



PMA13257-AA
DRAFT - PUBLIC

SPRINGCARD FUNKYGATE-IP

Integration and Configuration Guide

DOCUMENT IDENTIFICATION

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

File name	C:\Users\johann\Desktop\En cours\[PMA13257-AA] FunkyGate-IP NFC Integration and Configuration Guide.odt		
Date saved	18/11/13	Date printed	

REVISION HISTORY

Ver.	Date	Author	Valid. by		Approv. by	Details
			Tech.	Qual.		
AA	29/09/13	JDA				Created from PMA959P

CONTENTS

1. INTRODUCTION.....	6	6.3.3-PASS AUTHENTICATION.....	24
1.1. ABSTRACT.....	6	6.3.1.Reader's HELO.....	24
1.2. SUPPORTED PRODUCT.....	6	6.3.2.Host's HELO-Auth.....	24
1.3. AUDIENCE.....	7	6.3.3.Authentication, step 1.....	25
1.4. SUPPORT AND UPDATES.....	7	6.3.4.Authentication, step 2.....	25
1.5. RELATED DOCUMENTS.....	7	6.3.5.Authentication, step 3.....	26
2. HARDWARE INSTALLATION.....	8	6.3.6.Host's HELO-OK.....	26
2.1. SPRINGCARD FUNKYGATE-IP NFC.....	8	6.4. SESSION KEY.....	26
2.2. SPRINGCARD FUNKYGATE-IP+POE NFC.....	8	6.5. NEW AUTHENTICATION – GENERATION OF A NEW SESSION KEY.....	26
3. DEFINE READER'S IP ADDRESS.....	9	6.6. PRESENTATION LAYER AFTER AUTHENTICATION.....	27
3.1. ASSIGN AN IP ADDRESS USING NDDU SOFTWARE.....	9	6.6.1. Block format.....	27
3.1.1. Download and install the NDDU software.....	9	6.6.2. Description of the fields.....	27
3.1.2. Run the NDDU software.....	9	6.6.3. Size of the blocks.....	27
3.1.3. Discovered devices.....	10	6.6.4. Format of the TYPE byte.....	28
3.1.4. Configure a Reader.....	11	6.7. SEQUENCE NUMBERS.....	28
3.1.5. Verify the new configuration.....	12	6.8. CHECKSUM, PADDING, CIPHERING.....	29
3.2. ASSIGN AN IP ADDRESS USING A MASTER CARD.....	13	6.8.1. Checksum.....	29
4. TELNET ACCESS TO THE READER.....	14	6.8.2. Padding.....	29
4.1. READER'S CONSOLE.....	14	6.8.3. Ciphering.....	30
4.1.1. Open a Telnet session to the reader.....	14	6.8.4. Chaining.....	30
4.1.2. Sending a command to the Reader.....	15	6.9. GENERAL COMMUNICATION FLOW.....	30
4.1.3. List of Console commands.....	16	6.9.1. Nominal dialogue.....	30
5. TCP CLIENT/SERVER PROTOCOL – LOW LAYERS, PLAIN MODE.....	17	6.9.2. Timings.....	30
5.1. ABSTRACT.....	17	6.9.3. Chaining.....	31
5.2. PRESENTATION LAYER.....	17	6.10. ERROR HANDLING AND RECOVERY.....	31
5.2.1. Block format.....	17	6.10.1. For the Reader.....	31
5.2.2. Description of the fields.....	18	6.10.2. For the Host.....	31
5.2.3. Size of the blocks.....	18	6.10.3. Recovery.....	31
5.2.4. Format of the TYPE byte.....	19	6.11. APPLICATION LAYER.....	32
5.3. GENERAL COMMUNICATION FLOW.....	20	7. APPLICATION LAYER PROTOCOL.....	33
5.3.1. Session establishment.....	20	7.1. PRINCIPLES.....	33
5.3.2. Nominal dialogue.....	20	7.2. HOST → READER, AVAILABLE WITH BOTH ADMINISTRATION AND OPERATION KEYS.....	33
5.3.3. Timings.....	21	7.2.1. Get Global Status.....	33
5.3.4. Chaining.....	21	7.2.2. Start/Stop Reader.....	33
5.4. ERROR HANDLING AND RECOVERY.....	21	7.2.3. Clear LEDs command.....	34
5.4.1. For the Reader.....	21	7.2.4. Set LEDs command.....	34
5.4.2. For the Host.....	21	7.2.5. Start LED sequence command.....	34
5.4.3. Recovery.....	21	7.2.6. Buzzer command.....	35
5.5. APPLICATION LAYER.....	22	7.3. HOST → READER, AVAILABLE WITH ADMINISTRATION KEY ONLY.....	35
6. TCP/CLIENT SERVER PROTOCOL – LOW LAYERS, AUTHENTICATED MODE.....	23	7.3.1. Write Configuration Register.....	35
6.1. ABSTRACT.....	23	7.3.2. Erase Configuration Register.....	35
6.2. SUPPORTED CIPHER PROTOCOLS.....	23	7.3.3. Reset the Reader.....	35
6.2.1. Blowfish.....	23	7.4. READER → HOST.....	36
6.2.2.3DES2K.....	24	7.4.1. Reader Identifier.....	36
6.2.3. AES.....	24	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

