between 0x0 and 0x7)
b3 – b6		Block number (value between 0x0 and 0xE)
b7	(0)b	Number of the key to be used for authentication

#### Response

Data	SW1 SW2
1 byte of data	0x90 0x00

### 8.5.1.4. Write-Erase Byte (WRITE-E)

#### Description

This commands erases and then writes the value of an individual memory byte within the static memory model area of blocks 0x0-0xE.

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x56	0x00	Byte Address	0x01	1 byte of data to be written

Where P2 is coded as follow

Bit #	Value	Description
b0 – b2		Byte number to be addressed(value between 0x0 and 0x7)
b3 – b6		Block number (value between 0x0 and 0xE)
b7	(0)b	Number of the key to be used for authentication

#### Response

Data	SW1 SW2
Byte value that has been written	0x90 0x00



### 8.5.1.5. Write-No-Erase Byte (WRITE-NE)

#### Description

This command writes a byte value on an individual memory byte within the static memory model area of blocks 0x0-0xE.

This command does not erase the value of the targeted byte before writing the new data. Execution time of this command by NFC Forum tags type 1, is approximately half that of the normal write command (WRITE-E). Using this command, EEPROM bits can be set but not reset.

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x58	0x00	Byte Address	0x01	1 byte of data to be written

Where P2 is coded as follow

Bit #	Value	Description
b0 – b2		Byte number to be addressed(value between 0x0 and 0x7)
b3 – b6		Block number (value between 0x0 and 0xE)
b7	(0)b	Number of the key to be used for authentication

#### Response

Data	SW1 SW2
Value of the memory byte after execution	0x90 0x00

#### Example

Sending the following command to an NFC Forum type 1 tag that has the value 0x39 in the first EEPROM byte of block 0x1 of its static memory model area

CLA	INS	P1	P2	P3	Data
0xFF	0x58	0x00	0x10	0x01	0xA8

Will give the answer

Data	SW1 SW2
0xB9	0x90 0x00

0x39=(00111001)b

0xA8=(10101000)b

0xB9=(10111001)b

### 8.5.1.6. Read Segment (RSEG)

#### Description

This command reads out a complete segment of memory.

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x5A	0x00	Segment Address	0x00	-

Where P2 is coded as shown below

Bit #	Value	Description
b0 – b3	(0000)b	RFU
b4 – b7		Segment address (value between 0x0 and 0xF)

#### Response

Data	SW1 SW2
128 bytes of data	0x90 0x00

### 8.5.1.7. Read 8 bytes (READ8)

#### Description

This command reads out a block of memory.

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x5C	0x00	Block Address	0x00	-

P2 – Block Address - b8 - b1 - General block (0x00 -0xFF)

#### Response

Data	SW1 SW2
8 bytes of data	0x90 0x00

### 8.5.1.8. Write-Erase 8 bytes (WRITE-E8)

#### Description

This command erases a memory block and then writes a value to it.

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x5E	0x00	Block Address	0x08	8 bytes of data to be written

Where P2 codes the block address (value between 0x00 and 0xFF)

#### Response

Data	SW1 SW2
8 bytes of data that have been written	0x90 0x00

### 8.5.1.9. Write-No-Erase 8 bytes (WRITE-NE8)

#### Description

This command writes with no erase to a block of memory.

This command does not erase the value of the targeted block before writing the new data. Using this command, EEPROM bits can be set but not reset.

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x60	0x00	Block Address	0x08	8 bytes of data to be written

Where P2 codes the block address (value between 0x00 and 0xFF).

#### Response

Data	SW1 SW2
8 bytes of data	0x90 0x00

#### Example

Sending the following command to an NFC Forum type 1 tag that has the value (0x01 0x02 0x03 0x04 0x00 0x00 0x00 0x00) in the first EEPROM block

CLA	INS	P1	P2	P3	Data
0xFF	0x60	0x00	0x00	0x08	0x00 0x01 0x03 0x04 0x05 0x06 0x07 0x08

Will give the answer

Data	SW1 SW2
0x01 0x03 0x03 0x04 0x05 0x06 0x07 0x08	0x90 0x00

### 8.5.2. Commands for communicating with NFC Forum Tags Type 2

To interact with an NFC Forum tag type 2 the commands STORAGE\_CMDS\_READ\_BINARY and STORAGE\_CMDS\_WRITE\_BINARY previously described in this manual should be used.

Please refer to *NFC Forum tag type 2* specification for definition of the read and write procedures.

### 8.5.3. Commands for communication with NFC Forum Tags Type 3

This section describes APDUs Identive defined for the following FeliCa™ non-secure commands. For further details on FeliCa™ the reader should contact Sony corporation. Some description can also be found in the JIS X 6319-4 (Japanese Industry Standard) or the ISO18092 standards

- REQC
- Request Service
- Request Response
- Read
- Write

For further details on processing NFC Forum tag type 3, please refer to NFC Forum tag type 3 specification.

#### 8.5.3.1. REQC

##### Description

This command is used to detect the presence of a Type C(i.e NFC Forum Type 3/FeliCa) card in the RF field.

##### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x40	0x00	0x00	0x04	2 bytes of Service Code, 1 byte RFU, 1 byte TSN

##### Response

Data	SW1 SW2
16 bytes of NFCID2 + 2 bytes of System Code (sent only if the RFU byte is 0x01)	0x90 0x00

### 8.5.3.2. Request Service

#### Description

This command is used to know the area key version of the specified area and the service key version of the specified service of FeliCa card

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x42	Number of services/areas	0x00	2 * P1	Service Code List / Area Code List

#### Response

Data	SW1 SW2
8 bytes IDm + No. of Service or areas(n) + Service version or area version list (2*n)	0x90 0x00

### 8.5.3.3. Request response

#### Description

This command is used to know the current mode (Mode 0/1/2) of the FeliCa card

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x44	0x00	0x00	0x00	-

#### Response

Data	SW1 SW2
8 bytes IDm + Mode	0x90 0x00

### 8.5.3.4. Read

#### Description

This command is used to read the record value of the specified service of the FeliCa card

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x46	Number of services	Number of blocks	2*(P1 + P2)	Service Code List, Block List

#### Response

Data	SW1 SW2
8 bytes IDm + Status Flag 1 + Status Flag 2 + No. of blocks(n) + Block data (n*16)	0x90 0x00

### 8.5.3.5. Write

#### Description

This command is used to write the records of the specified service to the FeliCa card

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x48	Number of services	Number of blocks	$2 \times (P1 + P2) + (16 \times P2)$	Service Code List, Block List, Block Data

#### Response

Data	SW1 SW2
8 bytes IDm + Status Flag 1 + Status Flag 2	0x90 0x00

### 8.5.3.6. Request System Code

#### Description

This command searches for the system code registered in the card and returns its value. When the card is logically segmented, multiple system codes are returned in the form of a list.

