Basic Tutorial for MIFARE Classic 1K using C#, C++ etc.

28. January 2016 at 9:25
(https://www.mifare.net/support/forum/topic/basic-tutorial-for-mifare-classic-1k-using-c-c-etc/#post-14732)

(https://www.mifare.net/support/forum/users/niyazi-torosgmail-com/)

NTMS
(https://www.mifare.net/support/forum/users/niyazi-torosgmail-com/)

★★★★★

Hi,

I download FSP_MIFC1K_v5.03.pdf and read it. I also bought an ACR890 POS terminal that comes with 5 pieces Mifare Classic 1K card and device SDK. I am very new for Mifare programming.
All I need is simple tutorial with a documentation that can point me;
1. How to connect to card when user tabs the RFID part of the terminal.
2. How to I write information in to the card
3. How to read information from card
4. How to successfully close the connection
My device SDK uses Ubuntu 12.04 Linux platform and I am using QT Creator Opensource to develop app for my ACR890 device. My device operating system is embedded Linux.
So, Where I can get basic Tutorial for MIFARE Classic 1K using C#, C++ etc?

+ 0 | - 0

28. January 2016 at 11:36

Re: Basic Tutorial for MIFARE Classic 1K using C#, C++ etc.

(https://www.mifare.net/support/forum/topic/basic-tutorial-for-mifare-classic-1k-using-c-c-etc/#post-14734)

(https://www.mifare.net/support/forum/users/mifaresdknxp-com/)

mifaresdk

(https://www.mifare.net/support/forum/users/mifaresdknxp-com/)

★★★★★

Hi,

what you have to do depends on the card reader you want to use and the driver and the interface of this driver. When you work on Linux you can use several interfaces to communicate with the MIFARE Classic card.

Let us assume you use the PC/SC interface and you have a driver which support the PCSC-Lite interface on Linux. Then you know how to write connect to the card, find out that a card is currently connected to the reader and to write and read data packets to the card. This data packets are usually so called APDU commands. You see in the MIDARE Classic data sheet that some commands are defined, like Read Data and Write Data.

In the datasheet you see also that the MIFARE Classic also defines 16 byte data blocks you can use for writing. 4 blocks are bundled together to a sector. For each sector you have a password and access bits which defined the protection level of the sector. You have to authenticate first, before you can access to a data block of a sector.

With this information you can start your investigations into the world of Smartcards.

Kind regards
The MIFARE team

+ 0 | - 0

28. January 2016 at 11:37

Re: Basic Tutorial for MIFARE Classic 1K using C#, C++ etc.

(https://www.mifare.net/support/forum/topic/basic-tutorial-for-mifare-classic-1k-using-c-c-etc/#post-14737)

(https://www.mifare.net/support/forum/users/niyazi-torosgmail-com/)

NTMS (https://www.mifare.net/support/forum/users/niyazi-torosgmail-com/)

★★★★★

Hi mifaresdk, ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲

I use QT Creator opensource to develop my embedded Linux terminal app. I upload my app using Wi-Fi and FileZilla.
My terminal has physical RFID module that I can use this module to write and read Mifare 1K Classic contactless card.
There is also C++ mifare demo example that shows how to read from card and update info to the card. To get better understanding I need some basic information to start.
Because there isn't and documentation for the below example. I can read the code and make some sense out of it. But I am very confuse.
Example shows below code;

This is the main cpp.

```
DialogMifareCardProgramming::DialogMifareCardProgramming(QWidget *parent) :
QDialog(parent),
ui(new Ui::DialogMifareCardProgramming)
{
ui->setupUi(this);

cCard_.openReader();
}
```

There is a button on initializeForm that connects to the card. I guess…….

```
void DialogMifareCardProgramming::on_pushButtonConnect_clicked()
{
try
{
cCard_.connect();

initializeControls(true);

if (cCard_._eCardType == CARD_TYPE_MIFARE_1K)
ui->labelStatus->setText("Connected to Mifare 1K");
else if (cCard_._eCardType == CARD_TYPE_MIFARE_4K)
ui->labelStatus->setText("Connected to Mifare 4K");
else
ui->labelStatus->setText("Connected to unknown card");
```

```
}
catch(AcsException cExpection)
{
ui->labelStatus->setText(cExpection.sMessage);
initializeControls(false);
}
catch(...)
{
ui->labelStatus->setText("Connection failed");
initializeControls(false);
}
}
```

▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲

This is load key module. I don’t understand what is happening here.

```
void DialogMifareCardProgramming::on_pushButtonLoadKey_clicked()
{
MIFARE_KEY_STORE eKeyStore;

char aKey[6];
char aAsciiKey[12];
int iCounter = 0;
bool bNumeric = false;

try
{
ui->lineEditKeyStoreNumberInput->text().toInt(&bNumeric);
if (!bNumeric)
{
ui->labelStatus->setText("Invalid key store number");
return;
}

if (ui->lineEditKeyStoreNumberInput->text().toInt() == 0)
eKeyStore = MIFARE_KEY_STORE_0;
else if (ui->lineEditKeyStoreNumberInput->text().toInt() == 1)
eKeyStore = MIFARE_KEY_STORE_1;
else
{
ui->labelStatus->setText("Invalid key store number");
return;
}
```

```
if (ui->lineEditKey1->text().length() != 2 ||
ui->lineEditKey2->text().length() != 2 ||
ui->lineEditKey3->text().length() != 2 ||
ui->lineEditKey4->text().length() != 2 ||
ui->lineEditKey5->text().length() != 2 ||
ui->lineEditKey6->text().length() != 2)
{
ui->labelStatus->setText("Invalid key");
return;
}
```

▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲

```
memcpy(aAsciiKey, ui->lineEditKey1->text().toAscii().data(), 2);
memcpy(aAsciiKey + 2, ui->lineEditKey2->text().toAscii().data(), 2);
memcpy(aAsciiKey + 4, ui->lineEditKey3->text().toAscii().data(), 2);
memcpy(aAsciiKey + 6, ui->lineEditKey4->text().toAscii().data(), 2);
memcpy(aAsciiKey + 8, ui->lineEditKey5->text().toAscii().data(), 2);
memcpy(aAsciiKey + 10, ui->lineEditKey6->text().toAscii().data(), 2);

for (iCounter = 0; iCounter labelStatus->setText("Invalid key");
return;
}
}

cHelper_.getBytes(ui->lineEditKey1->text().toAscii().data(), aKey);
cHelper_.getBytes(ui->lineEditKey2->text().toAscii().data(), aKey + 1);
cHelper_.getBytes(ui->lineEditKey3->text().toAscii().data(), aKey + 2);
cHelper_.getBytes(ui->lineEditKey4->text().toAscii().data(), aKey + 3);
cHelper_.getBytes(ui->lineEditKey5->text().toAscii().data(), aKey + 4);
cHelper_.getBytes(ui->lineEditKey6->text().toAscii().data(), aKey + 5);

cCard_.loadKey(aKey, eKeyStore);

ui->labelStatus->setText("Load key succeeded");
}
catch(AcsException cException)
{
char *pMessage = new char[cException.sMessage.length() + 10];
sprintf(pMessage, "%02X %02X - %s", cException.aStatusWord[0],
cException.aStatusWord[1], cException.sMessage.toUtf8().constData());
ui->labelStatus->setText(pMessage);
delete pMessage;
}
catch(...)
{
ui->labelStatus->setText("Load key failed");
}
}
```