You'll find any details regarding hardware and physical characteristics of each reader in the corresponding datasheet.

Document ref.	Content
<i>To be written</i>	FunkyGate-IP NFC product flyer and datasheet
<i>To be written</i>	FunkyGate-IP+POE NFC product flyer and datasheet

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

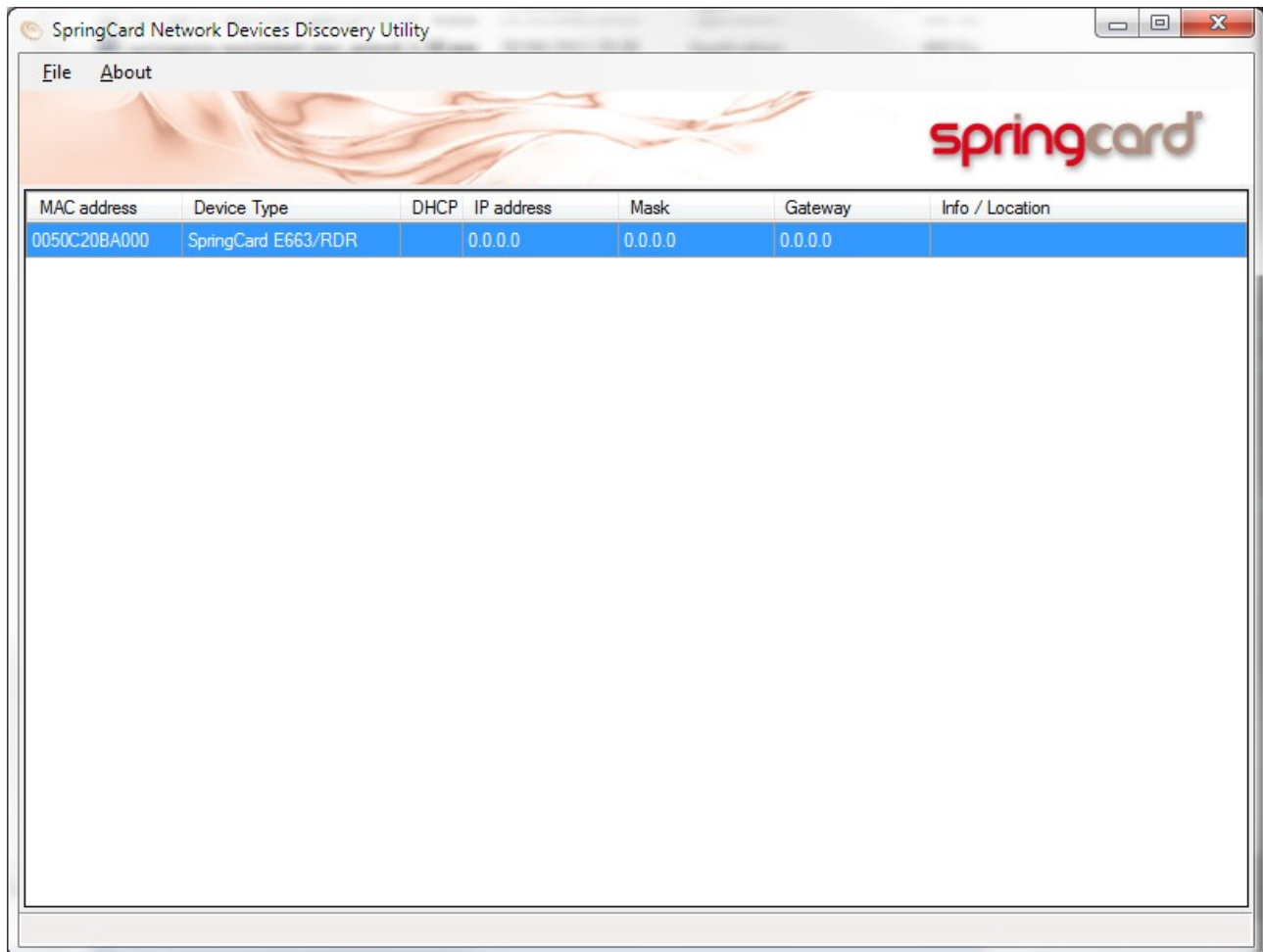
Make sure your Windows account has administrative privileges.

