PMA13257-CA
FINAL - PUBLIC

SpringCard FunkyGate-IP NFC

Guide d'intégration et de configuration

www.springcard.com

Document Identification

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

File name	V:\Dossiers\SpringCard\A-Notices\RFID scanners et lecteurs\FunkyGate-IP\[PMA13257-CA] FunkyGate-IP NFC Integration and Configuration Guide FR.odt		
Date saved	21/08/18	Date printed	25/07/16

Revision History

Ver.	Date	Author	Valid. by		Approv. by	Details
			Tech.	Qual.		
AA	29/09/13	JDA				Created from PMA959P
AB	24/04/14	JDA				Rewritten the authentication scheme
AC	28/05/14	JDA				Added the HTTP server and the REST API
BA	30/07/14	JDA			JDA	Network part moved to PMA14166
BB	29/07/15	JIZ				Changed reset command: §7.2.1 and §7.4.3
CA	08/07/16	JDA				Added the HTTP client

Contents

1. Introduction

1.1. Résumé

Le SpringCard FunkyGate-IP NFC est un lecteur mural RFID (13.56MHz) et NFC, pour les applications de contrôle d'accès. Le **SpringCard FunkyGate-IP NFC** dispose d'une interface TCP/IP par Ethernet exclusive.

Le design attractif et l'efficacité de l'interface Ethernet font de ce produit le permier choix pour les envrionnements de bureau. Le support avancé d'une large gamme de technologies et les caractéristiques de sécurité exclusives permettent aux schémas de contrôle d'accès haut de gamme d'être déployés facilement.

Grâce au **système de modèles polyvalents** (partagés avec tous les autres lecteurs et scanners RFID/NFC de **SpringCard**), le **SpringCard FunkyGate-IP NFC** est capable de lire le numéro de série ou n'importe quelle données provenant des cartes de proximité du standard ISO/IEC 14443, et des tags ou étiquettes du standard ISO/IEC 15693. Il est également capable de trouver les données NDEF des puces RFID formatées suivant les spécifications du NFC Forum, et de recevoir les données NDEF provenant d'un “peer-to-peer” NFC Forum (SNEP server on top of LLCP).

Le **SpringCard FunkyGate-IP+POE NFC** comprend également le “powered by network” (POE).

Ce document fournit toutes les informations nécessaires pour configurer le **FunkyGate-IP NFC** et le **FunkyGateIP-POE + NFC**, et pour développer un logiciel qui pourra gérer les données venant du lecteur, ainsi que pour diriger et re-configurer le lecteur lorsque ce sera nécessaire.

1.2. Public

Ce document est destiné aux développeurs d'applications et aux intégrateurs systèmes. Il suppose que le lecteur a de solides connaissances du développement informatique, des réseaux TCP/IP et des technologies RFID/NFC.

1.3. Support et mises à jour

Les documents liés (fiche technique de produits, notes d'applications, échantillon logiciel, HOWTO et FAQs…) sont disponibles sur le site web SpringCard:

www.springcard.com

Les nouvelles versions de ce document et d'autres sont mises en ligne dès qu'elles sont disponibles. Pour les demandes de support technique, merci de joindre notre support technique via cette page web:

www.springcard.com/support

1.4. Documents liés

1.4.1. Spécificités du produit

Vous trouverez la liste des options et caractéristiques techniques dans le document correspondant.

Document ref.	Content
PFL13276	FunkyGate NFC family leaflet

1.4.2. Documents communs

a. Intégration et configuration réseau

Le lecteur **SpringCard FunkyGate-IP NFC** partage le même protocole de communication réseau (en plus du TCP/IP) et la même manière de configurer le réseau que le module I/O **SpringCard HandyDrummer-IP**. Un document partagé par les deux produits couvre cette partie.

Document ref.	Content
PMA14166	FunkyGate-IP NFC, E663/RDR, HandyDrummer-IP, E663/MIO Network Integration and Configuration

b. Lecture et traitement des cartes

Tous les lecteurs et scanners RFID SpringCard partagent le même système de traitement de cartes via & à 4 modèles de traitement. La manière dont le lecteur traite la carte est détaillé dans un document partagé par tous les produits de cette famille.

Document ref.	Content
PMA13205	Readers / RFID Scanners Template System

2. Définir l'adresse IP du lecteur

Le lecteur sort de l'usine sans adresse IP. Cela signifie qu'il faut lui en donner une avant d'accéder à votre lecteur via Telnet link (chapitre 3) ou utiliser le protocole client/serveur TCP décrit dans le chapitre 6.

Utiliser le **SpringCard Network Device Discovery Utility (NDDU)** est la meilleure méthode pour assigner une adresse IP au lecteur.

> *Ce lecteur ne supporte pas le Dynamic Host Configuration Protocol (DHCP). Seules les adresses fixes IPv4 sont supportées.*

2.1. Assigner une adresse IP en utilisant le logiciel NDDU

SpringCard Network Device Discovery Utility (NDDU) est un logiciel Windows qui découvre et configure les appareils SpringCard connectés sur le même réseau local (LAN) lorsque l'ordinateur fonctionne.

> *Merci d'utiliser une connexion réseau filaire, et vérifier que le(s) lecteur(s) que vous voulez configuré est bien sûr le même LAN que votre ordinateur. NDDU utilise les frames de diffusion UDP pour découvrir et configurer le lecteur, cela ne fonctionnera pas derrière un routeur ou une gateway.*

2.1.1. Télécharger et installer un logiciel NDDU

Vérifiez que votre compte Windows a un accès administrateur.

Télécharger l'installateur avec cet URL

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

Installer le logiciel.

> *Ce logiciel est basé sur le .NET framework version 4. Merci de télécharger et d'installer ce framework de Microsoft s'il n'est pas déployé sur votre ordinateur.*