#### Format

CLA	INS	P1	P2	P3	Data
0xFF	0x4A	0x00	0x00	0x00	-

#### Response

Data	SW1 SW2
8 bytes IDm + No. of System Codes (n) + System Code List (2n)	0x90 0x00

### 8.5.4. Commands for communicating with NFC Forum Tags Type 4

To interact with NFC Forum tag type 4 tags, ISO/IEC 7816-4-defined APDU are used and sent through SCM3712 using the T=CL command described earlier in this manual.

The reader can find in *NFC Forum tag type 4* specification both the definition of the APDU commands to be used and the processing methods.

## 8.6. Escape IOCTL's supported in SCM3712

The reader behavior can be configured with the help of below given IOCTL's. The ScardControl method of PC/SC (see [http://msdn.microsoft.com/en-us/library/aa379474\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa379474(VS.85).aspx)) should be used to send those IOCTLs. Code the API as given below.

```
#define IOCTL_CCID_ESCAPE SCARD_CTL_CODE(3500)

SCardControl(
    __in SCARDHANDLE hCard, // Handle obtained through ScardConnect
    __in DWORD dwControlCode, // Should be set to IOCTL_CCID_ESCAPE
    __in LPCVOID lpInBuffer, // First BYTE contains IOCTL code followed by arguments if any
    __in DWORD nInBufferSize, // Total input buffer length
    __out LPVOID lpOutBuffer, // Response buffer
    __in DWORD nOutBufferSize, // Response buffer size
    __out LPDWORD lpBytesReturned // Total number of BYTES returned from the driver.
)
```

### 8.6.1. READER\_CNTLESS\_GET\_ATS\_ATQB

#### 8.6.1.1. Description

This escape command can be used to retrieve the ATS bytes of the type A or the ATQB bytes of the type B card present in front of the SCM3712.

#### 8.6.1.2. Input buffer

Byte #	Value	Description
B0	0x93	Escape command code

#### 8.6.1.3. Output buffer

The output buffer is

Byte #	Output buffer
...	ATS or ATQB

## 8.6.2. READER\_GET\_CARD\_TYPE\_POLLING

### 8.6.2.1. Description

Using this escape command one can retrieve the type of the technology which the reader is configured to poll for.

### 8.6.2.2. Input buffer

Byte #	Value	Description
B0	0x94	Escape command code

### 8.6.2.3. Output buffer

Byte #	Output buffer
B0	Configuration register

The output buffer contains 1 byte which is coded as follow:

b7	b6	b5	b4	b3	b2	b1	b0
RFU	RFU	RFU	FeliCa 424	FeliCa 212	Topaz	Type B	Type A

Bit value 1 means SCM3712 will poll for that technology; bit value 0 means SCM3712 will not poll for this value.

A reader configured to poll only for type A and type B will therefore answer 0x03 – i.e. (00000011)<sub>b</sub> to this command.

## 8.6.3. READER\_CNTLESS\_SET\_TYPE

### 8.6.3.1. Description

This escape command can be used to configure the polling loop of SCM3712. Applications may use this to optimize the detection speed performance of their system.

### 8.6.3.2. Input buffer

Byte #	Value	Description
B0	0x95	Escape command code
B1	Configuration register	

The configuration register is 1 byte which is coded as follow:

b7	b6	b5	b4	b3	b2	b1	b0
RFU	RFU	RFU	FeliCa 424	FeliCa 212	Topaz	Type B	Type A

Bit value 1 means SCM3712 will poll for that technology; bit value 0 means SCM3712 will not poll for this value. To poll only for FeliCa 424 and type A, B1=0x11.

### 8.6.3.3. Output buffer

Output buffer
NULL



### 8.6.4. READER\_CNTLESS\_RF\_SWITCH

#### 8.6.4.1. Description

This escape message ID can be used to retrieve the current RF state (ON/OFF) of SCM3712 as well as to switch the RF state (ON/OFF).

#### 8.6.4.2. Input buffer

Byte #	Value	Description
B0	0x96	Escape command code
B1	Configuration parameter	

Configuration parameter byte can take the following values

Value	Description
0x00	Switch the RF OFF
0x01	Switch the RF ON
0xFF	Get the current RF field state

#### 8.6.4.3. Output buffer

Byte #	Configuration parameter value from input buffer	Output buffer
NA	0x00 or 0x01	NULL
B0	0xFF	0x00 if the field is OFF
		0x01 if the field is ON

### 8.6.5. READER\_CNTLESS\_DISABLE\_PPS

#### 8.6.5.1. Description

Using this escape command one can enable/disable the default automatic PPS behavior of SCM3712. When automatic PPS is disabled communication happens at the lowest baud rate commonly supported by SCM3712 and the user token.

#### 8.6.5.2. Input buffer

Byte #	Value	Description
B0	0x99	Escape command code
B1	Enable automatic PPS	0x00
	Disable automatic PPS	0x01

#### 8.6.5.3. Output buffer

Output buffer
NULL

### 8.6.6. READER\_ENABLE\_DISABLE\_848

#### 8.6.6.1. Description

This escape message can be used to enable/disable 848kbps support and to get the current state of the 848kbps support. Applications may call this function, to enable/disable 848kbps support.

#### 8.6.6.2. Input buffer

Byte #	Value	Description
B0	0x9D	Escape command code
B1	0x00	Disable 848kbps
	0x01	Enable 848kbps
	0xFF	Get current state

#### 8.6.6.3. Output buffer

Byte #	Configuration parameter value from input buffer	Output buffer
NA	0x00 or 0x01	NULL
B0	0xFF	0x00 if 848kbps disabled
		0x01 if 848kbps enabled

### 8.6.7. READER\_CNTLESS\_BAUDRATE

#### 8.6.7.1. Description

This escape message can be used to get the actual communication baud rate between SCM3712 and the user token.

