PMA13257-AB
DRAFT - PUBLIC

SpringCard FunkyGate-IP NFC

Integration and Configuration Guide

www.springcard.com

Document Identification

Category	Owner's Manual		
Family/Customer	FunkyGate NFC Series		
Reference	PMA13257	Version	AB
Status	draft	Classification	Public
Keywords			
Abstract			

File name	C:\Users\johann\Documents\En cours\Notices\Lecteurs\[PMA13257-AB] FunkyGate-IP NFC Integration and Configuration Guide.odt		
Date saved	26/02/14	Date printed	

Revision History

Ver.	Date	Author	Valid. by		Approv. by	Details
			Tech.	Qual.		
AA	29/09/13	JDA				Created from PMA959P
AB						Rewritten the authentication scheme in chapter 6

Contents

1.INTRODUCTION....6
1.1.Abstract....6
1.2.Supported product....6
1.3.Audience....7
1.4.Support and updates....7
1.5.Related documents....7
2.HARDWARE INSTALLATION....8
2.1.SpringCard FunkyGate-IP NFC....8
2.2.SpringCard FunkyGate-IP+POE NFC....8
3.DEFINE READER'S IP ADDRESS....9
3.1.Assign an IP address using NDDU software....9
3.1.1.Download and install the NDDU software....9
3.1.2.Run the NDDU software....9
3.1.3.Discovered devices....10
3.1.4.Configure a Reader....11
3.1.5.Verify the new configuration....12
3.2.Assign an IP address using a Master Card....13
4.TELNET ACCESS TO THE READER....14
4.1.Reader's Console....14
4.1.1.Open a Telnet session to the reader....14
4.1.2.Sending a command to the Reader....15
4.1.3.List of Console commands....16
5.TCP CLIENT/SERVER PROTOCOL – LOW LAYERS, PLAIN MODE....17
5.1.Abstract....17
5.2.Presentation layer....17
5.2.1.Block format....17
5.2.2.Description of the fields....18
5.2.3.Size of the blocks....18
5.2.4.Format of the TYPE byte....19
5.3.General communication flow....20
5.3.1.Session establishment....20
5.3.2.Nominal dialogue....20
5.3.3.Timings....21
5.3.4.Chaining....21
5.4.Error handling and recovery....21
5.4.1.For the Reader....21
5.4.2.For the Host....21
5.4.3.Recovery....21
5.5.Application layer....22
6.TCP/CLIENT SERVER PROTOCOL – LOW LAYERS, AUTHENTICATED MODE....23
6.1.Abstract....23
6.2.Supported cipher protocols....23
6.2.1.Blowfish....23
6.2.2.3DES2K....24
6.2.3.AES....24
6.3.3-pass authentication....24
6.3.1.Reader's HELO....24
6.3.2.Host's HELO-Auth....24
6.3.3.Authentication, step 1....25
6.3.4.Authentication, step 2....25
6.3.5.Authentication, step 3....26
6.3.6.Host's HELO-OK....26
6.4.Session key....26
6.5.New authentication – Generation of a new session key....26
6.6.Presentation layer after authentication....27
6.6.1.Block format....27
6.6.2.Description of the fields....27
6.6.3.Size of the blocks....27
6.6.4.Format of the TYPE byte....28
6.7.Sequence numbers....28
6.8.Checksum, padding, ciphering....29
6.8.1.Checksum....29
6.8.2.Padding....29
6.8.3.Ciphering....30
6.8.4.Chaining....30
6.9.General communication flow....30
6.9.1.Nominal dialogue....30
6.9.2.Timings....30
6.9.3.Chaining....31
6.10.Error handling and recovery....31
6.10.1.For the Reader....31
6.10.2.For the Host....31
6.10.3.Recovery....31
6.11.Application layer....32
7.APPLICATION LAYER PROTOCOL....33
7.1.Principles....33
7.2.Host → Reader, available with both Administration and Operation keys....33
7.2.1.Get Global Status....33
7.2.2.Start/Stop Reader....33
7.2.3.Clear LEDs command....34
7.2.4.Set LEDs command....34
7.2.5.Start LED sequence command....34
7.2.6.Buzzer command....35
7.3.Host → Reader, available with Administration key only....35
7.3.1.Write Configuration Register....35
7.3.2.Erase Configuration Register....35
7.3.3.Reset the Reader....35
7.4.Reader → Host....36
7.4.1.Reader Identifier....36
7.4.2.Tamper Status....36
7.4.3.Card Read....36
7.4.4.Card Inserted....36
7.4.5.Card Removed....36
8.EDITING READER'S CONFIGURATION....38
8.1.Through the Telnet link....38

8.1.1.Reading Configuration Registers.................................38
8.1.2.Writing Configuration Registers..................................39
8.2.Using Master Cards..39
8.3.Using NFC peer-to-peer...39
8.4.Through the TCP Client/Server interface.......................39

9.GLOBAL CONFIGURATION OF THE READER........................40

9.1.General options...40
9.2.Delays and repeat...41
9.3.LEDs and Buzzer..41
9.4.TCP configuration..42
9.4.1.IPv4 address, mask, and gateway................................42
9.4.2.Server port...43
9.4.3.Server security settings and keys................................43
9.5.Security options...44
9.6.Password...45

10.THE TEMPLATE SYSTEM..46

1. Introduction

1.1. Abstract

SpringCard FunkyGate-IP NFC is a RFID (13.56MHz) and NFC wall-mount Reader, for access control applications. **SpringCard FunkyGate-IP NFC** features an exclusive TCP/IP over Ethernet interface.

The attractive styling and the efficiency of the Ethernet interface make it the preferred choice for corporate environments. Advanced support of the widest range of technologies and exclusive security features allow high-end access control schemes to be deployed seamlessly.

Thanks to a **versatile Template System** (shared with all other **SpringCard** Readers and RFID/NFC Scanners), **SpringCard FunkyGate-IP NFC** is able to read either a serial number or virtually any data coming from standard ISO/IEC 14443 proximity cards, ISO/IEC 15693 vicinity labels or tags. It is also able to fetch NDEF data from RFID chips formatted according to one the NFC Forum Tag specifications, and to receive NDEF data from a NFC Forum “peer-to-peer” (SNEP server on top of LLCP).

The **SpringCard FunkyGate-IP+POE NFC** version provides the “powered by the network” (POE) feature.

This document provides all necessary information to configure both the **FunkyGate-IP NFC** and **FunkyGateIP-POE + NFC** Readers, and to develop a software that will handle data coming from the Reader, and to drive or re-configure the Reader when needed.

1.2. Supported product

Order code	Product
FPF13253	**FunkyGate-IP NFC:** new generation wall-mount RFID/NFC/contactless card Reader, with Ethernet interface (10 or 100 Mbit/s)
FPF13254	**FunkyGate-IP+POE NFC**: new generation wall-mount RFID/NFC/contactless card Reader, with Ethernet interface (10 or 100 Mbit/s), powered by the network

1.3. Audience

This manual is designed for use by application developers and system integrators. It assumes that the reader has a good knowledge of computer development, TCP/IP networks, and a good knowledge of the RFID/NFC technologies.

1.4. Support and updates

Useful related materials (product datasheets, application notes, sample software, HOWTOs and FAQs…) are available at SpringCard's web site:

www.springcard.com

Updated versions of this document and others are posted on this web site as soon as they are available.

For technical support enquiries, please refer to SpringCard support page, on the web at

www.springcard.com/support

1.5. Related documents

1.5.1. Products' specifications

You'll find the feature-list and the technical characteristics of every product in the corresponding leaflet.