2.1.2. Faire fonctionner le logiciel NDDU

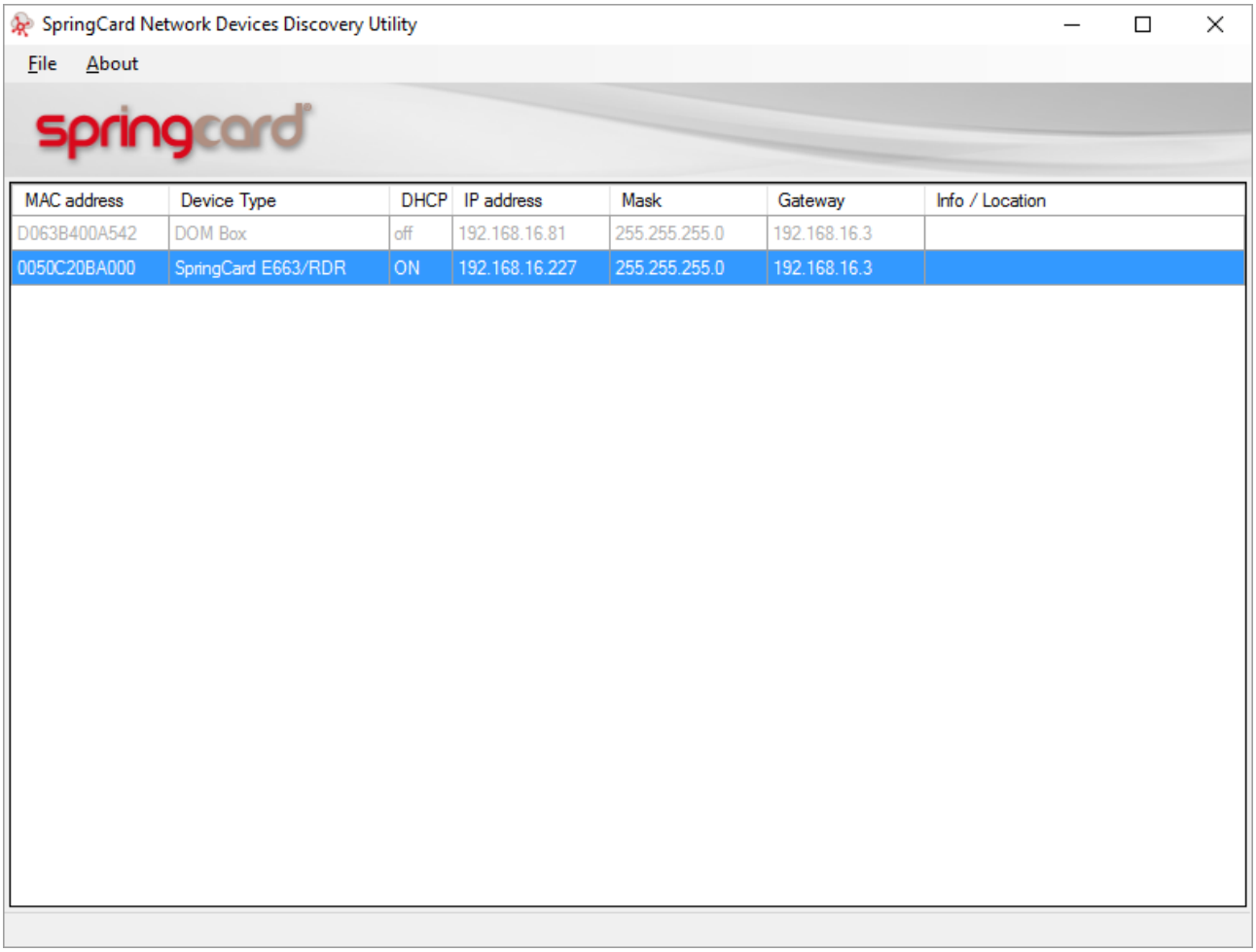
Vérifiez que votre compte Windows à les accès administateur.

Lancer le logiciel: Start Menu → SpringCard → Network Discovery → Network Device Discovery Utility.

Lors du premier lancement, vous devriez être averti par le Firewall Windows si vous souhaitez que NDDU accède à votre réseau. Merci de confirmer.

2.1.3. Appareils découverts

Après quelques secondes, NDDU affiche une liste d'appareils trouvés sur le LAN.



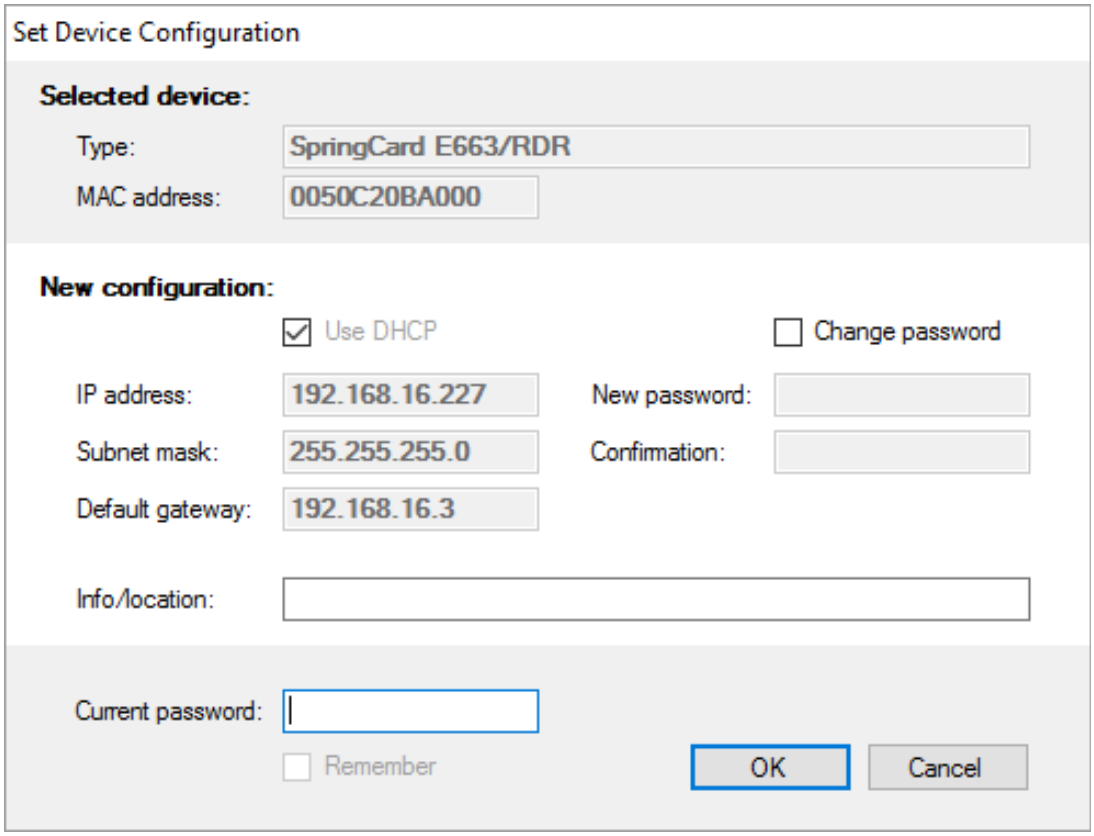
L'écran principal du logiciel affiche 7 colonnes:

- L'adresse MAC (l'adresse Ethernet et le numéro de série) de chaque appareil SpringCard trouvé sur le LAN,
- Le type d'appareil. **FunkyGate-IP NFC** et **FunkyGate-IP+POE** apparaîssent sous le type **SpringCard E663/RDR**,
- Si le DHCP est activé ou non (DHCP n'est pas supporté sur les premières versions du firmware),
- L'adresse IP actuelle de l'appareil, le masque du réseau local, et la gateway par défaut. Jusqu'à ce que l'appareil soit bien configuré, ces entrées afficheront “0.0.0.0”,

- Un fil définit par l'utilisateur appelé “Info / localisation”, qui sera utilisé comme une trace pour identifier l'appareil dans votre propre système.

2.1.4. Configurer un lecteur

Double-clicker l'un des appareils de la liste. Le formulaire de configuration apparaît:



Le formulaire montre la configuration actuelle de l'appareil. Entrer la nouvelle configuration.

a. Utiliser le DHCP?

DHCP signifie Dynamic Host Configuration Protocol. Activez le DHCP sur l'appareil seulement s'il y a un serveur DHCP qui fonctionne sur le réseau.