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	

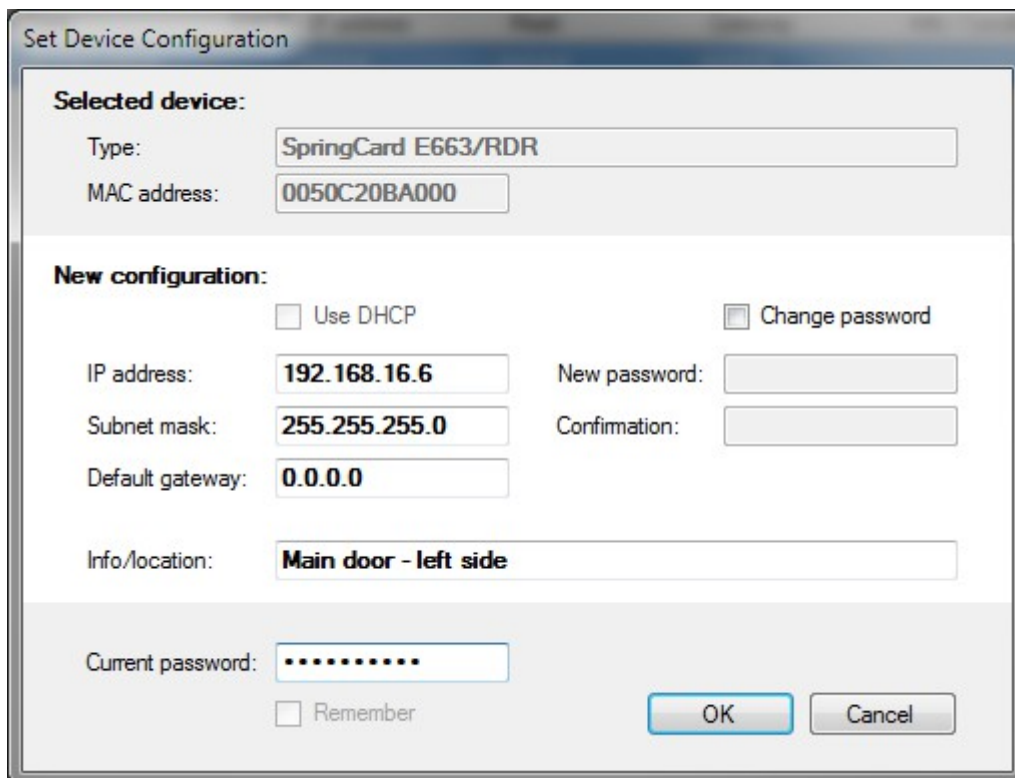
The software's main screen shows 7 columns:

- 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:



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.