#### 8.6.7.2. Input buffer

Byte #	Value	Description
B0	0x9E	Escape command code

#### 8.6.7.3. Output buffer

Byte #	Value	Comment										
B0	0xXY	<p>Nibble X corresponds to the baudrate from user token to SCM3712</p> <p>Nibble Y corresponds to the baudrate from SCM3712to user token</p> <table><tr><th>Baudrate</th><th>Nibble value</th></tr><tr><td>106kbps</td><td>0x0</td></tr><tr><td>212kbps</td><td>0x1</td></tr><tr><td>424kbps</td><td>0x2</td></tr><tr><td>848kbps</td><td>0x3</td></tr></table>	Baudrate	Nibble value	106kbps	0x0	212kbps	0x1	424kbps	0x2	848kbps	0x3
Baudrate	Nibble value											
106kbps	0x0											
212kbps	0x1											
424kbps	0x2											
848kbps	0x3											

## 8.6.8. READER\_FORCE\_BAUDRATE

### 8.6.8.1. Description

This escape command is used to force baud rate between the SCM3712 and the user token.

Once sent, the card needs to be disconnected and reconnected before the specific setting is adopted.

### 8.6.8.2. Input buffer

Byte #	Value	Description								
B0	0xAD	Escape command code								
B1	0x00	Apply baudrate specified by the card								
	0x01	Force baudrate								
B2	0xAB	<p>Byte present only if B1=0x01</p> <p>Nibble A is the baudrate between SCM3712 and user token</p> <p>Nibble B is the baudrate between user token and SCM3712</p> <p>Nibbles are coded as follow</p> <table><tr><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr><tr><td>0</td><td>848</td><td>424</td><td>212</td></tr></table>	b3	b2	b1	b0	0	848	424	212
b3	b2	b1	b0							
0	848	424	212							

### 8.6.8.3. Output buffer

Byte #	Value
NA	NULL

### 8.6.9. READER\_DISABLE\_NAK\_POLLING

#### 8.6.9.1. Description

This escape command can be used to enable/disable NAK Polling by SCM3712 once a user token has been selected.

#### 8.6.9.2. Input buffer

Byte #	Value	Description
B0	0xAC	Escape command code
B1	0x00	Enable NAK Polling
	0x01	Disable NAK Polling
	0xFF	Gets Current state of NAK polling.

#### 8.6.9.3. Output buffer

Byte #	Configuration parameter value from input buffer	Output buffer
NA	0x00 or 0x01	NULL
B0	0xFF	0x00 NAK polling enabled
		0x01 NAK polling disabled

### 8.6.10. FELICA\_PASSTHROUGH

#### 8.6.10.1. Description

This escape command can be used to send FeliCa commands as defined in JIS X 6319-4 specification. SCM3712 will add the transport level protocol data required.

#### 8.6.10.2. Input buffer

Byte #	Value	Description
B0	0xF3	Escape command code
B1	Cmd code	FeliCa command code
B2...B <sub>N+2</sub>	N bytes	Data – depends on the cmd code

#### 8.6.10.3. Output buffer

Byte #	Output buffer

Depends on the command code

## 8.7. Vendor IOCTL used for P2P mode of operation

Please note that some of the below mentioned commands do not work on all SCM3712 product versions.

IOCTL	Code	Description
<b>Generic IOCTL</b>		
IOCTL_GET_CARD_TYPE	SCARD_CTL_CODE(0x900)	Returns the type of the card detected by the reader.
<b>P2P Mode – Generic IOCTL<sup>7</sup></b>		
IOCTL_GET_DEVICE_CAPAB	SCARD_CTL_CODE(0x901)	Returns the supported target and initiator modes.
IOCTL_GET_OR_SET_RW_P2P_MODES	SCARD_CTL_CODE(0x906)	Switches the device to Reader/Write or P2P mode.
<b>P2P Initiator Mode IOCTL<sup>8</sup></b>		
IOCTL_INITIATOR_POLL	SCARD_CTL_CODE(0x902)	Polls for target devices in P2P initiator mode
IOCTL_INITIATOR_CONNECT	SCARD_CTL_CODE(0x903)	Connects to a target device in P2P initiator mode
IOCTL_INITIATOR_TRANSCEIVE	SCARD_CTL_CODE(0x904)	Sends and receives data to/from target device
IOCTL_INITIATOR_DISCONNECT	SCARD_CTL_CODE(0x905)	Disconnects from target device
<b>P2P Target Mode IOCTL<sup>9</sup></b>		
IOCTL_TARGET_RECEIVE	SCARD_CTL_CODE(0x907)	Receive data from initiator in target mode
IOCTL_TARGET_SEND	SCARD_CTL_CODE(0x908)	Send data to initiator in target mode

### Generic IOCTLs

#### 8.7.1. IOCTL\_GET\_CARD\_TYPE

##### 8.7.1.1. Description

This vendor IOCTL is used to get the type of the card or the target detected by the reader.

##### 8.7.1.2. Input buffer

NULL.

<sup>7</sup>Only applicable for SCM3712 NFC and SCM3712 EA

<sup>8</sup>Only applicable for SCM3712 NFC and SCM3712 EA

<sup>9</sup>Only applicable for SCM3712 NFC and SCM3712 EA

### 8.7.1.3. Output buffer

Byte #	Value	Description
B0	0x00	Type of card or target detected by the reader UNKNOWN
	0x01	MIFARE 1K
	0x02	MIFARE 4K
	0x03	MIFARE UL
	0x04	ISO14443-4A/MIFARE DESFIRE
	0x05	FeliCa
	0x06	JEWEL / TOPAZ
	0x07	ISO14443-4B
	0x08	NFC TARGET

### 8.7.1.4. Return Value

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small

## 8.7.2. P2P Mode – Generic IOCTL

### 8.7.2.1. IOCTL\_GET\_DEVICE\_CAPAB

#### 8.7.2.2. Description

This vendor IOCTL is used to get the capabilities of the reader in initiator mode and in target mode.