> *Note: souvent, le logiciel utilisateur se connectera comme un client à un service serveur fonctionnant dans le lecteur. Si le lecteur utilise DHCP, son adresse changera fréquemment et le logiciel client devra être reconfiguré. Il est recommandé de laisser un bail permanent au serveur DHCP pour éviter ce problème.*

b. Configuration statique

L'adresse IPv4 et le masque subnet sont des données obligatoires qui ne peuvent pas être laissées vides. La gateway par défaut est optionnelle, si les appareils n'utilisent pas de gateway “0.0.0.0”.

c. Info/Localisation

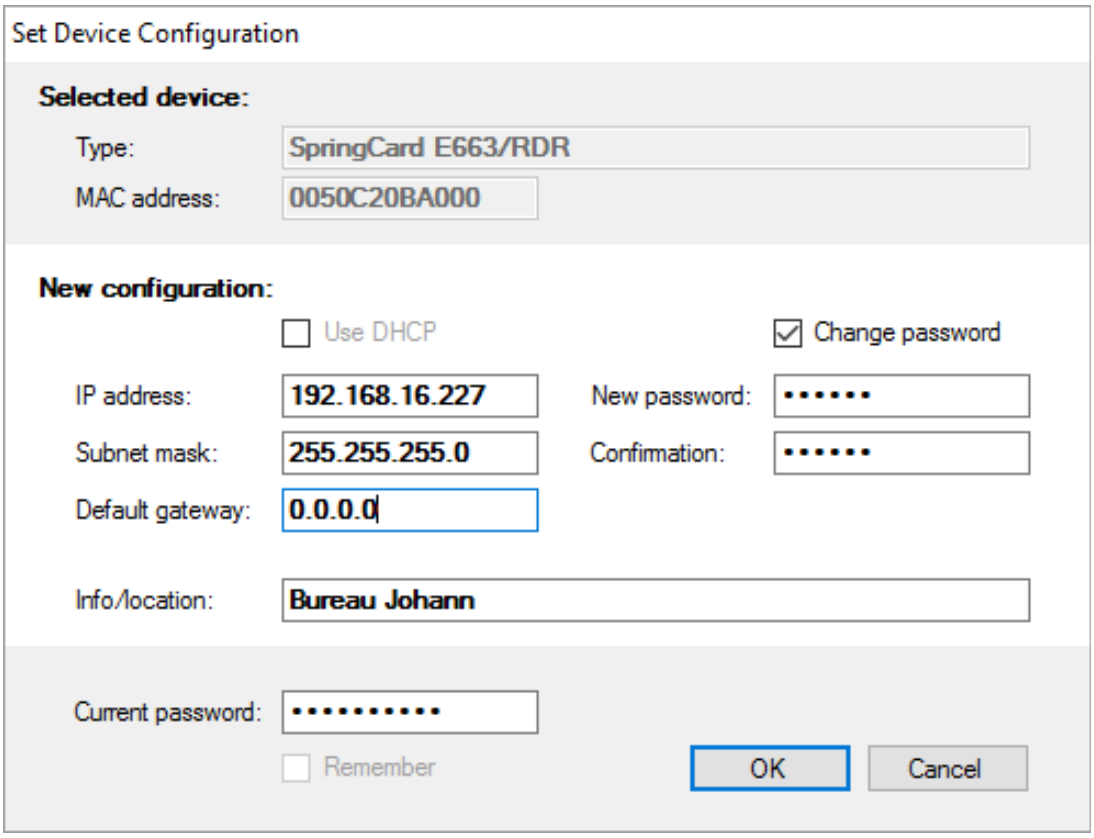
Dans le champ “Info/Location”, entrer un fil court (moins de 32 caractères) comme rappel de la localisation ou du rôle de l'appareil.

d. Mot de passe

Vérifier la boîte “changer de mot de passe” et entrer un nouveau mot de passe deux fois si vous souhaitez changer le mot de passe de l'appareil.

Terminer en entrant le mot de passe actuel de l'appareil pour confirmer que vous êtes autorisé à changer la configuration de cet appareil.

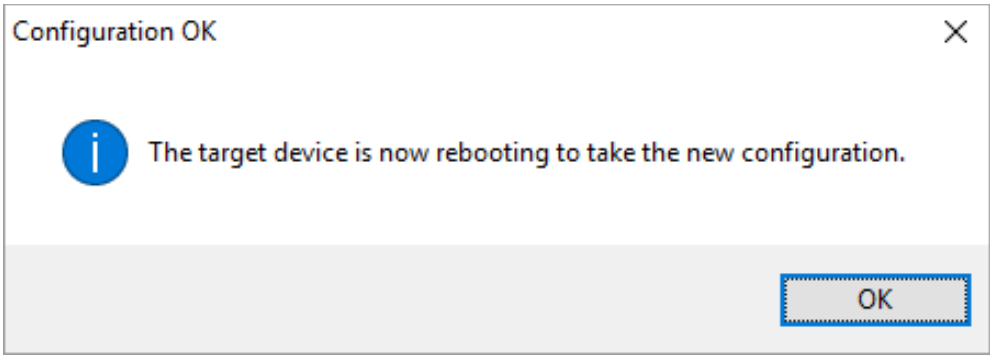
> *Le mot de passe par défaut de nos appareils est **springcard**.*



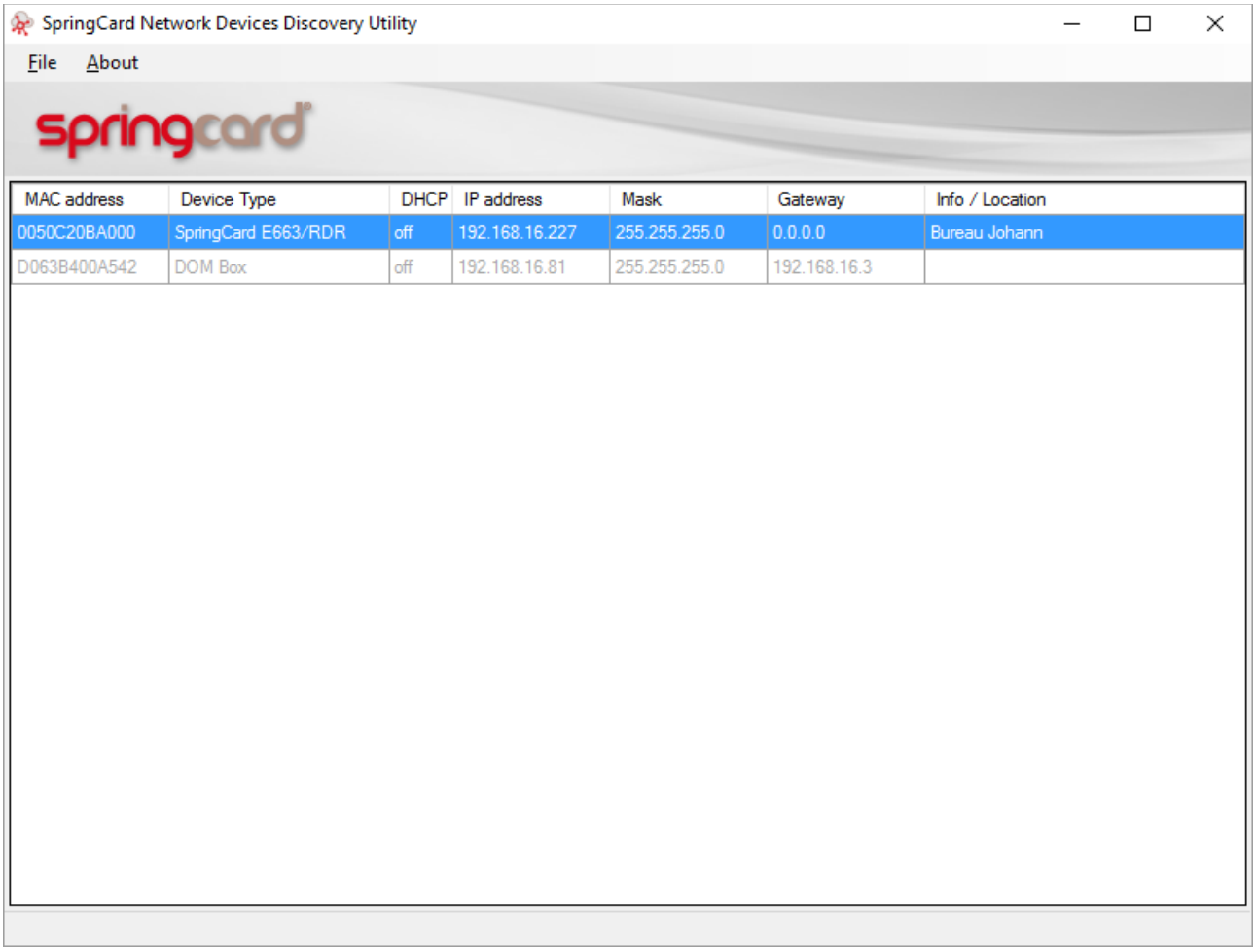
Losque c'est prêt, cliquer sur “OK”.