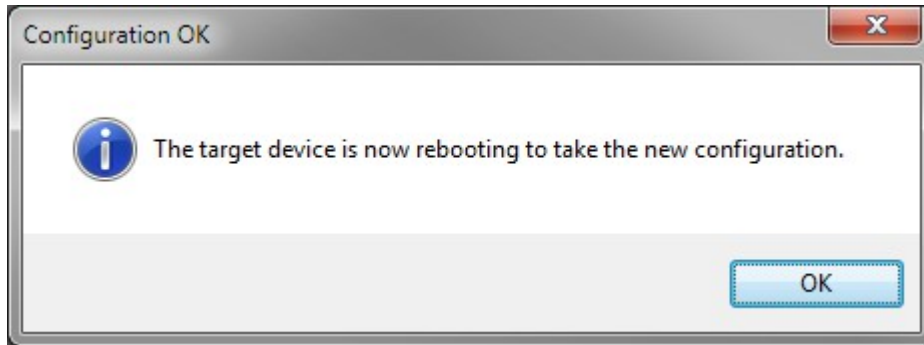
*The default password for all devices is **springcard**.*

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

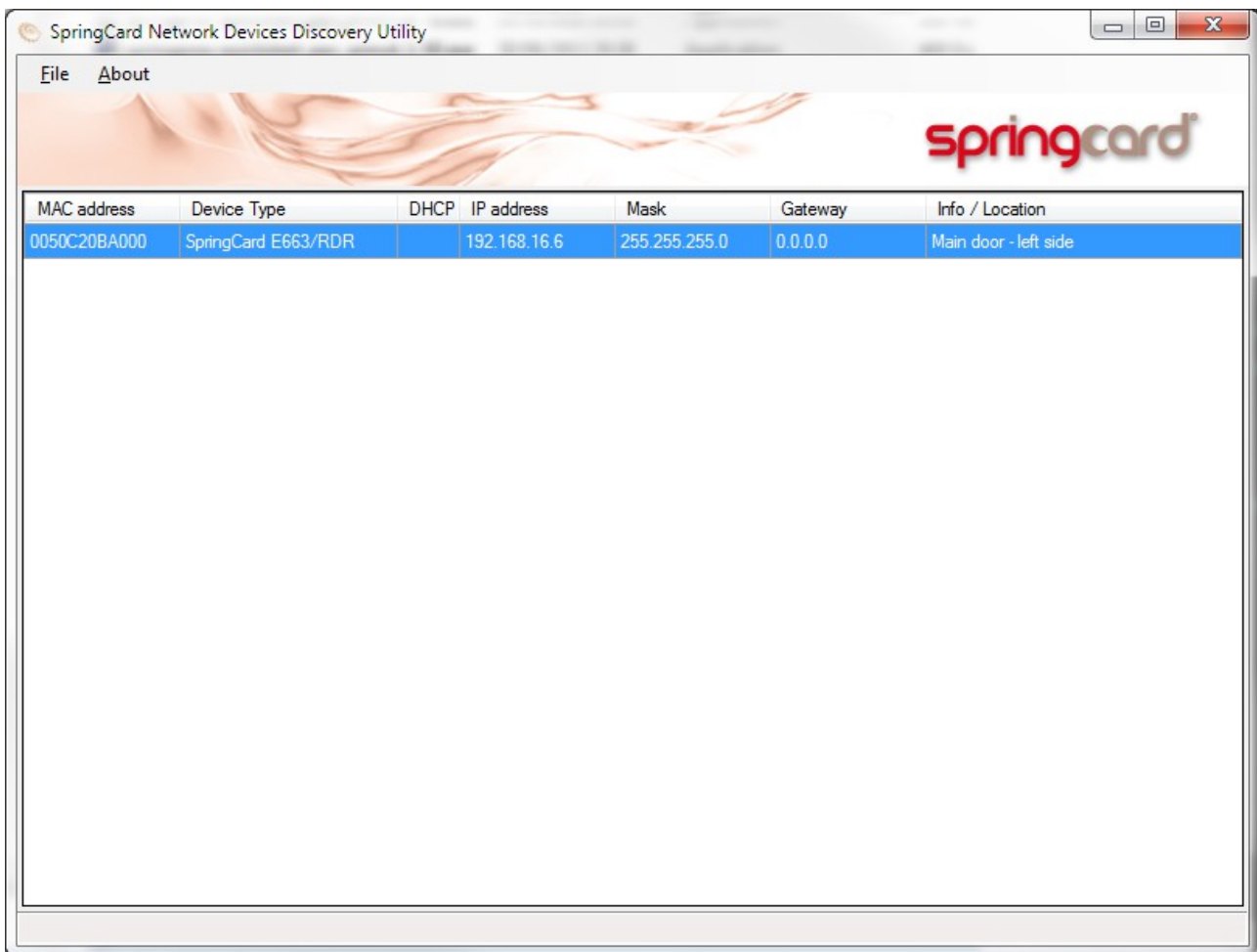
When ready, click “OK”.