### 8.7.2.3. Input buffer

Null.

#### 8.7.2.4. Output buffer

```
typedef struct _DEVICE_CAPABILITIES
{
    BYTE  byInitiatorModes;
    BYTE  byTargetModes;

} DEVICE_CAPABILITIES, *PDEVICE_CAPABILITIES
```

The bit mask used in byInitiatorModes and byTargetModes is as follows

MIFARE UL	- 0x01
MIFARE STD	- 0x02
ISO14443-4A	- 0x04
ISO14443-4B	- 0x08
FeliCa	- 0x10
NFC	- 0x20
JEWEL / TOPAZ	- 0x40

#### 8.7.2.5. Return Value

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small

### 8.7.3. IOCTL\_GET\_OR\_SET\_RW\_P2P\_MODES

#### 8.7.3.1. Description

IOCTL\_GET\_OR\_SET\_RW\_P2P\_MODES is used to switch the device from Reader/Writer mode to P2P mode and vice versa. It can also be used to retrieve the current mode of the device. By default, the device is in Reader/Writer mode. When switched to P2P mode, it can be defined whether the device shall operate in active or in passive mode.

#### 8.7.3.2. Input buffer

Byte #	Value	Description
B0	0x00	RW mode
	0x01	P2P mode
	0x02	Get the current mode
B1	0x00	<b>Passive mode</b> - The Initiator generates the RF field and the Target responds to an Initiator command in a load modulation scheme
	0x01	<b>Active mode</b> - Mode in which both the Initiator and the Target use their own RF field to enable the communication

#### 8.7.3.3. Output buffer

NULL, if the IOCTL is to switch the modes.

If the IOCTL is used to get the current mode of the device

Byte #	Value	Description
B0	0x00	RW mode
	0x01	P2P mode

#### 8.7.3.4. Return Value

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small



## 8.8. P2P Initiator Mode IOCTLs

### 8.8.1. IOCTL\_INITIATOR\_POLL

#### 8.8.1.1. Description

This IOCTL is sent to the device that would act as the initiator. On receiving this IOCTL, the initiator firmware / driver would

- Check if the device is already in the P2P mode
- If not, automatically switch the device to P2P mode
- Start polling for the target device.

The response would contain the target presence / absence status.

#### 8.8.1.2. Input Buffer

```
typedef struct _INITIATOR_POLL_PARAMS
```

```
{
```

```
    BYTE  byMaxNoOfTargets;
```

```
    BYTE  byActivePassive;
```

```
    BYTE  byBrTy;
```

```
} INITIATOR_POLL_PARAMS, *PINITIATOR_POLL_PARAMS;
```

Byte #	Value	Description
B0	0x00	Maximum number of targets – 0x01
B1	0x00	<b>Passive mode</b> - The Initiator generates the RF field and the Target responds to an Initiator command in a load modulation scheme
	0x01	<b>Active mode</b> - Mode in which both the Initiator and the Target use their own RF field to enable the communication
B2	0x00	Baud rate and modulation type to be used during the initialization 106kbps
	0x01	212kbps
	0x02	424kbps

#### 8.8.1.3. Output Buffer

Byte #	Value	Description
B0	0x00	Card / target absent
	0x01	Card / target present

#### 8.8.1.4. Return Value

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small

#### 8.8.2. IOCTL\_INITIATOR\_CONNECT

##### 8.8.2.1. Description

This IOCTL is sent to the device that would act as the initiator. On receiving this IOCTL, the initiator firmware / driver would

- Check whether the device is already in P2P mode.
- If not, switch the device to P2P mode automatically.
- Check if the Poll command was previously sent and target successfully detected.
- If not, poll for wPollTimems, detect the target device and connect it.

If wPollTime parameter is non-zero, the firmware / driver would poll for the target device for wPollTime milliseconds. Otherwise, the firmware / driver would poll for the target device for 3000ms.

##### 8.8.2.2. Input Buffer

typedef struct \_INITIATOR\_CONNECT\_PARAMS

```
{
    BYTE  byActivePassive, // 0x00 – Passive; 0x01 – Active
    BYTE  byBrTy;          // 0x00 – 106kbps; 0x01 – 212kbps; 0x02 – 424kbps
    BYTE  byNext;          // b0–NFCID3 present; b1–Gi present
    BYTE  byNFCID3i[10]; // NFC ID of the Initiator
    WORD  wPollTime;       // Range from 1000ms to 5000ms
    BYTE  byGiLen;         // Length of the general bytes
    BYTE  byGi[48];        // General bytes shall be optional and designate general information
} INITIATOR_CONNECT_PARAMS, *PINITIATOR_CONNECT_PARAMS;
```

##### 8.8.2.3. Output Buffer

typedef struct \_INITIATOR\_CONNECT\_RESP

```
{
    BYTE  byTgNum; // Logical number attributed to the activated target
    BYTE  byNFCID3t[10]; // Random identifier of the target
    BYTE  byDIDt; // DID byte sent by the target
    BYTE  byBSt; // Supported send-bit rate by the target
    BYTE  byBRt; // Supported receive-bit rate of the target
    BYTE  byTO; // Timeout value of the target in transport protocol
    BYTE  byPPT; // Optional parameters of the target
```

BYTE byGtLen;// Length of the general bytes

BYTE byGt[47];//Generalbytesshall be optional and designate general information.

} INITIATOR\_CONNECT\_RESP, \*PINITIATOR\_CONNECT\_RESP;

#### 8.8.2.4. Return Value

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small

### 8.8.3. IOCTL\_INITIATOR\_TRANSCEIVE

#### 8.8.3.1. Description

This IOCTL is sent to the initiator device with the data from the host application. The initiator device would in turn send this data to the target device. The initiator then receives the response from the target and sends it to the application.

#### 8.8.3.2. Input Buffer

Input buffer should contain data to be sent to the target.

#### 8.8.3.3. Output Buffer

Output buffer will hold target's output data.

#### 8.8.3.4. Return Value

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small

### 8.8.4. IOCTL\_INITIATOR\_DISCONNECT

#### 8.8.4.1. Description

This IOCTL is sent to the device that would act as the initiator. On receiving this IOCTL, the initiator would disconnect the target device.

