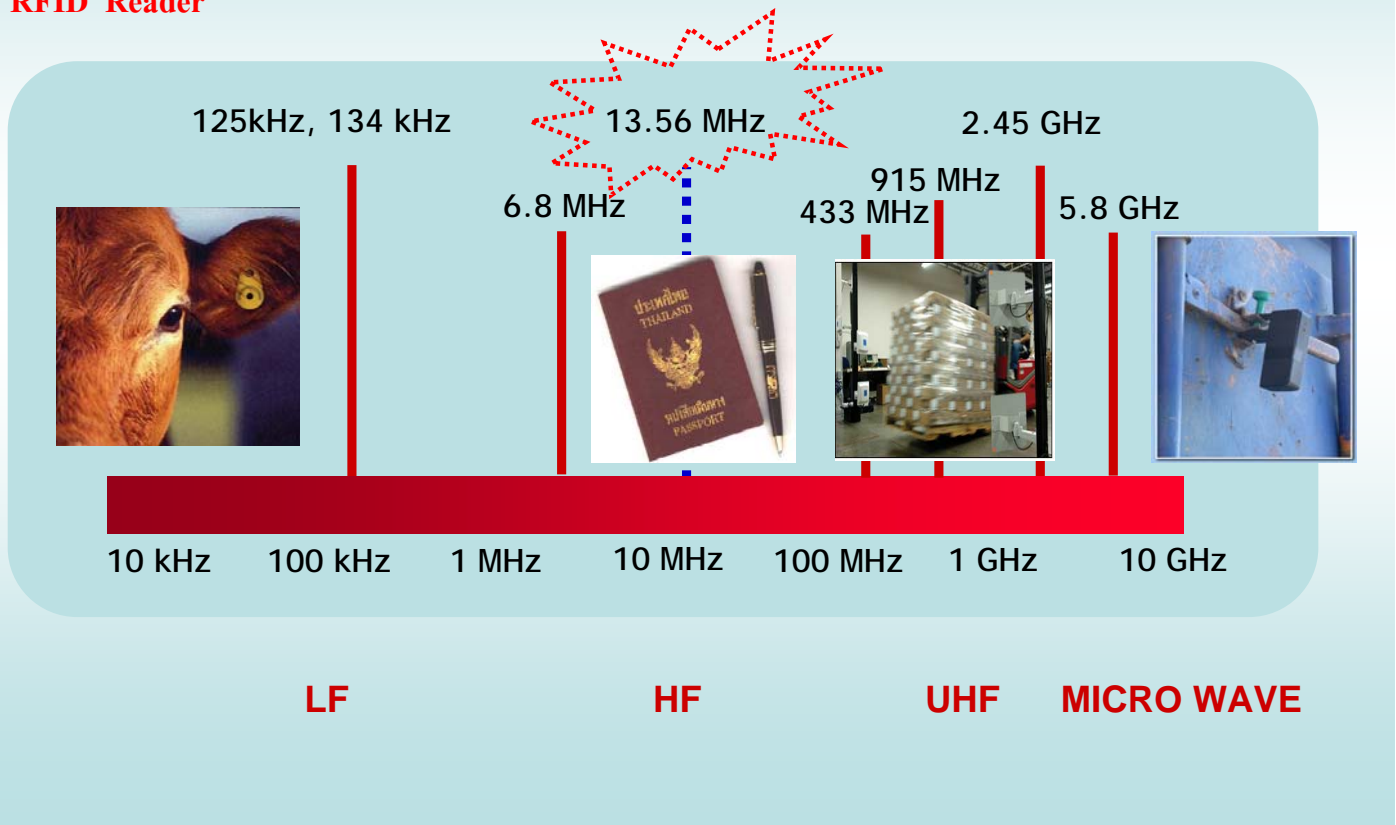
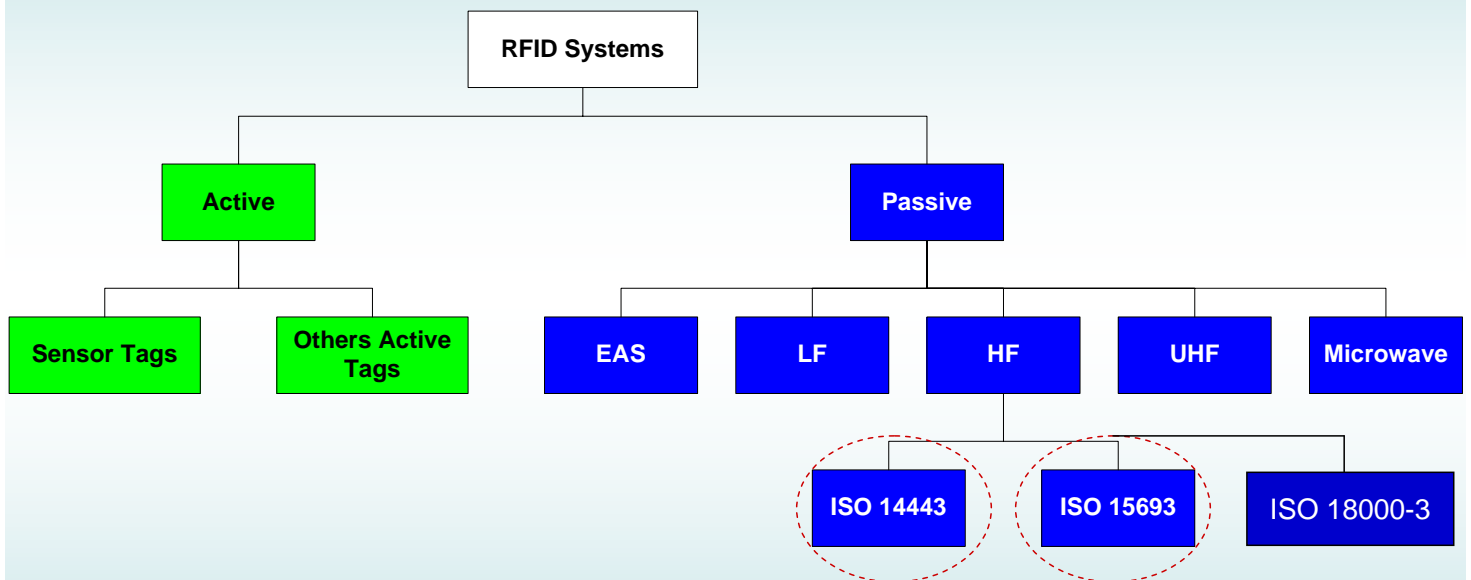




