



## DSock คืออะไร

WinSock คืออะไร เป็นส่วนที่ออกแบบมาสำหรับผู้พัฒนาโปรแกรม เพื่อติดต่อสื่อสารผ่านเครือข่าย บนระบบปฏิบัติการไมโครซอฟท์วินโดวส์

DSock.lib ถูกพัฒนาและออกแบบโดยกลุ่ม DMP ประมาณปี ค.ศ. 1998(2541) DSock คือระบบปฏิบัติการคอส TCP / IP ไลบรารีสำหรับผู้ซึ่งผู้ใช้ต้องการที่จะโปรแกรมซอฟต์แวร์ที่ติดต่อเครือข่ายภายใต้ระบบปฏิบัติการคอส ซึ่งจะมีการจัดเตรียม C ฟังก์ชันสำหรับนักเขียนโปรแกรมที่จะสร้างโปรแกรมอินเทอร์เน็ต V0.10 – V0.61(Support Ethernet in Vortex86SX)

Down load test Vesion : <http://www.dmp.com.tw/tech/dmp-lib/dsock/>



## Specifications

### Hardware Requirement

- DM&P M6117D/Vortex86/Vortex86SX signal board computers
- Ethernet interface.
- 512 KB RAM
- 64 KB Disk space
- Realtek 8019/8139/AX88796 10/100 Base-T.

### Packet Driver Requirement

- RTL8019/8139/NE2000 compatible
- AX88796
- R6040 in Vortex86SX

### Operating System

- X-DOS
- DR-DOS
- MS-DOS
- FreeDOS

### Protocols

- IP / ICMP / ARP
- TCP / UDP
- BOOTP / DHCP
- SMTP / HTTP / FTP / TELNET (Example)



## การติดตั้งไควเวอร์ก่อนใช้งาน (ติดตั้งทุกครั้งเมื่อเปิดเครื่องใหม่)

### Vortex86SX

For Vortex86SX with 100M LAN, you should use R6040 packet driver:

```
C:\DSOCK\DEMO\EXE>r6040pd 0x62
```

where software interrupt 62H can be any number between 60H and 70H. Or run pktdrv4.bat in directory "EXE" to load packet driver:

## Library Functions(initialize and close Dsock lib)

Function	Description
DSock_Open() ★	Open the socket library.
DSock_Close() ★	Close the socket library.
DSock_DoBootp()	Get Network setup from BOOTP/DHCP server.
DSock_LoadConfigFile() ★	Load configuration file.



## Library Functions Use DSock functions to set/get network configuration when you start Dsock

Function	Description
DSock_GetMacAddr() ★	Get MAC address from network card.
DSock_GetHostIp()	Get host IP address.
DSock_SetHostIp()	Set host IP address.
DSock_GetNetmask()	Get netmask of local host.
DSock_SetNetmask()	Set netmask of local host.
DSock_GetGateway()	Get gateway IP address.
DSock_AddGateway()	Add gateway to DSock.
DSock_GetDomainNameServer()	Get domain name server from DSock.
DSock_AddDomainNameServer()	Add domain name server to DSock.
DSock_Resolve()	Convert domain name to IP address.



Those **functions help** programmer to convert host to network byte ordering and get IP address

Function	Description
inet_ntoa()	Convert a network address into a string in dot notation. ★
inet_addr()	Convert a string containing a dotted IP address into a DWORD. ★
ntohs()	Convert a word from network to host byte order.
ntohl()	Convert a DWORD word from network to host byte order.
htons()	Convert a word from host to network byte order.
htonl()	Convert a double word from host to network byte order.



**Socket Functions** Use those function to establish a connection. See server & client section to get more

Function	Description
SocketCreate()	Create a socket. ★
SocketDestory()	Release a socket. ★
SocketClose()	Close a socket. ★
SocketAbort()	Abort a socket.
SocketBind()	Associate a local address with a socket. ★
SocketListen()	Establish a socket to listen for incoming connection. ★
SocketAccept()	Accept a connection on a socket. It's a non-blocking function. ★
SocketConnect()	Establish a connection with a peer.
SocketIsConnected()	Check connection of a socket.
SocketIsTcpPortUsed()	Is this port number used by DSOCK TCP sockets?
SocketIsUdpPortUsed()	Is this port number used by DSOCK UDP sockets?
SocketFindFreeTcpPort()	Find a free TCP port.
SocketFindFreeUdpPort()	Find a free UDP port.
SocketFlush()	Send pending data.
SocketFlushNext()	Cause next transmission to have a flush.



