RFID READER PROGRAMMING IN FAST PAYMENT SYSTEM

Katamadze Sofio, Topuria Nino
Georgian Technical University

Abstract

RFID stands for Radio Frequency Identification. As mentioned RFID devices use radio frequency waves to communicate. The common method used to identify person/object is to use serial number for each tag. Other information can also be customized as per the requirement. RFID tag usually will have smaller micro-chip along with the antenna. RFID tags communicate with the reader with the help of antenna. The reader converts reflected or transmitted information from tags to useful digital information which can be further processed by software running in the computer. The RFID reader simply plugs into a USB port. Once plugged in, the hardware side of the example is set up. It can't get easier than that. This paper presents brief RFID reader integration with the payment machine.

Keywords: Radio Frequency Identification. Payment machine. One-Time-Programmable bits. .Net technology.

1. Introduction

In different IT companies, various kinds of RFID readers are used. In our company, called HTS (Hizli Tahsilat Sistemleri) we used CN6XX(S)-X series reader. It is the preferred equipment in reader industry, because it is simple atmosphere, simple to use, powerful and wide application. It is a contactless Desktop reader to read/write 13.56MHz technology, including ISO14443A/B, Mifare, DesFire EV1, Mifare Plus, ISO15693 and read Sony Felica Compliant with ISO18092 NFC for peer to peer mode. It is as well with build-in SAM socket compliant to ISO7816 (T=0, T=1) to enable high strength encrypted transmission.

We use RFID reader to insert money in transport card called – Istanbul Card. Istanbul Card is widely used in big metropolicy.

2. Main Part

RFID reader gets information from the card and can write data in it. Well it sounds quite easy, but integration part requires a lot of work.

Istanbul Card has its protective mechanism and couple of steps to achieve the final step. It doesn't work offline, after successful data delivery, information is sent to provider by web service.

To implement device functions, we need to know not just device specifications (shown in Table) but also software development kit details. No let's interpret main commands we used for the card connection and their sequences [1]:

REQA command use ISO14443A REQA command

- WAKE-UP command use ISO14443A WAKE-UP command

- Anticoll command use ISO14443A MF_Anticoll command
- Select command use ISO14443A MF_Select command
- Anticol2 command use ISO14443A MF_Anticol2 command
- SelectL2 command use ISO14443A MF_Selectl2 command
- RATS command uses the same API as DESFire EV1 Card Commands, RATS
- PPS command use the same API as DESFire EV1 Card Commands, PPS
- DESELECT command use the same API as DESFire EV1 Card Commands, DESELECT
- int MF_Plus_WritePerso(int DeviceAddress,unsigned char CID,unsigned short BNr,unsigned char*Data)

SPECIFICATIONS **Tab.1**

Model	CN613(S)-X	CN670(S)-X
Operating Frequency	13.56MHz; APDU command	
Contactless Standard	ISO1443A 4(part1-4), Mifare, Mifare Plus, DESFire EV1	
	ISO1443B, ISO15693	ISO1443B, Felica;ISO18092NFC
Contactless Card Interface	Mifare1K/4K,Ultralight Mifare Plus,DESfire EV1,ISO7816	
	I Code II,Type B memory	Felica, Type B CPU
Contact Standard	ISO15693,ISO14443A/B,ISO7816	
SAM Slot	One build-in SAM slot as option	
Dimension	116*67*14mm	
Power Supply	5V DC,Power supplied from USB	
Current Consumption	<120mA	
Communication Rate	106-424kbps High speed transaction	
Baud Rate	9600-115200bit/s(default 115200)	
Indication	Controllable two-color LED, one buzzer: Red LED:Power state; Green LED: Normal Reading state	
Interface	USB or RS232(with USB for power)	
Communication Cable	1.5 meter long USB communication cable 1.5 meter long RS232 communication cable with splitter for the USB	
Reader Driver	USB 2.0 WIN7, XP, Vista, Linux, CCID compliant(CN6XX(S)-XC)	
Operating System	Windows98,2000, Vista, XP, 7, Server 2003&2008,ME98, WINCE6, 0/CE, net,	
Operating Distance	30-100mm(depending antenna, transponder)	
Operating Temperature	0℃- +60℃/+32° F ~+140°F	
Storage Temperature	- 10℃~ +65℃/+22° F~+149° F	
Operating Humidity	5~90% relative humidity non-condensing	
Special Feature	USB/RS232 for Easy firmware update. The Customized firmware could be implemented to meet the user's special	