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:



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



3.2. ASSIGN AN IP ADDRESS USING A MASTER CARD

To be written

4. TELNET ACCESS TO THE READER

4.1. READER'S CONSOLE

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

Note that the SEC Configuration Register (h6E, § 9.5) 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.

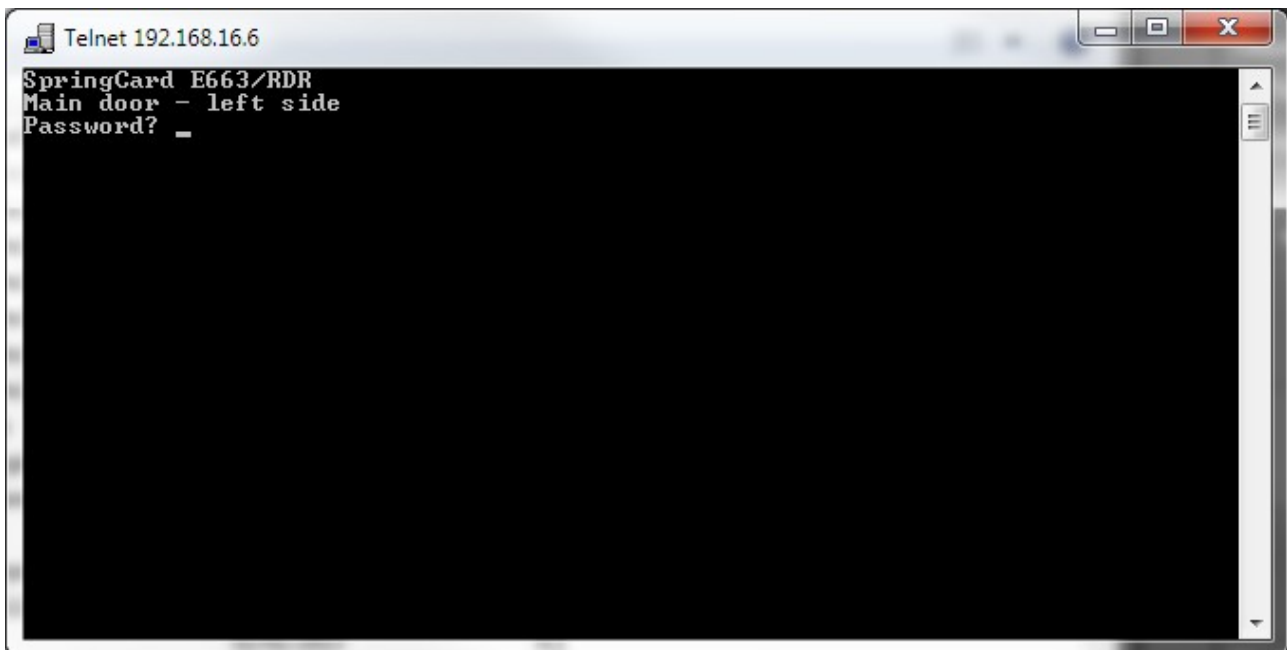
*Windows Vista / 7 / 8 : the Telnet client may be missing from your OS default install. Go to **Control Panel, Programs and Features** section, and then enable **Telnet client** in the **Turn Windows features on or off** tab.*

*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 lines 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**.*



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 $_hYY...YY$ to Configuration Register $_hXX$
cfgXX=!!	Erase Configuration Register $_hXX$
cfgXX	Read Configuration Register $_hXX$
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: <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 0 for Host → Reader • 1 for Reader → Host
6	Shall be set to 0
5	Shall be set to 0
4	Chaining <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 0 for Host → Reader • 1 for Reader → Host
6	Shall be set to 1
5	Block type: <ul style="list-style-type: none"> • _b00: HELO (Reader's "hello" block) • _b01: HELO-OK (Host's "hello" acknowledge) • _b10: RFU, do not use • _b11: HELO-AUTH (see § 6.3)
4	
3	Protocol Version for HELO block
2	
1	
0 (lsb)	
	Key Number for the first HELO-AUTH block
	0000 for the other blocks.

c. Protocol Version

The Reader sets this field to 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
h08	hC0	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
h02	h50	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} , § 9.4.3).

