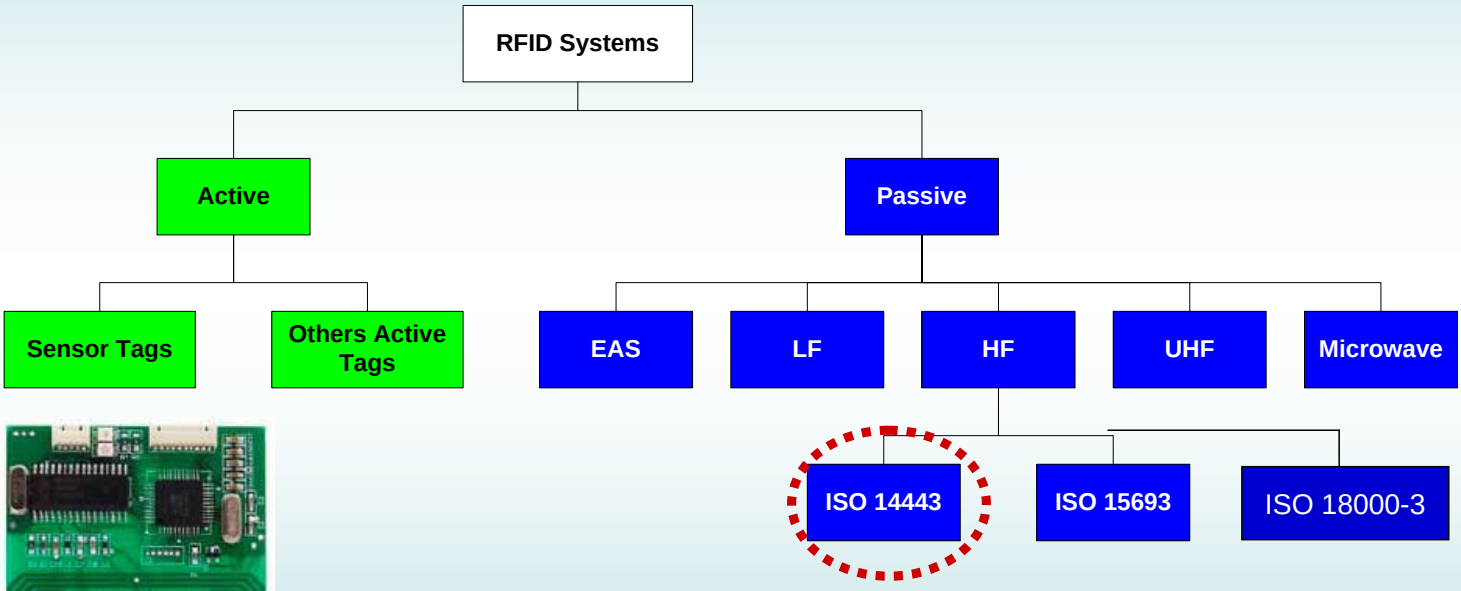


MIFARE ISO/IEC 14443A(NXP)

หน่วยวิจัยเทคโนโลยีอาร์เอฟไอดี โปรแกรมวิชาวิศวกรรมและเทคโนโลยี
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

RFID Reader

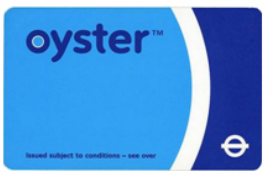
ISO14443A	Mifare	NXP
ISO14443B	CryptoRF	Motorola/Atmel
ISO14443C	Felica	Sony
ISO14443D	-	OTI
ISO14443E	-	Cubic
ISO14443F	LEGIC	KABA
ISO15693	Tag-IT	Texas Instruments

Mifare Classic

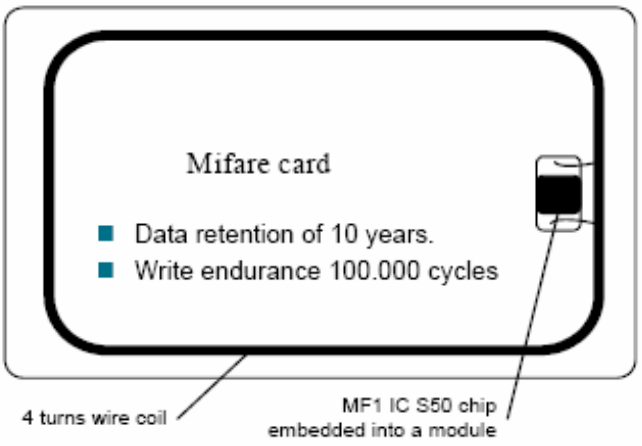
- Many chips in the Mifare (ISO14443A) family
  - Mifare Ultralight
  - Mifare Classic
  - Mifare DESFire
  - Mifare Plus
  - Mifare EV1
  - Mifare SMART MX
- Most popular: Mifare Classic
  - over 1 billion sold
  - over 200 million in use
  - 80% of contactless smartcard market

Mifare Classic Applications

- Public transport ticketing systems
- Access control
- Wireless payment systems

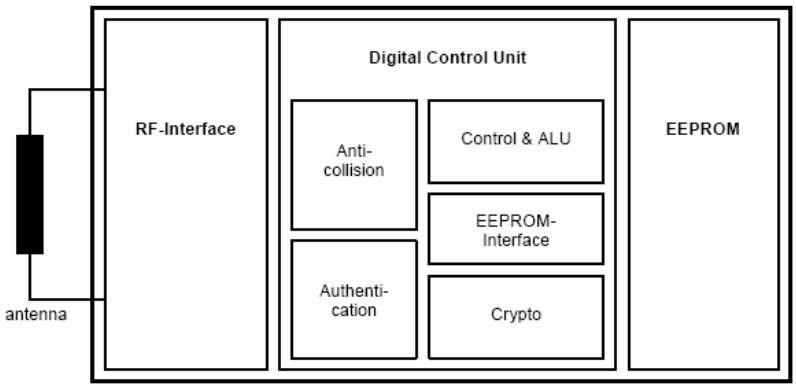


Memory organization



Mifare card reader

Block diagram



Memory organization

Sector	Block	Byte Number within a Block 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Description
15	3	Key A | Access Bits | Key B	Sector Trailer 15
	2		Data
	1		Data
	0		Data
14	3	Key A | Access Bits | Key B	Sector Trailer 14
	2		Data
	1		Data
	0		Data
:	:		
:	:		
:	:		
1	3	Key A | Access Bits | Key B	Sector Trailer 1
	2		Data
	1		Data
	0		Data
0	3	Key A | Access Bits | Key B	Sector Trailer 0
	2		Data
	1		Data
	0		Manufacturer Block

แบ่งเป็น 16 sector

แต่ละ sector มี 4 block

ในแต่ละ block มี 16 byte

Block สุดท้ายของแต่ละ sector เรียกว่า Sector Tailor เป็นข้อมูลของ key และเงื่อนไขการเข้าถึงข้อมูลภายใน sector นั้นๆ

Memory organization

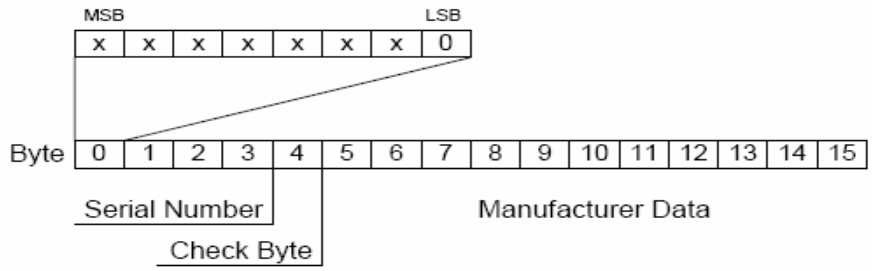


Fig 5. Manufacturer block

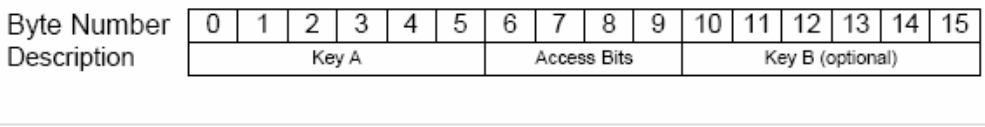


Fig 7. Sector trailer

หน่วยวิจัยเทคโนโลยีอาร์เอฟไอดี โปรแกรมวิชาวิศวกรรมและเทคโนโลยี
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

Memory organization

Data blocks

Value Blocks

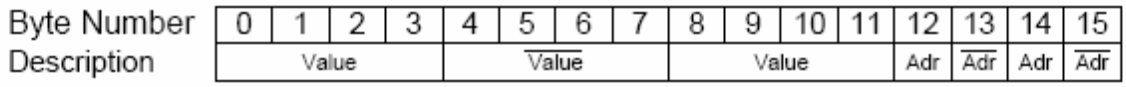


Fig 6. Value blocks

Memory organization

Data blocks

Value Blocks

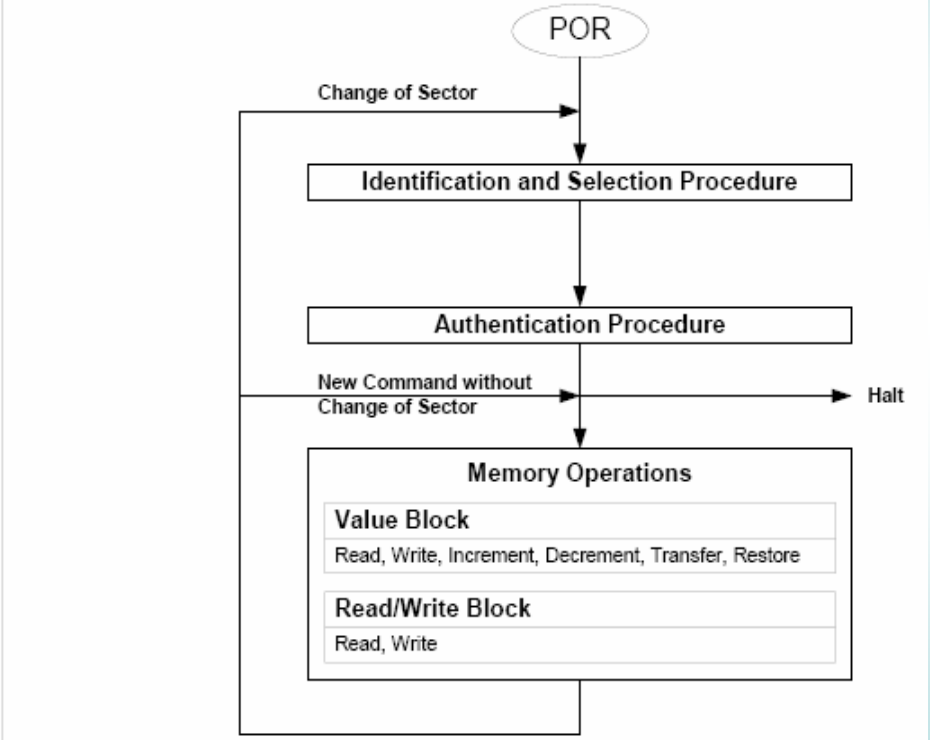


Fig 8. Memory access

RFID Reader

Mifare

- Tag supported: Mifare 1K, Mifare 4K, Mifare UltraLight
- Auto detecting tag
- Integrated antenna
- RS232 interface, baud rate 9,600 ~ 115,200 bps
- DC4.5V to DC5.5V VDD operating
- Operating distance: Up to 80mm, depending on tag
- Storage temperature: -40 ºC ~ +85 ºC
- Operating temperature: -20 ºC ~ +70 ºC
- Dimension: 85 × 55 × 7 mm
- Two LEDs, green led is auto light when tag in detection range, red led is controlled by host
- The PA1 pin at low level indicates tag in detective range, and high level indicating tag out

RFID Reader

Mifare

4-1. Communication Setting

The communication protocol is byte oriented. Both sending and receiving bytes are in hexadecimal format. The communication parameters are as follows,

Baud rate:	9,600 ~ 115,200 bps
Data:	8 bits
Stop:	1 bit
Parity:	None
Flow control:	None

4-2. Communication Format

Host to Reader:

Preamble	Len	Command	Data	Checksum

Preamble:	1 byte, 0xBA.
Len:	Byte length counting from Command to Checksum inclusively, 1 byte.
Command:	Command, 1 byte.
Data:	Data, variable length depends on the command type.
Checksum:	XOR result from Preamble to Data inclusively, 1 byte.