RFID Reader

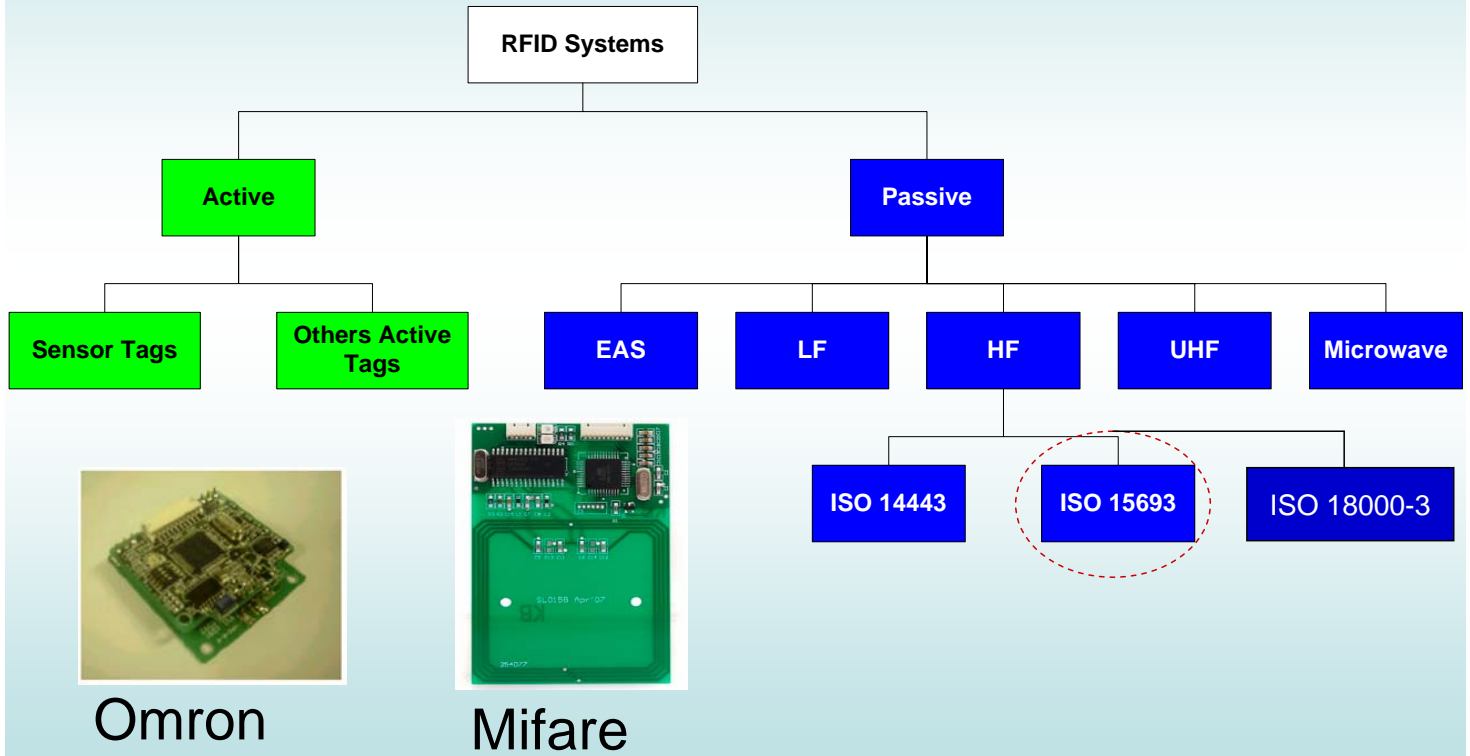


RFID Reader





RFID Reader



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

PIN	SYMBOL	TYPE	DESCRIPTION
1	TXD	Output	Serial output port
2	RXD	Input	Serial input port
3	VCC	PWR	Power Supply
4	GND	PWR	Ground
5	GND	PWR	Ground
6	VCC	PWR	Power Supply
7	PA0	Output	
8	PA1	Output	Tag detect signal:low level indicating tag in detection range, high level indicating tag out
9	PA2	Output	
10	PA3	Output	
11	PA4	Output	
12	PA5	Output	
13	PA6	Output	
14	PA7	Output	
15	GND	PWR	Ground



RFID Reader

4-2. Communication Format



Mifare

Host to Reader:

Header	Len	Command	Data	Checksum
--------	-----	---------	------	----------

Header: Communication header, 1 byte.
From host to module: 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 Header to Data inclusively, 1 byte.

Reader to Host:

Header	Len	Command	Status	Data	Checksum
--------	-----	---------	--------	------	----------

Header: Communication header, 1 byte.
From module to host: 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 Header to Data inclusively, 1 byte.



RFID Tag I.CODE

	Byte 0	Byte 1	Byte 2	Byte 3	
Block -4	UID0	UID1	UID2	UID3	Unique Identifier (lower bytes)
Block -3	UID4	UID5	UID6	UID7	Unique Identifier (higher bytes)
Block -2	Internally used	EAS	AFI	DSFID	EAS, AFI, DSFID
Block -1	00	00	00	00	Write Access Conditions
Block 0	X	X	X	X	User Data
Block 1	X	X	X	X	
Block 2	X	X	X	X	
Block 3	X	X	X	X	
Block 4	X	X	X	X	
Block 5	X	X	X	X	
Block 6	X	X	X	X	
Block 7	X	X	X	X	
...		...			
...		...			
...		...			
Block 27	X	X	X	X	User Data

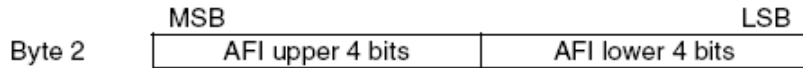
Read/write Block 0-27
รวม 28 page หรือ 112 byte



**หน่วยวิจัยเทคโนโลยีอาร์เอฟไอดี โปรแกรมวิศวกรรมและเทคโนโลยี
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม**

RFID Reader

AFI is a special area for enabling the user to identify a tag that is suitable to a specific user application.