#### 8.8.4.2. Input Buffer

Byte #	Value	Description
B0	0xXX	Logical number of the Target

### 8.8.4.3. Output Buffer

NULL.

### 8.8.4.4. Return Value

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small

## P2P Target Mode IOCTLs

### 8.8.5. IOCTL\_TARGET\_RECEIVE

#### 8.8.5.1. Description

This IOCTL is used by the application to receive the data from the target device.

The target firmware / driver would do the following

- Automatically switch to P2P mode if the device is not in P2P mode. For this the target parameters that are required to initialize the target must be sent by the application through this IOCTL.
- Automatically send the response data to the initiator device for any incoming connection requests. Based on the Status byte of the IOCTL\_TARGET\_RECEIVE, the application can decide whether to send IOCTL\_TARGET\_SEND or not.
- Receive timeout can be specified in milliseconds in the range of 1000ms to 5000ms in the input buffer offsets 27 and 28. If this parameter is 0, the firmware / driver would use default receive timeout of 1000ms.

**8.8.5.2. Input Buffer**

Byte #	Value	Description
B0	0x00	RW mode
	0x01	P2P mode
B1	0x00	Passive mode
	0x01	Active mode
B2 – B3		MIFARE SENS Response
B4 – B6		MIFARE NFCIDt
B7		MIFARE SEL Response
B8 – B9		FeliCa polling response
B10 – B15		FeliCa NFCID2t
B16 – B23		FeliCa PAD bytes
B24 – B25		FeliCa System Code
B26		NFCID3t
B27 – B28		Receive timeout ranging from 1000ms to 5000ms
B29		General bytes length
B30 onwards		General bytes (max 47 bytes)

**8.8.5.3. Output Buffer**

Byte #	Value	Description
B0	0x00	STATUS_SUCCESS
	0x01	STATUS_TGT_STARTED
	0x02	STATUS_TGT_CONNECTED
	0x03	STATUS_TGT_DISCONNECTED
B1 onwards	0xXX	Data from the initiator device

**8.8.5.4. Return Value**

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small

### 8.8.6. IOCTL\_TARGET\_SEND

#### 8.8.6.1. Description

This IOCTL is used by the application to send the data to the initiator device as response to the command sent by the initiator.

#### 8.8.6.2. Input Buffer

Input buffer should contain the data to be sent to initiator.

#### 8.8.6.3. Output Buffer

NULL.

#### 8.8.6.4. Return Value

Return Value	Value	Description
ERROR_SUCCESS	0	Successful execution
ERROR_GEN_FAILURE	31	Operation failed
ERROR_INSUFFICIENT_BUFFER	122	Input or output buffer too small

## 9. Annexes

### 9.1. Annex A

#### 9.1.1. Status words table

SW1	SW2	Description
0x90	0x00	NO ERROR
0x67	0x00	LENGTH INCORRECT
0x6D	0x00	INVALID INSTRUCTION BYTE
0x6E	0x00	CLASS NOT SUPPORTED
0x6F	0x00	UNKNOWN COMMAND
0x63	0x00	AUTHENTICATION ERROR
0x65	0x81	STATUS_COMMAND_FAILED
0x65	0x91	STATUS_SECURITY_STATUS_NOT_MET
0x68	0x00	CLASS BYTE INCORRECT
0x6A	0x81	FUNCTION NOT SUPPORTED
0x6B	0x00	WRONG PARAMETER P1-P2

#### 9.1.2. Further information about PC/SC

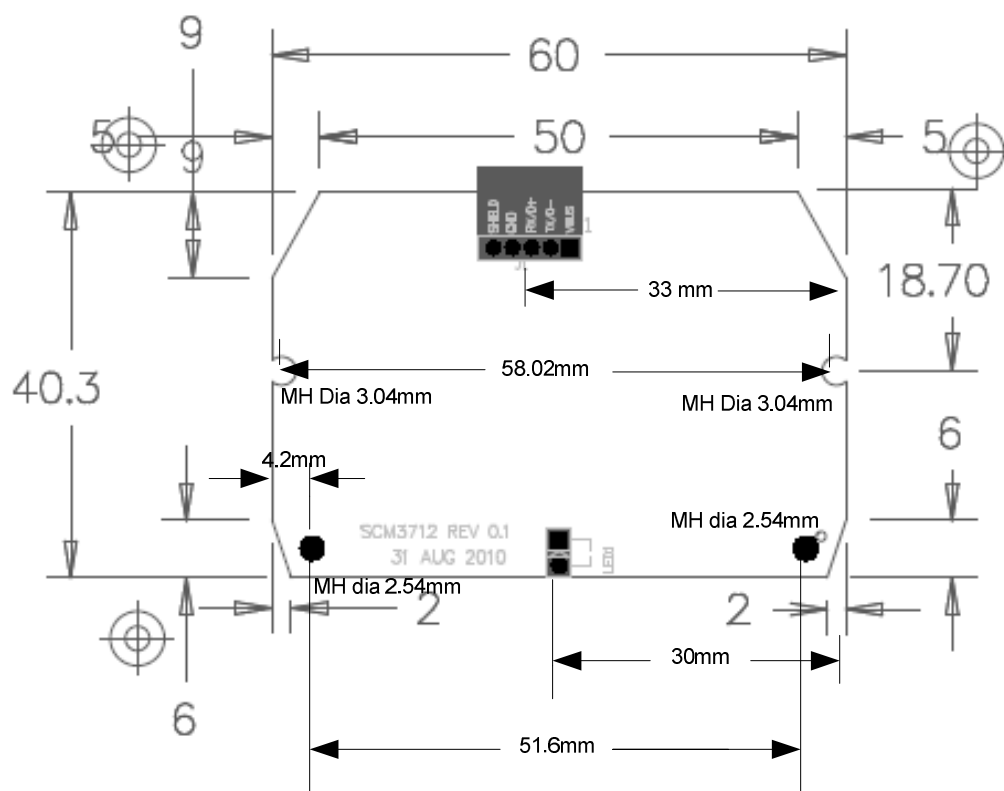
The PC/SC specifications can be downloaded from the PC/SC workgroup web site: [www.pcscworkgroup.com](http://www.pcscworkgroup.com).