Document ref.	Content
PFL13274	**FunkyGate-IP NFC** product leaflet
PFL13275	**FunkyGate-IP+POE NFC** product leaflet

1.5.2. Common documentations

All SpringCard Readers and RFID Scanners products share the same "card processing" system through 1 to 4 processing templates. How the Reader process the card is therefore detailed in a document shared among all products in the family.

Document ref.	Content
PMA13205	**Readers / RFID Scanners Template System**

2. Hardware installation

2.1. SpringCard FunkyGate-IP NFC

To be written

2.2. SpringCard FunkyGate-IP+POE NFC

To be written

3. Define Reader's IP Address

The Reader comes out of factory without an IP address. This means that you must assign it an IP address before being able to access it either through Telnet link (chapter 4) or using the TCP client/server protocol depicted in chapters 5 and 6.

Using **SpringCard Network Device Discovery Utility (NDDU)** is the preferred method to assign an IP address to the Reader.

> *This Reader does not support the Dynamic Host Configuration Protocol (DHCP). Only fixed IPv4 addresses are supported.*

3.1. Assign an IP address using NDDU software

SpringCard Network Device Discovery Utility (NDDU) is a Windows-based software that discovers and configures SpringCard Device connected on same the Local Area Network (LAN) as the computer it is running on.

> *Please use a wired network connection, and make sure the Reader(s) you want to configure are on the same LAN as your computer. NDDU makes use of broadcast UDP frames to discover and configure the Readers; therefore, it won't work behind a router or gateway.*

3.1.1. Download and install the NDDU software

Make sure your Windows account has administrative privileges.

Download the installer from URL

www.springcard.com/download/find/file/sn13210

Install the software.

> *This software relies on the .NET framework version 4. Please download and install this framework from Microsoft's in case it hasn't already been deployed onto your computer.*

3.1.2. Run the NDDU software

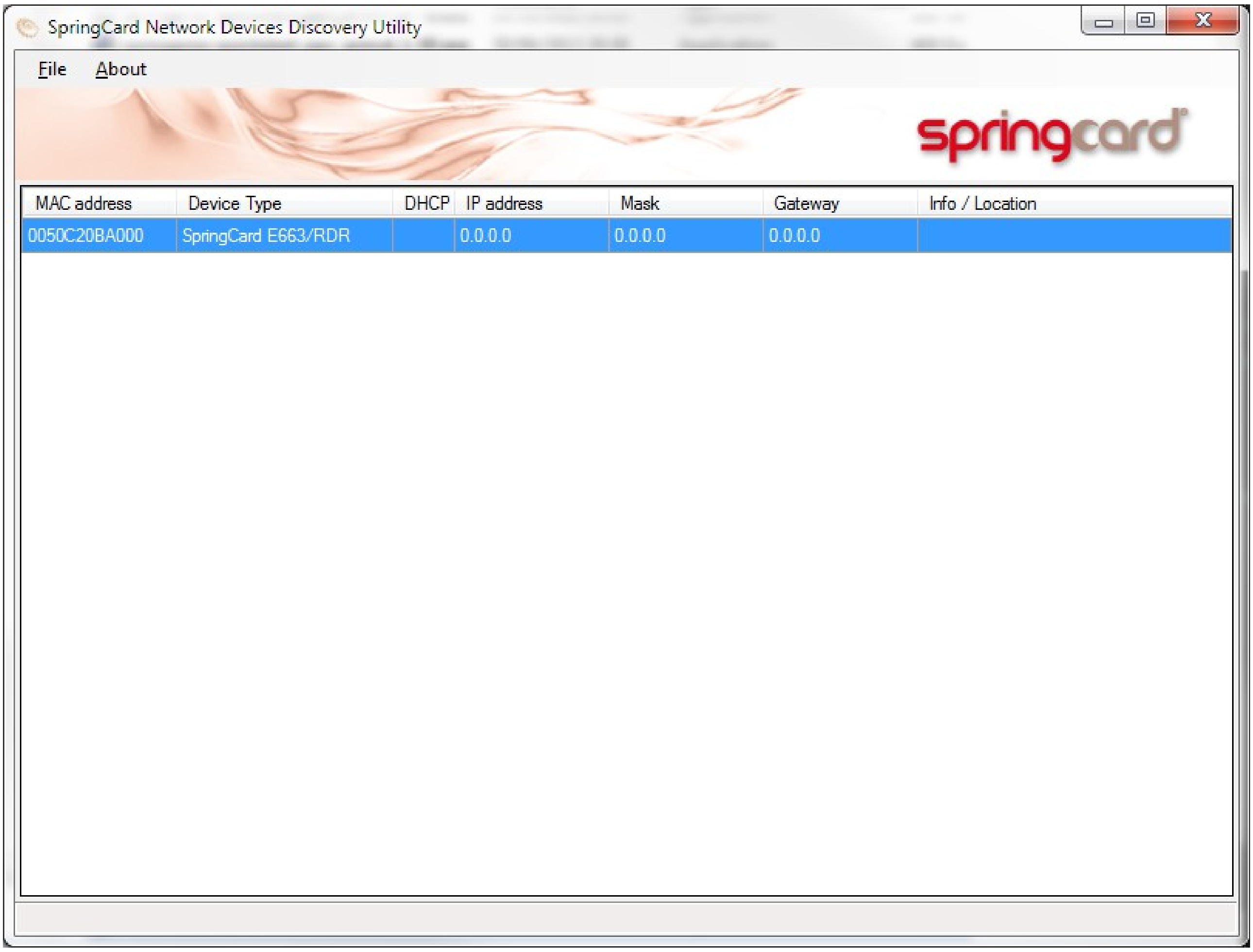
Make sure your Windows account has administrative privileges.

Launch the software: Start Menu → SpringCard → Network Discovery → Network Device Discovery Utility.

On first startup, you should be prompted by Windows Firewall whether you want to allow NDDU to access the network. Please confirm.

3.1.3. Discovered devices

After a few seconds, NDDU displays the list of devices it has found on the LAN.



MAC address	Device Type	DHCP	IP address	Mask	Gateway	Info / Location
0050C20BA000	SpringCard E663/RDR		0.0.0.0	0.0.0.0	0.0.0.0	

The software's main screen shows 7 columns:

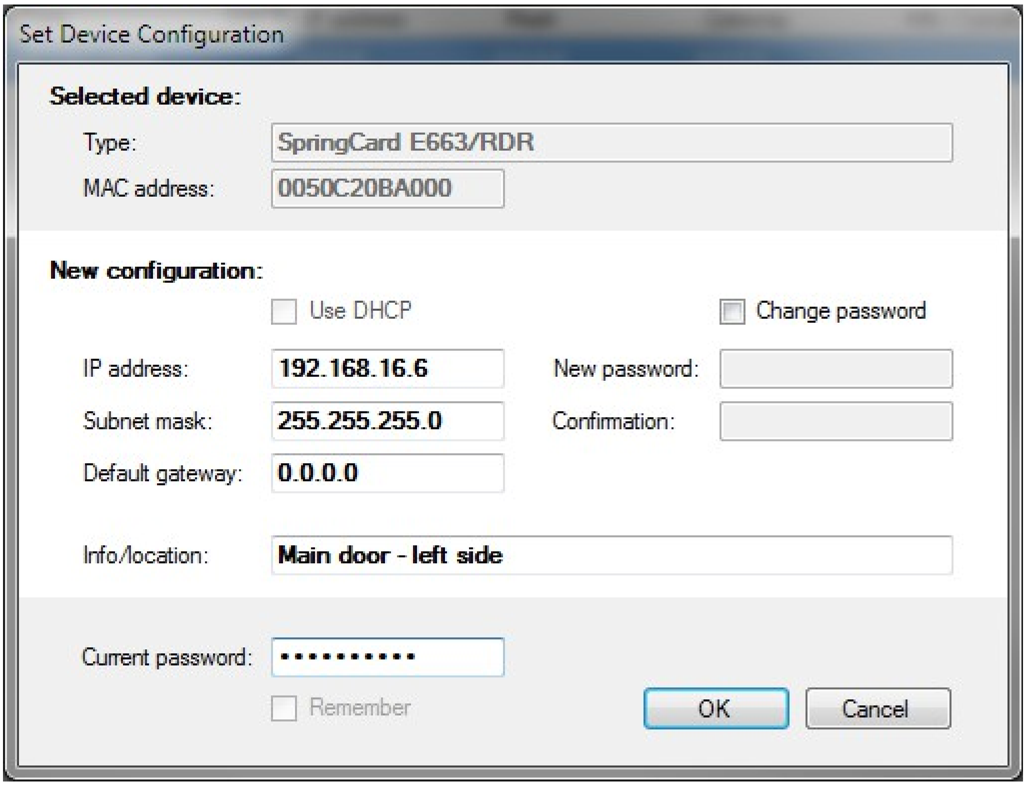
- The MAC address (Ethernet address and also serial number) of every SpringCard Device found on the LAN,
- The device type (code name **SpringCard E663/RDR** for **SpringCard FunkyGate-IP NFC** and related products),
- Whether DHCP is enabled or not (DHCP is not supported by **SpringCard FunkyGate-IP NFC**),
- The device's current IP address, local network mask, and default gateway. Until the device has been properly configured, those entries show has “0.0.0.0”,