AFI upper 4 bits	AFI lower 4 bits	Application area	Example/reference
0	0	All areas	Area not identified
X	0	X area	Selected extensively
X	Y	Y category of X area	
0	Y	Limited to Y category	
1	0,Y	Transportation	Mass-transit, bus, air plane
2	0,Y	Finance	Bank
3	0,Y	Recognition	Access control
4	0,Y	Telecommunication	Public telephone, CSM
5	0,Y	Medical care	
6	0,Y	Multimedia	Internet
7	0,Y	Game	
8	0,Y	Data storage	Portable file
9	0,Y	Logistics	
A	0,Y	Home delivery services	
B	0,Y	Mail	
C	0,Y	Airplane luggage	
D	0,Y	Reservation	
E	0,Y	Reservation	
F	0,Y	Reservation	

*Remarks: X=1 to F Y=1 to F



**หน่วยวิจัยเทคโนโลยีอาร์เอฟไอดี โปรแกรมวิศวกรรมและเทคโนโลยี
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม**

RFID Reader

DSFID

DSFID indicates how the data is configured in the memory.

Byte 3 MSB LSB

DSFID

Write-access conditions




The pages are write-inhibited permanently if they are so indicated in the memory map. The factory settings are as follow. If the bit of a particular page is 1, that page is write-protected.

	MSB				LSB			
Byte 0	0	0	0	0	0	0	0	0
	Page 03	Page 02	Page 01	Page 00				
Byte 1	0	0	0	0	0	0	0	0
	Page 0B	Page 0A	Page 09	Page 08	Page 07	Page 06	Page 05	Page 04
Byte 2	0	0	0	0	0	0	0	0
	Page 13	Page 12	Page 11	Page 10	Page 0F	Page 0E	Page 0D	Page 0C
Byte 3	0	0	0	0	0	0	0	0
	Page 1B	Page 1A	Page 19	Page 18	Page 17	Page 16	Page 15	Page 14



4-3. Command Overview

RFI

Command	Description
0x31	Get tag information
0x32	Get block security status
0x33	Read blocks
0x34	Write a data block
0x35	Write AFI
0x36	Write DSFID
0x37	Lock block 
0x38	Lock AFI 
0x39	Lock DSFID 
0x40	Control PA status
0xFF	Reset

Status Overview

Status	Description
0x00	Operation succeed
0x01	No tag
0x04	Read fail
0x05	Write fail
0x06	Unable to read after write
0x07	Read after write error
0xF0	Checksum error
0xF1	Command code error

```

1 #include <dos.h>    #include <bios.h>    #include <conio.h>
2 #include <iostream.h>    #include <time.h>
3 #define COM1    .....
4 #define DATA_READY 0x100
5 #define SETTINGS (...|...|...|...) // RS232 Sets the communications
6 void main()
7 {
8     char key;
9     int i;
10    int rf_reset_cmd[3]={.....};
11    int rf_check_sum;
12    bioscom(0, SETTINGS, COM1);
13    do
14    {
15        clrscr();
16        rf_check_sum=0;
17        cout<<"\n\tSend Command Reset Reader \n\t";
18        for(i=0;i<3;i++)
19        {
20            bioscom(1, rf_reset_cmd[i], COM1);
21            rf_check_sum^=rf_reset_cmd[i];
22            cout<<rf_reset_cmd[i];
23            cout<<',';
24        }
25        bioscom(1, rf_check_sum, COM1);
26        cout<<rf_check_sum;
27        cout<<"\n\t";
28        cout<<"\n\n\tExit Program Key ESC"<<endl;
29        key=getch();
30    }while(key!=27); // ESC = Char(0x27)
31 }
    
```

4-4-11. Reset

0xBA	Len	0xFF	Checksum
------	-----	------	----------

No return

การทดลองที่ 5 :tc\bin\mifare\iso15693\reset.cpp

4-4-1. Get tag information

0xBA	Len	0x31	Checksum
------	-----	------	----------

Return:

0xBD	Len	0x31	Status	UID	AFI	DSFID	Type	Checksum
------	-----	------	--------	-----	-----	-------	------	----------

Status: 0x00: Operation succeed
 0x01: No tag
 0x04: Read fail
 0xF0: Checksum error

UID: The Unique Identifier of card, 8 bytes
 AFI: The Application Family Identifier, 1byte
 DSFID: The Data Storage Format Identifier, 1byte
 Type: 0x31: Tag_it
 0x32: I.CODE SLI

การทดลองที่ 6 : \tc\bin\mifare\iso15693\inf.cpp

```

1  #include <dos.h>      #include <bios.h>      #include <conio.h>
2  #include <iostream.h>  #include <time.h>
3  void reset();
4  #define COM1  0          #define DATA_READY 0x100
5  #define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
6  void main()
7  {   time_t timef, timel;
8      float difft;
9      int len,i,status,index_data;
10     int rf_gettag_cmd[3]={.....};
11     int rf_check_sum,rf_return_cmd,rf_data[21];
12     char key;
13     bioscom(0, SETTINGS, COM1);
14     clrscr();
15     reset();
16     do
17     {   clrscr();
18         rf_check_sum=0;
19         cout<<"\n\tSend Command Get tag information\n\t";
20         for(i=0;i<3;i++)
21         {
22             bioscom(1, rf_gettag_cmd[i], COM1);
23             rf_check_sum^=rf_gettag_cmd[i];
24             cout<<rf_gettag_cmd[i];
25             cout<<' ';
26         }
27         bioscom(1, rf_check_sum, COM1);
28         cout<<rf_check_sum;
29         cout<<" "<<endl;

```

4-4-1. Get tag information

0xBA	Len	0x31	Checksum
------	-----	------	----------

การทดลองที่ 6 : \tc\bin\mifare\iso15693\inf.cpp

```

31     index_data=0;
32     len = 100;
33     do
34     {
35         status = bioscom(3, 0, COM1);
36         if (status & DATA_READY)
37         {
38             rf_return_cmd = bioscom(2, 0, COM1);
39             rf_data[index_data]=rf_return_cmd;
40             index_data++;
41         }
42         if (index_data==2) {len=rf_return_cmd+.....;
43             }
44     }while(!kbhit()&&index_data!=len);
45     cout<<"\n\tReturn Command Get tag information \n\t";
46     for(i=0;i<index_data;i++)
47     {
48         if(i>0)
49         {
50             cout<<" , ";
51         }
52         cout<<rf_data[i];
53     }
54     cout<<" " <<endl;