Reader to Host:

Preamble	Len	Command	Status	Data	Checksum

Preamble:	1 byte, 0xBD.
Len:	Byte length counting from Command to Checksum inclusively, 1 byte.
Command:	Command, 1 byte.
Status:	Command status, 1 byte
Data:	Data, variable length depends on the command type.
Checksum:	XOR result from Preamble to Data inclusively, 1 byte.

RFID Reader

4-3. Command Overview

Command		Description
0x01	2	Select Mifare card
0x02	3	Login to a sector
0x03	4	Read a data block
0x04	5	Write a data block
0x05	7	Read a value block
0x06	6	Initialize a value block
0x07	11	Write master key (key A)
0x08	8	Increment value
0x09	9	Decrement value
0x0A	10	Copy value
0x10	X	Read a data page (UltraLight)
0x11	X	Write a data page (UltraLight)
0x40	X	Control PA status
0xFF	1	Reset

Status Overview

Status	Description
0x00	Operation succeed
0x01	No tag
0x02	Login succeed
0x03	Login fail
0x04	Read fail
0x05	Write fail
0x06	Unable to read after write
0x0A	Collision occur
0x0D	Not authenticate
0x0E	Not a value block
0xF0	Checksum error
0xF1	Command code error

4-4-14. Reset

0xBA	Len	0xFF	Checksum

No return

```
#include <dos.h>        #include <bios.h>       #include <conio.h>
#include <iostream.h>   #include <time.h>
void reset();
#define COM1        ....//Identifies the I/O port;  0 = COM1, 1 = COM2, etc.
#define DATA_READY 0x100
#define SETTINGS (...|...|...|...)  //  Sets the communications parameters
void main()
{   char key;
    bioscom(0, SETTINGS, COM1);
    clrscr();
    do
    {   reset();
        cout<<"\n\n\tExit Program Key ESC"<<endl;
        cout<<"\n\t===================="<<endl;
        key=getch();
    }while(key!=27);// ESC = Char(27)
}
```

การทดลองที่ 15 :\tc\bin\mifare\iso14443\reset.cpp


```
void reset()
{       time_t timef, timel;
        float difft;
        int status;
        int rf_check_sum,i;
        int rf_reset_cmd[3]={.....,.....,....};
        rf_check_sum=0;
        cout<<"\n\tSend Command Reset Reader :=";
        for(i=0;i<3;i++)
        { bioscom(1, rf_reset_cmd[i], COM1);
          rf_check_sum^=rf_reset_cmd[i];
          cout<<rf_reset_cmd[i];
          cout<<',';
        }
        bioscom(1, rf_check_sum, COM1);
        cout<<rf_check_sum;
        timef = time(NULL);
        do
        { status = bioscom(3, 0, COM1);
          if (status & DATA_READY)
              { i = bioscom(2, 0, COM1);  }
          timel = time(NULL);
          difft=difftime(timel,timef);
        }while(!kbhit()&&difft<=1);
        cout<<""<<endl;
        cout<<"\n\tExit Command Reset ";
        delay(1000);
}
```

4-4-14. Reset

0xBA	Len	0xFF	Checksum

No return

การทดลองที่ 15 :\tc\bin\mifare\iso14443\reset.cpp

4-4-1. Select Mifare card

0xBA	Len	0x01	Checksum

Return:

0xBD	Len	0x01	Status	UID	Type	Checksum

Status: 0x00: Operation succeed
0x01: No tag
0x0A: Collision occur
0xF0: Checksum error

UID: The uniquely serial number of Mifare card,
4 bytes for Mifare 1k & Mifare 4k, 7 bytes for UltraLight & DesFire

Type: 0x01: Mifare Standard 1K card
0x02: Mifare Pro card
0x03: Mifare UltraLight card
0x04: Mifare Standard 4K card
0x05: Mifare ProX card
0x06: Mifare DesFire card

การทดลองที่ 16 :\tc\bin\mifare\iso14443\check1.cpp


```
#include <dos.h>          #include <bios.h>        #include <conio.h>
#include <iostream.h>     #include <time.h>
void reset();
#define COM1  0           #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
void main()
{  time_t timef, timel;
   float difft;           int len,i,status,index_data;
   int rf_selectcard_cmd[3]={...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();
   reset();
   do
   {  clrscr();
      rf_check_sum=0;
      cout<<"\n\tSend Command  select Mifare card information\n\t";
      for(i=0;i<3;i++)
      { bioscom(1, rf_selectcard_cmd[i], COM1);
        rf_check_sum^=rf_selectcard_cmd[i];
        cout<<rf_selectcard_cmd[i];
        cout<<',';
      }
      bioscom(1, rf_check_sum, COM1);
      cout<<rf_check_sum;
      cout<<""<<endl;
```

4-4-1. Select Mifare card

0xBA	Len	0x01	Checksum

การทดลองที่ 16 :\tc\bin\mifare\iso14443\check1.cpp

```
30        index_data=0;     len = 100;
31        do
32        {  status = bioscom(3, 0, COM1);
33           if (status & DATA_READY)
34              {   rf_return_cmd = bioscom(2, 0, COM1);
35                  rf_data[index_data]=rf_return_cmd;
36                  index_data++;
37              }
38           if (index_data==2) { len=rf_return_cmd+...; }
39        }while(!kbhit()&&index_data!=len);
40        cout<<"\n\tReturn Command select Mifare card information \n\t";
41        for(i=0;i<index_data;i++)
42        { if(i>0) { cout<<","; }
43          cout<<rf_data[i];
44        }
45        cout<<""<<endl;
```

Return:

0xBD	Len	0x01	Status	UID	Type	Checksum

Reader to Host:

Preamble	Len	Command	Status	Data	Checksum

Preamble: 1 byte, 0xBD.
Len: Byte length counting from Command to Checksum inclusively, 1 byte.

การทดลองที่ 16 :\tc\bin\mifare\iso14443\check1.cpp


```
50        if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
51          {   cout<<"\n\tselect Mifare card information success"<<endl;
52              cout<<"\n\tUID "<<rf_data[4]<<":"<<rf_data[5]<<":"<<rf_data[6]<<":"<<rf_data[7]<<endl;
53              switch(rf_data[...])
54              {   case ....: cout<<"\n\tMifare Standard 1K card"<<endl; break;
55                  case 0x02: cout<<"\n\tMifare Pro card"<<endl; break;
56                  case 0x03: cout<<"\n\tMifare UltraLight card"<<endl; break;
57                  case 0x04: cout<<"\n\tMifare Standard 4K card"<<endl;break;
58                  case 0x05: cout<<"\n\tMifare ProX card"<<endl; break;
59                  case 0x06: cout<<"\n\tMifare DesFire card"<<endl; break;
60              }
61          }
62        else
63        {     cout<<"\n\tselect Mifare card information Fail"<<endl;
64              switch(rf_data[...])
65              {   case ....: cout<<"\n\tNo Tag"<<endl; break;
66                  case 0x0A: cout<<"\n\tCollision occur"<<endl; break;
67                  case 0xF0: cout<<"\n\tChecksum error"<<endl; break;
68              }
69        }
70        cout<<"\n\tExit Program Key ESC"<<endl;
71        key=getch();
72      }while(key!=27);
73  }
```

Status: 0x00: Operation succeed
0x01: No tag
0x0A: Collision occur
0xF0: Checksum error

Return:

0xBD	Len	0x01	Status	UID	Type	Checksum

```
76  void reset()
77  {
78
79  }
```

UID: The uniquely serial number of Mifare card,
4 bytes for Mifare 1k & Mifare 4k, 7 bytes for UltraLight & DesFire

การทดลองที่ 16 :\tc\bin\mifare\iso14443\check1.cpp

```
#include <dos.h>          #include <bios.h>          #include <conio.h>
#include <iostream.h>     #include <time.h>          #include <string.h>
void reset();             void select_card();
#define COM1 0            #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
char sn_tag_char[20]={" "}; // void select_card();
void main()
{  char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();
   reset();
   do
   {  select_card();
      cout<<"\n\tExit Program Key ESC"<<endl;
      key=getch();
    }while(key!=27);
}
```

การทดลองที่ 17 :\tc\bin\mifare\iso14443\check2.cpp


```
void select_card()
{  int len,loop,i,status,index_data;
   int rf_selectcard_cmd[3]={...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   rf_check_sum=0;
      for(i=0;i<3;i++)
      { bioscom(1, rf_selectcard_cmd[i], COM1); delay(10);
        rf_check_sum^=rf_selectcard_cmd[i];
      }
      bioscom(1, rf_check_sum, COM1);
      index_data=0;
      len=100;
      do
      { status = bioscom(3, 0, COM1);
        if (status & DATA_READY)
            {   rf_return_cmd = bioscom(2, 0, COM1);
                rf_data[index_data]=rf_return_cmd;
                index_data++;
                if (index_data==2) {len=rf_return_cmd+...;}
            }
      }while(!kbhit()&&index_data!=len);
      if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
        {   int j=0;
            strcpy(sn_tag_char," ");
            sn_tag_char[j++]='|';
            for(i=4;i<8;i++) {  sn_tag_char[j++]=...+(rf_data[i]/...);
                                sn_tag_char[j++]=...+((rf_data[i]%...)/...);
                                sn_tag_char[j++]=...+(rf_data[i]%...);
                                sn_tag_char[j++]='|';
                             }
            cout<<"\n\n\t\tSelect Mifare card succeed UID Card :"<<sn_tag_char<<endl;
        }
      else
      { cout<<"\n\n\t\tSelect Mifare card Fail"<<endl;  }
}
```