2.1.5. Vérifier la nouvelle configuration

Si tous est bon, ainsi que le mot de passe actuel, le logiciel NDDU sera capable de configurer le lecteur. Le message suivant confirme que la nouvelle configuration a été acceptée:



Après quelques secondes, la liste des appareils est mise à jour et montre la nouvelle configuration:



MAC address	Device Type	DHCP	IP address	Mask	Gateway	Info / Location
0050C20BA000	SpringCard E663/RDR	off	192.168.16.227	255.255.255.0	0.0.0.0	Bureau Johann
D063B400A542	DOM Box	off	192.168.16.81	255.255.255.0	192.168.16.3	

2.2. Assigner une adresse IP en utilisant une Master Card

Le lecteur peut être configuré en utilisant une Master Card sans contact.

Les Master Cards sont des cartes NXP Desfire formattées et programmées par **SpringCard Configuration Tool (ScMultiConf.exe, ref # SN14007)** pour Windows.

Merci de vous référer à la documentation de ce logiciel pour plus de détails.

3. Accès Telnet au lecteur

3.1. La console lecteur

Le lecteur dispose d'un processeur de commandes “humain” (shell ou console). Cette option est accessible via le protocole Telnet. Il est fait pour tester et réaliser des démonstrations. Seules les quelques commandes décrites dans ce chapitre peuvent être utilisées de manière sécurisée pour la configuration et le diagnostic.

Note que le SEC Configuration Register (h6E, § 9.4) peut être utilisé pour désactiver la console.

3.1.1. Ouvrir une session Telnet au lecteur

Dans la plupart des systèmes d'exploitation vous pouvez trouver un client Telnet dans les outils systèmes par défaut. Ouvrir une console et entrer

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

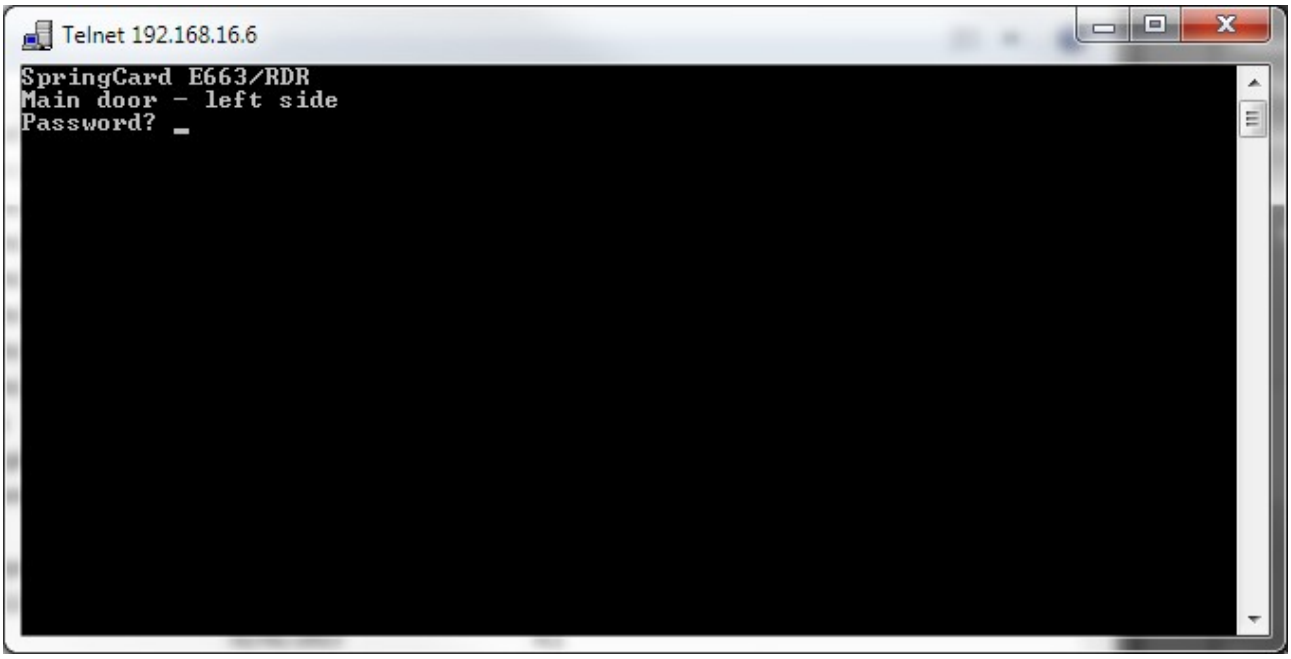
où xxx.xxx.xxx.xxx est l'adresse IP du lecteur définie dans le chapitre 2.

Windows Vista / 7 / 8 / 10 : le client Telnet peut être absent de l'installation de votre OS par défaut. Aller au ***tableau de contrôle, section programmes et options,*** *puis activer* ***le client Telnet*** *dans le tableau* ***Activer ou désactiver des options Windows.***
Vous pouvez également, télécharger un terminal client gratuit comme ***Putty****, qui est également un client Telnet.*

Le shell du Telnet du lecteur dit “SpringCard E663/RDR”, puis le fil d'info/localisation entré dans le chapitre 2, et enfin une demande de mot de passe.

Entrer le mot de passe du lecteur que vous avez défini dans le chapitre 2.

Si vous n'avez pas changer le mot de passe, le mot de passe par défaut est ***springcard****.*



3.1.2. Envoyer des commandes au lecteur

Écrire la ligne de commandes comme indiqué ci-dessous et terminer en entrant la clé ENTRER.

Noter que le lecteur répercute les caractères entrés.