**Socket Functions** When socket connection is established (UDP is not need connection), use those functions to transmit data

Function	Description
SocketSend()	Send data to a connected socket.
SocketSend2()	Non-blocking version of SocketSend().
SocketRecv()	Receive data from a socket.
SocketRecv2()	Non-blocking function of SocketRecv().
SocketPutChar()	Write a character to a socket. ★
SocketGetChar()	Read a character from a socket. ★
SocketGetString()	Read a string from a socket ★
SocketPutString()	Write a string to a socket. ★
SocketDataReady()	Check incoming data of a socket. ★



int SocketSendTo(SOCKET s,DWORD dwAddr,WORD wPort,BYTE *pby,int nLen)	
<b>Description:</b>	Use UDP socket to send data.
<b>Arguments:</b>	<p><b>Return</b> - TRUE is success and FALSE is error.</p> <p><b>s</b> - Socket descriptor you want to send to.</p> <p><b>dwAddr</b> - Remote IP.</p> <p><b>wPort</b> - Remote port.</p> <p><b>pby</b> - Buffer to send.</p> <p><b>nLen</b> - The size of the pby buffer.</p>
<b>Example:</b>	SocketSendTo(inet_addr("192.168.0.19"),1234,&c,1); ★

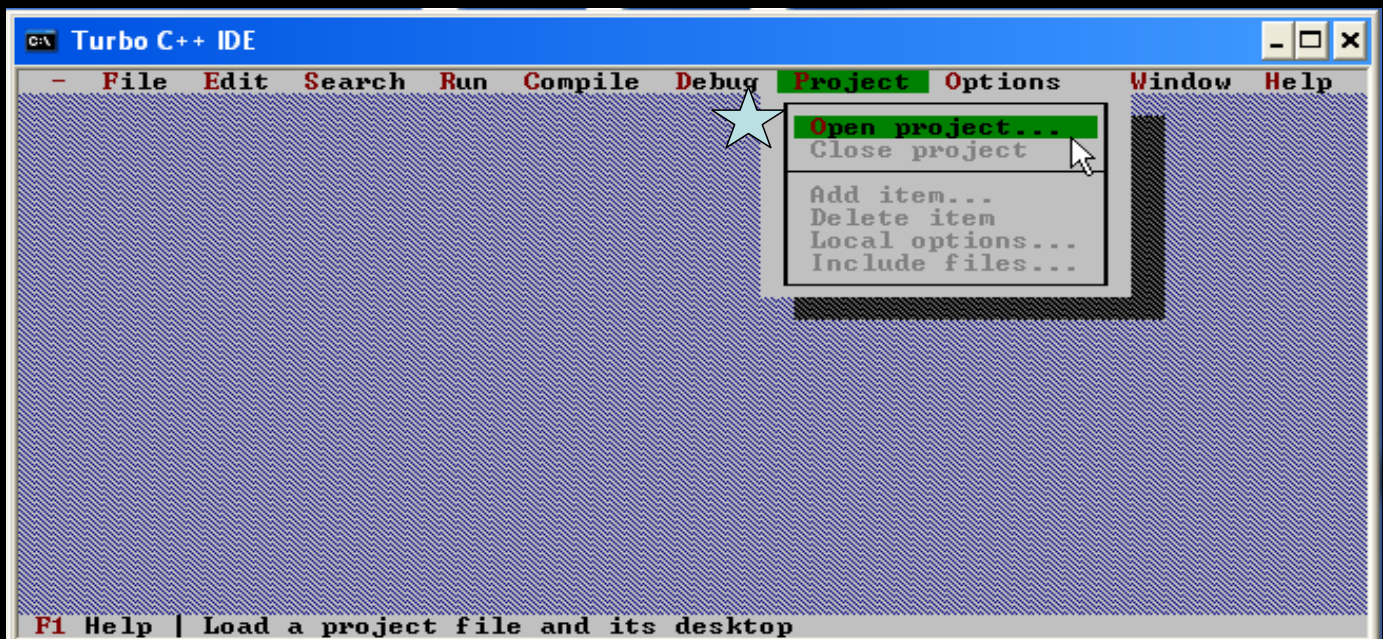


int SocketRecvFrom(SOCKET s,DWORD \*pdwAddr,WORD \*pwPort,BYTE \*pby,int nLen)