```

Reader to Host:

Header	Len	Command	Status	Data	Checksum
--------	-----	---------	--------	------	----------

Header: Communication header, 1 byte.
From module to host: 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 Header to Data inclusively, 1 byte.

```

58     if(index_data==.....&&rf_data[2]==.....&&rf_data[3]==.....)
59     {
60         cout<<"\n\tGet tag information success"<<endl;
61         cout<<"\n\tUID " <<rf_data[4]<<":"<<rf_data[5]<<":"<<rf_data[6]<<":"<<rf_data[7]
62             <<rf_data[8]<<":"<<rf_data[9]<<":"<<rf_data[10]<<":"<<rf_data[11]<<endl;
63         cout<<"\n\tAFI " <<rf_data[12]<<endl;
64         cout<<"\n\tDSFID " <<rf_data[13]<<endl;
65         switch(rf_data[.....])
66         {
67             case 0x31: cout<<"\n\tType Tag_it"<<endl; break;
68             case .....: cout<<"\n\tType I.CODE SLI"<<endl; break;
69         }
70     }
71     else
72     {
73         cout<<"\n\tGet tag information Fail"<<endl;
74         switch(rf_data[.....])
75         {
76             case .....: cout<<"\n\tNo Tag"<<endl; break;
77             case 0x04: cout<<"\n\tRead fail"<<endl; break;
78             case .....: cout<<"\n\tChecksum error"<<endl; break;
79         }
80     }
81     cout<<"\n\tExit Program Key ESC"<<endl;
82     key=getch();
83     }while(key!=27);
84 }

```

Return:

0xBD	Len	0x31	Status	UID	AFI	DSFID	Type	Checksum
------	-----	------	--------	-----	-----	-------	------	----------

Status: 0x00: Operation succeed

0x01: No tag

0x04: Read fail

0xF0: Checksum error

UID: The Unique Identifier of card, 8 bytes

AFI: The Application Family Identifier, 1byte

DSFID: The Data Storage Format Identifier, 1byte

Type: 0x31: Tag_it

0x32: I.CODE SLI

```

87 void reset()
88 {
89     time_t timef, timel;
90     float difft;
91     int status;
92     int rf_check_sum,i;
93     int rf_reset_cmd[3]={.....,.....,.....};
94     rf_check_sum=0;
95     cout<<"\n\tSend Command Reset Reader := ";
96     for(i=0;i<3;i++)
97     {
98         bioscom(1, rf_reset_cmd[i], COM1);
99         rf_check_sum^=rf_reset_cmd[i];
100        cout<<rf_reset_cmd[i];
101        cout<<' , ' ;
102    }
103    bioscom(1, rf_check_sum, COM1);
104    cout<<rf_check_sum;
105    timef = time(NULL);
106    do
107    {
108        status = bioscom(3, 0, COM1);
109        if (status & DATA_READY)
110        { i = bioscom(2, 0, COM1);
111        }
112        timel = time(NULL);
113        difft=difftime(timel,timef);
114    }while(!kbhit()&&difft<=2);
115    cout<<" " <<endl;
116    cout<<"\n\tExit Command Reset ";
117    delay(1000);
118 }

```

4-4-11. Reset

0xBA	Len	0xFF	Checksum
------	-----	------	----------

No return

การทดลองที่ 6 : \tc\bin\mifare\iso15693\inf.cpp

```

1 #include <dos.h>           #include <bios.h>   #include <conio.h>
2 #include <iostream.h>     #include <time.h>   #include <string.h>
3 #include <process.h>
4 void reset();
5 void get_sn_tag();
6 #define COM1             0
7 #define DATA_READY 0x100
8 #define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
9 int tag_sn_state; // void get_sn_tag();
10 int sn_tag[8]; // void get_sn_tag();
11 char sn_tag_char[40]={" "}; // void get_sn_tag();
12 void main()
13 { char key;
14   bioscom(0, SETTINGS, COM1);
15   clrscr();
16   reset();
17   clrscr();
18   do
19   { get_sn_tag();
20     if (tag_sn_state==1)
21     {
22         cout<<"\n\t\t Tag SN ="<<sn_tag_char<<"==\n" ;
23         sound(1000);delay(200);nosound();
24         sound(500);delay(200);nosound();
25     }
26     else
27     {
28         cout<<"\n\t\t Tag SN =ERROR==\n" ;
29         sound(500);delay(200);nosound();
30     }
31
32     cout<<"\n\n\tExit Program Key ESC" <<endl;
33     key=getch();
34
35 }while(key!=27);
36 }

```

การทดลองที่ 7 : \tc\bin\mifare\iso15693\read_sn.cpp


```

38 void get_sn_tag()
39 { int len,i,status,index_data;
40 int rf_gettag_cmd[3]={.....,.....,.....};
41 int rf_check_sum,rf_return_cmd,rf_data[21];
42 rf_check_sum=0;
43 for(i=0;i<3;i++)
44 { bioscom(1, rf_gettag_cmd[i], COM1);
45   rf_check_sum^=rf_gettag_cmd[i];
46 }
47 bioscom(1, rf_check_sum, COM1);
48 index_data=0;
49 len = 100;
50 do
51 { status = bioscom(3, 0, COM1);
52   if (status & DATA_READY)
53   { rf_return_cmd = bioscom(2, 0, COM1);
54     rf_data[index_data]=rf_return_cmd;
55     index_data++;
56   }
57   if (index_data==2) {len=rf_return_cmd+...;}
58 }while (!kbhit()&&index_data!=len);
59 if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
60 { int j=0; int k=0;
61   sn_tag_char[j++]='|';
62   for(i=4;i<12;i++)
63     { sn_tag[k]=rf_data[i];
64       sn_tag_char[j++]=...+(rf_data[i]/...);
65       sn_tag_char[j++]=...+((rf_data[i]%100)/10);
66       sn_tag_char[j++]=...+(rf_data[i]%10);
67       sn_tag_char[j++]='|';
68       k++;
69     }
70   tag_sn_state=1;
71 }
72 else {tag_sn_state=0;}
73 }