4-4-1. Select Mifare card

0xBA	Len	0x01	Checksum

Return:

0xBD	Len	0x01	Status	UID	Type	Checksum

```
void reset()
{
}
```

การทดลองที่ 17 :\tc\bin\mifare\iso14443\check2.cpp

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector: Sector need to login
Type: Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key: Password, 6 bytes

Return:

0xBD	Len	0x02	Status	Checksum

Status: 0x02: Login succeed
0x01: No tag
0x03: Login fail
0xF0: Checksum error

การทดลองที่ 18 :\tc\bin\mifare\iso14443\logic_s2.cpp


```
#include <bios.h>        #include <conio.h>       #include <iostream.h>
#include <time.h>        #include <dos.h>         void reset();
#define COM1 0           #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
void main()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();              reset();
   do
   {  clrscr();
      rf_check_sum=0;
      cout<<"\Send Command Login to a sector:"<<rf_login_cmd[3]<<" = ";
      for(i=0;i<11;i++)
      { bioscom(1,rf_login_cmd[i], COM1);
        rf_check_sum^=rf_login_cmd[i];
        cout<<rf_login_cmd[i];
        cout<<',';
      }
      bioscom(1, rf_check_sum, COM1);
      cout<<rf_check_sum<<endl;
```

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

การทดลองที่ 18 :\tc\bin\mifare\iso14443\logic_s2.cpp

```
28        index_data=0;
29        len=100;
30        do
31        {  status = bioscom(3, 0, COM1);
32          if (status & DATA_READY)
33          {   rf_return_cmd = bioscom(2, 0, COM1);
34              rf_data[index_data]=rf_return_cmd;
35              index_data++;
36              if (index_data==2) {len=rf_return_cmd+2;}
37          }
38        }while(!kbhit()&&index_data!=len);
39        cout<<"\nReturn Command Login to a sector:"<<rf_login_cmd[3]<<" = ";
40        for(i=0;i<index_data;i++)
41          {   if(i>0) {  cout<<",";   }
42              cout<< rf_data[i];
43          }
44        cout<<""<<endl;
45        if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
46          {   cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;  }
47        else
48          {   cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
49              switch(rf_data[...])
50                  {   case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
51                      case 0x03  :cout<<"\n\t\tLogin fail"<<endl;break;
52                      case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
53                  }
54          }
55        cout<<"\n\n\nExit Program Key ESC"<<endl;
56        key=getch();
57    }while(key!=27);
58  }
```

Return:

0xBD	Len	0x02	Status	Checksum

Status:	0x02:	Login succeed
	0x01:	No tag
	0x03:	Login fail
	0xF0:	Checksum error

117

```
60  void reset()
61  {
62  }
```

การทดลองที่ 18 :\tc\bin\mifare\iso14443\logic_s2.cpp

118

```
1  #include <bios.h>        #include <conio.h>      #include <iostream.h>
2  #include <time.h>        #include <dos.h>        void reset();
3  void login_sector();
4  #define COM1 0           #define DATA_READY 0x100
5  #define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
6  int login_state; // void login_sector();
7  void main()
8  {  char key;
9     bioscom(0, SETTINGS, COM1);
10    clrscr();            reset();
11    do
12    {  login_sector();
13       if(login_state==...)
14             {   cout<<"\n\n\t\t===login succeed==="; sound(1000);delay(200);nosound();  }
15       else  { cout<<"\n\n\t\t===login Fail==="; sound(500);delay(200);nosound();}
16
17       cout<<"\n\n\t\tExit Program Key ESC"<<endl;
18       key=getch();
19    }while(key!=27);
20 }
```

การทดลองที่ 19 :\tc\bin\mifare\iso14443\logic_s3.cpp

```
void login_sector()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[30];
   char key;
      rf_check_sum=0;
      for(i=0;i<11;i++)
      { bioscom(1, rf_login_cmd[i], COM1);
        rf_check_sum^=rf_login_cmd[i];
      }
      bioscom(1, rf_check_sum, COM1);
      index_data=0;
      len = 100;
      do
      { status = bioscom(3, 0, COM1);
        if (status & DATA_READY)
            {   rf_return_cmd = bioscom(2, 0, COM1);
                rf_data[index_data]=rf_return_cmd;
                index_data++;
                if (index_data==2) {len=rf_return_cmd+2;}
            }
      }while(!kbhit()&&index_data!=len);
      if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
      { cout<<"\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
        login_state=1;
      }
      else
      { cout<<"\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
        login_state=0;
      }
}
```

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Return:

0xBD	Len	0x02	Status	Checksum

Status: 0x02: Login succeed
0x01: No tag
0x03: Login fail
0xF0: Checksum error

```
void reset()
{
}
```

การทดลองที่ 19 :\tc\bin\mifare\iso14443\logic_s3.cpp

4-4-3. Read a data block

0xBA	Len	0x03	Block	Checksum

Block: The block number to be read, 1 byte

Return:

0xBD	Len	0x03	Status	Data	Checksum

Status: 0x00: Operation succeed
0x01: No tag
0x04: Read fail
0x0D: Not authenticate
0xF0: Checksum error

Data: Block data returned if operation succeeds, 16 bytes.

```
#include <stdio.h>      #include <bios.h>      #include <conio.h>
#include <iostream.h>   #include <time.h>      #include <dos.h>
#include <string.h>
void login_sector();    void reset();
#define COM1 0          #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int login_state;
void main()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_read_cmd[4]={...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   char data_in_block[30]={" "};
   bioscom(0, SETTINGS, COM1);
   clrscr();
   reset();
```

การทดลองที่ 20 :\tc\bin\mifare\iso14443\read_d3.cpp

```
21      do
22      {  login_state=0;
23         login_sector();
24         if(login_state==1)
25           {    cout<<"\n\nSend Command Read a data block:"<<rf_read_cmd[3]<<"  ";
26                rf_check_sum=0;
27                for(i=0;i<4;i++)
28                     {    bioscom(1,rf_read_cmd[i],COM1);
29                          rf_check_sum^=rf_read_cmd[i];
30                          cout<<rf_read_cmd[i];
31                          cout<<',';
32                     }
33                bioscom(1, rf_check_sum, COM1);
34                cout<<rf_check_sum<<endl;
35                index_data=0;
36                len=100;
37                do   {    status = bioscom(3,0,COM1);
38                          if (status & DATA_READY)
39                               {    rf_return_cmd = bioscom(2,0,COM1);
40                                    rf_data[index_data]=rf_return_cmd;
41                                    index_data++;
42                                    if (index_data==2) {len=rf_return_cmd+2;     }
43                               }
44                     }while(!kbhit()&&index_data!=len);
```

การทดลองที่ 20 :\tc\bin\mifare\iso14443\read_d3.cpp

```
47              cout<<"\nReturn Command Read a data block:"<<rf_read_cmd[3]<<"\n\t";
48              for(i=0;i<index_data;i++)
49                   {    if(i>0)
50                             { cout<<',';     }
51                        cout<<rf_data[i];
52                   }
53              cout<<""<<endl;
54              if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
55                   {    cout<<"\n\tRead a data block:"<<rf_read_cmd[3]<<"   = Success"<<endl;
56                        int k=0;
57                        for(i=4;i<20;i++) { data_in_block[k]=rf_data[i];     k++;     }
58                        cout<<"\n\tData In Block:"<<rf_read_cmd[3]<<"==="<<data_in_block<<"==={Char}";
59                   }
60              else
61                   {    cout<<"\n\tRead a data block:"<<rf_read_cmd[3]<<"   = Fail"<<endl;
62                        switch(rf_data[...])
63                             {    case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
64                                  case 0x04  :cout<<"\n\t\tRead fail"<<endl;break;
65                                  case 0x0D  :cout<<"\n\t\tNot authenticate"<<endl;break;
66                                  case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
67                             }
68                   }
69         }
70        cout<<"\n\n\nExit Program Key ESC"<<endl;
71        key=getch();
72      }while(key!=27);
73 }
```

Return:

0xBD	Len	0x03	Status	Data	Checksum

Status:
- 0x00: Operation succeed
- 0x01: No tag
- 0x04: Read fail
- 0x0D: Not authenticate
- 0xF0: Checksum error

Data: Block data returned if operation succeeds, 16 bytes.

การทดลองที่ 20 :\tc\bin\mifare\iso14443\read_d3.cpp

```
77  void login_sector()
78  {   time_t timef, time1;
79      float difft;
80      int len,i,status,index_data;
81      int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
82      int rf_check_sum,rf_return_cmd,rf_data[30];
83      char key;
84      bioscom(0, SETTINGS, COM1);
85      clrscr();
86         rf_check_sum=0;
87         cout<<"\Send Command Login to a sector:"<<rf_login_cmd[3]<<" = ";
88         for(i=0;i<11;i++)
89         { bioscom(1, rf_login_cmd[i], COM1);
90           rf_check_sum^=rf_login_cmd[i];
91           cout<<rf_login_cmd[i];
92           cout<<',';
93         }
94         bioscom(1, rf_check_sum, COM1);
95         cout<<rf_check_sum<<endl;
```

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

การทดลองที่ 20 :\tc\bin\mifare\iso14443\read_d3.cpp

```
98         index_data=0;
99         len = 100;
100        do
101        { status = bioscom(3, 0, COM1);
102          if (status & DATA_READY)
103              {   rf_return_cmd = bioscom(2, 0, COM1);
104                  rf_data[index_data]=rf_return_cmd;
105                  index_data++;
106                  if (index_data==2) {len=rf_return_cmd+2;}
107              }
108        }while(!kbhit()&&index_data!=len);
109        cout<<"\Return Command Login to a sector:"<<rf_login_cmd[3]<<" = ";
110        for(i=0;i<index_data;i++)
111          {   if(i>0) {  cout<<",";   }
112              cout<< rf_data[i];
113          }
114        cout<<""<<endl;
115        if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
116          {   cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
117              login_state=1;
118          }
119        else
120          {   cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
121              login_state=0;
122              switch(rf_data[3])
123                  {   case 1  :cout<<"\n\t\tNo Tag"<<endl;break;
124                      case 3  :cout<<"\n\t\tLogin fail"<<endl;break;
125                      case 240:cout<<"\n\t\tCard Type Ultralight Card"<<endl;break;
126                  }
127          }
128        cout<<"\nExit Login sector\n\n"<<endl;
129  }
```

Return:

0xBD	Len	0x02	Status	Checksum

Status:	0x02:	Login succeed
	0x01:	No tag
	0x03:	Login fail
	0xF0:	Checksum error

```
132 void reset()
133 {
134
135
136 }
```