]

- A user-defined string named “Info / location”, which will be used as an hint to identify the device in your own system.

3.1.4. Configure a Reader

Double-click one of the devices in the list. The configuration form appears:



Set Device Configuration

Selected device:

Type: SpringCard E663/RDR

MAC address: 0050C20BA000

New configuration:

□ Use DHCP

□ Change password

IP address: 192.168.16.6

New password:

Subnet mask: 255.255.255.0

Confirmation:

Default gateway: 0.0.0.0

Info/location: Main door - left side

Current password: ••••••••••

□ Remember

The form shows the device's current configuration. Enter the new configuration. IP address and subnet mask are mandatory data and couldn't be left empty. The default gateway is optional; if the devices won't need to use a gateway, leave this field to “0.0.0.0”.

In the “info/location” field, enter a short string (less than 32 characters) as a reminder of the device's location or role.

Terminate by entering the device's current password to confirm your allowed to change this device's configuration.

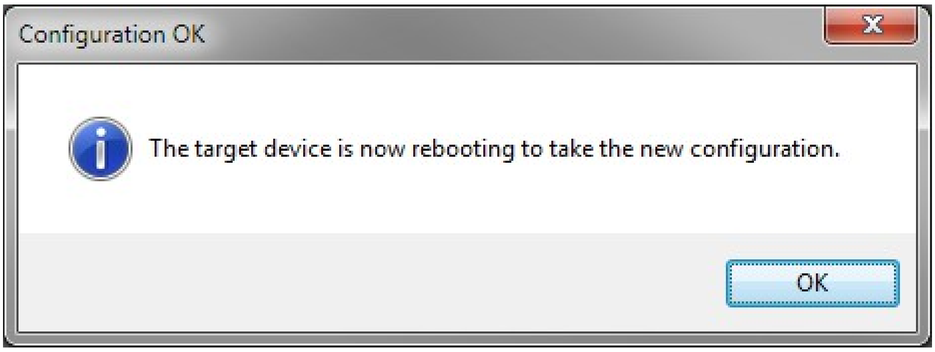
> *The default password for all devices is* ***springcard****.*

Check the box “change password” and enter a new password twice if you want to change it.

When ready, click “OK”.

3.1.5. Verify the new configuration

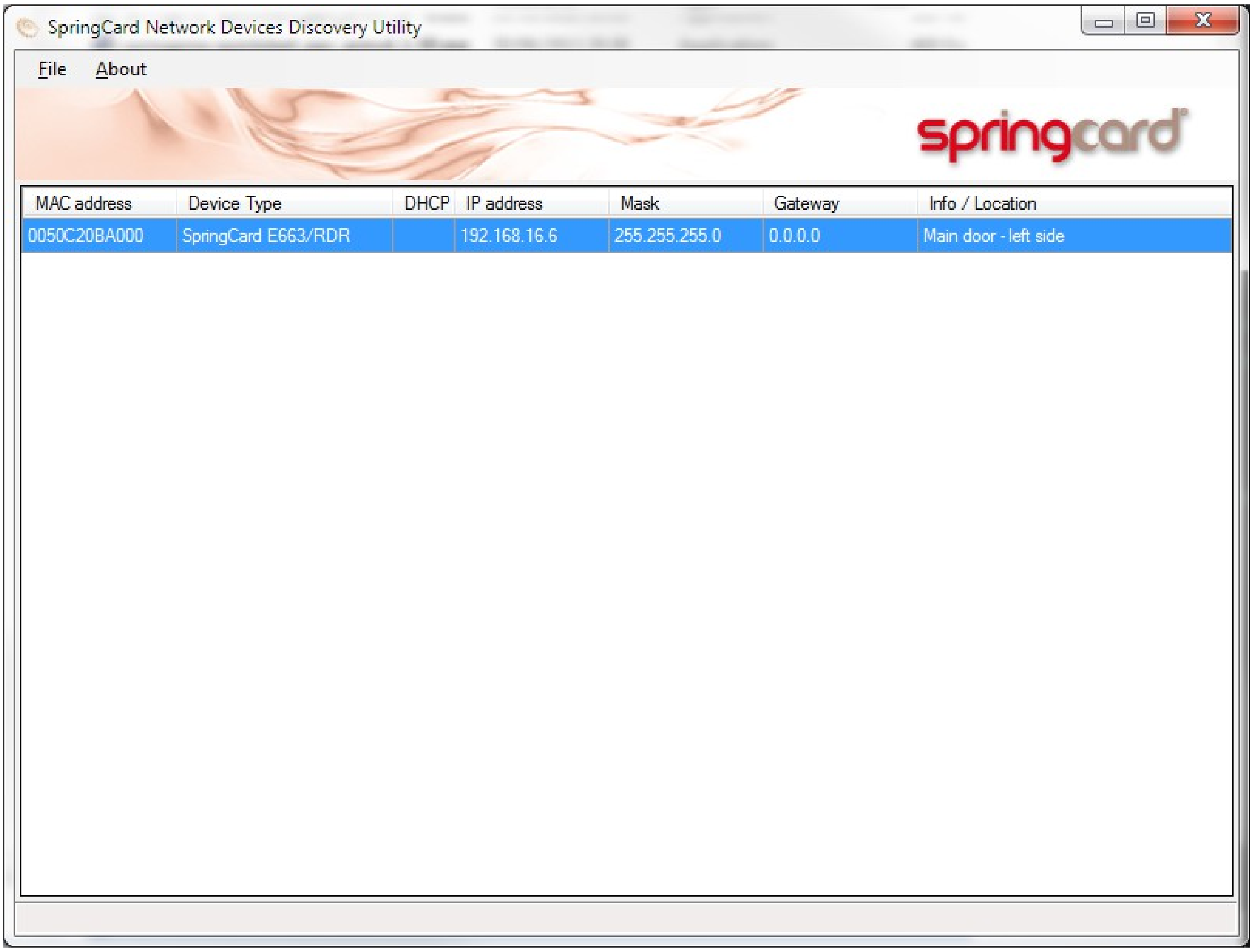
If everything is OK, including the current password, the NDDU software is able to configure the device. The following message confirms that the new configuration has been accepted:



Configuration OK

The target device is now rebooting to take the new configuration.