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. SUPPORTED CIPHER PROTOCOLS

The Reader supports 3 cipher protocols. The protocol to be used together with either key is defined in the IPS Configuration Register (h_{83} , § 9.4.3).

6.2.1. Blowfish

Blowfish is a widely used algorithm that provides a good encryption rate in software.

Blowfish has a fixed 64-bit (8 bytes) block size. CBC mode is used to cipher a frame that is longer than the block size. The initialization vector is reset after every frame.

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

6.2.2. 3DES2K

3DES2K or “Triple DES with 2 keys” is the combination of 3 DES rounds, the left-part of the key being used for rounds 1 and 3, and the right-part for round 2.

3DES2K has a fixed 64-bit (8 bytes) block size. CBC mode is used to cipher a content that is longer than the block size. The initialization vector is reset after every frame.

The Reader stores the 2 56-bit keys as a single 16-byte key.

6.2.3. AES

AES or Rijndael is the standard that supersedes DES and Triple DES.

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 is reset after every frame.

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

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 one (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
h02	h70	empty

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

LENGTH	TYPE	PAYLOAD
h02	h71	empty

6.3.3. Authentication, step 1

After receiving the HELO-Auth block from the Host,

- The Reader activates the selected secret key K_S ,
- The Reader generate a random challenge (C_R) on 16 bytes,
- The Reader sends to the Host a block containing $E (K_S, C_R)$.

Authentication, step 1: block Reader → Host

LENGTH	TYPE	PAYLOAD
h12	hF0	$E (K_S, C_R)$ on 16 bytes

6.3.4. Authentication, step 2

- The Host activates the secret key K_S ,
- The Host decipheres the payload received from the Reader, and retrieves C_R ,
- The Host computes $C_R' = C_R \ll 1 \mid \mid C_R \gg 127$ (shift left with carry),
- The Host generate a random challenge (C_H) on 16 bytes,
- The Host sends to the Reader a block containing $E (K_S, C_H \mid \mid C_R')$,

Authentication, step 2: block Host → Reader

LENGTH	TYPE	PAYLOAD
h22	h70	$E (K_S, C_H \mid \mid C_R')$ on 32 bytes

6.3.5. Authentication, step 3

- The Reader decipheres 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 $C_H' = C_H \ll 1 \mid \mid C_H \gg 127$ (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
$_{h}12$	$_{h}F0$	$E (K_S, C_H')$ on 16 bytes

6.3.6. Host's HELO-OK

- The Host decipheres 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 sends to the Reader a HELO-OK block.

HELO-OK block (Host → Reader)

LENGTH	TYPE	PAYLOAD
$_{h}02$	$_{h}50$	empty

6.4. SESSION KEY

The session key K_T is $C_H \oplus C_R$ (exclusive OR).

Further ciphering is performed using the same algorithm as during the authentication.

6.5. 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.

3-pass authentication proceeds as usual from § 6.3.2 to § 6.3.6.

6.6. PRESENTATION LAYER AFTER AUTHENTICATION

6.6.1. Block format

Every block transmitted in the channel is formatted as follow:

LENGTH	TYPE	CONTENT CIPHERED BY THE SESSION KEY			
		SEQUENCE	PAYLOAD	CHECKSUM	PADDING
1 byte	1 byte	4 bytes	Variable length	4 bytes	Variable length

6.6.2. 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
SEQUENCE	The SEQUENCE number (DWORD) provides 2 features: <ol style="list-style-type: none"> 1. Insert some "salt" in the ciphered part, so that identical payloads will always be ciphered differently 2. Prevent a defrauder to replay a valid block without being noticed
PAYLOAD	The PAYLOAD field is optional. When present, the PAYLOAD field conveys application data.
CHECKSUM	The CHECKSUM field is a standard CRC32, computed over the SEQUENCE and PAYLOAD fields
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.

6.6.3. 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 74 if the cipher's block size is 8 (Blowfish and 3DES2K),
- Between 18 and 82 if the cipher's block size is 16 (AES).