3.1.3. Liste des commandes console

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

4. Utiliser le lecteur en mode client HTTP

4.1. Résumé

Le lecteur fonctionne avec un petit client HTTP (web) embarqué capable de se connecter à un serveur web et d'envoyer ses données dans une demande POST. Le contenu des demandes est authentifié par une fonction à sens unique (HMAC-MD5). Le lecteur est capable d'analyser une réponse formattée JSON pour retrouver une commande LED ou buzzer.

Pour commencer avec HTTP et JSON, merci de lire

- *http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol*
- *http://en.wikipedia.org/wiki/JSON*

Cette option permet d'utiliser le lecteur dans des architectures basées sur le Cloud.

4.2. Activer et configurer le client HTTTP

Pour sélectionner le mode HTTP client, il faut modifier le registre général des options (OPT, h60, § 9.1).

Le client HTTP (web) doit être configuré avec attention avant utilisation. Voici les paramètres de configuration dont vous aurez besoin:

Parameter	Content	Constraints/Limits	Register
server	The fully-qualified DNS name of the HTTP server to send the Requests to. This could also be an IPv4 address in the form "xxx.xxx.xxx.xxx".	32 chars. Max *Do not add the protocol header ("http://") nor the port number to this field.*	h89
port	The port number. Default is 80.		h88 bytes 2-3
query	The path to the target script on the server. The Reader constructs the complete URL "http://**<servername>[:<port>]/<query>**"	32 chars. max Do not add the leading slash ("/") to this field.	h8A
hmac_key	The key used to authenticate the requests coming from the Reader.	16 byte value. If empty, authentication is disabled.	h8B
max_timeout	Time (in seconds) to wait for a response from the server. Default value is 30s.	The Reader stops reading until the server has answered or a timeout occurs.	h88 byte 0
keepalive	Interval (in seconds) after which a keepalive request is sent, to notify the server that the Reader is still there.	Minimum value is 2 * max_timeout. Set to 0 to disable this feature	h88 byte 1

4.3. Limites

Le lecteur supporte IPv4 et HTTP 1.1 seulement. Il ne supporte pas HTTPS.
Le lecteur ne peut pas envoyer une requête plus longue que 512B (avec les titres).
Le lecteur ne peut pas traiter une réponse plus longue que 512B (avec les titres).
Le lecteur traitera seulement les réponses avec le code HTTP 200 ("success"). Les autres réponses sont ignorées. Le code 301 ("moved permanently") ne déclenche pas de nouvelle requête du lecteur.
N'interrogez pas le serveur HTTP (web) du lecteur plus de 4 fois par seconde (i.e. un chaque 250ms) pour laisser le lecteur réaliser son travail de lecteur.

4.4. HTTP Client – POST requête

Field name	Description
Mandatory fields (always transmitted by the Reader)	
what	Reason of the request. Possible values are: "startup": transmitted when the Reader starts up "read": transmitted when a card has been read and Insert/Remove mode is disabled "insert": transmitted when a card is inserted "remove": transmitted when a card is removed "tamper": transmitted when the status of the tampers is changed "keepalive": transmitted periodically if this feature is enabled
sequence	Counter incremented after every startup
counter	Counter incremented after every request
mac	Serial number / MAC address of the Reader
tampers	Current status of the tampers
Optional fields (present only when required by the context)	
id	Card Identifier transmitted together with what="read" or what="insert"
info	Info / Location field transmitted only if this field is non-empty
auth	Authentication string (HMAC) transmitted only if the authentication key is non-empty

4.5. HTTP Client – JSON réponse

Field name	Description
led-red	Action on the red LED. Possible values are: “off” “on” (default) “blink-fast” “blink-slow”
led-green	Action on the green LED. Possible values are: “off” “on” (default) “blink-fast” “blink-slow”
led-time	Duration of the sequence on the LEDs, in seconds. 0 means forever.
buzz	Duration of the buzzer's sound, in tenth of second. 0 means no sound.

5. Utiliser un lecteur en mode serveur HTTP – REST API

5.1. Résumé

Le lecteur fait fonctionner un petit serveur HTTP (web) embarqué qui écoute sur le port TCP 80 et fournit le REST API basique. L'API fournit ses résultats en structures JSON, ou en script, contenant une seule fonction appel JavaScript (le paramètre de la fonction appel est le même que pour la structure JSON).

Pour commencer avec JSON et REST, merci de lire

- *http://en.wikipedia.org/wiki/Representational_State_Transfer*
- *http://en.wikipedia.org/wiki/JSON*

Cette option rend possible l'utilisation du lecteur dans une application qui embarque un composant HTTP client.

Le mode HTTP serveur est intrinsèquement non-sécurisé. Les applications de contrôle d'accès "sérieuses" doivent être construites sur des protocoles d'authentification client/serveur décrit dans le chapitre 6, et non pas sur du HTTP.

5.2. Activer le serveur HTTP

Pour sélectionner le mode serveur HTTP, il faut modifier le registre général des options (OPT, $_h$60, § 9.1).

5.3. Limites

Le lecteur supporte IPv4 et HTTP 1.1 seulement. Il ne supporte pas HTTPS.
N'interrogez pas le serveur HTTP (web) du lecteur plus de 4 fois par seconde (I.e un chaque 250ms) pour laisser le lecteur réaliser son travail de lecteur.

5.4. REST API – Liste de fonctions

Resource	Description	See §
Functions that returns a JSON response		
GET iwm2/read	Return the Card Identifier of the last Card that has been read. The Card Identifier is cleared afterwards.	5.5.1
GET iwm2/read/keep	Return the Card Identifier of the last Card that has been read. The Card Identifier is not cleared afterwards.	5.5.2
GET iwm2/buzz/{time_ms}	Drive the buzzer	5.6.1
GET iwm2/led/{color}/{mode}	Drive one LED (permanent)	5.6.2
GET iwm2/led/{color}/{mode}/{time_ms}	Drive one LED (temporary)	5.6.2
GET iwm2/leds/auto	Go back to default mode for both LEDs	5.6.3
Functions that returns a script (containing a JavaScript function call)		
GET iwm2_cb/read	Return the Card Identifier of the last Card that has been read. The Card Identifier is cleared afterwards.	5.5.1
GET iwm2_cb/read/keep	Return the Card Identifier of the last Card that has been read. The Card Identifier is not cleared afterwards.	5.5.2
GET iwm2_cb/buzz/{time_ms}	Drive the buzzer	5.6.1
GET iwm2_cb/led/{color}/{mode}	Drive one LED (permanent)	5.6.2
GET iwm2_cb/led/{color}/{mode}/{time_ms}	Drive one LED (temporary)	5.6.2
GET iwm2_cb/leds/auto	Go back to default mode for both LEDs	5.6.3

5.5. REST API – Boucle d'interrogation

5.5.1. iwm2/read and iwm2_cb/read

C'est le point d'entrée principal du lecteur. Invoquer cette fonction pour avoir l'identifiant de la dernière carte lue par le lecteur.

Cette fonction doit être invoquée au minimum toutes les 60 secondes sinon le lecteur signale qu'il est déconnecté grâce à ses LEDs.