After a few seconds, the list of devices is refreshed and shows the new configuration:



MAC address	Device Type	DHCP	IP address	Mask	Gateway	Info / Location
0050C20BA000	SpringCard E663/RDR		192.168.16.6	255.255.255.0	0.0.0.0	Main door - left side

3.2. Assign an IP address using a Master Card

The Master Cards are NXP Desfire cards formatted and programmed by **SpringCard Configuration Tool (ScMultiConf.exe, ref # SN14007)** for Windows.

Please refer to this software's documentation for details.

4. Telnet access to the Reader

4.1. Reader's Console

The Reader features a “human” command processor (shell or console). This feature accessible through the Telnet protocol. It is primarily made for testing and demonstration purpose. Only the few commands depicted in this chapter could safely be used for configuration and diagnostic.

> *Note that the SEC Configuration Register (h6E, § 9.4) may be used to disable the Console.*

4.1.1. Open a Telnet session to the reader

On most operating systems you could find a Telnet client in the default system tools. Open a console and enter

```
telnet xxx.xxx.xxx.xxx
```

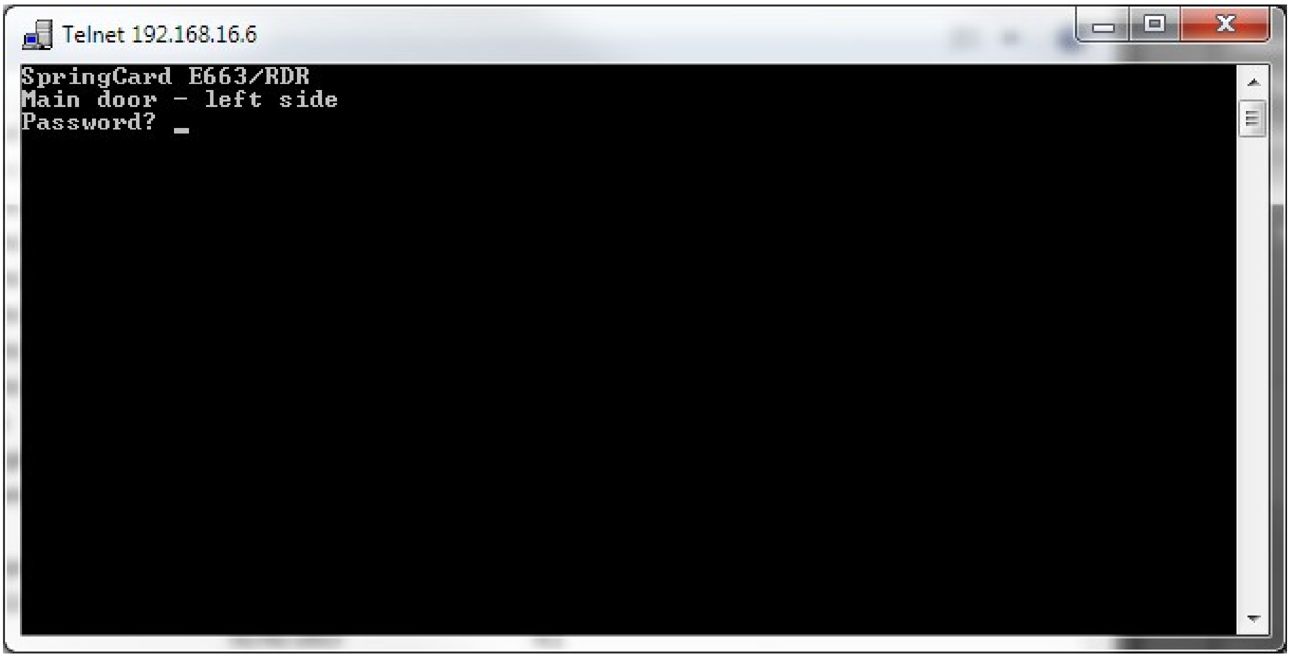
where *xxx.xxx.xxx.xxx* is the Reader's IP Address as defined in chapter 3.

> *Windows Vista / 7 / 8 : the Telnet client may be missing from you OS default install. Go to **Control Panel**, **Programs and Features** section, and then enable **Telnet client** in the **Turn Windows features on or off** tab.*
> *Alternatively, you may download a free terminal client such as **Putty**, that is also a Telnet client.*

The Reader's Telnet shell says “SpringCard E663/RDR”, then the Info / Location string that has been entered in chapter 3, and finally prompts for a password.

Enter the Reader's password that you've defined in chapter 3.

> *If you haven't changed the password, the default password is **springcard**.*



```
SpringCard E663/RDR
Main door - left side
Password? _
```

4.1.2. Sending a command to the Reader

Write the command line as documented below, and terminate by hitting the ENTER key.

Note that the Reader echoes the entered characters.

4.1.3. List of Console commands

Command	Meaning
version	Show the firmware version
info	Show the firmware information data
show	Show the current configuration
cfg	Dump all Configuration Registers written into persistent memory
cfgXX=YY...YY	Write value $_h$YY...YY to Configuration Register $_h$XX
cfgXX=!!	Erase Configuration Register $_h$XX
cfgXX	Read Configuration Register $_h$XX
exit	Terminate the Telnet session

5. TCP Client/Server protocol – Low layers, Plain mode

5.1. Abstract

The communication protocol is a Client / Server protocol, the Host being the Client, and the Reader being the Server:

- The Reader listen on a TCP port,
- The Host is responsible to connect on this port, and to restore the connection every-time it went down.

Note that the Reader is not able to accept more than one Client at the time. Trying to connect to the same Reader from two different Host is not supported, and shall not be tried. An undefined behaviour may occur.

The communication scheme is based on the transmission of variable-length blocks. The I-Block convey application-level frames that are defined in chapter 7.

5.2. Presentation layer

5.2.1. Block format

Every block transmitted in the channel is formatted as follow:

LENGTH	TYPE	PAYLOAD
1 byte	1 byte	Variable length

5.2.2. Description of the fields

Field	Description
LENGTH	The LENGTH byte is the total length of the block, this byte included.
TYPE	The TYPE byte is used to convey the information required to control the data transmission. There are three fundamental types of blocks: • I-block used to convey information for use by the upper layers • H-block used to exchange control information between the Server and the Client
PAYLOAD	The PAYLOAD field is optional. When present, the PAYLOAD field conveys application data.

5.2.3. Size of the blocks

The size of every block must be less or equal to 66 bytes.

This lead to a PAYLOAD between 0 and 64 bytes.

If the application layer needs to transmit more than 64 bytes, chaining shall be used.

5.2.4. Format of the TYPE byte

a. I-Block

Bit	Description
7 (msb)	Direction • 0 for Host → Reader • 1 for Reader → Host
6	Shall be set to 0
5	Shall be set to 0
4	Chaining • 0: no chaining – this block is the only one, or the last one in a sequence • 1: chaining enabled – more block(s) to come
3	Shall be set to 0000
2	
1	
0 (lsb)	

b. H-Block

Bit	Description
7 (msb)	Direction • 0 for Host → Reader • 1 for Reader → Host
6	Shall be set to 1
5	Block type: • $_b$00: HELO (Reader's “hello” block) • $_b$01: HELO-OK (Host's “hello” acknowledge) • $_b$10: RFU, do not use • $_b$11: HELO-AUTH (see § 6.3)
4	
3	Protocol Version for HELO block Key Number for the first HELO-AUTH block 0000 for the other blocks.
2	
1	
0 (lsb)	

c. Protocol Version

The Reader sets this field to $_b$0000. Any other value shall be interpreted by the Host as an error.

5.3. General communication flow

5.3.1. Session establishment

The Host tries to connect to one (or many) Reader.