6.6.4. Format of the TYPE byte

a. I_S-Block

Bit	Description
7 (msb)	Direction <ul style="list-style-type: none"> • 0 for Host → Reader • 1 for Reader → Host
6	Shall be set to 0
5	Shall be set to 1
4	Chaining <ul style="list-style-type: none"> • 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)	

6.7. SEQUENCE NUMBERS

Both the Reader and the Host maintain 2 sequence numbers:

- The sequence number Reader → Host (SEQUENCE_R),
- The sequence number Host → Reader (SEQUENCE_H).

Both numbers are initialized to _h00000000 every time the session key is generated.

a. SEQUENCE_R

The Reader sends SEQUENCE_R in its I_S-Block.

The Reader increments SEQUENCE_R after every I_S-Block transmitted without chaining.

The Host shall monitor SEQUENCE_R and shall drop the connection if an out-of-sequence block is received.

b. SEQUENCE_H

The Host sends SEQUENCE_H in its I_S-Block.

The Host shall increment SEQUENCE_H after every I_S-Block transmitted without chaining.

The Reader monitors SEQUENCE_H and drops the connection if an out-of-sequence block is received.

c. Overflow

If either $SEQUENCE_R$ or $SEQUENCE_H$ reaches $_{h}FFFFFFF$, the Reader drops the connection.

The Host shall periodically use the New authentication process (§ 6.5) to generate a new session key and reset both counters.

6.8. CHECKSUM, PADDING, CIPHERING

6.8.1. Checksum

The checksum is a CRC32 according to ITU-T V42, and computed over the SEQUENCE and PAYLOAD fields.

It is happened by the sender after the PAYLOAD (after the SEQUENCE if the PAYLOAD is empty).

The receiver uses the CRC32 (and the PADDING) to ensure that the block's content has been correctly recovered after deciphering.

$$CHECKSUM = CRC32 (SEQUENCE || PAYLOAD)$$

6.8.2. Padding

Before ciphering, the sender must make sure the block's content length is a multiple of the cipher's block size.

The receiver uses the PADDING (and the CRC32) to ensure that the block's content has been correctly recovered after deciphering.

$$PLAIN_CONTENT = SEQUENCE || PAYLOAD || CHECKSUM || PADDING$$

a. Blowfish and 3DES2K

The cipher's block size is 8 bytes. The sender adds 1 to 8 bytes, until the correct length is reached.

The value of the padding bytes is equal to the length of the padding: the sender adds $_{h}01$ if 1 byte is needed, $_{h}02 02$ if 2 bytes are needed... and $_{h}08 08 08 08 08 08 08 08$ if the length was already a multiple of 8 bytes.

b. AES

The cipher's block size is 16 bytes. The sender adds 1 to 16 bytes, until the correct length is reached.

The value of the padding bytes is equal to the length of the padding: the sender adds $_{h}01$ if 1 byte is needed, $_{h}02 02$ if 2 bytes are needed... and $_{h}10 10 \dots 10 10$ (16 times the $_{h}10$ value) if the length was already a multiple of 16 bytes.

6.8.3. Cipherng

The sender applies its cipherng algorithm, using the session key K_T , to generate the CIPHER_CONTENT.

$$\text{CIPHER_CONTENT} = E (K_T, \text{PLAIN_CONTENT})$$

The receiver retrieves

$$\text{SEQUENCE} \parallel \text{PAYLOAD} \parallel \text{CHECKSUM} \parallel \text{PADDING} = D (K_T, \text{CIPHER_CONTENT})$$

6.8.4. Chaining

When Chaining is used (§ 6.9.3), only the first blocks contains the SEQUENCE field, and only the last block contains the CRC32 and the PADDING fields.

Checksum, padding and cipherng operation shall be done on the “complete” content buffer, before splitting it into 64-B chunks to be sent in chained I_S -blocks.

6.9. GENERAL COMMUNICATION FLOW

6.9.1. 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).

6.9.2. 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.

6.9.3. 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.

6.10. ERROR HANDLING AND RECOVERY

6.10.1. 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.

6.10.2. 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.

6.10.3. 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.

6.11. 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
hD100	h01	seq.

- **seq:**
 - h00 buzzer OFF,
 - h01 buzzer ON,
 - h02 buzzer short sequence,
 - h03 buzzer long sequence.

7.3. HOST → READER, AVAILABLE WITH ADMINISTRATION KEY ONLY

7.3.1. Write Configuration Register

This command allows to write into any Configuration Register. <addr> is on one byte (valid values are h00 to hFE).