> *Le dernier identifiant est effacé lorsque la fonction revient. Utiliser iwm2/read/keep si l'identifiant doit rester visible pour une autre demande client.*

a. iwm2/read

Invoquer **iwm2/read**

Le lecteur renvoie une structure JSON. Trois cas sont possibles:

Aucune carte n'a été lue depuis la dernière demande – le dernier identifiant est vide

Le lecteur renvoie:

```
{ "iwm2": {
      "success": "true",
      "id": ""
}}
```

Une carte a été lue moins d'une seconde avant

Le lecteur renvoie:

```
{ "iwm2": {
      "success": "true",
      "id": "The card's identifier"
}}
```

Une carte a été lue il y a plus d'une seconde (et il y a moins de 60 secondes)

Le lecteur renvoie:

```
{ "iwm2": {
      "success": "true",
      "id": "The card's identifier",
      "seconds_ago": "Number of seconds elapsed since the card has been read"
}}
```

b. iwm2_cb/read

Invoquer **iwm2_cb/read**

Le lecteur renvoie un script: 3 cas sont possibles:

Aucune carte n'a été lue depuis la dernière invocation – le dernier identifiant est vide

Le lecteur renvoie:

```
iwm2_cb( { "iwm2": {
      "success": "true",
      "id": ""
}});
```

Une carte a été lue moins d'une seconde avant

Le lecteur renvoie:

```
iwm2_cb( { "iwm2": {
      "success": "true",
      "id": "The card's identifier"
}});
```

Une carte a été lue il y a plus d'une seconde (et moins de 60 secondes avant)

Le lecteur renvoie:

```
iwm2_cb( { "iwm2": {
      "success": "true",
      "id": "The card's identifier",
      "seconds_ago": "Number of seconds elapsed since the card has been read"
}});
```

5.5.2. iwm2/read/keep et iwm2_cb/read/keep

Comme pour **iwm2/read** et **iwm2_cb/read/keep** mais le dernier identifiant reste actif.

5.6. REST API – Envoyer des commandes au lecteur

5.6.1. iwm2/buzz et iwm2_cb/buzz

Il met le buzzer en position ON/OFF.

- Pour mettre le buzzer en position ON, invoquer **iwm2/buzz/{time_ms}** où **{time_ms}** est la valeur décimal de la durée du son exprimé en millisecondes (1 à 65534).
- Pour mettre le buzzer en position OFF, invoquer **iwm2/buzz/0**.

a. Iwm2/buzz

Invoquer **iwm2/buzz/{time_ms}**

Le lecteur renvoie:

```
{ "iwm2": {
      "success": "true"
}}
```

b. iwm2_cb/buzz

Invoquer **iwm2_cb/buzz/{time_ms}**

Le lecteur renvoie:

```
iwm2_cb( { "iwm2": {
      "success": "true"
}});
```

5.6.2. iwm2/led et iwm2_cb/led

Cette commande permet de gérer les LEDs du lecteur.

La syntaxe complète est **iwm2/led/{color}/{mode}/{time_ms}**

Les valeurs autorisées pour le paramètre **{color}** sont

- **red**
- **green**

La LED bleue ne peut pas être gérée explicitement.

Les valeurs autorisées pour le paramètre **{mode}** sont

- **on**
- **fast**
- **slow**
- **heart**

Le paramètre **{time_ms}** est la valeur décimale de la durée de la commande, exprimée en millisecondes (1 à 65534). Supprimer le paramètre **{time_ms}** (ou mettez le à 0) pour le rendre permanent.

a. Iwm2/led

Invoquer **iwm2/led/{color}/{mode}/{time_ms}**

Le lecteur renvoie:

```
{ "iwm2": {
      "success": "true"
}}
```

b. iwm2_cb/led

Invoquer **iwm2_cb/led/{color}/{mode}/{time_ms}**

Le lecteur renvoie:

```
iwm2_cb( { "iwm2": {
      "success": "true"
}});
```

5.6.3. iwm2/leds and iwm2_cb/leds

Invoquer la commande **iwm2/leds/auto** pour annuler une commande LED en attente (§ 5.6.3).

a. Iwm2/leds

Invoquer **iwm2/leds/auto**

Le lecteur répond:

```
{ "iwm2": {
      "success": "true"
}}
```

b. iwm2_cb/leds

Invoquer **iwm2_cb/leds/auto**

Le lecteur répond:

```
iwm2_cb( { "iwm2": {
      "success": "true"
}});
```

5.7. REST API – Erreurs

a. iwm2 namespace

Pour chaque erreur le lecteur renvoie:

```
{ "iwm2": {
      "success": "false"
}}
```

b. iwm2_cb namespace

Pour chaque erreur le lecteur renvoie:

```
iwm2_cb( { "iwm2": {
      "success": "false"
}});
```

6. Protocole réseau des appareils SpringCard C/S

6.1. Résumé

Le protocole réseau des appareils SpringCard C/S est un protocole réseau léger avec une bande passante efficace. Le lecteur est un serveur TCP et l'hôte (unité de contrôle d'accès ou l'ordinateur qui s'en charge) est le client.

Noter que le lecteur n'est pas capable d'accepter plus d'un client à la fois. Essayer de se connecter au même lecteurs par deux hôtes différents n'est pas supporté et ne doit pas être essayé. Une attitude inexpliquée pourrait apparaître.

Il existe deux modes de communication:

- **Mode simple**, sans sécurité
- **Le mode sécurisé**, basé sur la cryptographie AES.

Dans le mode sécurisé, il y a deux clés d'authentification, menant à deux niveaux d'authentification:

- La **clé d'opération** donne accès aux opérations basiques du lecteur (§ 7.3). C'est la clé qu'une unité de contrôle utiliserait pour faire fonctionner le lecteur.
- La **clé d'administration** permet de changer la configuration du lecteur (§ 7.4). Cette clé sera utilisée par un logiciel de configuration lors de l'installation du lecteur.

Le protocole est intégralement documenté dans [PMA14166], chapitres 5 et 6.

7. Appareil réseau SpringCard – Couche applicative du lecteur

7.1. Principes

Ce chapitre décrit les **couches applicatives du lecteur**, dont les datagrams du niveau application sont transmis au début du protocol décrit dans [PMA14166], chapitres 5 et 6.

Cette couche applicative est intégralement documentée dans [PMA14166], chapitre 7. Ce chapitre est uniquement un extrait qui résume les options du lecteur.

> *Le **protocole réseau des appareils SpringCard C/S** n'est pas un protocole demande/réponse; comme une chaîne TCP est complètement bidirectionnelle, l'hôte et le lecteur puvent parler à tout moment. L'hôte doit donc être prêt à traiter (ou au moins à mettre en attente) un datagramme du niveau applicatif envoyé par le lecteur à n'importe quel moment.*

7.2. Liste des codes-opération et identifiants de champs de données

7.2.1. Codes-opération (Hôte → Lecteur)

T (Tag)	Operation	See §
h00	Get Global Status	7.3.1
h0A	Start / Stop Reader	7.3.2
hD000	Clear LEDs Set LEDs Start LEDs	7.3.3 7.3.4 7.3.5
hD100	Buzzer	7.3.6
	Restricted operations (available only after authentication using Administration Key)	
h0C	Write Configuration Erase Configuration	7.4.1 7.4.2
h0B	Reset the Reader (to apply the Configuration)	7.4.3

7.2.2. Identifiants champs de données (Lecteur → Hôte)

T (Tag)	Operation	See §
h8100	Reader Identifier	7.5.1
h2F	Tamper Status	7.5.2
hB000	Card Read	7.5.3
hB100	Card Inserted Card Removed	7.5.4 7.5.5

7.3. Hôte → Lecteur, opérations basiques

Les opérations listées dans ce chapitre sont disponibles **quelque soit le mode**:

- Simple (pas d'authentification),
- Sécurisé, après authentification en utilisant la clé d'opération,
- Sécurisé, après authentification en utilisant la clé d'administration.