When a connection is established on the Reader, the Reader sends a HELO block. The payload of the HELO block is the Reader's MAC address on 6 bytes.

HELO block (Reader → Host)

LENGTH	TYPE	PAYLOAD
$_h$08	$_h$C0	Reader's MAC address on 6 bytes

The Host may check that the claimed MAC address is coherent with its records.

The Host may check that the Reader's Protocol Version is acceptable.

If everything is OK, the Host sends a HELO-OK block. The payload of the HELO-OK block is empty.

HELO-OK block (Host → Reader)

LENGTH	TYPE	PAYLOAD
$_h$02	$_h$50	empty

5.3.2. Nominal dialogue

The TCP channel is full-duplex; both the Reader and the Host may send at any time, and therefore must be ready to receive at any time.

The Host sends I-Blocks to transmit its commands or to query the Reader. An empty I-Block denotes a Keep Alive request.

The Reader sends I-Block to transmit its notifications or its answers. An empty I-Block denotes a Keep Alive response (when no other data is available).

5.3.3. Timings

The Reader ensures that it answer to every block coming from the Host by a response block within 2.5s. The Host may use a 3s-timeout to watch-out the Reader. This is also applicable to the HELO frame that is sent by the Reader immediately when the connection is opened.

The Reader expects to receive a block from the Host at least every 60s.

5.3.4. Chaining

If the application data buffer is longer than the max size for the PAYLOAD field, the data shall be divided onto multiple I-Blocks. In this case, the Chaining bit is set to 1 for every I-Block but the last one.

Chaining is not implemented in the current version of the Reader's firmware. The Host shall not use this feature (and the Reader will not use it).

5.4. Error handling and recovery

5.4.1. For the Reader

- **Bad sequence during session establishment:** is the Reader receives a frame before having transmitted its HELO, the Reader drops the connection,
- **Protocol error:** if the Reader receives an invalid block from the Host (LENGTH not coherent with actual length, or unallowed value for TYPE), the Reader drops the connection,
- **No more activity error:** if the Host remains silent for 60s, the Reader drops the connection.

5.4.2. For the Host

- **Bad sequence during session establishment:** is the first frame received by Host is not a valid HELO, or the Host receives another frame before having transmitted its HELO-OK, the Host shall drop the connection,
- **Protocol error:** if the Host receives an invalid block from a Reader (LENGTH not coherent with actual length, or unallowed value for TYPE), the Host shall drop the connection,
- **Timeout error:** if the Reader doesn't answer within 3s, the Host shall drop the connection.

5.4.3. Recovery

If the connection is dropped for any reason, the Host shall wait at least 5s before trying to connect again to the same Reader.

5.5. Application layer

Chapter 7 contains the application layer protocol. The application layer frames are conveyed within I-Blocks.

6. TCP/Client Server Protocol – Low layers, Authenticated mode

6.1. Abstract

As in chapter 5, the communication protocol is a Client / Server protocol, the Host being the Client, and the Reader being the Server.

In Authenticated mode,

- The Reader and the Host perform a 3-pass mutual authentication that proofs they share the very same authentication key (one of the 2 Reader's secret key), and in the same time allow to establish a one-time, random session key – that remains also a secret shared by both partners,
- The blocks conveyed between the two partners are ciphered and authenticated, i.e. their content remains undisclosed, and a defrauder could not insert its own packets in the sequence without being noticed.

The Reader has 2 secret keys. Both keys are defined in the IPS Configuration Register ($_h$83, § 9.5.3).

a. Administration Key

When authenticated using the Administration Key, the Host gains full access to the Reader's command set, including the ability to edit the configuration.

b. Operation Key

When authenticated using the Administration Key, the Host has no access to the Reader's command set.

6.2. Cryptographic background

6.2.1. Cipher

The Reader uses the **AES block cipher** (Rijndael) .

AES has a fixed 128-bit (16 bytes) block size. CBC mode is used to cipher a content that is longer than the block size. The initialization vector of the CBC is reset after every frame.

The Reader supports 128-bit keys (16 bytes) only.

In the following paragraphs,

- E (*K* , *p*) means “AES encrypt operation (cipher) over *p* using the key *K*”,

- D (*K* , *c*) means "AES decrypt operation (decipher) over *c* using the key *K*".

Both operations are valid only on a number of bytes that is multiple of 16. Before encryption, a padding is applied to ensure this length (see below). When more than one block are involved, the encrypt and decrypt operations are performed **in CBC (cipher block chaining) mode**.

6.2.2. CMAC

6.2.3. Padding

6.3. 3-PASS AUTHENTICATION

The 3-pass authentication is initiated by the Host after receiving the HELO frame from the Reader (§ 5.3.1)

6.3.1. Reader's HELO

HELO block (Reader → Host)

LENGTH	TYPE	PAYLOAD
$_h$08	$_h$C0	Reader's MAC address on 6 bytes

The HELO block contains the Reader's MAC address. This makes it possible for the Host

1. To check this Reader is the expected once (table IP address ←→ MAC address)
2. To select this Reader's secret key.

6.3.2. Host's HELO-Auth

The Host asks the Reader to open a secure session by sending an HELO-Auth block. The payload of the HELO-Auth block is empty. The low-order bit of the TYPE byte selects the key

HELO-Auth block (Host → Reader) using Operation Key

LENGTH	TYPE	PAYLOAD
$_h$02	$_h$71	empty

HELO-Auth block (Host → Reader) using Administration Key

LENGTH	TYPE	PAYLOAD

$_h$02	$_h$72	empty

6.3.3. Authentication, step 1

After receiving the HELO-Auth block from the Host,

- The Reader activates the selected **secret key K_{AUTH},**
- The Reader generate a random challenge (C_R) on 16 bytes,
- The Reader sends to the Host a block containing E (K_{Auth}, C_R).

Authentication, step 1: block Reader → Host

LENGTH	TYPE	PAYLOAD
$_h$12	$_h$F0	E (K_{AUTH}, C_R) on 16 bytes

> *The Init Vector of the AES cipher is cleared to (00..00) before computing E (K_{AUTH}, C_R).*
> *1 block is crypted, no padding is applied.*

6.3.4. Authentication, step 2

- The Host activates the secret key K_{AUTH},
- The Host decrypts the payload received from the Reader, and retrieves C_R,
- The Host computes C_R' = C_R << 1 || C_R >> 127 (shift left with carry),
- The Host generate a random challenge (C_H) on 16 bytes,
- The Host sends to the Reader a block containing E (K_{Auth}, C_H || C_R'),

Authentication, step 2: block Host → Reader

LENGTH	TYPE	PAYLOAD
$_h$22	$_h$70	E (K_{AUTH}, C_H || C_R') on 32 bytes

> *The Init Vector of the AES cipher is cleared to (00..00) before computing E (K_{AUTH}, C_H || C_R').*
> *2 block are crypted in CBC mode, no padding is applied.*

6.3.5. Authentication, step 3

- The Reader decrypts the payload received from the Host, and retrieves C_H and C_R',
- The Reader checks that C_R' is valid. This is the proof that the Host knows the secret key,
- The Reader computes $C_H' = C_H << 1 \,||\, C_H >> 127$ (shift left with carry),
- The Reader sends to the Host a block containing E (K_S, C_H'),

Authentication, step 3: block Reader → Host

LENGTH	TYPE	PAYLOAD
$_h$12	$_h$F0	E (K_{AUTH}, C_H') on 16 bytes

> *The Init Vector of the AES cipher is cleared to (00..00) before computing E (K_{AUTH}, C_H').*
> *1 block is crypted, no padding is applied.*

6.4. Generation of session keys

Two session keys are derived from the two random challenges:

- **K_{SESS} is the session cipher key**, used to ensure confidentiality over the TCP channel,
- **K_{CMAC} is the session CMAC key**, used to ensure confidentiality over the TCP channel,

6.4.1. Session cipher key

6.4.2. Session CMAC key

6.5. HELO-OK and synchronisation

6.5.1. Synchronisation of the Sequences

From now on, all operations are authenticated by a CMAC. The calculation of the CMAC is includes a Sequence number, to protect against the injection or the removal of frames. Both the Reader and the Host shall maintain 2 Sequence numbers:

- **SEQ_H is used by the Host to compute its outgoing CMAC**, and by the Reader to verify the its incoming CMAC. SEQ_H is incremented every time the Hosts sends a frame.
- **SEQ_R is used by the Reader to compute its outgoing CMAC**, and by the Host to verify the its incoming CMAC. SEQ_R is incremented every time the Hosts sends a frame.