Further information on the Microsoft resource manager API can be found online on [http://msdn.microsoft.com/en-us/library/aa380149\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa380149(VS.85).aspx).

## 9.2. Annex B – Mechanical drawings

### 9.2.1. SCM3712 and SCM3712 NFC

Board Size:	60mm X 40.3mm / 2,36 x 1,59 Inch
PCB thickness:	1.0mm $\pm$ 10% / 0,039 Inch $\pm$ 10%
Max PCB height with components:	10mm / 0,39 Inch
Mounting hole Diameter:	2.54mm / 0,1 Inch



All dimensions are in mm  
10mm  $\approx$  0,39 Inch

Fig.9-1: SCM3712 & SCM3712 NFC Reader PCB top view



### 9.2.2. SCM3712 EA and external antenna board

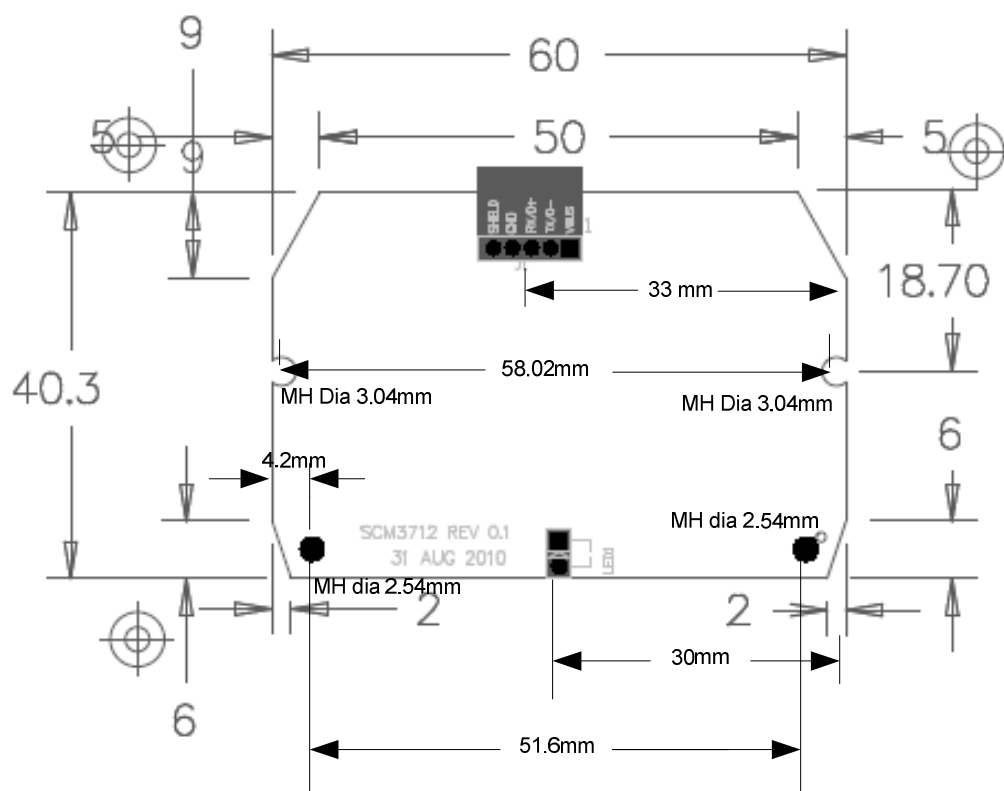
#### Main board

Board Size: 60mm X 40.3mm / 2,36 x 1,59 Inch

PCB thickness: 1.0mm  $\pm$  10% / 0,039 Inch  $\pm$  10%

Max PCB height with components: 10mm / 0,39 Inch

Mounting hole Diameter: 2.54mm / 0,1 Inch



*All dimensions are in mm  
10mm  $\approx$  0,39 Inch*

Fig.9-2: SCM3712 EA Reader PCB top view

**Antenna board**

Board Size: 92.7mm X 49.5mm / 3,65 x 1,95 Inch

PCB thickness: 1.6mm  $\pm$  10% / 0,063 Inch  $\pm$  10%

Max PCB height with components: 10mm / 0,39 Inch

Mounting hole Diameter: 3mm / 0,12 Inch

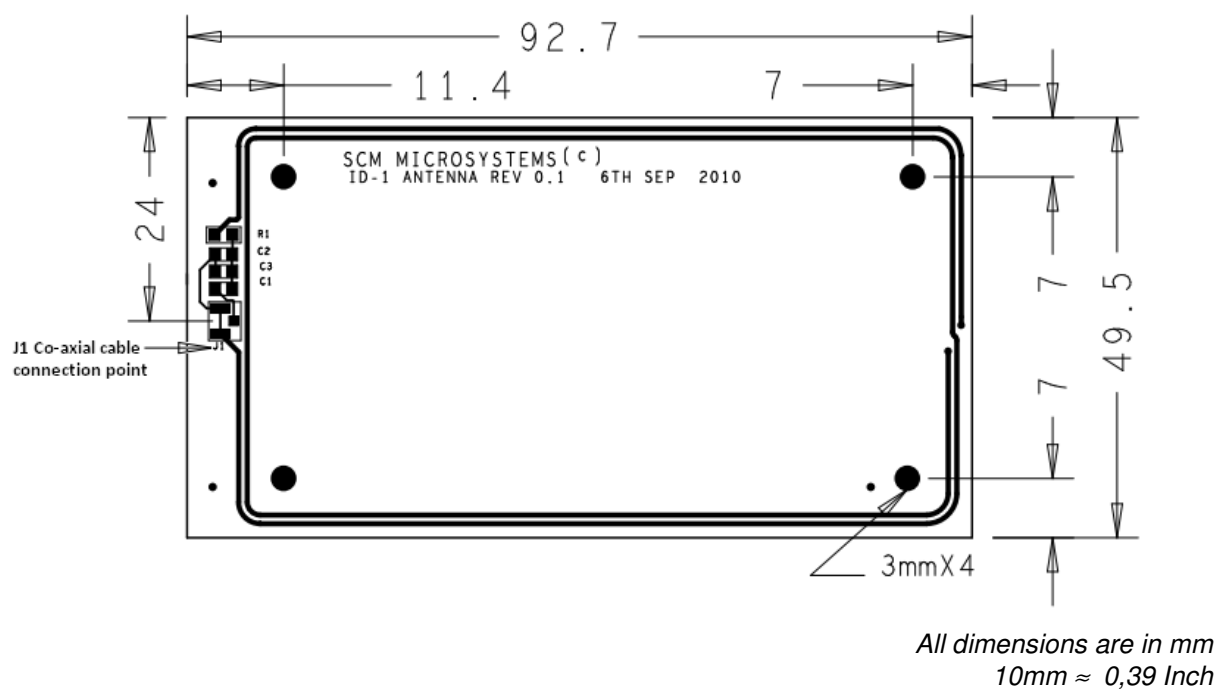


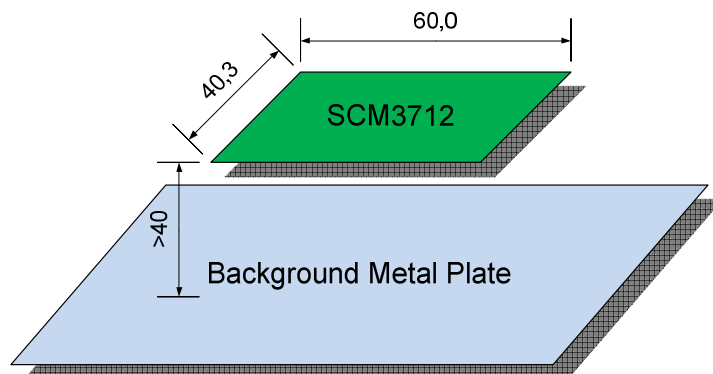
Fig.9-3: External Antenna PCB top view