```

```

76 void reset()
77 {
78     int rf_reset_cmd[3]={.....,.....};
79 }

```

การทดลองที่ 7 : \tc\bin\mifare\iso15693\read_sn.cpp

4-4-2. Get block security status

0xBA	Len	0x32	block	number	Checksum
------	-----	------	-------	--------	----------

block: Start block number
 number: Number of blocks to be read

Return:

0xBD	Len	0x32	Status	Data	Checksum
------	-----	------	--------	------	----------

Status: 0x00: Operation succeed
 0x01: No tag
 0x04: Read fail
 0xF0: Checksum error
 Data: Security status, 1 byte to 1 block

Write-access conditions

The pages are write-inhibited permanently if they are so indicated in the memory map. The factory settings are as follow. If the bit of a particular page is 1, that page is write-protected.

	MSB				LSB			
Byte 0	0	0	0	0	0	0	0	0
	Page 03	Page 02	Page 01	Page 00				
Byte 1	0	0	0	0	0	0	0	0
	Page 0B	Page 0A	Page 09	Page 08	Page 07	Page 06	Page 05	Page 04
Byte 2	0	0	0	0	0	0	0	0
	Page 13	Page 12	Page 11	Page 10	Page 0F	Page 0E	Page 0D	Page 0C
Byte 3	0	0	0	0	0	0	0	0
	Page 1B	Page 1A	Page 19	Page 18	Page 17	Page 16	Page 15	Page 14

การทดลองที่ 8 : \tc\bin\mifare\iso15693\security.cpp

```

1 #include <dos.h>           #include <bios.h>   #include <conio.h>
2 #include <iostream.h>     #include <time.h>
3 void reset();
4 #define COM1             0
5 #define DATA_READY 0x100
6 #define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
7 void main()
8 {
9     time_t timef, time1;
10    float difft;
11    int len,i,status,index_data;
12    int blockstart=...;
13    int numberblock=...;
14    int numdata;
15    int rf_readblock_cmd[5]={.....};
16    int rf_check_sum,rf_return_cmd,rf_data[21];
17    char key;
18    bioscom(0, SETTINGS, COM1);
19    rf_readblock_cmd[3]=blockstart;
20    rf_readblock_cmd[4]=numberblock;
21    numdata=numberblock+5;
22    clrscr();

```

0xBA	Len	0x32	block	number	Checksum
------	-----	------	-------	--------	----------

block: Start block number

number: Number of blocks to be read

การทดลองที่ 8 :tc\bin\mifare\iso15693\security.cpp

```

25 reset();
26 do
27 {
28     clrscr();
29     rf_check_sum=0;
30     cout<<"\n\tSend Command Get block security status blocks:"<<blockstart<<
31         " to "<<blockstart+numberblock<<"\n\t";
32     for(i=0;i<5;i++)
33     {
34         bioscom(1, rf_readblock_cmd[i], COM1);
35         rf_check_sum^=rf_readblock_cmd[i];
36         cout<<rf_readblock_cmd[i];
37         cout<<',';
38     }
39     bioscom(1, rf_check_sum, COM1);
40     cout<<rf_check_sum;
41     cout<<" "<<endl;
42     index_data=0;
43     len=100;
44     do
45     {
46         status = bioscom(3, 0, COM1);
47         if (status & DATA_READY)
48         {
49             rf_return_cmd = bioscom(2, 0, COM1);
50             rf_data[index_data]=rf_return_cmd;
51             index_data++;
52         }
53         if (index_data==2) {len=rf_return_cmd+.....;
54                             }
55     }while(!kbhit()&&index_data!=len);

```

การทดลองที่ 8 :tc\bin\mifare\iso15693\security.cpp

Reader to Host:

Header	Len	Command	Status	Data	Checksum
--------	-----	---------	--------	------	----------

Header: Communication header, 1 byte.
From module to host: 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 Header to Data inclusively, 1 byte.

```

58     cout<<"\n\tReturn Get block security status blocks:"<<blockstart<<" to "
59           <<blockstart+numberblock<<"\n\t";
60     for(i=0;i<index_data;i++)
61     {
62         if(i>0)
63         {
64             cout<<",";
65         }
66         cout<<rf_data[i];
67     }
68     cout<<" "<<endl;
69     if(index_data==numdata&&rf_data[2]==...&&rf_data[3]==...)
70     {
71         cout<<"\n\tGet block security status success"<<endl;
72     }
73     else
74     {
75         cout<<"\n\tGet block security status Fail"<<endl;
76         switch(rf_data[...])
77         {
78             case 0x01: cout<<"\n\tNo Tag"<<endl; break;
79             case 0x04: cout<<"\n\tRead fail"<<endl; break;
80             case 0xF0: cout<<"\n\tChecksum error"<<endl; break;
81         }
82     }
83     cout<<"\n\n\tExit Program Key ESC"<<endl;
84     key=getch();
85 }while(key!=27);
86 }

```

Return:					
0xBD	Len	0x32	Status	Data	Checksum
Status:	0x00:	Operation succeed			
	0x01:	No tag			
	0x04:	Read fail			
	0xF0:	Checksum error			
Data:	Security status, 1 byte to 1 block				

80

การทดลองที่ 8 :tc\bin\mifare\iso15693\security.cpp

4-4-3. Read blocks

0xBA	Len	0x33	block	number	Checksum
------	-----	------	-------	--------	----------

block: Start block number

number: Number of blocks to be read, max 16 blocks

Return:

0xBD	Len	0x33	Status	Data	Checksum
------	-----	------	--------	------	----------