And this is authentication module. Again I don't understand what is happening in here….

```
void DialogMifareCardProgramming::on_pushButtonAuthenticate_clicked()
{
MIFARE_KEY_TYPE eKeyType;
MIFARE_KEY_STORE eKeyStore;
bool bNumeric = false;

try
{
if (ui->radioButtonKeyA->isChecked())
eKeyType = MIFARE_KEY_TYPE_A;
else if (ui->radioButtonKeyB->isChecked())
eKeyType = MIFARE_KEY_TYPE_B;
else
{
ui->labelStatus->setText("Please select key type");
return;
}

ui->lineEditKeyStoreNumber->text().toInt(&bNumeric);
if (!bNumeric)
{
ui->labelStatus->setText("Invalid key store number");
return;
}

if (ui->lineEditKeyStoreNumber->text().toInt() == 0)
eKeyStore = MIFARE_KEY_STORE_0;
else if (ui->lineEditKeyStoreNumber->text().toInt() == 1)
eKeyStore = MIFARE_KEY_STORE_1;
else
{
ui->labelStatus->setText("Invalid key store number");
return;
}

ui->lineEditBlockNumber->text().toInt(&bNumeric);
if (!bNumeric || ui->lineEditBlockNumber->text().toInt() labelStatus->setText("Invalid block
number");
return;
}

if (cCard_._eCardType == CARD_TYPE_MIFARE_1K)
{
if (ui->lineEditBlockNumber->text().toInt() > 63)
```

```
if (ui->lineEditBlockNumber->text().toInt() > 255)
{
ui->labelStatus->setText("Card does not have block " + ui->lineEditBlockNumber->text());
return;
}
}
else
{
if (ui->lineEditBlockNumber->text().toInt() > 255)
{
ui->labelStatus->setText("Card does not have block " + ui->lineEditBlockNumber->text());
return;
}
}

cCard_.authenticate(eKeyType, ui->lineEditBlockNumber->text().toInt(), eKeyStore);

ui->labelStatus->setText("Authentication succeeded");
}
catch(AcsException cException)
{
char *pMessage = new char[cException.sMessage.length() + 10];
sprintf(pMessage, "%02X %02X - %s", cException.aStatusWord[0],
cException.aStatusWord[1], cException.sMessage.toUtf8().constData());
ui->labelStatus->setText(pMessage);
delete pMessage;
}
catch(...)
{
ui->labelStatus->setText("Authentication failed");
}
}
```

And this is update module. I guess like a write into the card.

```
void DialogMifareCardProgramming::on_pushButtonDataUpdate_clicked()
{
bool bNumeric = false;

try
{
ui->lineEditDataBlockNumber->text().toInt(&bNumeric);
if (!bNumeric || ui->lineEditDataBlockNumber->text().toInt() labelStatus->setText("Invalid
block number");
return;
}
```

```
if (cCard_.CardType == ACR_TOPIC_MIFARE_1K)
{
if (ui->lineEditDataBlockNumber->text().toInt() > 63)
{
ui->labelStatus->setText("Card does not have block " + ui->lineEditDataBlockNumber-
>text());
return;
}
}
else
{
if (ui->lineEditDataBlockNumber->text().toInt() > 255)
{
ui->labelStatus->setText("Card does not have block " + ui->lineEditDataBlockNumber-
>text());
return;
}
}

ui->lineEditDataLength->text().toInt(&bNumeric);
if (!bNumeric || ui->lineEditDataLength->text().toInt() lineEditDataLength->text().toInt() %
16) != 0)
{
ui->labelStatus->setText("Invalid data length");
return;
}

if (ui->lineEditDataLength->text().toInt() != ui->textEditData->toPlainText().length())
{
ui->labelStatus->setText("Data length does not match");
return;
}

cCard_.updateBlock(ui->lineEditDataBlockNumber->text().toInt(), ui->lineEditDataLength-
>text().toInt(), ui->textEditData->toPlainText().toAscii().data());

ui->textEditData->setText("");

ui->labelStatus->setText("Update block succeeded");
}
catch(AcsException cException)
{
ui->textEditData->setText("");

char *pMessage = new char[cException.sMessage.length() + 10];
sprintf(pMessage, "%02X %02X - %s", cException.aStatusWord[0],
cException.aStatusWord[1], cException.sMessage.toUtf8().constData());
```

```
ui->labelStatus->setText(pMessage);
delete pMessage;
}
catch(...)
{
ui->textEditData->setText("");

ui->labelStatus->setText("Update block failed");
}
}
```

▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲

Is it possible to tell me what is happening in these modules? I need to order say 1000 Mifare card for the pass through authentication services. So I have to write each card one by one. I don’t want to order card that has data and I have to update the data using above update module.

I will be happy if you can help me to understand the above code.

+ 0 | - 0

Re: Basic Tutorial for MIFARE Classic 1K using C#, C++ etc.

29. January 2016 at 8:57 (https://www.mifare.net/support/forum/topic/basic-tutorial-for-mifare-classic-1k-using-c-c-etc/#post-14738)

(https://www.mifare.net/support/forum/users/mifaresdknxp-com/)

mifaresdk (https://www.mifare.net/support/forum/users/mifaresdknxp-com/)

★★★★★

Hi NTMS,

I do not know the document "FSP_MIFCK1Kv5.03.pdf" and neither the ACR8900 terminal. There are a lot of reader devices and several drivers to operate with the contactless cards. The Qt framework allows it to use any library on Linux where a C header file and a shared library is available. But you will lose the platform independency if the library is not also available for other platforms. As I know Qt does not support Smardcard programming with it's library.