### 9.3. Annex C – Installation Guidelines

The SCM3712 reader is a contactless reader working on 13.56MHz. In order to enable the maximum performance the reader ideally needs to be installed in an metal free environment. The mechanical integration may not allow for metal plates or metal rings in close proximity. Smaller metallic units like single screws or rivets for mounting the reader can be used without remarkable influence on the overall performance. Ideally also the mounting is done by non-metallic devices only. Of course also the housing where the reader is being built in may not be made of metal or metalized plastics. In any case the housing needs to have a metal free window in front of the reader around the reader antenna. The bigger the distance is between metal and the reader the less the influence on performance will be noticed.

The below drawings show the principles. Please note even at the mentioned distances there will be some influence on the reader performance. The performance degradation is in a range were the end user shouldn't see any negative influence. We strongly recommend performing detailed tests before a product roll out.

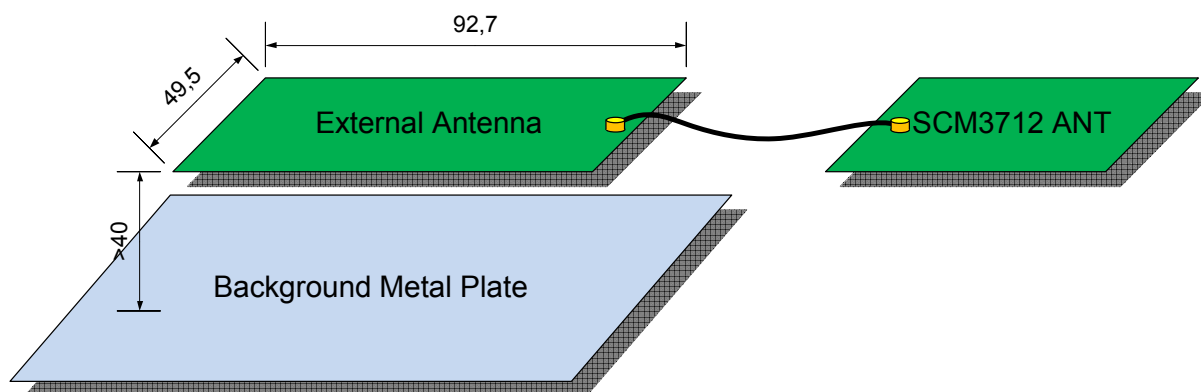
### 9.3.1. SCM3712 & SCM3712 NFC Mount on Metal Instruction



*All dimensions are in mm  
10mm  $\approx$  0,39 Inch*

Fig.9-4: SCM3712 & SCM3712 NFC Mount on Metal Instruction

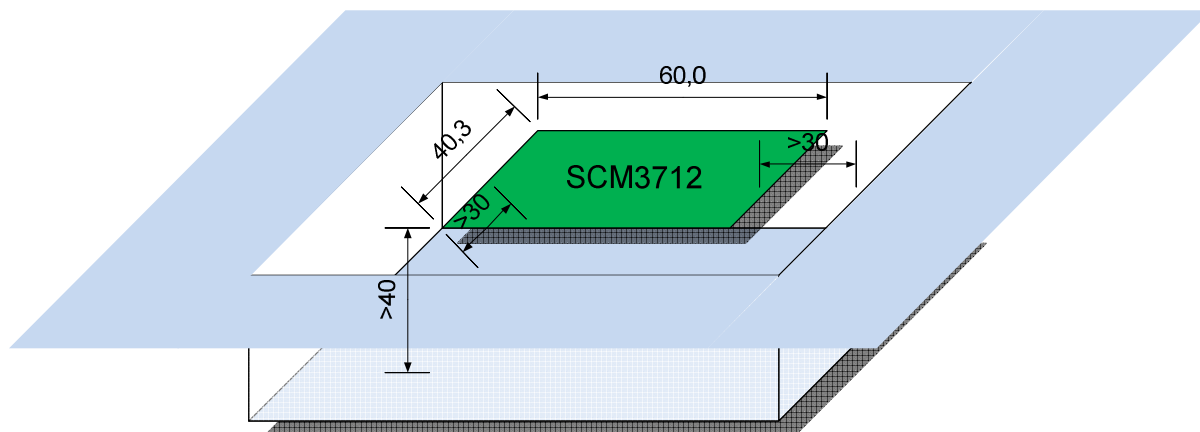
### 9.3.2. SCM3712 ANT Mount on Metal Instruction