การทดลองที่ 20 :\tc\bin\mifare\iso14443\read_d3.cpp

```
#include <stdio.h>      #include <bios.h>      #include <conio.h>
#include <iostream.h>   #include <time.h>      #include <dos.h>
#include <string.h>
void read_expire_date();void login_sector();    void reset();
#define COM1 0          #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int login_state;
int expire_date_check; // void read_expire_date();
char data_in_block[10]={" "};// void read_expire_date();
void main()
{   char key;
    clrscr();
    reset();
    do
    {     login_sector();
          read_expire_date();
          cout<<"\n\t\tExpire_date_check ="<<expire_date_check;
          cout<<"\n\n\t\tExit Program Key ESC"<<endl;
          key=getch();
    }while(key!=27);
}
```


การทดลองที่ 21 :\tc\bin\mifare\iso14443\read_d4.cpp

การทดลองที่ 21 :\tc\bin\mifare\iso14443\read_d4.cpp

```
void read_expire_date()
{  int DD,MM,YY;        int len,i,status,index_data;
   int rf_read_cmd[4]={...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
      rf_check_sum=0;
      for(i=0;i<4;i++)
      {  bioscom(1,rf_read_cmd[i],COM1);
         rf_check_sum^=rf_read_cmd[i];
      }
      bioscom(1, rf_check_sum, COM1);
      index_data=0;     len=100;
      do
      {  status = bioscom(3,0,COM1);
         if (status & DATA_READY)
         {  rf_return_cmd = bioscom(2,0,COM1);
            rf_data[index_data]=rf_return_cmd;
            index_data++;
            if (index_data==2) {len=rf_return_cmd+2;}
         }
      }while(!kbhit()&&index_data!=len);
      if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
      { cout<<"\n\t\tRead a data block:"<<rf_read_cmd[3]<<"  = Success ";
        int k=0;
        for(i=4;i<12;i++)   {data_in_block[k]=rf_data[i];
                             k++;
                             cout<<rf_data[i];
                            }
        cout<<"\n\t\tData In Block:"<<rf_read_cmd[3]<<"==="<<data_in_block<<"==={Char}";
        DD = ((rf_data[...]-48)*10)+(rf_data[...]-48);
        MM = ((rf_data[...]-48)*10)+(rf_data[...]-48);
        YY = ( ((rf_data[...]-48)*1000)+((rf_data[...]-48)*100)
              +((rf_data[...]-48)*10) +(rf_data[...]-48) );
       cout<<"\n\t\tExpire_date_check:"<<rf_read_cmd[3]<<"   DD:"<<DD<<" MM:"<<MM<<" YY:"<<YY;
```

```
        struct date d;
        getdate(&d);
            if (YY>...)
          {
           expire_date_check=1;
          }
      else if(YY==...)
          {
             if (MM>...)
                {
                expire_date_check=1;
                }
            else if (MM==...)
                    {
                     if (DD>=...) expire_date_check=1;
                     else expire_date_check=0;
                    }
               else expire_date_check=0;
           }
       else  expire_date_check=0;
       }
       else
       {
          cout<<"\n\t\tRead a data block:"<<rf_read_cmd[3]<<"   = Fail"<<endl;
          expire_date_check=0;
       }
}
```

getdate example

```
#include <dos.h>
#include <stdio.h>

int main(void)
{
   struct date d;

   getdate(&d);
   printf("The current year is: %d\n", d.da_year);
   printf("The current day is: %d\n", d.da_day);
   printf("The current month is: %d\n", d.da_mon);
   return 0;
}
```

การทดลองที่ 21 :\tc\bin\mifare\iso14443\read_d4.cpp


```
 88 void login_sector()
 89 {
 90    time_t timef, timel;
 91    float difft;
 92    int len,i,status,index_data;
 93    int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
 94    int rf_check_sum,rf_return_cmd,rf_data[30];
 95    char key;
 96       if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
 97       {  cout<<"\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
 98          login_state=1;
 99       }
100       else
101       {  cout<<"\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
102          login_state=0;
103       }
104 }
105
106
107 void reset()
108 {
109
110 }
111
```

การทดลองที่ 21 :\tc\bin\mifare\iso14443\read_d4.cpp

4-4-4. Write a data block

0xBA	Len	0x04	Block	Data	Checksum

Block: The block number to be written, 1 byte.
Data: The data to write, 16 bytes.

Return:

0xBD	Len	0x04	Status	Data	Checksum

Status:
- 0x00: Operation succeed
- 0x01: No tag
- 0x05: Write fail
- 0x06: Unable to read after write
- 0x0D: Not authenticate
- 0xF0: Checksum error

Data: Block data written if operation succeeds, 16 bytes.

```
1  #include <stdio.h>      #include <bios.h>      #include <conio.h>
2  #include <iostream.h>   #include <time.h>      #include <dos.h>
3  void login_sector();    void reset();
4  #define COM1 0
5  #define DATA_READY 0x100
6  #define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
7  int login_state;
```

การทดลองที่ 22 :\tc\bin\mifare\iso14443\write_d4.cpp


```
9  void main()
10 {  time_t timef, timel;
11    float difft;
12    int len,i,status,index_data;
13    int rf_write_cmd[20]={..,..,..,..,' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}
14    int rf_check_sum,rf_return_cmd,rf_data[21];
15    char key;
16    bioscom(0, SETTINGS, COM1);
17    clrscr();
18    reset();
19    do
20    {  login_sector();
21       if(login_state==1)
22       { cout<<"\nSend Command Write a data block:="<<rf_write_cmd[3]<<"  ";
23         rf_check_sum=0;
24         for(i=0;i<20;i++)
25             {   bioscom(1, rf_write_cmd[i], COM1);
26                 rf_check_sum^=rf_write_cmd[i];
27                 cout<<rf_write_cmd[i];
28                 cout<<',';
29             }
30         bioscom(1, rf_check_sum, COM1);
31         cout<<rf_check_sum<<endl;
```

4-4-4. Write a data block

0xBA	Len	0x04	Block	Data	Checksum

Block: The block number to be written, 1 byte.
Data: The data to write, 16 bytes.

การทดลองที่ 22 :\tc\bin\mifare\iso14443\write_d4.cpp

```
        index_data=0;   len=100;
        do
        {   status = bioscom(3, 0, COM1);
            if (status & DATA_READY)
                {   rf_return_cmd = bioscom(2, 0, COM1);
                    rf_data[index_data]=rf_return_cmd;
                    index_data++;
                    if (index_data==2) {len=rf_return_cmd+2;}
                }
        }while(!kbhit()&&index_data!=len);
        cout<<"\nReturn Command Write a data block:"<<rf_write_cmd[3]<<"   ";
        for(i=0;i<index_data;i++)
            {   if(i>0) { cout<<','; }
                cout<<rf_data[i];
            }
        cout<<""<<endl;
        if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
            {   cout<<"\n\n\t\tWrite a data block:"<<rf_write_cmd[3]<<" = Success"<<endl;   }
        else
            {   cout<<"\n\n\t\tWrite a data block:"<<rf_write_cmd[3]<<" = Fail"<<endl;
                switch(rf_data[...])
                    {   case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
                        case 0x05  :cout<<"\n\t\tWrite fail"<<endl;break;
                        case 0x06  :cout<<"\n\t\tUnable to read after write"<<endl;break;
                        case 0x0D  :cout<<"\n\t\tNot authenticate"<<endl;break;
                        case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
                    }
            }
      }
      cout<<"\nExit Program Key ESC"<<endl;
      key=getch();
   }while(key!=27);
}
```

การทดลองที่ 22 :\tc\bin\mifare\iso14443\write_d4.cpp

Return:

0xBD	Len	0x04	Status	Data	Checksum

Status:	0x00:	Operation succeed
	0x01:	No tag
	0x05:	Write fail
	0x06:	Unable to read after write
	0x0D:	Not authenticate
	0xF0:	Checksum error
Data:	Block data written if operation succeeds, 16 bytes.	


```
void login_sector()
{
   int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[30];
   char key;
      if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
      {
         cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
         login_state=1;
      }
      else
      {
        cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
         login_state=0;
      }
      cout<<"\nExit Login sector\n\n"<<endl;
}

void reset()
{

}
```

Return:

0xBD	Len	0x02	Status	Checksum

Status:	0x02:	Login succeed
	0x01:	No tag
	0x03:	Login fail
	0xF0:	Checksum error

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector: Sector need to login

Type: Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)

Key: Password, 6 bytes

การทดลองที่ 22 :\tc\bin\mifare\iso14443\write_d4.cpp

4-4-6. Initialize a value block

0xBA	Len	0x06	Block	Value	Checksum

Block: The block number to be initialized, 1 byte.
Value: The value to write, 4 bytes.

Return:

0xBD	Len	0x06	Status	Value	Checksum

Status:
- 0x00: Operation succeed
- 0x01: No tag
- 0x05: Write fail
- 0x06: Unable to read after write
- 0x0D: Not authenticate
- 0xF0: Checksum error

Value: Value written if the operation succeeds, 4 bytes.

```
1  #include <stdio.h>      #include <bios.h>      #include <conio.h>
2  #include <iostream.h>   #include <time.h>      #include <dos.h>
3  void login_sector();    void reset();
4  #define COM1 0          #define DATA_READY 0x100
5  #define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
6  int login_state;
7  void main()
8  {  time_t timef, timel;
9     float difft;
10    int len,i,status,index_data;
11    int rf_initializa_cmd[8]={...,...,...,...,...,...,...,...};
12    int rf_check_sum,rf_return_cmd,rf_data[21];
13    char key;
14    bioscom(0, SETTINGS, COM1);
15    clrscr();            reset();
```

การทดลองที่ 23 :\tc\bin\mifare\iso14443\inti_v6.cpp


```
18    do
19    {  clrscr();
20       login_state=0;
21       login_sector();
22       if(login_state==1)
23       {   cout<<"\nSend Command initializa a value block:"<<rf_initializa_cmd[3]<<"   ";
24           rf_check_sum=0;
25           for(i=0;i<8;i++)
26             {   bioscom(1, rf_initializa_cmd[i], COM1);
27                 rf_check_sum^=rf_initializa_cmd[i];
28                 cout<<rf_initializa_cmd[i];
29                 cout<<',';
30             }
31           bioscom(1, rf_check_sum, COM1);
32           cout<<rf_check_sum<<endl;
33           index_data=0;     len=100;
34           do
35             {   status = bioscom(3, 0, COM1);
36                 if (status & DATA_READY)
37                     {   rf_return_cmd = bioscom(2, 0, COM1);
38                         rf_data[index_data]=rf_return_cmd;
39                         index_data++;
40                         if (index_data==2) {len=rf_return_cmd+2;  }
41                     }
42             }while(!kbhit()&&index_data!=len);
```

การทดลองที่ 23 :\tc\bin\mifare\iso14443\inti_v6.cpp

```
45          cout<<"\nReturn Command initializa a value block:"<<rf_initializa_cmd[3]<<"   ";
46          for(i=0;i<index_data;i++)
47            {   if(i>0)
48                    { cout<<',';  }
49                cout<<rf_data[i];
50            }
51          cout<<""<<endl;
52          if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
53            {   cout<<"\n\n\n\t\tInitializa a value block:"<<rf_initializa_cmd[3]
54                <<" =  success"<<endl;
55            }
56          else
57            {   cout<<"\n\n\n\t\tInitializa a value block:"<<rf_initializa_cmd[3]<<" fail"<<endl;
58                switch(rf_data[...])
59                    {   case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
60                        case 0x05  :cout<<"\n\t\tWrite fail"<<endl;break;
61                        case 0x06  :cout<<"\n\t\tUnable to read after write"<<endl;break;
62                        case 0x0D  :cout<<"\n\t\tNot authenticate"<<endl;break;
63                        case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
64                    }
65            }
66       }
67       cout<<"\n\nExit Program Key ESC"<<endl;
68       key=getch();
69    }while(key!=27);
70  }
```

Return:

0xBD	Len	0x06	Status	Value	Checksum

Status:	0x00:	Operation succeed
	0x01:	No tag
	0x05:	Write fail
	0x06:	Unable to read after write
	0x0D:	Not authenticate
	0xF0:	Checksum error
Value:	Value written if the operation succeeds, 4 bytes.	