Parameter	Description
DeviceAddress	Device Address of the reader
CID	The logical number of the addressed, you can active multiple cards simultaneously CID is assigned by API RATS (), the CID number must be in the range 0-14.
BNr	Indicate the key and block number of plus card, the following values are in hex format. 9000: Card Master Key 9001: Card Configuration Key 9002: Level 2 Switch Key 9003: Level 3 Switch Key 9004: SL1 Card Authentication Key B000: MFP Configuration Block B001: Installation Identifier B002: ATS Information B003: Field Configuration Block 0000-00FF: Mifare Blocks (Sector 0-39) 4000-404F: AES Sector Keys for sector 0-39, the second byte defines the sector number and which key (Key A or Key B) is used. Key A=Sector number multiplied by 2, Key B=sector number multiplied by 2+1, E.g. Key A for sector 2 has the number :4004

Before the plus card is switched to higher security levels, the data or key of BNr must be changed: 9000, 9001, 9002, 9003, 9004, this is mandatory, and we recommend writing all other keys and configuration blocks too Plus S card does not support SL2, therefore the BNr 9004 of block is unnecessary for plus S card.

Many functions are present in current firmware, it gives us big opportunity to change, add and even remove configurations. Now illustrate how we can achieve it:

int PN532_SAMConfiguration (int DeviceAddress, unsigned char Mode, unsigned char Timeout, unsigned char IRQ=1)

Parameter	Description
DeviceAddress	Device Address of the reader
Mode	Defines the way of using the SAM9(Security Access Module) **0x01**: **Normal mode**, the SAM is not used; this is the default mode. **0x02**: **Virtual Card**, the couple PN532+SAM is seen as only one contactless SAM card from the external world.

	0x03: Wired Card, the host controller can access to the SAM with standard PCD command (InListPassiveTarget, InDataExchange…) 0x04: Dual Card, both the PN532 and the SAM are visible from the external world as two separated targets. Virtual, Wired and Dual Card mode are only valid with 106 kbps ISO 14443-3 and 4 type A and Mifare
Timeout	Defines the time-out only in Virtual card configuration(Mode=0x02). In Virtual Card mode, this field is mandatory; whereas in the other mode, it is optional.
IRQ	Specifies if the PN532 takes care of the P70_IRQ pin or not. If the value is null(IRQ=0x00), the P70_IRQ pin remains at high level; whereas if the value is 0x01, the P70_IRQ pin is driven by the PN532. If the P70_IRQ parameter is not present, the default value is 0x01.
Return	0x00 – Successful (Refer to the API return code for other values)
Description	A SAM companion chip can be used to bring security. It is connected to the PN532 by using a S2C interface(SigIn(pin#36), SigOut(pin#35) and CLAD(pin#34)). The CLAD line is optional.

The API is used to write the data to the specified area with the static memory model area of blocks 0-Eh. It relates to an individual memory byte within the static memory model area of blocks 0-Eh. This command does not erase the target byte before writing the new data, and the execution time is approximately half that of the 'normal' write command (NFC_T1T_Write). Bits can be set but not reset (i.e., data bits previously set to a '1' cannot be reset to a '0'). If any of Block1 to BlockC are locked, the command is barred from that block, but it's not barred from BlockE to allow setting of lock and OTP bits. If the command is barred, the write-no-erase cycle is skipped-no write operation occurs and the tag will return to the "READY" state and wait for a new command. As a pre-condition, this command requires that the tag be in the READY state and afterward, the tag remains in READY state. This command has three main purposes:

- Lock – to set the 'lock bit' for a block.
- OTP- to set One-Time-Programmable bits (bytes 2 – 7 of BlockE), where between
- one and eight OTP bits can be set with a single command.
- A fast-write in order to reduce overall time to write data to memory blocks for the

first time given that the original condition of memory is zero.