7.3.1. Avoir le statut global

T	L
$_h$00	$_h$00

Le lecteur répond par 2 frames:

1. Identifiant lecteur
2. Status du piratage

7.3.2. Lecteur Start/Stop R

T	L	V
$_h$0A	$_h$01	**mode**

- **mode:** start/stop commande
  - $_h$00 lecteur position OFF (RF field OFF, pas d'activité en RF)
  - $_h$01 lecteur position ON

7.3.3. Enlever les commandes LEDs

Les deux LEDs sont en position OFF.

T	L
$_h$D000	$_h$00

7.3.4. Configurer les comandes LEDs

Les deux LEDs sont gérées – jusqu'à ce qu'une commande “enlever les commandes LEDs” soit reçue.

T	L	V	
$_h$D000	$_h$02	red	green

- **red:** command for red LED
  - $_h$00 OFF
  - $_h$01 ON
  - $_h$02 blinks slowly
  - $_h$03 blinks quickly
- **green:** command for green LED
  - $_h$00 OFF
  - $_h$01 ON
  - $_h$02 blinks slowly
  - $_h$03 blinks quickly

7.3.5. Commencer une séquence de commande LED

Les deux LEDs sont gérées – jusqu'à ce qu'une commande “enlever les commandes LEDs” soit reçue ou que le délai soit dépassé.

T	L	V		
$_h$D000	$_h$04	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.

7.3.6. Commande buzzer

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.

7.4. Hôte → Lecteur, opérations confidentielles

Les opérations listées dans ce chapitre sont disponibles uniquement en **mode sécurisé après authentification en utilisant la clé d'administration**.

7.4.1. Écrire un registre de configuration

L'attitude du lecteur est définie par les registres de configuration documentés dans les chapitres 9 et 10. La commande écrire un registre de configuration permet d'écrire dans un registre de configuration si l'on a son adresse.

<addr> est le nombre du registre sur un byte (les valeurs valides sont $_h$00 à $_h$FE).

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

7.4.2. Effacer un registre de configuration

L'attitude du lecteur est définie par les registres de configuration documentés dans les chapitres 9 et 10. La commande effacer un registre de configuration vous permet d'effacer n'importe quel registre de configuration si vous avez son adresse. Une fois qu'un registre est effacé c'est la valeur par défaut de ce registre qui est enregistrée.

<addr> est le nombre du registre sur un byte (les valeurs valides sont $_h$00 à $_h$FE).

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

7.4.3. Réinitialiser le lecteur

Le lecteur doit être réinitialisé pour qu'une nouvelle configuration soit prise en compte. Lorsqu'il reçoit cette commande le lecteur se déconnecte et se réinitialise.

T	L	V	
$_h$0B	$_h$02	$_h$DE	$_h$AD

7.5. Lecteur → Hôte

7.5.1. Identifiant du lecteur

Ce T,L,V est transmis en réponse à la commande **avoir le statut global**.

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

7.5.2. Statut de piratage

Ce T,L,V est transmis en réponse à la commande **avoir le statut global** ou lorsque l'un des tampers est cassé ou restoré.

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.

7.5.3. Carte lue

Ce T,L,V est transmis lorsque le lecteur a lu une carte, si le mode insérer/retirer est désactivé.

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

7.5.4. Carte insérée

Ce T,L,V est transmis lorsque le lecteur a lu une carte si le mode insérer/retirer est activé.

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

7.5.5. Carte retirée

Ce T,L,V est transmis lorsqu'une carte est retirée, si le mode insérer/retirer est activé.

T	**L**
$_h$B100	$_h$00

8. Modifier la configuration du lecteur

La configuration du lecteur est stockées dans un ensemble de registres de configuration non-volatile. Il existe deux groupes de registres:

- Les registres qui contrôlent l'attitude du lecteur qui sont intégralement documentés dans le chapitre 9. Certains sont communs à plusieurs lecteurs SpringCard, mais d'autres sont spécifiques au **SpringCard FunkyGate-IP NFC**.
- Les registres qui contrôlent le système de modèle sont partagés par tous les lecteurs SpringCard. Le chapitre 10 est dont un espace redirigeant vers le document qui décrit précisément le système des modèles.

Mais cette subtile distinction entre les deux groupes nous permet de réduire la taille de ce document et de faciliter le changement d'un lecteur à l'autre. De manière technique, tous les registres sont définit (et accessibles) de la même manière.

Il y a quatre manières de modifier les registres de configuration d'un lecteur:

1. Grâce à Telnet link
2. En utilisant des Master Cards
3. En utilisant le protocole réseau des appareils SpringCard C/S, après authentification avec la clé d'administration.

> *Noter que le registre de configuration SEC ($_h$6E, § 9.4) peut être utilisé pour désactiver les accès aux registres de configuration.*
> *La clé d'administration est définie par le registre de configuration IPS ($_h$83, § 9.5.3)*

8.1. Grâce au Telnet link

Ouvrir une session Telnet du lecteur comme indiqué dans § 3.1.

8.1.1. Lire les registres de configuration

Entrer “cfg” pour lister les registres de configuration actuellement définis (les registres qui ne sont pas explicitement définis gardent leur valeur par défaut).

Entrer “cfgXX” pour lire la valeur du registre de configuration $_h$XX.

Noter que les registres de configuration $_h$55, $_h$56, $_h$6E et $_h$6F qui contiennent des données sensibles (les clés utilisées par les Master Cards et les clés secrètes et mot de passe du lecteur) sont masqués.

8.1.2. Écrire des registres de configuration

Entrer “cfgXX=YYYY” pour mettre à jour le registre de configuration $_h$XX avec la valeur $_h$YYYY. YYYY peut être de n'importe quelle longueur entre 1 et 32 bytes.

Entrer “cfgXX=!!” pour effacer le registre de configuration $_h$XX.

8.2. Utiliser les Master Cards

Les Master Cards sont des cartes NXP Desfire formatées et programmées par **SpringCard Configuration Tool (ScMultiConf.exe, ref # SN14007)** pour Windows.

Merci de vous référer à la documentation de ce logiciel pour plus de détails.

8.3. Grâce au protocole SpringCard Network Device C/S

Merci de vous référer à [PMA14166].