T	L	V	
h0C	<var.>	<addr>	<value>

7.3.2. Erase Configuration Register

This command allows to erase any Configuration Register, to go back to default value. <addr> is on one byte (valid values are h00 to hFE).

T	L	V
h0C	h01	<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
h0C	h00

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
h8100	h1C	SpringCard FunkyGate II 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
h2F	h01	Bit field, the broken tampers are denoted by the corresponding bit set to 1. V = h00 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
hB000	<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
hB100	<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
hB100	h00

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 (^h6E, § 9.5) may be used to disable either way to access the Configuration Registers.

Administration Key is defined in the IPS Configuration Register (_h83, § 9.4.3)

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 explicitly defined keep their default value).

Enter "cfgXX" to read the value of the Configuration register _hXX.

Note that Configuration registers $h55$, $h56$, $h6E$ and $h6F$ 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 hXX with value $hYYYY$. YYYYY can be any length between 1 and 32 bytes.

Enter “cfgXX=!!” to erase Configuration Register hXX .

8.2. USING MASTER CARDS

Preliminary information – not available yet.

The Master Cards are NXP Desfire cards formatted and programmed by **SpringCard Master Card Creation Tool** for Windows. Please refer to this software's documentation for details.

8.3. USING NFC PEER-TO-PEER

Preliminary information – not available yet.

The **SpringCard Reader Configuration Tool** for Android makes it possible to program SpringCard Readers from using an Android smartphone's or tablet's NFC adapter. Please refer to this software's documentation for details.

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

Default value: _b00001100 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: _b00001111

Buzzer control bits

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

Default value : $\text{b}00010010$

9.4. TCP CONFIGURATION

9.4.1. IPv4 address, mask, and gateway

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

IPv4 configuration bytes

Bytes	Contains	Remark
0	Address, 1 st byte	Reader's IPv4 Address. If these bytes are missing, the default IP Address hC0 A8 00 FA (192.168.0.250) is taken.
1	Address, 2 nd byte	
2	Address, 3 rd byte	
3	Address, 4 th byte	
4	Mask, 1 st byte	Network Mask. If these bytes are missing, the default Mask hFF FF FF FF (255.255.255.0) is taken.
5	Mask, 2 nd byte	
6	Mask, 3 rd byte	
7	Mask, 4 th byte	

Default value: $\text{hC0 A8 00 FA FF FF FF 00}$

(*address = 192.168.0.250, mask = 255.255.255.0*)

9.4.2. Server port

Name	Tag	Description	Size
IPP	_h 81	Listen TCP port for the server (2 bytes, MSB-first)	2

Default value: _h0F 9F (server TCP port = 3999)

9.4.3. 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)
4-3	00	The Administration Key is disabled
	01	The Administration Key uses the Blowfish cipher
	10	The Administration Key uses the 3DES2K cipher
	11	The Administration Key uses the AES cipher
2-1	00	The Operation Key is disabled
	01	The Operation Key uses the Blowfish cipher
	10	The Operation Key uses the 3DES2K cipher
	11	The Operation Key uses the AES cipher
0	0	Plain communication is allowed
	1	Secure communication is mandatory

Default value: _b00001010

(both keys use the Blowfish cipher, plan communication is allowed)

9.5. 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	Telnet access is enabled
	1	Telnet access is disabled
6	0	<i>RFU (set to 0)</i>
5	0	<i>RFU (set to 0)</i>
4	0	<i>RFU (set to 0)</i>
3	0	<i>RFU (set to 0)</i>
Tampers		
2	0	Tampers must be OK at power up
	1	Report alarm only if a tamper is broken later on
1	0	Signal tamper alarms on buzzer
	1	Do not signal tamper alarms on buzzer
0	0	Reader stops reading when a tamper is broken
	1	Reader keeps on reading when a tamper is broken

Default value: _b00000111

9.6. PASSWORD

Laetitia à recopier

10. THE TEMPLATE SYSTEM

SpringCard FunkyGate-IP NFC provides 4 “Card Processing Templates” that tells the Reader which data shall be retrieved from the cards/tags, and how the Card Identifier shall be constructed 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 your **SpringCard FunkyGate-IP NFC**.

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.

Copyright © PRO ACTIVE SAS 2013, all rights reserved.

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