การทดลองที่ 23 :\tc\bin\mifare\iso14443\inti_v6.cpp


```
73  void login_sector()
74  {
75     time_t timef, timel;
76     float difft;
77     int len,i,status,index_data;
78     int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
79     int rf_check_sum,rf_return_cmd,rf_data[30];
80     char key;
81
82        if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
83        { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
84          login_state=1;
85        }
86        else
87        { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
88          login_state=0;
89          switch(rf_data[3])
90              {   case 1   :cout<<"\n\t\tNo Tag"<<endl;break;
91                  case 3   :cout<<"\n\t\tLogin fail"<<endl;break;
92                  case 240:cout<<"\n\t\tCard Type Ultralight Card"<<endl;break;
93              }
94        }
95        cout<<"\nExit Login sector\n\n"<<endl;
96  }
97
98
99  void reset()
100 {
101
102 }
```

Return:

0xBD	Len	0x02	Status	Checksum

Status:	0x02:	Login succeed
	0x01:	No tag
	0x03:	Login fail
	0xF0:	Checksum error

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector:	Sector need to login
Type:	Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key:	Password, 6 bytes

การทดลองที่ 23 :\tc\bin\mifare\iso14443\inti_v6.cpp

4-4-5. Read a value block

0xBA	Len	0x05	Block	Checksum

Block: The block number to be read, 1 byte.

Return:

0xBD	Len	0x05	Status	Value	Checksum

Status: 0x00: Operation succeed
0x01: No tag
0x04: Read fail
0x0D: Not authenticate
0x0E: Not a value block
0xF0: Checksum error

Value: Value returned if the operation succeeds, 4 bytes.

```
#include <stdio.h>      #include <bios.h>      #include <conio.h>
#include <iostream.h>   #include <time.h>      #include <dos.h>
void login_sector();    void reset();
#define COM1 0          #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int login_state;
void main()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_readvalue_cmd[4]={...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();
   reset();
```

การทดลองที่ 24 :\tc\bin\mifare\iso14443\read_v5.cpp


```
   do
   {  clrscr();
      login_state=0;
      login_sector();
      if(login_state==1)
          {   cout<<"\nSend Command Read a value block:"<<rf_readvalue_cmd[3]<<"   ";
              rf_check_sum=0;
              for(i=0;i<4;i++)
                {   bioscom(1, rf_readvalue_cmd[i], COM1);
                    rf_check_sum^=rf_readvalue_cmd[i];
                    cout<<rf_readvalue_cmd[i];
                    cout<<',';
                }
              bioscom(1, rf_check_sum, COM1);
              cout<<rf_check_sum<<endl;
              index_data=0;
              len=100;
              do
                {   status = bioscom(3, 0, COM1);
                    if (status & DATA_READY)
                        {   rf_return_cmd = bioscom(2, 0, COM1);
                            rf_data[index_data]=rf_return_cmd;
                            index_data++;
                            if (index_data==2) {len=rf_return_cmd+2;}
                        }
                }while(!kbhit()&&index_data!=len);
```

การทดลองที่ 24 :\tc\bin\mifare\iso14443\read_v5.cpp

```
49              cout<<"\nReturn Command Read a value block:"<<rf_readvalue_cmd[3]<<"   ";
50              for(i=0;i<index_data;i++)
51                {   if(i>0) { cout<<','; }
52                    cout<<rf_data[i];
53                }
54              cout<<""<<endl;
55              if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
56                {cout<<"\n\n\n\t\tRead a value block:"<<rf_readvalue_cmd[3]<<" = success"<<endl;}
57              else
58                {cout<<"\n\n\n\t\tRead a value block:"<<rf_readvalue_cmd[3]<<" fail"<<endl;
59                    switch(rf_data[...])
60                        {   case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
61                            case 0x04  :cout<<"\n\t\tRead fail"<<endl;break;
62                            case 0x0D  :cout<<"\n\t\tNot authenticate"<<endl;break;
63                            case 0x0E  :cout<<"\n\t\tNot a value block"<<endl;break;
64                            case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
65                        }
66                }
67          }
68        cout<<"\n\nExit Program Key ESC"<<endl;
69        key=getch();
70    }while(key!=27);
71 }
```

Return:

0xBD	Len	0x05	Status	Value	Checksum

Status:	0x00:	Operation succeed
	0x01:	No tag
	0x04:	Read fail
	0x0D:	Not authenticate
	0x0E:	Not a value block
	0xF0:	Checksum error
Value:	Value returned if the operation succeeds, 4 bytes.	

การทดลองที่ 24 :\tc\bin\mifare\iso14443\read_v5.cpp


```
73  void login_sector()
74  {
75     time_t timef, timel;
76     float difft;
77     int len,i,status,index_data;
78     int rf_login_cmd[11]={186,10,2,...,170,255,255,255,255,255,255};
79     int rf_check_sum,rf_return_cmd,rf_data[30];
80     char key;
81     bioscom(0, SETTINGS, COM1);
82
83        if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
84        { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
85          login_state=1;
86        }
87        else
88        { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
89          login_state=0;
90          switch(rf_data[3])
91              {   case 1   :cout<<"\n\t\tNo Tag"<<endl;break;
92                  case 3   :cout<<"\n\t\tLogin fail"<<endl;break;
93                  case 240:cout<<"\n\t\tCard Type Ultralight Card"<<endl;break;
94              }
95        }
96        cout<<"\nExit Login sector\n\n"<<endl;
97  }
98
99  void reset()
100 {
101
102 }
```

Return:

0xBD	Len	0x02	Status	Checksum

Status:	0x02:	Login succeed
	0x01:	No tag
	0x03:	Login fail
	0xF0:	Checksum error

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector:	Sector need to login
Type:	Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key:	Password, 6 bytes

การทดลองที่ 24 :\tc\bin\mifare\iso14443\read_v5.cpp

```
#include <stdio.h>      #include <bios.h>      #include <conio.h>
#include <iostream.h>   #include <time.h>      #include <dos.h>
void check_value();     void login_sector();
void reset();
#define COM1 0          #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int login_state;
int count,check_value_state; // void check_value();
char count_char[5]={" "};    // void check_value();
void main()
{  char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();
   reset();
   do
   {  login_sector();
      check_value();
      cout<<"\n\n\t\tCheck_value_state :"<<check_value_state<<endl;
      cout<<"\n\nExit Program Key ESC"<<endl;
      key=getch();
   }while(key!=27);
}
```

การทดลองที่ 25 :\tc\bin\mifare\iso14443\read_v6.cpp


```
void check_value()
{  int len,i,status,index_data;
   int rf_readvalue_cmd[4]={...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
      rf_check_sum=0;
      for(i=0;i<4;i++)
      { bioscom(1, rf_readvalue_cmd[i], COM1);
        rf_check_sum^=rf_readvalue_cmd[i];
      }
      bioscom(1, rf_check_sum, COM1);
      index_data=0;
      len=100;
      do
      { status = bioscom(3, 0, COM1);
        if (status & DATA_READY)
         {  rf_return_cmd = bioscom(2, 0, COM1);
            rf_data[index_data]=rf_return_cmd;
            index_data++;
            if (index_data==2) {len=rf_return_cmd+2;  }
         }
      }while(!kbhit()&&index_data!=len);
```

4-4-5. Read a value block

0xBA	Len	0x05	Block	Checksum

Block: The block number to be read, 1 byte.

การทดลองที่ 25 :\tc\bin\mifare\iso14443\read_v6.cpp

```
50        if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
51        { count=(rf_data[5]*...)+rf_data[...];
52          check_value_state=1;
53          cout<<"\n\n\t\tRead a value block:"<<rf_readvalue_cmd[3]<<" = success >>"<<count<<endl;
54          count_char[0]=48+(count/100); count_char[1]=48+((count%100)/10);
55          count_char[2]=48+((count%10)%10);
56        }
57        else
58        { cout<<"\n\n\t\tRead a value block:"<<rf_readvalue_cmd[3]<<" fail"<<endl;
59          check_value_state=0;
60        }
61
62  }
```

Return:

0xBD	Len	0x05	Status	Value	Checksum

Status:
- 0x00: Operation succeed
- 0x01: No tag
- 0x04: Read fail
- 0x0D: Not authenticate
- 0x0E: Not a value block
- 0xF0: Checksum error

Value: Value returned if the operation succeeds, 4 bytes.

```
64  void login_sector()
65  {   int rf_login_cmd[11]={186,10,2,....,170,255,255,255,255,255,255};
66      if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
67        { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
68          login_state=1;
69        }
70        else
71        { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
72          login_state=0;
73        }
74  }
75
76
77  void reset()
78  {
79  }
```

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector: Sector need to login
Type: Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key: Password, 6 bytes

การทดลองที่ 25 :\tc\bin\mifare\iso14443\read_v6.cpp

4-4-8. Increment value

0xBA	Len	0x08	Block	Value	Checksum

Block: The block number to be increased, 1 byte.
Value: The value to be increased by, 4 bytes.

Return:

0xBD	Len	0x08	Status	Value	Checksum

Status:
- 0x00: Operation succeed
- 0x01: No tag
- 0x05: Write fail
- 0x06: Unable to read after write
- 0x0D: Not authenticate
- 0x0E: Not a value block
- 0xF0: Checksum error

Value: The value after increment if the operation succeeds, 4 bytes

การทดลองที่ 26 :\tc\bin\mifare\iso14443\incre_v8.cpp

```
#include <stdio.h>      #include <bios.h>      #include <conio.h>
#include <iostream.h>   #include <time.h>      #include <dos.h>
void login_sector();    void reset();
#define COM1 0          #define DATA_READY 0x100
#define SETTINGS (0xE0| 0x03 | 0x00 | 0x00)
int login_state;
void main()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_incrementvalue_cmd[8]={...,...,...,...,...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();
   reset();
   do
   {  clrscr();
      login_state=0;
      login_sector();
      if(login_state==1)
      { cout<<"\nSend Command Increment value block:"<<rf_incrementvalue_cmd[3]<<"   ";
        rf_check_sum=0;
        for(i=0;i<8;i++)
        {   bioscom(1, rf_incrementvalue_cmd[i], COM1);
            rf_check_sum^=rf_incrementvalue_cmd[i];
            cout<<rf_incrementvalue_cmd[i];
            cout<<',';
        }
        bioscom(1, rf_check_sum, COM1);
        cout<<rf_check_sum<<endl;
```

4-4-8. Increment value

0xBA	Len	0x08	Block	Value	Checksum

Block: The block number to be increased, 1 byte.
Value: The value to be increased by, 4 bytes.