<b>Description:</b>	Receive data from UDP socket.
<b>Arguments:</b>	<p><b>Return</b> - Bytes received, return -1 is error.</p> <p><b>s</b> - Socket descriptor you want to receive from.</p> <p><b>pdwAddre</b> - Pointer of DWORD buffer to save remote IP address.</p> <p><b>pwPort</b> - Pointer of WORD buffer to save remote port.</p> <p><b>pby</b> - Buffer for received data.</p> <p><b>nLen</b> - The size of the pby buffer.</p>
<b>Example:</b>	<pre>DWORD dwAddr; WORD wPort; char c; SocketRecvFrom(s, &amp;dwAddr, &amp;wPort,&amp;c,1); printf("From %s:%d, send '%c' back.\n", inet_ntoa(szBuf, dwAddr), wPort, c);</pre>

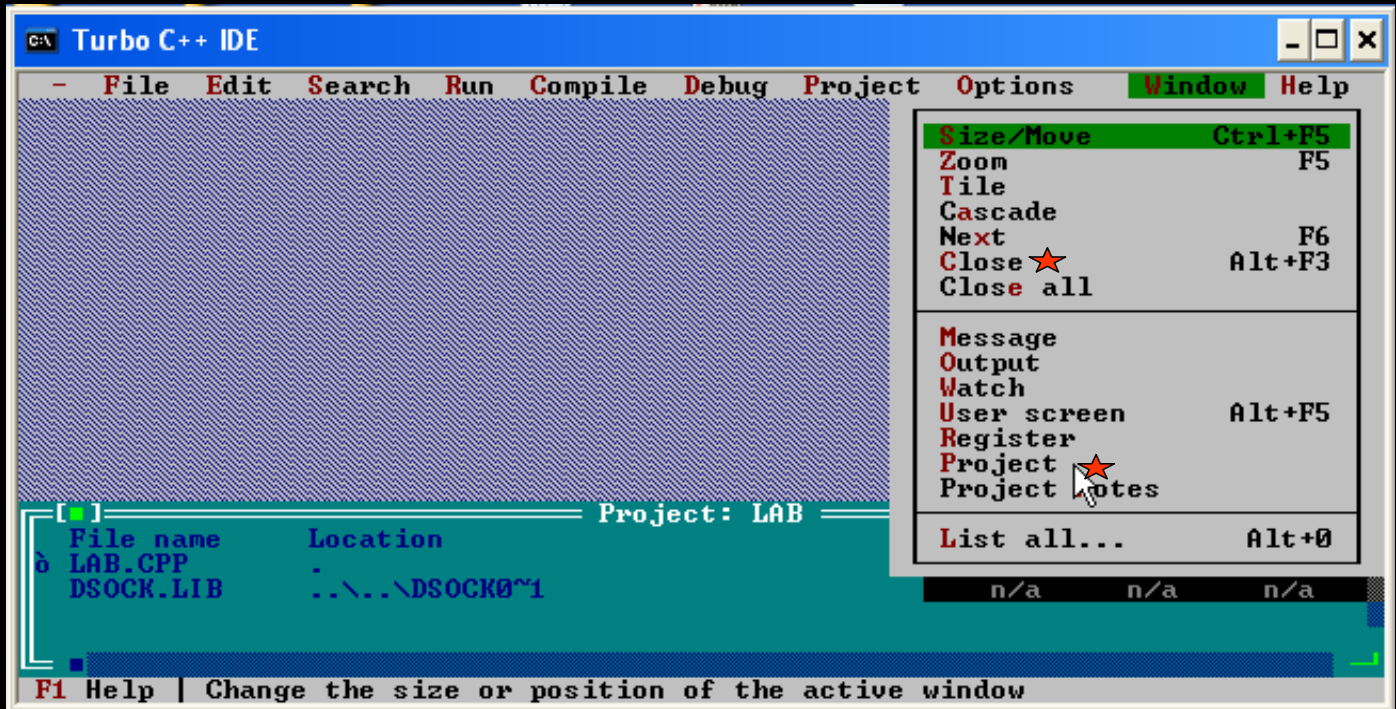


## Project File ( XXX.CPP + DSOCK.LIB )

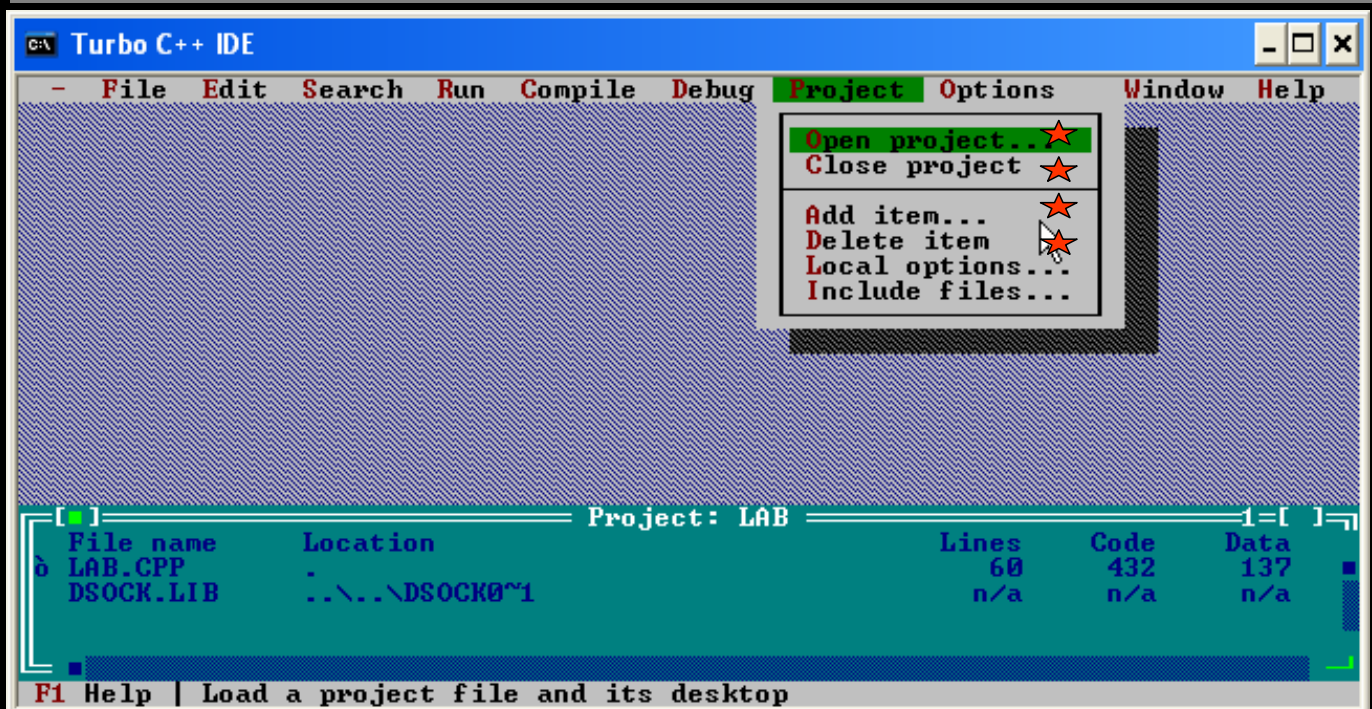




Project File ( XXX.CPP + DSOCK.LIB )



Project File ( XXX.CPP + DSOCK.LIB )





## ตัวอย่างการใช้งาน Dsock

```
#include "..\dsock.h"
```

```
#include <stdio.h>  
#include <conio.h>  
#include <stdlib.h>  
#include <values.h>  
#include <dos.h>  
#include <iostream.h>  
#include <bios.h>  
#include <conio.h>  
#include <string.h>  
#include <time.h>  
#include <dos.h>
```

ส่วนประกาศหรือหัวโปรแกรม

File : XXX.CPP

File : tc30/demo/test/Lab.CPP



## ตัวอย่างการใช้งาน Dsock

## เปิด Dsock

```
if(DSock_Open() == FALSE)  
{  