Status: 0x00: Operation succeed

0x01: No tag

0x04: Read fail

0xF0: Checksum error

Data: Blocks data returned if operation succeeds, 4 bytes to 1 block

81

การทดลองที่ 9 :tc\bin\mifare\iso15693\read.cpp

```

1 #include <dos.h> #include <bios.h> #include <conio.h> #include <iostream.h>
2 #include <time.h>
3 void reset();
4 #define COM1 0
5 #define DATA_READY 0x100
6 #define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
7 void main()
8 {
9     time_t timef, time1; float difft; int len,i,status,index_data;
10    int blockstart=....;
11    int numberblock=....;
12    int numdata;
13    int rf_readblock_cmd[5]={.....};
14    int rf_check_sum,rf_return_cmd,rf_data[21];
15    char data_in_block[64]={" "};
16    char key;
17    bioscom(0, SETTINGS, COM1);
18    rf_readblock_cmd[3]=blockstart;
19    rf_readblock_cmd[4]=numberblock;
20    numdata=(numberblock*4)+5;
21    clrscr();
22    reset();
23    do
24    { clrscr();
25      rf_check_sum=0;
26      cout<<"\n\tSend Command Read blocks:"<<blockstart<<" to "<<blockstart+numberblock<<"\n\t";
27      for(i=0;i<5;i++)
28      {
29        bioscom(1, rf_readblock_cmd[i], COM1);
30        rf_check_sum^=rf_readblock_cmd[i];
31        cout<<rf_readblock_cmd[i];
32        cout<<',';
33      }
34      bioscom(1, rf_check_sum, COM1);
35      cout<<rf_check_sum;
36      cout<<" "<<endl;

```

4-4-3. Read blocks

0xBA	Len	0x33	block	number	Checksum
------	-----	------	-------	--------	----------

block: Start block number

number: Number of blocks to be read, max 16 blocks

การทดลองที่ 9 :Atc\bin\mifare\iso15693\read.cpp

```

40    index_data=0; len=100;
41    do
42    { status = bioscom(3, 0, COM1);
43      if (status & DATA_READY)
44      { rf_return_cmd = bioscom(2, 0, COM1);
45        rf_data[index_data]=rf_return_cmd;
46        index_data++;
47      }
48      if (index_data==2) {len=rf_return_cmd+2;}
49    }while(!kbhit()&&index_data!=len);
50    cout<<"\n\tReturn Read blocks:"<<blockstart<<" to "<<blockstart+numberblock<<"\n\t";
51    for(i=0;i<index_data;i++)
52    { if(i>0)
53      {cout<<",";};
54      cout<<rf_data[i];
55    }
56    cout<<" "<<endl;
57    if(index_data==numdata&&rf_data[2]==.....&&rf_data[3]==.....)
58    { cout<<"\n\tRead blocks success"<<endl;
59      int k=0;
60      for(i=4;i<index_data-1;i++) { data_in_block[k]=rf_data[i]; k++; }
61      cout<<"\n\tData In Block:"<<blockstart<<" to "<<blockstart+numberblock<<"\n\t=="
62      <<data_in_block<<"=={Char}";
63    }
64    else
65    { cout<<"\n\tRead blocks Fail"<<endl;
66      switch(rf_data[....])
67      { case 0x01: cout<<"\n\tNo Tag"<<endl; break;
68        case 0x04: cout<<"\n\tRead fail"<<endl; break;
69        case 0xF0: cout<<"\n\tChecksum error"<<endl; break;
70      }
71    }
72    cout<<"\n\n\tExit Program Key ESC"<<endl;
73    key=getch();
74    }while(key!=27);
75 }

```

Return:

0xBD	Len	0x33	Status	Data	Checksum
Status:	0x00:	Operation succeed			
	0x01:	No tag			
	0x04:	Read fail			
	0xF0:	Checksum error			
Data:	Blocks data returned if operation succeeds, 4 bytes to 1 block				

```

81 void reset()
82 {
83
84 }

```

การทดลองที่ 9 :Atc\bin\mifare\iso15693\read.cpp

```

1 #include <dos.h>      #include <bios.h>      #include <conio.h>      #include <iostream.h>
2 #include <time.h>     #include <string.h>
3 void read_name();
4 void reset();
5 #define COM1          0
6 #define DATA_READY  0x100
7 #define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
8 int read_id_name_state; //void read_name();
9 char id_tag_char[15]={" "}; // void read_name();
10 char name_tag_char[64]={" "}; // void read_name();
11 void main()
12 {
13     char key;   bioscom(0, SETTINGS, COM1);
14     reset();
15     do
16     { read_name();
17       cout<<"\n\t\tRead_id_name_state : "<<read_id_name_state<<endl;
18       if ((read_id_name_state==1))
19       { cout<<"\t\tTag ID ="<<id_tag_char<<"==\n";
20         cout<<"\t\tTag Name ="<<name_tag_char<<"==\n";
21         sound(1000);delay(200);nosound();
22         sound(500);delay(200);nosound();
23       }
24       else
25       { cout<<"\n\t\tTag ID = Read Data Error==\n";
26         sound(500);delay(200);nosound();
27       }
28       cout<<"\n\n\tExit Program Key ESC"<<endl;
29       key=getch();
30     }while(key!=27);
31 }

```

การทดลองที่ 10 : \tc\bin\mifare\iso15693\read_na.cpp

```

33 void read_name()
34 { int len,i,status,index_data;
35   int blockstart=....;   int numberblock=....;
36   int numdata;           int rf_readblock_cmd[5]={.....};
37   int rf_check_sum,rf_return_cmd,rf_data[40];
38   bioscom(0, SETTINGS, COM1);
39   rf_readblock_cmd[3]=blockstart;   rf_readblock_cmd[4]=numberblock;
40   numdata=(numberblock*4)+5;   strcpy(name_tag_char, " ");   strcpy(id_tag_char, " ");
41   rf_check_sum=0;
42   for(i=0;i<5;i++)
43   { bioscom(1, rf_readblock_cmd[i], COM1);
44     rf_check_sum^=rf_readblock_cmd[i];
45   }
46   bioscom(1, rf_check_sum, COM1);
47   index_data=0;
48   len=100;
49   do
50   { status = bioscom(3, 0, COM1);
51     if (status & DATA_READY)
52     {
53       rf_return_cmd = bioscom(2, 0, COM1);
54       rf_data[index_data]=rf_return_cmd;
55       index_data++;
56     }
57     if (index_data==2) {len=rf_return_cmd+2; }
58   }while(!kbhit()&&index_data!=len);
59   if(index_data==numdata&&rf_data[2]==.....&&rf_data[3]==.....)
60   { int k=0;
61     for(i=4;i<.....;i++){ if(rf_data[i]!=' ')
62                           { id_tag_char[k]=rf_data[i];   k++; }
63                       }
64     k=0;
65     for(i=16;i<...;i++) { name_tag_char[k]=rf_data[i];   k++; }
66     read_id_name_state=1;
67   }
68   else
69   { read_id_name_state=0; }
70 }