Your code snippet shows me that you use a library which offer a Smardcard interface (

```
if (cCard_._eCardType == CARD_TYPE_MIFARE_1K)
```

). What library do you use in your code?

The method

```
DialogMifareCardProgramming::on_pushButtonAuthenticate_clicked()
```

shows an authenticate with keys from a key store. In general, the Authenticate() is a method where you proof to the card that you are the allowable user and able to read (or write) protected content on the card. The Authenticate() method needs the cards keys and the keys are usually handled in a so called *key store* .

The regular procedure to use MIFARE cards is, to personalize the blank card with customer data and initialize the keys. "Initialize the key" means to overwrite the factory-set default keys with keys created in your office e.g.: from a random number generator. After this personalizing you hand-out the card to the end user.

Now the user operate with the card on a terminal and your software. You know the card key and the Authenticate() is a verification that your software operates only with cards you have personalized before.

Kind regards,
The MIFARE Team

+ 0 | - 0

29. January 2016 at 11:42

Re: Basic Tutorial for MIFARE Classic 1K using C#, C++ etc.

(https://www.mifare.net/support/forum/topic/basic-tutorial-for-mifare-classic-1k-using-c-c-etc/#post-14766)

(https://www.mifare.net/support/forum/users/niyazi-torosgmail-com/)

NTMS (https://www.mifare.net/support/forum/users/niyazi-torosgmail-com/)

★★★★★

Hi mifaresdk,

I have no idea which library they used. I have a mifare.h and mifare.cpp in the sample project. But I couldn't find any mifare.so library file in the project.

Here is the header file.

```
#ifndef MIFARECLASSIC_H
#define MIFARECLASSIC_H

#include "AcsIncludes.h"

#define MAXIMUM_VALUE ************

enum MIFARE_KEY_TYPE {
MIFARE_KEY_TYPE_A = 0x60,
MIFARE_KEY_TYPE_B = 0x61
};

enum MIFARE_KEY_STORE {
MIFARE_KEY_STORE_0 = 0x00,
MIFARE_KEY_STORE_1 = 0x01
};

enum CARD_TYPE {
CARD_TYPE_UNKOWN = 0x00,
CARD_TYPE_MIFARE_1K= 0x01,
CARD_TYPE_MIFARE_4K = 0x02
};

class MifareClassic
{
public:
```

```
MifareClassic();
CARD_TYPE _eCardType;

void openReader();
void closeReader();
void connect();
void disconnect();
void loadKey(char *pKey, MIFARE_KEY_STORE eKeyStore);
void authenticate(MIFARE_KEY_TYPE eKeyType, uint8_t uBlockNumber,
MIFARE_KEY_STORE eKeyStore);
void readBlock(uint8_t uBlockNumber, uint8_t uLength, char *pData);
void updateBlock(uint8_t uBlockNumber, uint8_t uLength, char *pData);
void storeValue(uint8_t uBlockNumber, uint64_t uValue);
void incrementValue(uint8_t uBlockNumber, uint64_t uValue);
void decrementValue(uint8_t uBlockNumber, uint64_t uValue);
void readValue(uint8_t uBlockNumber, uint64_t *pValue);
void restoreValue(uint8_t uSourceBlock, uint8_t uTargetBlock);

private:

};

#endif // MIFARECLASSIC_H
```

+ 0 | - 0

Re: Basic Tutorial for MIFARE Classic 1K using C#, C++ etc.

29. January 2016 at 17:04 (https://www.mifare.net/support/forum/topic/basic-tutorial-for-mifare-classic-1k-using-c-c-etc/#post-14771)

(https://www.mifare.net/support/forum/users/mifaresdknxp-com/)

mifaresdk (https://www.mifare.net/support/forum/users/mifaresdknxp-com/)

★★★★★

Hi NTMS,

it seems you are using a proprietary class library. Let us assume you have a blank MIFARE Classic card. On a MIFARE Classic you have sectors with 4 blocks. Each block has the capacity of 16 bytes. The last block in a sector is the sector trailer and contains the keys. So you can use only the first 3 blocks for saving your data.