*All dimensions are in mm  
10mm  $\approx$  0,39 Inch*

Fig.9-5: SCM3712 EA Mount on Metal Instruction

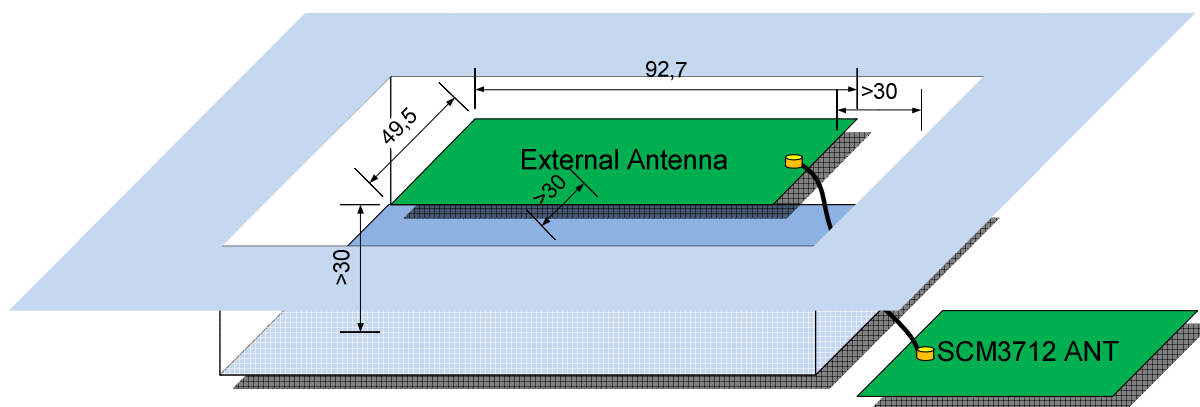
### 9.3.3. SCM3712 & SCM3712 NFC Mount in Metal Environment Instruction



All dimensions are in mm  
10mm  $\approx$  0,39 Inch

Fig.9-6: SCM3712 & SCM3712 NFC Mount in Metal Environment Instruction

### 9.3.4. SCM3712 ANT Mount in Metal Environment Instruction



All dimensions are in mm  
10mm  $\approx$  0,39 Inch

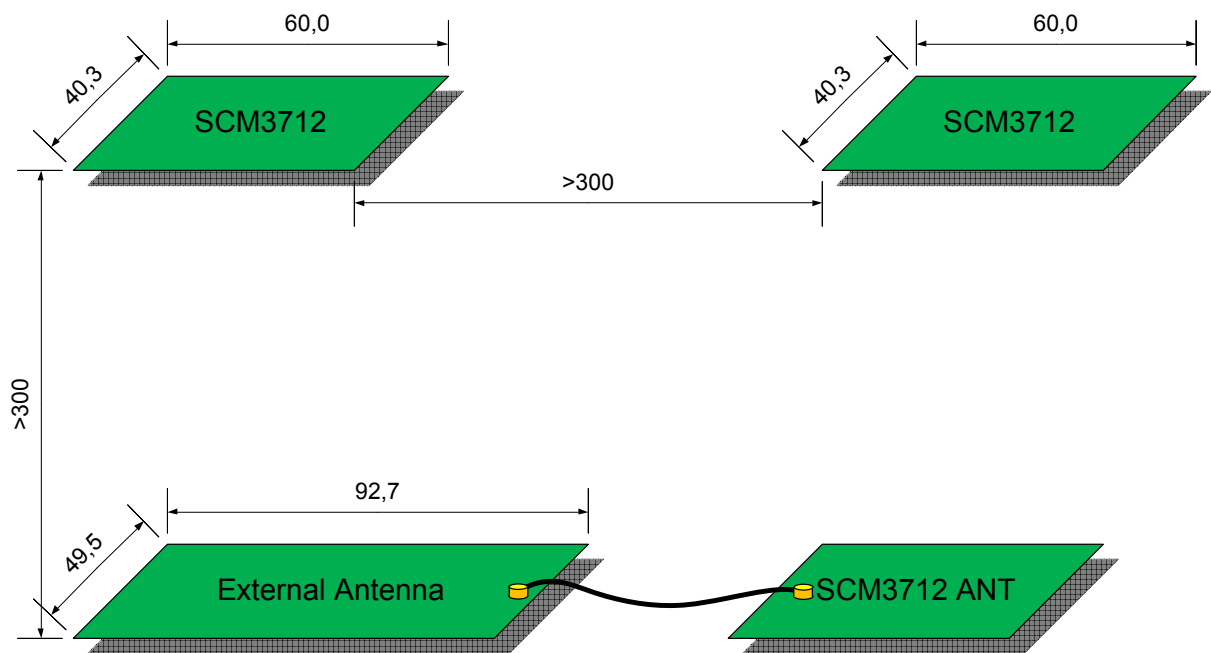
Fig.9-7: SCM3712 EA Mount in Metal Environment Instruction

### 9.3.5. Recommended distance between readers

Contactless readers are being influenced in performance by their environment. Other contactless devices working in the same or a similar frequency range also can have an influence on the overall performance. The below drawing illustrates the minimum recommended distance between two readers without influencing each other. A closer distance than the recommended distance may result in a reduced read range or a negatively influenced interoperability. It makes no real difference which SCM3712 reader boards are in proximity to each other.

The below dimensions are good as a guideline and valid for reading from and writing to any kind of card or token.

Of course for NFC Peer-To-Peer (P2P) functionality readers may be put in close proximity. In this case the used RF-protocol helps to minimize the influence as the command flow is being synchronized in that case unlike the normal ISO14443 read/write mode.



*All dimensions are in mm  
10mm ≈ 0,39 Inch*

Fig.9-8: SCM3712 EA Mount in Metal Environment Instruction

## 9.4. Annex D – Certifications

As a service for our customers, Identive already successfully has performed the EMI pretests to ensure the SCM3712 readers are ready for best compliance against applicable international radio certification rules like CE, FCC, VCCI or Japan Radio certification. As the reader is intended to be integrated into a final device, the final device needs to be tested against the specific rules as a complete device. Even if the reader board would come with a certification, it was invalid once integrated into the final device.