    printf("Unable to initialize socket library.\n");  
    exit(1);  
}  
Else  
{  
    printf("Open DSock library.\n");  
}
```

Open Sock Lib

File : tc30/demo/test/Lab.CPP



ตัวอย่างการใช้งาน Dsock โหลด Max และปิด Dsock

```

BYTE *max_add = DSocket_GetMacAddr();
max0=max_add[0];
max1=max_add[1];
max2=max_add[2];
max3=max_add[3];
max4=max_add[4];
max5=max_add[5];
cout<<"max0 :="<<max0<<endl;
cout<<"max1 :="<<max1<<endl;
cout<<"max2 :="<<max2<<endl;
cout<<"max3 :="<<max3<<endl;
cout<<"max4 :="<<max4<<endl;
cout<<"max5 :="<<max5<<endl;
DSock_Close();

```

Get MAC address from network card

File : tc30/demo/test/Lab.CPP

Close the socket library



การติดตั้ง Config ให้หัวอ่าน File : tc30/demo/test/config1.cpp

```

1 #include "....." // #include "..\dsock.h"
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 dsock_config();
6 char ip[30]={ " " },netmask[30]={ " " },gateway[30]={ " " }; // void dsock_config();
7 char server[30]={ " " },cport[30]={ " " }; // void dsock_config();
8 long netmask_int,cport_int; // void dsock_config();
9 SOCKET s; // void dsock_config();
10 DWORD dwAddr; // void dsock_config();
11 WORD wPort; // void dsock_config();
12 char szBuf[32]; // void dsock_config();
13 FILE * fptr;
14
15 void main()
16 { clrscr();
17 dsock_config();
18 getch();
19 }