การทดลองที่ 26 :\tc\bin\mifare\iso14443\incre_v8.cpp


```
        index_data=0;   len = 100;
        do
        {   status = bioscom(3, 0, COM1);
            if (status & DATA_READY)
                {   rf_return_cmd = bioscom(2, 0, COM1);
                    rf_data[index_data]=rf_return_cmd;
                    index_data++;
                    if (index_data==2) {len=rf_return_cmd+2;}
                }
        }while(!kbhit()&&index_data!=len);
        cout<<"\nReturn Command Increment value block:"<<rf_incrementvalue_cmd[3]<<"   ";;
        for(i=0;i<index_data;i++)
        {   if(i>0) {   cout<<',';  }
            cout<<rf_data[i];
        }
        cout<<""<<endl;
        if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
        { cout<<"\n\n\n\t\tIncrement value block:"<<rf_incrementvalue_cmd[3]<<"= success"<<endl;}
        else
        {   cout<<"\n\n\n\t\tIncrement a value block:"<<rf_incrementvalue_cmd[3]<<"  = fail"<<endl;
            switch(rf_data[...])
            {   case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
                case 0x05  :cout<<"\n\t\tWrite fail"<<endl;break;
                case 0x06  :cout<<"\n\t\tUnable to read after write"<<endl;break;
                case 0x0D  :cout<<"\n\t\tNot authenticate"<<endl;break;
                case 0x0E  :cout<<"\n\t\tNot a value block"<<endl;break;
                case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
            }
        }
       }
        cout<<"\n\nExit Program Key ESC"<<endl;
        key=getch();
   }while(key!=27);
}
```

Return:

0xBD	Len	0x08	Status	Value	Checksum

Status: 0x00: Operation succeed
0x01: No tag
0x05: Write fail
0x06: Unable to read after write
0x0D: Not authenticate
0x0E: Not a value block
0xF0: Checksum error
Value: The value after increment if the operation succeeds, 4 bytes

การทดลองที่ 26 :\tc\bin\mifare\iso14443\incre_v8.cpp

```
69  void login_sector()
70  {   int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
71      if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
72        { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
73          login_state=1;
74        }
75        else
76        { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
77          login_state=0;
78          switch(rf_data[3])
79              {   case 1   :cout<<"\n\t\tNo Tag"<<endl;break;
80                  case 3   :cout<<"\n\t\tLogin fail"<<endl;break;
81                  case 240:cout<<"\n\t\tCard Type Ultralight Card"<<endl;break;
82              }
83        }
84        cout<<"\nExit Login sector\n\n"<<endl;
85  }
86
87
88  void reset()
89  {
90
91  }
```


Return:

0xBD	Len	0x02	Status	Checksum

Status: 0x02: Login succeed
0x01: No tag
0x03: Login fail
0xF0: Checksum error

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector: Sector need to login
Type: Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key: Password, 6 bytes

การทดลองที่ 26 :\tc\bin\mifare\iso14443\incre_v8.cpp

4-4-9. Decrement value

0xBA	Len	0x09	Block	Value	Checksum

Block: The block number to be decreased, 1 byte
Value: The value to be decreased by, 4 bytes

Return:

0xBD	Len	0x09	Status	Value	Checksum

Status: 0x00: Operation succeed
0x01: No tag
0x05: Write fail
0x06: Unable to read after write
0x0D: Not authenticate
0x0E: Not a value block
0xF0: Checksum error
Value: The value after decrement if the operation succeeds, 4 bytes

การทดลองที่ 27 :\tc\bin\mifare\iso14443\decre_v9.cpp

```
#include <stdio.h>      #include <bios.h>      #include <conio.h>
#include <iostream.h>   #include <time.h>      #include <dos.h>
void reset();           void login_sector();
#define COM1 0          #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int login_state;
void main()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_decrementvalue_cmd[8]={...,...,...,...,...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();
   reset();
   do
   {  clrscr();
      login_state=0;
      login_sector();
      if(login_state==1)
      { cout<<"\nSend Command Decrement value block:"<<rf_decrementvalue_cmd[3]<<"   ";
        rf_check_sum=0;
        for(i=0;i<8;i++)
        {   bioscom(1, rf_decrementvalue_cmd[i], COM1);
            rf_check_sum^=rf_decrementvalue_cmd[i];
            cout<<rf_decrementvalue_cmd[i];
            cout<<',';
        }
        bioscom(1, rf_check_sum, COM1);
        cout<<rf_check_sum<<endl;
```

4-4-9. Decrement value

0xBA	Len	0x09	Block	Value	Checksum

Block: The block number to be decreased, 1 byte
Value: The value to be decreased by, 4 bytes

การทดลองที่ 27 :\tc\bin\mifare\iso14443\decre_v9.cpp


```
        index_data=0;   len=100;
        do
        {   status = bioscom(3, 0, COM1);
            if (status & DATA_READY)
            {   rf_return_cmd = bioscom(2, 0, COM1);
                rf_data[index_data]=rf_return_cmd;
                index_data++;
                if (index_data==2) {len=rf_return_cmd+2; }
            }
        }while(!kbhit()&&index_data!=len);
        cout<<"\nReturn Command Decrement value block:"<<rf_decrementvalue_cmd[3]<<"   ";
        for(i=0;i<index_data;i++)
        {   if(i>0)
            {   cout<<',';  }
            cout<<rf_data[i];
            }
        cout<<""<<endl;
        if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
        {   cout<<"\n\n\n\t\tDecrement value block:"<<rf_decrementvalue_cmd[3]<<"= success"<<endl; }
        else
        {   cout<<"\n\n\n\t\tDecrement a value block:"<<rf_decrementvalue_cmd[3]<<"= fail"<<endl;
            switch(rf_data[...])
            {   case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
                case 0x05  :cout<<"\n\t\tWrite fail"<<endl;break;
                case 0x06  :cout<<"\n\t\tUnable to read after write"<<endl;break;
                case 0x0D  :cout<<"\n\t\tNot authenticate"<<endl;break;
                case 0x0E  :cout<<"\n\t\tNot a value block"<<endl;break;
                case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
            }
        }
      }
      cout<<"\n\nExit Program Key ESC"<<endl;
      key=getch();
   }while(key!=27);
}
```

Return:

0xBD	Len	0x09	Status	Value	Checksum

Status: 0x00: Operation succeed
0x01: No tag
0x05: Write fail
0x06: Unable to read after write
0x0D: Not authenticate
0x0E: Not a value block
0xF0: Checksum error
Value: The value after decrement if the operation succeeds, 4 bytes

การทดลองที่ 27 :\tc\bin\mifare\iso14443\decre_v9.cpp

```
71 void login_sector()
72 {
73    int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
74    if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
75     {   cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
76         login_state=1;
77     }
78     else
79     {   cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
80         login_state=0;
81         switch(rf_data[3])
82             {   case 1  :cout<<"\n\t\tNo Tag"<<endl;break;
83                 case 3  :cout<<"\n\t\tLogin fail"<<endl;break;
84                 case 240:cout<<"\n\t\tCard Type Ultralight Card"<<endl;break;
85             }
86      }
87      cout<<"\nExit Login sector\n\n"<<endl;
88 }
89
90
91 void reset()
92 {
93
94 }
```

Return:

0xBD	Len	0x02	Status	Checksum

Status: 0x02: Login succeed
0x01: No tag
0x03: Login fail
0xF0: Checksum error

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector: Sector need to login
Type: Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key: Password, 6 bytes

การทดลองที่ 27 :\tc\bin\mifare\iso14443\decre_v9.cpp


```
1  #include <stdio.h>      #include <bios.h>      #include <conio.h>
2  #include <iostream.h>   #include <time.h>      #include <dos.h>
3  void dec_value();       void reset();
4  void login_sector();
5  #define COM1 0          #define DATA_READY 0x100
6  #define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
7  int login_state;
8  int price;
9  int dec_value_state;    //void dec_value();
10 char dec_char[5]={" "}; //void dec_value();
11 void main()
12 {  char key;
13    bioscom(0, SETTINGS, COM1);
14    clrscr();
15    reset();
16    do
17    {  clrscr();
18       cout<<"\n\t\tPrice := ";
19       cin>>price;
20       cout<<"\n\t\t===Insert Card===";
21       login_sector();
22       dec_value();
23       cout<<"\n\nExit Program Key ESC"<<endl;
24       key=getch();
25    }while(key!=27);
26 }
```

การทดลองที่ 28 :\tc\bin\mifare\iso14443\decrev10.cpp

```
void dec_value()
{  int len,i,status,index_data;
   int rf_decrementvalue_cmd[8]={...,...,...,...,0,0,0,0};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   rf_decrementvalue_cmd[4]=price;
   char key;
      rf_check_sum=0;
      for(i=0;i<8;i++)
      {  bioscom(1, rf_decrementvalue_cmd[i], COM1);
         rf_check_sum^=rf_decrementvalue_cmd[i];
      }
      bioscom(1, rf_check_sum, COM1);
      index_data=0;     len=100;
      do
      {  status = bioscom(3, 0, COM1);
         if (status & DATA_READY)
         {  rf_return_cmd = bioscom(2, 0, COM1);
            rf_data[index_data]=rf_return_cmd;
            index_data++;
            if (index_data==2) {len=rf_return_cmd+2;  }
         }
      }while(!kbhit()&&index_data!=len);
      if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
      {   int temp_dec;
          temp_dec=(rf_data[5]*256)+rf_data[4];
          cout<<"\n\tDecrement a value block:"<<rf_decrementvalue_cmd[3]
              <<"=success >>"<<temp_dec<<endl;
          dec_char[0]=48+(temp_dec/100); dec_char[1]=48+((temp_dec%100)/10);
          dec_char[2]=48+((temp_dec%10)%10);
          dec_value_state=1;
      }
      else
      {  cout<<"\n\t\tDecrement a value block:"<<rf_decrementvalue_cmd[3]<<"  = fail"<<endl;
          dec_value_state=0;
      }
}
```

4-4-9. Decrement value

0xBA	Len	0x09	Block	Value	Checksum

Block: The block number to be decreased, 1 byte
Value: The value to be decreased by, 4 bytes

Return:

0xBD	Len	0x09	Status	Value	Checksum

Status:
- 0x00: Operation succeed
- 0x01: No tag
- 0x05: Write fail
- 0x06: Unable to read after write
- 0x0D: Not authenticate
- 0x0E: Not a value block
- 0xF0: Checksum error

Value: The value after decrement if the operation succeeds, 4 bytes

การทดลองที่ 28 :\tc\bin\mifare\iso14443\decrev10.cpp


```
void login_sector()
{
   int len,i,status,index_data;
   int rf_login_cmd[11]={186,10,2,...,170,255,255,255,255,255,255};
      if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
      { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
        login_state=1;
      }
      else
      { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
        login_state=0;
        switch(rf_data[3])
            {   case 1   :cout<<"\n\t\tNo Tag"<<endl;break;
                case 3   :cout<<"\n\t\tLogin fail"<<endl;break;
                case 240:cout<<"\n\t\tCard Type Ultralight Card"<<endl;break;
            }
      }
      cout<<"\nExit Login sector\n\n"<<endl;
}


void reset()
{

}
```

Return:

0xBD	Len	0x02	Status	Checksum

Status:
- 0x02: Login succeed
- 0x01: No tag
- 0x03: Login fail
- 0xF0: Checksum error

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector: Sector need to login
Type: Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key: Password, 6 bytes

การทดลองที่ 28 :\tc\bin\mifare\iso14443\decrev10.cpp

4-4-10. Copy value

0xBA	Len	0x0A	Source	Destination	Checksum

Source: The source block copy from, 1 byte
Destination: The destination copy to, 1 byte
The source and destination must in the same sector

Return:

0xBD	Len	0x0A	Status	Value	Checksum

Status:
- 0x00: Operation succeed
- 0x01: No tag
- 0x05: Write fail
- 0x06: Unable to read after write
- 0x0D: Not authenticate
- 0x0E: Not a value block (Source)
- 0xF0: Checksum error

Value: The value after copy if the operation succeeds, 4 bytes

การทดลองที่ 29 :\tc\bin\mifare\iso14443\copy_v10.cpp


```
#include <stdio.h>      #include <bios.h>      #include <conio.h>
#include <iostream.h>   #include <time.h>      #include <dos.h>
void login_sector();    void reset();
#define COM1 0
#define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int login_state;
void main()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_copyvalue_cmd[5]={...,...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();
   reset();
   do
   {  clrscr();
      login_state=0;
      login_sector();
      if(login_state==1)
      { cout<<"\nSend Command copy value block:"<<rf_copyvalue_cmd[3]
          <<" to block:"<<rf_copyvalue_cmd[4]<<"   ";
        rf_check_sum=0;
        for(i=0;i<5;i++)
        {   bioscom(1, rf_copyvalue_cmd[i], COM1);
            rf_check_sum^=rf_copyvalue_cmd[i];
            cout<<rf_copyvalue_cmd[i];
            cout<<',';
        }
        bioscom(1, rf_check_sum, COM1);
        cout<<rf_check_sum<<endl;
```

4-4-10. Copy value

0xBA	Len	0x0A	Source	Destination	Checksum

Source: The source block copy from, 1 byte
Destination: The destination copy to, 1 byte
The source and destination must in the same sector

การทดลองที่ 29 :\tc\bin\mifare\iso14443\copy_v10.cpp

```
        index_data=0;   len=100;
        do
        {   status = bioscom(3, 0, COM1);
            if (status & DATA_READY)
            {   rf_return_cmd = bioscom(2, 0, COM1);
                rf_data[index_data]=rf_return_cmd;
                index_data++;
                if (index_data==2) {len=rf_return_cmd+2;}
            }
        }while(!kbhit()&&index_data!=len);
        cout<<"Return Command copy v block:"<<rf_copyvalue_cmd[3]<<"to"<<rf_copyvalue_cmd[4]<<" ";
        for(i=0;i<index_data;i++)
        {   if(i>0)
            {   cout<<',';  }
            cout<<rf_data[i];
        }
        cout<<""<<endl;
        if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
        {cout<<"Copy v block:"<<rf_copyvalue_cmd[3]<<"to"<<rf_copyvalue_cmd[4]<<"= success"<<endl;}
        else
        {   cout<<"Copy v block:"<<rf_copyvalue_cmd[3]<<"to"<<rf_copyvalue_cmd[4]<<"= fail"<<endl;
            switch(rf_data[...])
            {   case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
                case 0x05  :cout<<"\n\t\tWrite fail"<<endl;break;
                case 0x06  :cout<<"\n\t\tUnable to read after write"<<endl;break;
                case 0x0D  :cout<<"\n\t\tNot authenticate"<<endl;break;
                case 0x0E  :cout<<"\n\t\tNot a value block(source)"<<endl;break;
                case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
            }
        }
      }
      cout<<"\n\nExit Program Key ESC"<<endl;
      key=getch();
   }while(key!=27);
}
```

Return:

0xBD	Len	0x0A	Status	Value	Checksum

Status:	0x00:	Operation succeed
	0x01:	No tag
	0x05:	Write fail
	0x06:	Unable to read after write
	0x0D:	Not authenticate
	0x0E:	Not a value block (Source)
	0xF0:	Checksum error
Value:	The value after copy if the operation succeeds, 4 bytes	

การทดลองที่ 29 :\tc\bin\mifare\iso14443\copy_v10.cpp


```
void login_sector()
{
   int len,i,status,index_data;
   int rf_login_cmd[11]={186,10,2,...,170,255,255,255,255,255,255};
      if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
      { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
        login_state=1;
      }
      else
      { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
        login_state=0;
        switch(rf_data[3])
            {   case 1   :cout<<"\n\t\tNo Tag"<<endl;break;
                case 3   :cout<<"\n\t\tLogin fail"<<endl;break;
                case 240:cout<<"\n\t\tCard Type Ultralight Card"<<endl;break;
            }
      }
      cout<<"\nExit Login sector\n\n"<<endl;
}


void reset()
{

}
```

Return:

0xBD	Len	0x02	Status	Checksum

Status:	0x02:	Login succeed
	0x01:	No tag
	0x03:	Login fail
	0xF0:	Checksum error

4-4-2. Login to a sector

0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector: Sector need to login
Type: Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key: Password, 6 bytes

การทดลองที่ 29 :\tc\bin\mifare\iso14443\copy_v10.cpp

4-4-7. Write master key (key A)

0xBA	Len	0x07	Sector	Key	Checksum

Sector: The sector number to be written, 1 byte.
Key: Authentication key, 6 bytes

Return:

0xBD	Len	0x07	Status	Key	Checksum

Status: 0x00: Operation succeed
0x01: No tag
0x05: Write fail
0x0D: Not authenticate
0xF0: Checksum error

Key: Authentication key written if the operation succeeds, 6 bytes.

การทดลองที่ 30 :\tc\bin\mifare\iso14443\write_m7.cpp


```
#include <stdio.h>      #include <bios.h>      #include <conio.h>
#include <iostream.h>   #include <time.h>      #include <dos.h>
void login_sector();    void reset();
#define COM1 0          #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int login_state;
void main()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_write_masterkey_cmd[10]={...,...,...,...,...,...,...,...,...,...};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   clrscr();       reset();
   do
   {  clrscr();
      login_state=0;
      login_sector();
      if(login_state==1)
        {   cout<<"\nSend Command Write master key(key A)Sector:="<<rf_write_masterkey_cmd[3]<<" ";
            rf_check_sum=0;
            for(i=0;i<10;i++)
                {   bioscom(1, rf_write_masterkey_cmd[i], COM1);
                    rf_check_sum^=rf_write_masterkey_cmd[i];
                    cout<<rf_write_masterkey_cmd[i];
                    cout<<',';
                }
            bioscom(1, rf_check_sum, COM1);
            cout<<rf_check_sum<<endl;
```

4-4-7. Write master key (key A)

0xBA	Len	0x07	Sector	Key	Checksum

Sector: The sector number to be written, 1 byte.
Key: Authentication key, 6 bytes

การทดลองที่ 30 :\tc\bin\mifare\iso14443\write_m7.cpp

```
            index_data=0;   len = 100;
            do
            {   status = bioscom(3, 0, COM1);
                if (status & DATA_READY)
                    {   rf_return_cmd = bioscom(2, 0, COM1);
                        rf_data[index_data]=rf_return_cmd;
                        index_data++;
                        if (index_data==2) {len=rf_return_cmd+2;  }
                    }
            }while(!kbhit()&&index_data!=len);
            cout<<"\nReturn Command Write master (key A)Sector:"<<rf_write_masterkey_cmd[3]<<"   ";
            for(i=0;i<index_data;i++)
                {   if(i>0) {   cout<<',';    }
                    cout<<rf_data[i];
                }
            cout<<""<<endl;
            if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
                { cout<<"\tWrite master(key A)Sector:"<<rf_write_masterkey_cmd[3]<<"=success"<<endl;
                }
            else
                { cout<<"\tWrite master(key A)Sector:"<<rf_write_masterkey_cmd[3]<<"= fail"<<endl;
                    switch(rf_data[...])
                        {   case 0x01  :cout<<"\n\t\tNo Tag"<<endl;break;
                            case 0x05  :cout<<"\n\t\tWrite fail"<<endl;break;
                            case 0x0D  :cout<<"\n\t\tNot authenticate"<<endl;break;
                            case 0xF0  :cout<<"\n\t\tChecksum error"<<endl;break;
                        }
                }
        }
        cout<<"\n\nExit Program Key ESC"<<endl;
        key=getch();
    }while(key!=27);
}
```

Return:

0xBD	Len	0x07	Status	Key	Checksum

Status: 0x00: Operation succeed
0x01: No tag
0x05: Write fail
0x0D: Not authenticate
0xF0: Checksum error

Key: Authentication key written if the operation succeeds, 6 bytes.