Both SEQ_H and SEQ_R are cleared at the end of the authentication.

6.5.2. Host's HELO-OK

- The Host deciphers the payload received from the Reader, and retrieves C_H',
- The Host checks that C_H' is valid. This is the proof that the Reader knows the secret key,
- The Host generates a random nounce (N_H) on 16 bytes
- The Host sends to the Reader a HELO-OK block containing E (K_{SESS}, N_H)

HELO-OK block (Host → Reader)

LENGTH	TYPE	PAYLOAD
$_h$22	$_h$50	E (K_{SESS}, N_H || CMAC || PADD) on 32 bytes

> *The CMAC is computed as specified in 6.2.2 . The sequence number of the CMAC is cleared to 0 before computing the CMAC. The initial Length (before CMAC and Padding) is $_h$12 (size of header + size of N_H).*
> *After CMAC, the Length is $_h$1A (size of CMAC is 8 bytes).*
> *The Padding is applied as specified in 6.2.3 to reach 32 bytes of payload, i.e. a final Length of $_h$22.*
> *The Init Vector of the AES cipher is cleared to (00..00) before computing E (K_{SESS}, ...). 2 blocks are crypted in CBC mode.*

6.5.3. Synchronisation of the Init Vectors using HELO-OK

From now on, all operation are performed in CBC mode. The Init Vector is preserved among all operations on both sides. Both the Reader and the Host shall maintain 2 Init Vectors for the CBC:

- **IV_H is used by the Host to send** (encrypt), and by the Reader to receive (decrypt),
- **IV_R is used by the Reader to send** (encrypt), and by the Reader to receive (decrypt).

When receiving the HELO-OK block, the Reader decrypts the received cryptogram after starting from a clear IV (00..00). As a consequence, the Reader's IV_H becomes synchronised with the Host's IV_H . At this step, both **the Reader and the Host copy IV_H into IV_R.**

Afterwards, IV_H into IV_R will evolve independently.

6.6. New authentication – Generation of a new session key

The Host may require a new authentication at any time, by sending a new HELO-Auth block as specified in § 6.3.2 .

6.7. Presentation layer after authentication

6.7.1. Block format

Every block transmitted in the channel is formatted as follow:

LENGTH	TYPE	CONTENT CRYPTED USING K_{SESS} (CBC MODE, IV KEPT)		
		PAYLOAD	**CMAC**	**PADDING**
1 byte	1 byte	Variable length	8 bytes	Variable length

6.7.2. Description of the fields

Field	Description
LENGTH	The LENGTH byte is the total length of the block, this byte included.
TYPE	The TYPE byte is used to convey the information required to control the data transmission. After authentication, only I_S-Blocks could be transmitted
PAYLOAD	The PAYLOAD field is optional. When present, the PAYLOAD field conveys application data.
CMAC	The CMAC field is computed over the initial LENGTH (before CMAC and Padding), the TYPE, the SEQUENCE and the PAYLOAD fields, as specified in 6.2.2 .
PADDING	The cipher algorithm uses fixed-size blocks. Therefore a PADDING shall be applied to ensure that the size of content to be ciphered is a multiple of the cipher's block size. The PADDING is specified in 6.2.3 .

6.7.3. Size of the blocks

If the application layer needs to transmit more than 64 bytes, chaining shall be used.

With a PAYLOAD between 0 and 64 bytes, the size of every block is between 18 and 82.

6.7.4. Format of the TYPE byte

a. I_S-Block

Bit	Description
7 (msb)	Direction • 0 for Host → Reader • 1 for Reader → Host
6	Shall be set to 0
5	Shall be set to 1
4	Chaining • 0: no chaining – this block is the only one, or the last one in a sequence • 1: chaining enabled – more block(s) to come
3	Shall be set to 0000
2	
1	
0 (lsb)	

6.8. General communication flow

6.8.1. Nominal dialogue

The TCP channel is full-duplex; both the Reader and the Host may send at any time, and therefore must be ready to receive at any time.

The Host sends I_S-Blocks to transmit its commands or to query the Reader. An empty I_S-Block denotes a Keep Alive request.

The Reader sends I-Block to transmit its notifications or its answers. An empty I_S-Block denotes a Keep Alive response (when no other data is available).

6.8.2. Timings

The Reader ensures that it answer to every block coming from the Host by a response block within 2.5s. The Host may use a 3s-timeout to watch-out the Reader. This is also applicable to the HELO frame that is sent by the Reader immediately when the connection is opened.

The Reader expects to receive a block from the Host at least every 60s.

6.8.3. Chaining

If the application data buffer is longer than the max size for the PAYLOAD field, the data shall be divided onto multiple I_S-Blocks.

In this case,

- The Chaining bit is set to 1 for every I_S-Block but the last one,
- Only the first I_S-Block contains the SEQUENCE field,
- Only the last I_S-Block contains the CRC32 and PADDING fields.

6.9. Error handling and recovery

6.9.1. For the Reader

- **Bad sequence during session establishment:** is the Reader receives a frame before having transmitted its HELO, the Reader drops the connection,
- **Protocol error:** if the Reader receives an invalid block from the Host (LENGTH not coherent with actual length, or unallowed value for TYPE), the Reader drops the connection,
- **No more activity error:** if the Host remains silent for 60s, the Reader drops the connection.

6.9.2. For the Host

- **Bad sequence during session establishment:** is the first frame received by Host is not a valid HELO, or the Host receives another frame before having transmitted its HELO-OK, the Host shall drop the connection,
- **Protocol error:** if the Host receives an invalid block from a Reader (LENGTH not coherent with actual length, or unallowed value for TYPE), the Host shall drop the connection,
- **Timeout error:** if the Reader doesn't answer within 3s, the Host shall drop the connection.

6.9.3. Recovery

If the connection is dropped for any reason, the Host shall wait at least 5s before trying to connect again to the same Reader.

6.10. Application layer

Chapter 7 contains the application layer protocol. The application layer frames are conveyed within I-Blocks.

7. Application layer protocol

7.1. Principles

The application-level communication uses the T,L,V scheme:

- **T (Tag):** this is the operation-code of a command, or the identifier of a data field. The Tag is on either 1 or 2 bytes,
- **L (Length)**: this is the length of the following Value, on 1 byte. Allowed values are $_{h}$00 to $_{h}$7F,
- **V (Value):** the parameters to the command, or the data field itself. The length is specified by L, from 0 to 127 bytes.

7.2. Host → Reader, available with both Administration and Operation keys

7.2.1. Get Global Status

T	L
$_{h}$00	$_{h}$00

The Reader answers by 2 frames:

1. Reader Identifier
2. Tamper Status

7.2.2. Start/Stop Reader

T	L	V
$_{h}$0A	$_{h}$01	**mode**

- **mode:** start/stop command
  - $_{h}$00 Reader goes OFF (RF field OFF, no activity on RF)
  - $_{h}$01 Reader goes ON

7.2.3. Clear LEDs command

Both LEDs go OFF.