For the details, please visit: http://www.nxp.com/documents/data_sheet/MF1S50YYX_V1.pdf (http://www.nxp.com/documents/data_sheet/MF1S50YYX_V1.pdf)

I do not know how the key store is implemented in your library but the scheme for interacting with the card in general could look like this:

```
char *pDefaultKey = "FFFFFFFFFFFF";

char *pDataToWrite = "00112233445566778899AABBCCDDEEFF"
    ▲       ▲       ▲       ▲       ▲       ▲       ▲       ▲       ▲       ▲
char readBuffer[16];



openReader();    // Open the reader interface

connect();       // Connect to the card, if no card is detected, an error
 is returned!



authenticate( MIFARE_KEY_TYPE_A, 4, ... ); // Authenticate to sector 1 (bl
ock 4) with key A

updateBlock( 4, sizeof( pDataToWrite ), pDataToWrite ); // Write data to b
lock 4

readBlock( 4, sizeof( readBuffer ), readBuffer );      // Read data from b
lock 4



authenticate( MIFARE_KEY_TYPE_A, 8, ... ); // Authenticate to sector 2 (bl
ock 8) with key A

readBlock( 8, sizeof( readBuffer ), readBuffer );      // Read data from b
lock 8



disconnect();    // Close connection

closeReader();   // Close interface
```

The variable

```
pDefaultKey
```

contains the factory key for a blank card. You have to authenticate to each sector you want to have access. But you can change the access conditions in the sector trailer – and also the keys.

Your library deals with key over the key store. So you have to load the default key into the key store and use the key store reference for the authentication method.

The MIFARE Team

+ 0 | - 0

▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲

Re: Basic Tutorial for MIFARE Classic 1K using C#, C++ etc.

1. February 2016 at 8:00 (https://www.mifare.net/support/forum/topic/basic-tutorial-for-mifare-classic-1k-using-c-c-etc/#post-14774)

(https://www.mifare.net/support/forum/users/niyazi-torosgmail-com/)

NTMS (https://www.mifare.net/support/forum/users/niyazi-torosgmail-com/)

★★★★★

Hi mifaresdk,

Now I am getting somewhere. I will read the "proprietary class library" again. I will be post here the basic sample code later.
Your last post most helpful and clear to me.

Thank you very much.

Kind Regards,

+ 0 | - 0

Viewing 7 posts - 1 through 7 (of 7 total)

You must be logged in to reply to this topic.

Products

MIFARE products have an excellent reputation as the leading contactless solution

- MIFARE ICs (/products/chip-card-ics)
- MIFARE 2GO (/products/mobile-solutions/mifare-2go)
- TapLinx SDK (/products/tools/taplinx)
- Reader-Writer Kit for MIFARE ICs (https://www.mifare.net/en/products/tools/reader-kit/)
- NFC Reader ICs for MIFARE Products (https://www.mifare.net/products/infrastructure-components/mifare-nfc-reader-ics/)
- MIFARE4Mobile (https://www.mifare.net/en/products/mobile-solutions/mifare4mobile-2/)
- MIFARE SAM (https://www.mifare.net/products/infrastructure-components/mifare-sam/)

Applications

MIFARE products can be used in more than 40 different applications

- Smart Mobility (https://www.mifare.net/solutions/mifare-applications/smart-mobility/)
- Access (https://www.mifare.net/solutions/mifare-applications/access/)
- Loyalty (https://www.mifare.net/solutions/mifare-applications/loyalty/)
- Micropayment (https://www.mifare.net/solutions/mifare-applications/micropayment/)
- Electronic Identity (https://www.mifare.net/solutions/mifare-applications/electronic-identity/)
- Smart Media/Consumer Interaction (https://www.mifare.net/solutions/mifare-applications/smart-media-consumer-interaction)
- Gaming (https://www.mifare.net/solutions/mifare-applications/gaming/)

Companies

Discover the latest updates from our MIFARE Partners

- CITA SOLUTIONS MIDDL… (https://www.mifare.net/company/cita-solutions-middle-east/)
- Tachy Limited (https://www.mifare.net/company/tachy-limited/)
- TAO Digital (https://www.mifare.net/company/tao-software/)
- YouCard Kartensystem… (https://www.mifare.net/company/youcard-kartensysteme-gmbh-4/)
- Impro Technologies (https://www.mifare.net/company/impro-technologies/)
- Clyde Technologies (https://www.mifare.net/company/clyde-technologies/)
- Uniware Systems Ltd (https://www.mifare.net/company/uniware-systems-ltd/)

More

Find useful information about the MIFARE brand and our contact details