We have discussed some firmware functions and its parameters. Documentation is really good to have but we need to map everything to codes and debug and test it many times. We have written integration part in .Net technology. Each call of firmware method ends up calling to web service. Afterwards service responses to the machine, and the answer is written in the card (using corresponding

command). Well it sounds that code should look big, so for illustration, let's have a look of connection part with the RFID reader. Listing 1 below shows C# function.

3. Conclusion

We demonstrated some snippets how to integrate RFID reader software development kit to the desired project. Device programming is really entertaining and not that difficult as it seems. All we need is good documentation of the machine, and good programming skills.

```
public bool Connect2Chip()
{
    HidDeviceLoader loader = new HidDeviceLoader();
    var deviceList = loader.GetDevices().ToArray();
    device = loader.GetDevices(4292, 7).FirstOrDefault(d => d.MaxInputReportLength == 65);
    if (device == null)
    {
        return false;
    }
    if (!device.TryOpen(out stream))
    {
        return false;
    }
    byte[] response = null;
    bool res = sendJinmuyuUsbCommand(stream, searchCard, out response);
    if(!res)
    {
        return false;
    }
    if(response[4] == 0x41 && response[5] == 0x7)
    {
        byte[] uid = new byte[7];

        Array.Copy(response, 6, uid, 0, 7);

        uidstring = Util.ByteArrayToString(uid);

        return true;
    }
    else
    {
        return false;
    }
}
```

Listing 1: Connection to RFID Reader

ლიტერატურა - References - Литература:

1. CV/NReader API Reference, Civintec Global Co.
2. RFID Programming Made Simple and Cheap (2009), Bradley Jones
3. CN6XX(S)-X Desktop Reader User Manual, Civintec Global Co.

რადიოსიხშირული საიდენტიფიკაციო ბარათით დაპროგრამება სწრაფი გადახდის სისტემაში

ქათამაძე სოფიო, თოფურია ნინო
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

RFID- ის მოწყობილობებს იყენებენ რადიოსიხშირული ტალღების კომუნიკაციისთვის. თითოეულ ობიექტს აქვს სერიული ნომერი, რომელიც გამოიყენება მისი იდენტიფიკაციისთვის. RFID tag-ს აქვს პატარა მიკრო ჩიპი ანტენასთან ერთად. RFID tag-თან დაკავშირება ხდება ანტენის დახმარებით. მოწყობილობა აკონვერტირებს მოწოდებულ ინფორმაციას ციფრულ ინფორმაციაში, რომელიც შეიძლება შემდგომ დამუშავდეს კომპიუტერში პროგრამული უზრუნველყოფის გაშვების გზით. RFID კომპიუტერს USB პორტით უკავშირდება. ნაშრომში განხილულია RFID მოწყობილობის პროგრამული ინტეგრაცია გადახდის აპარატთან. RFID იღებს ინფორმაციას, ამუშავებს და აგზავნის შესაბამის ვებ სერვისში. ბარათზე არსებული ინფრორმაციის წაკითხვა/ჩაწერა მოიცავს რამდენიმე ეტაპს და ყოველ ნაბიჯზე ხდება სერვერთან ინფორმაციის გადამოწმება. აქ გამოყენებულია თანამედროვე ტექნოლოგიები რომელიც უზრუნველყოფს პროგრამის უსაფრთხოებას.

ПРОГРАМИРОВАНИЕ С ПОМОЩЬЮ RFID-УСТРОЙСТВА В СИСТЕМЕ БЫСТРЫХ ОПЛАТ

Катамадзе С., Топурия Н.
Грузинский Технический Университет

Резюме

RFID обозначает радиочастотную идентификацию. Как упоминалось, RFID-устройства используют радиочастотные волны для связи. Общий метод, используемый для идентификации человека/объекта - использовать серийный номер для каждого тега. Другая информация также может быть настроена в соответствии с требованием. Обычно метка RFID будет иметь меньший микрочип вместе с антенной. RFID-метки общаются с читателем с помощью антенны. Считыватель преобразует отраженную или переданную информацию из тегов в полезную цифровую информацию, которая может быть дополнительно обработана программным обеспечением, запущенным на компьютере. RFID-считыватель просто подключается к USB-порту. После подключения к компьютеру настраивается аппаратная часть. В статье представлена краткая интеграция RFID-считывателя с платежной машиной.