T	L
hD000	h00

7.2.4. Set LEDs command

Both LEDs are driven – until a Clear LEDs command is received.

T	L	V	
hD000	h02	red	green

- **red:** command for red LED
  - h00 OFF
  - h01 ON
  - h02 blinks slowly
  - h03 blinks quickly
- **green:** command for green LED
  - h00 OFF
  - h01 ON
  - h02 blinks slowly
  - h03 blinks quickly

7.2.5. Start LED sequence command

Both LEDs are driven – until a Clear LEDs command is received or a timeout occurs.

T	L	V		
hD000	h04	red	green	time (sec)

- **red:** same as above,
- **green:** same as above,
- **time:** time (in seconds, MSB-first) before returning to all-LED-OFF state.

7.2.6. Buzzer command

T	L	V
${}_h$D100	${}_h$01	seq.

- **seq:**
  - ${}_h$00 buzzer OFF,
  - ${}_h$01 buzzer ON,
  - ${}_h$02 buzzer short sequence,
  - ${}_h$03 buzzer long sequence.

7.3. Host → Reader, available with Administration key only

7.3.1. Write Configuration Register

The Reader's behaviour is defined by Configuration Registers documented in chapters 9 and 10. The Write Configuration Register command allows to write into any Configuration Register given its address.

<addr> is the Register number on one byte (valid values are ${}_h$00 to ${}_h$FE).

T	L	V	
${}_h$0C	<var.>	<addr>	<value>

7.3.2. Erase Configuration Register

The Reader's behaviour is defined by Configuration Registers documented in chapters 9 and 10. The Erase Configuration Register command allows to delete any Configuration Register given its address. Once a Register is deleted, the default value for this Register is used.

<addr> is the Register number on one byte (valid values are ${}_h$00 to ${}_h$FE).

T	L	V
${}_h$0C	${}_h$01	<addr>

7.3.3. Reset the Reader

The Reader must be re-setted in order for the new configuration to take effect. When receiving this command, the Reader drops the connection and resets.

T	**L**
$_h$0C	$_h$00

7.4. Reader → Host

7.4.1. Reader Identifier

This T,L,V is transmitted in response to the **Get Global Status** command.

T	**L**	**V**
$_h$8100	$_h$1C	`SpringCard E663/RDR x.xx`

7.4.2. Tamper Status

This T,L,V is transmitted in response to the **Get Global Status** command or when one of the tampers is broken/restored.

T	**L**	**V**
$_h$2F	$_h$01	Bit field, the broken tampers are denoted by the corresponding bit set to 1. V = $_h$00 when all tampers are OK.

7.4.3. Card Read

This T,L,V is transmitted when the Reader has read a card, if the Insert/Remove mode is disabled.

T	**L**	**V**
$_h$B000	<var.>	Card Identifier

7.4.4. Card Inserted

This T,L,V is transmitted when the Reader has read a card, if the Insert/Remove mode is enabled.

T	L	V
$_h$B100	<var.>	Card Identifier

7.4.5. Card Removed

This T,L,V is transmitted when the card is removed, if the Insert/Remove mode is enabled.

T	L
$_h$B100	$_h$00

8. Editing Reader's configuration

The Reader's configuration is stored in a set of non-volatile Configuration Registers. There are two groups of Registers:

- The Registers that control the behaviour of the Reader are fully documented in chapter 9. Some of them are common to various SpringCard Readers, but some of them are very specific to the **SpringCard FunkyGate-IP NFC**.
- The Registers that control the Template System are shared among all SpringCard Readers. Chapter 10 is therefore a place-holder that redirects to the document describing this Template System precisely.

But this subtle distinction between these two groups is only there to keep the documents short, and to ease switching from one Reader to the other. Technically speaking, all Registers are defined (and accessed) the same way.

There are four ways to edit the Reader's Configuration Registers:

1. Through the Telnet link
2. Using Master Cards
3. Using NFC peer-to-peer
4. Through the TCP Client/Server interface, after authentication with Administration Key.

> *Note that the SEC Configuration Register (^{h}6E, § 9.4) may be used to disable either way to access the Configuration Registers.*
> *Administration Key is defined in the IPS Configuration Register ($_h$83, § 9.5.3)*

8.1. Through the Telnet link

Open a Telnet session to the Reader as instructed in § 4.1.

8.1.1. Reading Configuration Registers

Enter “cfg” to list all Configuration registers currently defined (registers that are not explicitely defined keep their default value).

Enter “cfgXX” to read the value of the Configuration register $_h$XX.

Note that Configuration registers $_h$55, $_h$56, $_h$6E and $_h$6F that hold sensitive data (the keys used by Master Cards and the Reader's secret keys and password) are masked.

8.1.2. Writing Configuration Registers

Enter “cfgXX=YYYY” to update Configuration Register $_h$XX with value $_h$YYYY. YYYY can me any length between 1 and 32 bytes.

Enter “cfgXX=!!” to erase Configuration Register $_h$XX.

8.2. Using Master Cards

The Master Cards are NXP Desfire cards formatted and programmed by **SpringCard Configuration Tool (ScMultiConf.exe, ref # SN14007)** for Windows.

Please refer to this software's documentation for details.

8.3. Through the TCP Client/Server interface

Please refer to § 7.3.

9. Global configuration of the Reader

9.1. General options

Name	Tag	Description	Size
OPT	$_h$60	General option, see table below	1 or 2

General options bits

Bits	Value	Meaning
		Byte 0
7	0 1	Normal mode Power saving mode (the Reader is slower)
6	0 1	Track the cards by their ID only Keep the RF field active to track the cards (works with Random IDs)
5 - 4	 00 01 10 11	***Anti-collision mode*** Read every card one after the other *RFU* Read only one card at a time (ignore the other ones) Prevent reading when there's more than one card in the field
3 - 2	 00 01 10 11	***Master Card and NFC configuration*** Disable configuration by Master Card or NFC Allow configuration by Master Card or NFC at power up only *RFU* Allow configuration by Master Card or NFC all the time
1	0	*RFU (set to 0)*
0	0	*RFU (set to 0)*
		Byte 1 (optional)
7	0	*RFU (set to 0)*
6	0	*RFU (set to 0)*
5	0	*RFU (set to 0)*
4	0 1	Insert/Remove mode is disabled (§ 7.4.3) Insert/Remove mode is enabled (§ 7.4.4 and § 7.4.5)
3	0	*RFU (set to 0)*
2	0	*RFU (set to 0)*
1	0	*RFU (set to 0)*
0	0 1	Reader is active on startup Reader is not active on startup (Host must send an activation command)

Default value: $_b$00001100 00000000

9.2. Delays and repeat

Name	Tag	Description	Min	Max
ODL	$_h$61	Min. delay between 2 consecutive outputs (0.1s)	0	100
RDL	$_h$62	Min. delay between 2 consecutive identical outputs (0.1s) A value of 255 means that the card must be removed from the field –and re-inserted into– before being read again	0	100

Default value: ODL = 5 (1ms) RDL = 20 (2s)

9.3. LEDs and Buzzer

Name	Tag	Description	Size
CLD	$_h$63	LEDs control, see table below	1
CBZ	$_h$64	Buzzer control, see table below	1

LEDs control bits

Bits	Value	Meaning
7	0	Short LED sequences (3 seconds)
	1	Long LED sequences (10 seconds)
6 - 5	00	When idle, blue LED blinks slowly (“heart beat” sequence)
	01	When idle, blue LED is always on
	10	When idle, blue LED is always off
	11	RFU
4	0	Green LED stays OFF
	1	Green LED blinks when a valid card has been processed
3	0	Red LED stays OFF
	1	Red LED blinks when an unsupported card has been processed
2	0	Green LED stays OFF
	1	Green LED blinks as soon as a card is seen in the field
1 - 0	00	*RFU, do not use*
	01	LED driven by Host commands only
	10	*RFU, do not use*
	11	LED driven by internal state machine and Host commands