9. Configuration globale du lecteur

9.1. Options générales

Name	Tag	Description	Size
OPT	h60	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	*RFU*
	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	*RFU*
	11	Allow configuration by Master Card or NFC all the time
1 - 0		***Communication mode***
	00	The Reader uses the SpringCard Network Device C/S Protocol (§ 6)
	01	The Reader runs in HTTP server mode (§ 5)
	10	The Reader runs in HTTP client mode (§ 4)
	11	*RFU*
		Byte 1 (optional)
7	0	Insert/Remove mode is disabled (§ 7.5.3)
	1	Insert/Remove mode is enabled (§ 7.5.4 and § 7.5.5)
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	*RFU (set to 0)*
1	0	*RFU (set to 0)*

0	0	Reader is active on startup
	1	Reader is not active on startup (Host must send an activation command)

Default value: $_b$00001100 00000000

9.2. Délais et répétitions

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 et 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

Bits de contrôle buzzer

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		*RFU*
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 : $_b$00010010

9.4. Options de sécurité

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

Security option bits

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

Default value: $_b$10010100

[1] SpringField Colorado is a NFC-enabled application running on Android, or embedded in a specific NFC Tag, that retrieves and shows the Reader's data: firmware name and version, serial number, IP address etc.

9.5. ConfigurationTCP

9.5.1. Adresse IPv4, masque et gateway

Name	Tag	Description	Size
IPA	$_h$80	IPv4 configuration bytes, see table below	4 to 20

IPv4 configuration bytes

Bytes	Contains	Remark
0	Address, 1st byte	Device'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	
8	Gateway, 1st byte	Default Gateway. If these bytes are missing, the value $_h$00 00 00 00 (0.0.0.0) is taken, meaning that there's no Gateway.
9	Gateway, 2nd byte	
10	Gateway, 3rd byte	
11	Gateway, 4th byte	
12	DNS server 1, 1st byte	Address of 1st DNS server. If these bytes are missing, the value $_h$00 00 00 00 (0.0.0.0) is taken, meaning that there's no DNS server.
13	DNS server 1, 2nd byte	
14	DNS server 1, 3rd byte	
15	DNS server 1, 4th byte	
16	DNS server 2, 1st byte	Address of 2nd DNS server. If these bytes are missing, the value $_h$00 00 00 00 (0.0.0.0) is taken, meaning that there's no DNS server.
17	DNS server 2, 2nd byte	
18	DNS server 2, 3rd byte	
19	DNS server 2, 4th byte	

Default value: $_h$C0 A8 00 FA FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

(address = 192.168.0.250, mask = 255.255.255.0, no gateway, no DNS servers)

9.5.2. Protocol SpringCard Network Device C/S – Port serveur

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. Protocole SpringCard Network Device C/S – Paramètres de sécurité et clés d'authentification

Name	Tag	Description	Size
IPS	$_h$84	Server security settings bits, see table below	1

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	The Administration Key is enabled
	1	The Administration Key is disabled
1	0	The Operation Key is enabled
	1	The Operation Key is disabled
0	0	Plain communication is allowed
	1	Secure communication is mandatory

Default value: $_b$00000100

(only Operation Key is enabled, plain communication is allowed)

9.5.4. Protocole SpringCard Network Device C/S – Clé d'opération

Name	Tag	Description	Size
IPK.OPE	$_h$85	C/S Protocol Operation Key	16

Default value: $_h$00 … $_h$00

9.5.5. Protocole SpringCard Network Device C/S – Clé d'administration

Name	Tag	Description	Size
IPK.ADM	$_h$86	C/S Protocol Administation Key	16

Default value: $_h$00 … $_h$00

9.5.6. Configuration HTTP client

Name	Tag	Description	Size
HTC	$_h$88	HTTP client configuration bytes. See table below	1

HTTP client configuration bytes

Bytes	Contains	Remark
0	Network timeout, in seconds	The same timeout applies to DNS queries, TCP channel openings, and HTTP exchanges. Default is 30s
1	Keep-alive interval, in seconds	Default is 0 (keep-alive is disabled)
2	TCP port of the HTTP server, MSB	Default port is 80 ($_h$0050)
4	TCP port of the HTTP server, LSB	

Default value: $_h$1E 00 00 50

9.5.7. HTTP client – Nom de serveur

Name	Tag	Description	Size
HTS	$_h$89	HTTP client – name of the remote HTTP server	0 to 32

9.5.8. HTTP client – Fil d'interrogation

Name	Tag	Description	Size
HTQ	$_h$8A	HTTP client – query string on the remote HTTP server	0 to 32

9.5.9. Configuration Ethernet

Name	Tag	Description	Size
ETC	$_h$8D	Ethernet configuration bits. See table below	1

Ethernet configuration 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	*RFU (set to 0)*
1	0	*RFU (set to 0)*
0	0	Use auto-configuration (10/100Mbps, half or full-duplex)
	1	Force bitrate = 10Mbps, half-duplex

Default value: $_b$00000000

9.5.10. Info / Localisation

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

Default value: empty

Le fil **Info / Localisation** est une valeur texte (ASCII) qui apparaît

- Lorsque quelqu'un essaie de se connecter à Telnet,
- Dans le logiciel NDDU (§ 2.1.3).

Utiliser ce fil comme un rappel de là où votre lecteur est installé, ou quel est son rôle dans votre système de contrôle d'accès.

9.5.11. Mot de passe pour l'accès Telnet

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

Default value: “springcard”

Le **mot de passe pour accéder au fil Telnet** est une valeur texte (ASCII) qui protège l'accès au lecteur en utilisant le protocole Telnet.

Le mot de passe est obligatoire. Si le registre n'est pas configuré la valeur par défaut “springcard” est utilisée.

10. Le système des modèles

SpringCard FunkyGate-IP NFC fournit 4 “Modèles de traitement de cartes” qui définissent comment le lecteur atteindra les données des différentes cartes/tags, et comment l'identifiant de cette carte sera construit avec ses données avant d'être envoyé à l'hôte.

Le système de modèle est entièrement décrit dans le document **[PMA13205] “RFID/NFC Scanners Template System**”.

Merci d'utiliser ce document comme référence afin de configurer la partie lecteur de votre **SpringCard FunkyGate-IP NFC.**

11. Licences des parties tierce

SpringCard FunkyGate-IP NFC a été développé en utilisant des composants logiciels open-source.

11.1. FreeRTOS

FreeRTOS est un système d'exploitation en temps réel (ou RTOS) de Real Time Engineers Ltd. **SpringCard FunkyGate-IP NFC** fonctionne avec FreeRTOS v7.5.2.

FreeRTOS est distribué sous une licence modificée GNU General Public License (GPL) qui autorise son usage commercial, et les produits à source-fermée.

Pour plus d'information, ou pour télécharger le code source de FreeRTOS, visitez

www.freertos.org

11.2. uIP

µIP est un stack open-source TCP/IP développé par Adam Dunkels et sous licence BSD.

SpringCard FunkyGate-IP NFC utilise FreeTCPIP, une version modifiée de µIP qui est liée à FreeRTOS. Pour respecter la licence originale de µIP, nous avons copié le texte complet ici:

> Copyright (c) 2001-2003, Adam Dunkels.
>
> All rights reserved.
>
> Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
>
> 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
>
> 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
>
> 3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.
>
> THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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 SPRINGCARD 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 SPRINGCARD 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. SPRINGCARD reserves the right to change the information at any time without notice.

SPRINGCARD doesn't warrant any results derived from the use of the products described in this document. SPRINGCARD 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. SPRINGCARD customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify SPRINGCARD for any damages resulting from such improper use or sale.

Copyright notice

Editor's information

SPRINGCARD 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