```

4-4-3. Read blocks

0xBA	Len	0x33	block	number	Checksum
block: Start block number					
number: Number of blocks to be read, max 16 blocks					

Return:

0xBD	Len	0x33	Status	Data	Checksum
Status: 0x00: Operation succeed					
0x01: No tag					
0x04: Read fail					
0xF0: Checksum error					
Data: Blocks data returned if operation succeeds, 4 bytes to 1 block					

```

74 void reset()
75 {
76
77 }

```

การทดลองที่ 10 : \tc\bin\mifare\iso15693\read_na.cpp

4-4-4. Write data to a block

0xBA	Len	0x34	Block	Data	Checksum
------	-----	------	-------	------	----------

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

Data: The data to write, 4 bytes.

Return:

0xBD	Len	0x34	Status	Data	Checksum
------	-----	------	--------	------	----------

Status: 0x00: Operation succeed
 0x01: No tag
 0x05: Write fail
 0x06: Unable to read after write
 0x07: Read after write error
 0xF0: Checksum error

Data: Block data written if operation is succeeds, 4 bytes.

การทดลองที่ 11 :tc\bin\mifare\iso15693\write.cpp

```

1 #include <dos.h>           #include <bios.h>           #include <conio.h>
2 #include <iostream.h>     #include <time.h>
3 void reset();
4 #define COM1              0
5 #define DATA_READY 0x100
6 #define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
7 void main()
8 {
9     time_t timef, timel;    float diff;
10    int len,i,status,index_data;
11    int rf_writeblock_cmd[8]={...,'.',...,'.',...,'.',...,'.',...};
12    int rf_check_sum,rf_return_cmd,rf_data[21];
13    char key;
14    bioscom(0, SETTINGS, COM1);
15    clrscr();    reset();
16    do
17    {
18        clrscr();    rf_check_sum=0;
19        cout<<"\n\tSend Command Write blocks:"<<rf_writeblock_cmd[3]<<"\n\t";
20        for(i=0;i<8;i++)
21        { bioscom(1, rf_writeblock_cmd[i], COM1);
22          rf_check_sum^=rf_writeblock_cmd[i];
23          cout<<rf_writeblock_cmd[i];
24          cout<<',';
25        }
26        bioscom(1, rf_check_sum, COM1);
27        cout<<rf_check_sum;
28        cout<<" " <<endl;

```

4-4-4. Write data to a block

0xBA	Len	0x34	Block	Data	Checksum
------	-----	------	-------	------	----------

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

Data: The data to write, 4 bytes.

การทดลองที่ 11 :tc\bin\mifare\iso15693\write.cpp

```

32     index_data=0;
33     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         }
41       if (index_data==2) {len=rf_return_cmd+2;}
42     }while(!kbhit()&&index_data!=len);
43     cout<<"\n\tReturn Write blocks:"<<rf_writeblock_cmd[3]<<"\n\t";
44     for(i=0;i<index_data;i++)
45     {   if(i>0)
46         { cout<<","; }
47         cout<<rf_data[i];
48     }
49     cout<<" ";<<endl;
50     if(index_data==...&&rf_data[2]==...&&rf_data[3]==...)
51     { cout<<"\n\tWrite blocks:"<<rf_writeblock_cmd[3]<<" = success"<<endl;
52     }
53     else
54     { cout<<"\n\tWrite blocks:"<<rf_writeblock_cmd[3]<<" = Fail"<<endl;
55       switch(rf_data[3])
56       {
57         case 0x01 :cout<<"\n\t\tNo Tag"<<endl;break;
58         case 0x05 :cout<<"\n\t\tWrite fail"<<endl;break;
59         case 0x06 :cout<<"\n\t\tUnable to read after write"<<endl;break;
60         case 0x07 :cout<<"\n\t\tRead after write error"<<endl;break;
61         case 0xF0 :cout<<"\n\t\tChecksum error"<<endl;break;
62       }
63     }
64     cout<<"\n\tExit Program Key ESC"<<endl;
65     key=getch();
66 }while(key!=27);
67 }

```

Return:

0xBD	Len	0x34	Status	Data	Checksum
Status: 0x00:			Operation succeed		
0x01:			No tag		
0x05:			Write fail		
0x06:			Unable to read after write		
0x07:			Read after write error		
0xF0:			Checksum error		
Data:	Block data written if operation is succeeds, 4 bytes.				

88

```

70 void reset()
71 {
72
73 }

```

การทดลองที่ 11 : \tc\bin\mifare\iso15693\write.cpp

```

1 #include <dos.h>           #include <bios.h>           #include <conio.h>
2 #include <iostream.h>     #include <time.h>
3 void config_gpio();
4 void check_tag();
5 #define COM1 0
6 #define DATA_READY 0x100
7 #define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
8 int data;// void check_tag();
9 void main()
10 { config_gpio();
11   clrscr();
12   do
13   {   check_tag();
14       cout<<"\n\t\tGPIO PORT[7..0] >>"<<data;
15       delay(100);
16   }while(!kbhit());
17 }
18
19 void reset()
20 {
21 }
22
23 void config_gpio()
24 { outportb(0x22,...); // Unlock Configuration
25   outportb(0x23,...); // Unlock Configuration
26   outportb(0x22,...); // Set GPIO[7..0]
27   outportb(0x23,...); // Set 1111 1110= IN
28 }
29
30 void check_tag()
31 { outportb(0x22,...); // Set GPIO[7..0] input port
32   data= inportb(0x23); // data = GPIO[7..0]
33 }

```

GPIO CONFIG
Unlock Config

outportb(0x22,...0x13...);
outportb(0x23,...0xc5...);

IN/OUT Set

outportb(0x22,0x4e);
outportb(0x23,0xff); | outportb(0x23,0x00);

outportb(0x22,0x47); | outportb(0x22,0x46);
outportb(0x23, data.); | data = inportb(0x23);