Default value: $_b$00001111

Buzzer control bits

Bits	Value	Meaning
7	0 1	Buzzer short pulse = 0,2 sec Buzzer short pulse = 0,5 sec
6	0 1	Buzzer long pulse = 0,7 sec Buzzer long pulse = 1,5 sec
5		*RFU*
4	0 1	No action on buzzer before specified by host controller Short pulse when a valid card has been processed
3	0 1	No action on buzzer for unsupported cards Long pulse when an unsupported card has been processed
2	0 1	No action on buzzer before processing is achieved Short pulse as soon as a card is seen in the field
1 - 0	00 01 10 11	Buzzer is disabled, other settings are ignored Buzzer controlled by serial commands, other settings are ignored Buzzer controlled by internal software, serial commands are ignored Buzzer controlled by both internal software and serial commands

Default value : $_b$00010010

9.4. Security options

Name	Tag	Description	Size
SEC	$_h$6E	Security option bits. See table a below	1

Security option bits

Bits	Value	Meaning
7	0 1	Telnet access is disabled Telnet access is enabled
6	0	*RFU (set to 0)*
5	0	*RFU (set to 0)*
4	0	*RFU (set to 0)*
3	0	*RFU (set to 0)*
		Tampers
2	0 1	Do not signal tamper alarms on buzzer Signal tamper alarms on buzzer
1	0 1	Reader keeps on reading even if a tamper is broken Reader stops reading when a tamper is broken
0	0 1	Do not raise alarm if a tamper is broken at power up Raise alarm on tamper broken even at power up

Default value: $_b$10000100

9.5. TCP configuration

9.5.1. IPv4 address, mask, and gateway

Name	Tag	Description	Size
IPA	$_{h}$80	IPv4 configuration bytes, see table below	4 or 8

IPv4 configuration bytes

Bytes	Contains	Remark
0	Address, 1st byte	Reader's IPv4 Address. If these bytes are missing, the default IP Address $_{h}$C0 A8 00 FA (192.168.0.250) is taken.
1	Address, 2nd byte	
2	Address, 3rd byte	
3	Address, 4th byte	
4	Mask, 1st byte	Network Mask. If these bytes are missing, the default Mask $_{h}$FF FF FF FF (255.255.255.0) is taken.
5	Mask, 2nd byte	
6	Mask, 3rd byte	
7	Mask, 4th byte	

Default value: $_{h}$C0 A8 00 FA FF FF FF 00

(address = 192.168.0.250, mask = 255.255.255.0)

9.5.2. Server port

Name	Tag	Description	Size
IPP	$_{h}$81	Listen TCP port for the server (2 bytes, MSB-first)	2

Default value: $_{h}$0F 9F (server TCP port = 3999)

9.5.3. Server security settings and keys

Name	Tag	Description	Size
IPS	$_h$82	Server security settings and keys bytes, see table below	1, 17 or 33

Server security settings and keys bytes

Bytes	Contains	Remark
0	Security settings bits	See table below
1-16	Operation Key	If these bytes are missing, the default Key is $_h$00 … 00
17-32	Administration Key	If these bytes are missing, the default Key is $_h$00 … 00

Security settings bits

Bits	Value	Meaning
7	0	*RFU (set to 0)*
6	0	*RFU (set to 0)*
5	0	*RFU (set to 0)*
4	0	*RFU (set to 0)*
3	0	*RFU (set to 0)*
2	0 1	The Administration Key is disabled The Administration Key is enabled
1	0 1	The Operation Key is disabled The Operation Key is enabled
0	0 1	Plain communication is allowed Secure communication is mandatory

Default value: $_b$00000110

(both keys are enabled, plain communication is allowed)

9.5.4. Info / Location

Name	Tag	Description	Size
ILI	$_h$8E	Info / Location string	Var. 0-30

Default value: empty

The **Info / Location** string is a text value (ASCII) that appears

- When someone tries to connect on Telnet,
- In the NDDU software (§ 3.1.3).

Use this string as a reminder of where your Reader is installed, or what is its role in your access-control system.

9.5.5. Password for Telnet access

Name	Tag	Description	Size
ITP	$_h$8F	Password for Telnet access string	Var. 0-16

Default value: “springcard”

The **Password for Telnet access** string is a text value (ASCII) that protects the access to the Reader using Telnet protocol.

The password is mandatory. If this registry is not set, default value “springcard” is used.

10. The Template System

SpringCard FunkyGate-IP NFC provides 4 “Card Processing Templates” that defines how the Reader which fetch data from various cards/tags, and how the Card Identifier will be constructed from these data before being sent to the Host.

The template system is fully described in document # **PMA13205 “RFID/NFC Scanners Template System**”.

Please use this document as reference to configure the “Reader part” of your **SpringCard FunkyGate-IP NFC.**

11. 3RD-PARTY LICENSES

SpringCard FunkyGate-IP NFC has been developed using open-source software components.

11.1. FreeRTOS

FreeRTOS is a market leading real time operating system (or RTOS) from Real Time Engineers Ltd. **SpringCard FunkyGate-IP NFC** runs on FreeRTOS v7.5.2.

FreeRTOS is distributed under a modified GNU General Public License (GPL) that allows to use it in commercial, closed-source products.

For more information, or to download the source code of FreeRTOS, please visit

www.freertos.org

11.2. uIP

µIP is an open-source TCP/IP stack initially developed by Adam Dunkels and licensed under a BSD style license.

SpringCard FunkyGate-IP NFC uses FreeTCPIP, a modified version of µIP that comes with FreeRTOS. To comply with the original license of µIP, we have to copy the full text here:

Disclaimer

This document is provided for informational purposes only and shall not be construed as a commercial offer, a license, an advisory, fiduciary or professional relationship between PRO ACTIVE and you. No information provided in this document shall be considered a substitute for your independent investigation.

The information provided in document may be related to products or services that are not available in your country.

This document is provided "as is" and without warranty of any kind to the extent allowed by the applicable law. While PRO ACTIVE will use reasonable efforts to provide reliable information, we don't warrant that this document is free of inaccuracies, errors and/or omissions, or that its content is appropriate for your particular use or up to date. PRO ACTIVE reserves the right to change the information at any time without notice.

PRO ACTIVE doesn't warrant any results derived from the use of the products described in this document. PRO ACTIVE will not be liable for any indirect, consequential or incidental damages, including but not limited to lost profits or revenues, business interruption, loss of data arising out of or in connection with the use, inability to use or reliance on any product (either hardware or software) described in this document.

These products are not designed for use in life support appliances, devices, or systems where malfunction of these product may result in personal injury. PRO ACTIVE customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify PRO ACTIVE for any damages resulting from such improper use or sale.

Copyright notice

All information in this document is either public information or is the intellectual property of PRO ACTIVE and/or its suppliers or partners.

You are free to view and print this document for your own use only. Those rights granted to you constitute a license and not a transfer of title : you may not remove this copyright notice nor the proprietary notices contained in this documents, and you are not allowed to publish or reproduce this document, either on the web or by any mean, without written permission of PRO ACTIVE.

Editor’s information

PRO ACTIVE SAS company with a capital of 227 000 €

RCS EVRY B 429 665 482

Parc Gutenberg, 2 voie La Cardon

91120 Palaiseau – FRANCE

Contact information

For more information and to locate our sales office or distributor in your country or area, please visit

www.springcard.com