การทดลองที่ 30 :\tc\bin\mifare\iso14443\write_m7.cpp


```
void login_sector()
{
    time_t timef, timel;
    float difft;
    int len,i,status,index_data;
    int rf_login_cmd[11]={...,...,...,...,...,...,...,...,...,...,...};
       if(index_data==5&&rf_data[2]==2&&rf_data[3]==2)
       { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = success"<<endl;
         login_state=1;
       }
       else
       { cout<<"\n\n\n\t\tLogin to a sector:"<<rf_login_cmd[3]<<" = Fail"<<endl;
         login_state=0;
         switch(rf_data[3])
             {   case 1   :cout<<"\n\t\tNo Tag"<<endl;break;
                 case 3   :cout<<"\n\t\tLogin fail"<<endl;break;
                 case 240:cout<<"\n\t\tCard Type Ultralight Card"<<endl;break;
             }
       }
       cout<<"\nExit Login sector\n\n"<<endl;
}

void reset()
{

}
```

Return:

0xBD	Len	0x02	Status	Checksum

Status: 0x02: Login succeed
0x01: No tag
0x03: Login fail
0xF0: Checksum error

4-4-2. Login to a sector

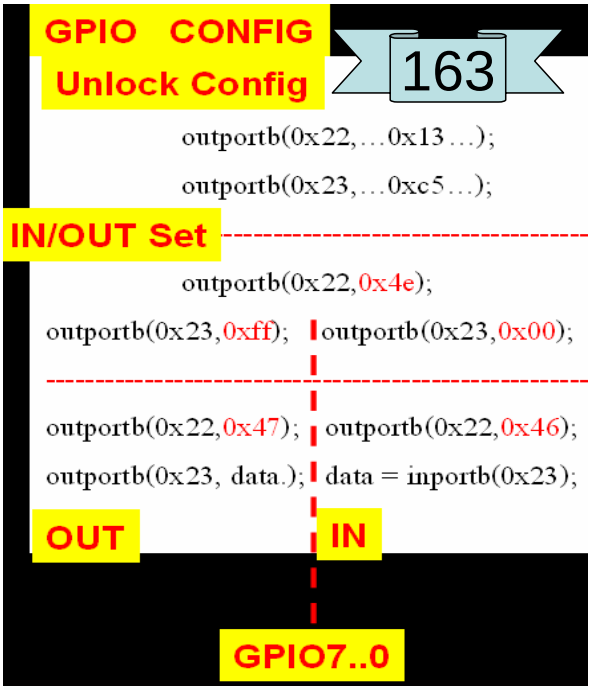
0xBA	Len	0x02	Sector	Type	Key	Checksum

Sector: Sector need to login
Type: Key type (0xAA: authenticate with KeyA, 0xBB: authenticate with KeyB)
Key: Password, 6 bytes

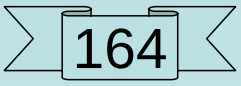
การทดลองที่ 30 :\tc\bin\mifare\iso14443\write_m7.cpp

```c
#include <dos.h>    #include <bios.h>
#include <conio.h>  #include <iostream.h>
#include <time.h>
void config_gpio(); void check_tag();
int data;         //void check_tag();
void main()
{  config_gpio();
   clrscr();
   do
   {    check_tag();
        cout<<"\n\t\tGPIO[7..0]  >>"<<data;
        delay(100);
    }while(!kbhit());
}
void config_gpio()
{ outportb(0x22,...); // Unlock Configuration
  outportb(0x23,...); // Unlock Configuration
  outportb(0x22,...); // Set GPIO[7..0]
  outportb(0x23,...); // Set 1111 1110 >>>     0 = IN PUT
}

void check_tag()
{  outportb(0x22,...); // Read Data GPIO[7..0]
   data= inportb(0x23); // data = GPIO[7..0]
}
```



การทดลองที่ 31 :\tc\bin\mifare\iso14443\gpio_1.cpp




```c
#include <dos.h>          #include <bios.h>        #include <conio.h>
#include <iostream.h>     #include <time.h>
void reset();             void config_gpio();
void check_tag();
#define COM1 0            #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int data; // void check_tag();
void main()
{  time_t timef, timel;
   float difft;
   int len,i,status,index_data;
   int rf_selectcard_cmd[3]={186,2,1};
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   config_gpio();
   clrscr();
   reset();
   do
   {  clrscr();
      rf_check_sum=0;
      cout<<"\n\tSend Command  select Mifare card information\n\t";
      for(i=0;i<3;i++)
      { bioscom(1, rf_selectcard_cmd[i], COM1);
        rf_check_sum^=rf_selectcard_cmd[i];
        cout<<rf_selectcard_cmd[i];
        cout<<',';
      }
      bioscom(1, rf_check_sum, COM1);
      cout<<rf_check_sum;
      cout<<""<<endl;
```

การทดลองที่ 32 :\tc\bin\mifare\iso14443\checktag.cpp

```
        index_data=0;
        len = 100;
        do
        { status = bioscom(3, 0, COM1);
          if (status & DATA_READY)
              {   rf_return_cmd = bioscom(2, 0, COM1);
                  rf_data[index_data]=rf_return_cmd;
                  index_data++;
              }
          if (index_data==2) {len=rf_return_cmd+2;}
        }while(!kbhit()&&index_data!=len);
        cout<<"\n\tReturn Command select Mifare card information \n\t";
        for(i=0;i<index_data;i++)
          {   if(i>0) {   cout<<",";  }
              cout<<rf_data[i];
          }
        cout<<""<<endl;
        if(index_data==10&&rf_data[2]==1&&rf_data[3]==0)
          {   cout<<"\n\tselect Mifare card information success"<<endl;     }
        else
          {   cout<<"\n\tselect Mifare card information Fail"<<endl;       }
        cout<<"\n\tExit Program Key ESC"<<endl;
        do{   check_tag();
              cout<<"\n\t\tWait Tag Out loop1 >>"<<data;
              delay(100);
          }while(!kbhit()&&data!=...);

         do{  check_tag();
              cout<<"\n\t\tWait Insert loop1 >>"<<data;
              delay(100);
          }while(!kbhit()&&data!=...);
      if(kbhit()) key=getch();
    }while(key!=27);
}
```

การทดลองที่ 32 :\tc\bin\mifare\iso14443\checktag.cpp

```
void reset()
{

}
void config_gpio()
{
  outportb(0x22,...); // Unlock Configuration
  outportb(0x23,...); // Unlock Configuration
  outportb(0x22,...); // Set GPIO[7..0]
  outportb(0x23,...); // Set 1111 1110= IN
}

void check_tag()
{
   outportb(0x22,...); // Set GPIO[7..0] input port
   data= inportb(0x23); // data = GPIO[7..0]
}
```


GPIO CONFIG

Unlock Config

outportb(0x22,…0x13…);

outportb(0x23,…0xc5…);

IN/OUT Set

outportb(0x22,0x4e);

OUT	IN
outportb(0x23,0xff);	outportb(0x23,0x00);
outportb(0x22,0x47);	outportb(0x22,0x46);
outportb(0x23, data.);	data = inportb(0x23);

GPIO7..0

การทดลองที่ 32 :\tc\bin\mifare\iso14443\checktag.cpp

```
#include <dos.h>          #include <bios.h>          #include <conio.h>
#include <iostream.h>     #include <time.h>          #include <process.h>
#include <string.h>       #include <stdio.h>
void select_card();       void read_expire_date();
void check_value();       void dec_value();
void check_value();       void login_sector();
void reset();             void config_gpio();
void check_tag();         void save_flie();
#define COM1 0            #define DATA_READY 0x100
#define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
int data;                 int price;                int login_state;
int count,check_value_state;                        int dec_value_state;
int expire_date_check;                              char data_in_block[10]={" "};
char sn_tag_char[20]={" "};                         char price_char[5]={" "};
char count_char[5]={" "};                           char dec_char[5]={" "};
void main()
{  int len,i,status,index_data;
   int rf_check_sum,rf_return_cmd,rf_data[21];
   char key;
   bioscom(0, SETTINGS, COM1);
   config_gpio();         clrscr();                 reset();
   do
   {clrscr();
    cout<<"\n\t\tPrice := ";                       cin>>price;
    cout<<"\n\t\t===Insert Card===";
    price_char[0]=48+(price/100); price_char[1]=48+((price%100)/10);
    price_char[2]=48+((price%10)%10);
    do{  check_tag();
      }while(!kbhit()&&data!=254);
    if(kbhit()) key=getch();
```

การทดลองที่ 33 :\tc\bin\mifare\iso14443\food1.cpp

ตัวอย่างการพัฒนาระบบศูนย์อาหาร


```
    if(data!=255)
      {select_card();
       login_sector();
       if(login_state==1)
        {
         read_expire_date();
         if(expire_date_check==1)
            {
               check_value();
               if(check_value_state==1)
             {if(price<=count)
                {dec_value();
                   if(dec_value_state==1)
                 {
                    save_flie();
                    sound(1000);delay(200);nosound();
                    sound(500);delay(200);nosound();
                 }
                 else { cout<<"\n\n\t===Decrement value Fail==="; sound(500);delay(200);nosound();}
                }
                else { cout<<"\n\n\t===Not Enough Fail==="; sound(500);delay(200);nosound(); }
             }
             else { cout<<"\n\n\t===Check Value Fail==="; sound(500);delay(200);nosound();}
            }
            else { cout<<"\n\n\t===Expice Date Fail==="; sound(500);delay(200);nosound(); }
        }
        else { cout<<"\n\n\t===login Fail==="; sound(500);delay(200);nosound();}
      }
    do{
       check_tag();
      }while(!kbhit()&&data!=255);
   }while(key!=27);
}
```

การทดลองที่ 33 :\tc\bin\mifare\iso14443\food1.cpp

ตัวอย่างการพัฒนาระบบศูนย์อาหาร

```
68  void select_card()
69  {
70     //Copy From File : CHECK2.CPP
71  }
72  void read_expire_date()
73  {
74     //Copy From File : READ_D4.CPP
75  }
76  void dec_value()
77  {
78     //Copy From File : DECREV10.CPP
79  }
80  void check_value()
81  {
82     //Copy From File : READ_V6.CPP
83  }
84  void login_sector()
85  {
86     //Copy From File : LOGIN3.CPP
87  }
88  void reset()
89  {
90    //Copy From File : CHECK2.CPP
91  }
92  void config_gpio()
93  {
94    //Copy From File : CHECKTAG.CPP
95  }
96  void check_tag()
97  {
98    //Copy From File : CHECKTAG.CPP
99  }
```



```
102 void save_flie()
103 {FILE *fptr;         char time_w[10]={" "};  char date_w[20]={" "};
104 struct time t;       struct date d;
105 gettime(&t);
106 strcpy(time_w," ");strcpy(date_w," ");
107 time_w[0]=...+(t.ti_hour/10); time_w[1]=...+(t.ti_hour%10); time_w[2]=':';
108 time_w[3]=...+(t.ti_min/10);  time_w[4]=...+(t.ti_min%10);  time_w[5]=':';
109 time_w[6]=...+(t.ti_sec/10);  time_w[7]=...+(t.ti_sec%10);
110 getdate(&d);
111 date_w[0]=48+(d.da_day/10); date_w[1]=48+(d.da_day%10); date_w[2]=':';
112 date_w[3]=48+(d.da_mon/10); date_w[4]=48+(d.da_mon%10); date_w[5]=':';
113 date_w[6]=48+(d.da_year/...);
114 date_w[7]=48+((d.da_year%...)/...);
115 date_w[8]=48+((d.da_year%...)/...);
116 date_w[9]=48+((d.da_year%...)/...);
117 fptr=.....(".......","...");
118 fputc('S',fptr);
119 .....('|',fptr);
120 .....(date_w,fptr);
121 fputc('|',fptr);
122 fputs(time_w,fptr);
123 fputs(sn_tag_char,fptr);
124 fputs(count_char,fptr);
125 fputc('|',fptr);
126 fputs(price_char,fptr);
127 fputc('|',fptr);
128 fputs(dec_char,fptr);
129 fputc('|',fptr);
130 fputc('Q',fptr);
131 fputc(0x0d,fptr);
132 .....(fptr);
133 }
```