OUT

IN

GPIO7..0

การทดลองที่ 12 : \tc\bin\mifare\iso15693\checktag.cpp

```

1 #include <dos.h>           #include <bios.h>       #include <conio.h>
2 #include <iostream.h>     #include <time.h>
3 void config_gpio();
4 void check_tag();
5 #define COM1              0
6 #define DATA_READY 0x100
7 #define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
8 int data; // void check_tag();
9 void main()
10 {
11     config_gpio();         clrscr();
12     do
13     {
14         do{
15             check_tag();
16             cout<<"\n\t\tWait Insert Tag loop1 >>"<<data;
17             delay(100);
18         }while(!kbhit()&&data!=....);
19         cout<<"\n\t\tProcess loop2 >>" ;
20         delay(100);
21         cout<<"\n\t\tProcess loop2 >>" ;
22         do{
23             check_tag();
24             cout<<"\n\t\tCheck Tag Out loop3 >>"<<data;
25             delay(100);
26         }while(!kbhit()&&data!=....);
27         delay(100);
28     }while(!kbhit());
29 }
--

```

```

32 void reset()
33 {
34 }
35 }
36 void config_gpio()
37 {
38     outporth(0x22,....); // Unlock Configuration
39     outporth(0x23,....); // Unlock Configuration
40     outporth(0x22,....); // Set GPIO[7..0]
41     outporth(0x23,....); // Set 1111 1110= IN
42 }
43 }
44 void check_tag()
45 {
46     outporth(0x22,....); // Set GPIO[7..0] input port
47     data= inporth(....); // data = GPIO[7..0]
48 }

```

การทดลองที่ 13 : \tc\bin\mifare\iso15693\ch_tag1.cpp

ตัวอย่างการพัฒนาระบบบลงเวลา

```

2 #include <dos.h>           #include <bios.h>       #include <conio.h>
3 #include <iostream.h>     #include <time.h>       #include <string.h>
4 #include <process.h>      #include <stdio.h>
5 void read_name();        void get_sn_tag();      void config_gpio();
6 void check_tag();       void save_flie();      void reset();
7 #define COM1            0
8 #define DATA_READY 0x100
9 #define SETTINGS ( 0xE0 | 0x03 | 0x00 | 0x00)
10 int data;
11 int tag_sn_state,read_id_name_state;
12 int sn_tag[8];
13 char sn_tag_char[40]={ " " };      char name_tag_char[64]={ " " };      char id_tag_char[15]={ " " };

```

การทดลองที่ 14 : \tc\bin\mifare\iso15693\timein1.cpp

ตัวอย่างการพัฒนาาระบบบลูทูธ

```

17 void main()
18 { char key;
19   bioscom(0, SETTINGS, COM1);
20   config_gpio();
21   clrscr();      reset();          clrscr();
22   cout<<"\n\t\t\tExit Program Key ESC"<<endl;
23   do
24   {   do{ check_tag();
25       }while(!kbhit()&&data!=254);
26       if(kbhit()) key=getch();
27       if (data==254)
28       {   get_sn_tag();
29           read_name();
30           if ((tag_sn_state==...)&&(read_id_name_state==...))
31           {   cout<<"\n\t\t\tTag SN ="<<sn_tag_char<<"==\n" ;
32               cout<<"\n\t\t\tTag Name ="<<id_tag_char<<"==\n" ;
33               cout<<"\n\t\t\tTag Name ="<<name_tag_char<<"==\n" ;
34               save_flie();
35               sound(1000);delay(200);nosound();
36               sound(500);delay(200);nosound();
37           }
38           else
39           {   cout<<"\n\t\t\tTag ID = Read Data Error==\n" ;
40               sound(500);delay(200);nosound();
41           }
42       }
43       do{ check_tag();
44           }while(!kbhit()&&data!=255);
45       if(kbhit()) key=getch();
46   }while(key!=27);
47 }

```

การทดลองที่ 14 : \tc\bin\mifare\iso15693\timein1.cpp

ตัวอย่างการพัฒนาาระบบบลูทูธ

```

52 void read_name()
53 {
54     //Copy From File : READ_NA.CPP
55 }
56
57 void get_sn_tag()
58 {
59     //Copy From File : READ_SN.CPP
60 }
61 }
62
63 void reset()
64 {
65     //Copy From File : READ_SN.CPP
66 }
67 void config_gpio()
68 {
69     //Copy From File : CH_TAG1.CPP
70 }
71
72 void check_tag()
73 {
74     //Copy From File : CH_TAG1.CPP
75 }

```

การทดลองที่ 14 : \tc\bin\mifare\iso15693\timein1.cpp

```
77 void save_flie()
78 { FILE *fptr;
79   char time_w[10]={" "};
80   char date_w[20]={" "};
81   struct time t;
82   struct date d;
83   gettime(&t);
84   strcpy(time_w, " ");strcpy(date_w, " ");
85   time_w[0]=...+(t.ti_hour/10); time_w[1]=...+(t.ti_hour%10); time_w[2]=': ';
86   time_w[3]=...+(t.ti_min/10); time_w[4]=...+(t.ti_min%10); time_w[5]=': ';
87   time_w[6]=...+(t.ti_sec/10); time_w[7]=...+(t.ti_sec%10);
88   getdate(&d);
89   date_w[0]=48+(d.da_day/10); date_w[1]=48+(d.da_day%10); date_w[2]=': ';
90   date_w[3]=48+(d.da_mon/10); date_w[4]=48+(d.da_mon%10); date_w[5]=': ';
91   date_w[6]=48+(d.da_year....);
92   date_w[7]=48+(d.da_year....);
93   date_w[8]=48+(d.da_year....);
94   date_w[9]=48+(d.da_year....);
95   fptr=fopen(".....",".....");
96   .....('S',fptr);
97   .....(sn_tag_char,fptr);
98   .....(id_tag_char,fptr);
99   fputc('|',fptr);
100  fputs(name_tag_char,fptr);
101  fputc('|',fptr);
102  fputs(time_w,fptr);
103  fputc('|',fptr);
104  fputs(date_w,fptr);
105  fputc('|',fptr);
106  fputc('Q',fptr);
107  fputc(0x0d,fptr);
108  .....(fptr);
109 }
```