```



```

21 void dsock_config()
22 { fptr=fopen(".....","r");
23
24 while(fgets(ip,30,fptr) != NULL)
25     { cout<<" ip = "<<ip<<endl; }
26 fclose(fptr);
27 fptr=fopen(".....","r");
28
29 while(fgets(netmask,30,fptr) != NULL)
30     { cout<<" netmask = "<<netmask<<endl; }
31 fclose(fptr);
32 sscanf(netmask,"%d",&netmask_int);
33 cout<<"--netmask----"<<netmask_int<<endl;
34
35 fptr=fopen(".....","r");
36 while(fgets(gateway,30,fptr) != NULL)
37     { cout<<" gateway = "<<gateway<<endl; }
38 fclose(fptr);
39
40 fptr=fopen(".....","r");
41 while(fgets(server,30,fptr) != NULL)
42     { cout<<" server = "<<server<<endl; }
43 fclose(fptr);
44
45 fptr=fopen(".....","r");
46 while(fgets(cport,30,fptr) != NULL)
47     { cout<<" port = "<<cport<<endl; }
48 fclose(fptr);
49 sscanf(cport,"%d",&cport_int);

```

การติดตั้ง Config ให้หัวอ่าน

File : tc30/demo/test/config1.cpp

```

52 /* Open DSocket Library */
53 if(.....() == FALSE)
54 {
55     printf("Unable to initialize socket library.\n");
56     exit(1); // return 1;
57 }else{
58     printf("Open DSocket library.\n");
59 }
60
61 DSocket_LoadConfigFile(".....");
62 printf("Load configuration dsocket.cfg.\n");
63 s = SocketCreate(.....);
64 if(...== INVALID_SOCKET)
65 {
66     printf("SocketCreate() error\n");
67     exit(1); // return 1;
68 }else{
69     printf("Socket create. ");
70 }
71 if(.....(s,inet_addr(netmask),cport_int) == FALSE)
72 {
73     printf("SocketBind() error\n");
74     exit(1); //return 1;
75 }else
76 {
77     printf("Socket bind on port \n");
78 }
79 }

```

DSocket\_Open()

DSocket\_LoadConfigFile("dsocket.cfg");

UDP\_SOCKET  
(s == INVALID\_SOCKET)

SocketBind

การติดตั้ง Config ให้หัวอ่าน

File : tc30/demo/test/config1.cpp



ตัวอย่างการใช้งาน Dsock

การส่ง/รับ ข้อมูลแบบ UDP(ทีละตัวอักษร)

```
FILE * fptr;
clrscr();
```

```
fptr=fopen("ip.cfg", "r");
while(fgets(ip,30,fptr) != NULL)
{
    cout<<" ip = "<<ip<<endl;
}
fclose(fptr);
```

1

```
fptr=fopen("gateway.cfg", "r");
while(fgets(gateway,30,fptr) != NULL)
{
    cout<<" gateway = "<<gateway<<endl;
}
fclose(fptr);
```

3

```
fptr=fopen("server.cfg", "r");
while(fgets(server,30,fptr) != NULL)
{
    cout<<" server = "<<server<<endl;
}
fclose(fptr);
```

4

```
fptr=fopen("netmask.cfg", "r");
while(fgets(netmask,30,fptr) != NULL)
{
    cout<<" netmask = "<<netmask<<endl;
}
fclose(fptr);
sscanf(netmask, "%d", &netmask_int);
```

2

```
fptr=fopen("port.cfg", "r");
while(fgets(cport,30,fptr) != NULL)
{
    cout<<" port = "<<cport<<endl;
}
fclose(fptr);
sscanf(cport, "%d", &cport_int);
```

5

File : tc30/demo/test/UDP1.CPP



ตัวอย่างการใช้งาน Dsock

การส่ง/รับ ข้อมูลแบบ UDP(ทีละตัวอักษร)

```
if(DSock_Open() == FALSE)
{
    printf("Unable to initialize socket library.\n");
    exit(1);// return 1;
}
Else
{
    printf("Open DSock library.\n");
}
```

1

```
DSock_LoadConfigFile("dsock.cfg");
printf("Load configuration dsock.cfg.\n");
```

2

```
s = SocketCreate(UDP_SOCKET);
if(s == INVALID_SOCKET)
{
    printf("SocketCreate() error\n");
    exit(1);// return 1;
}
Else
{
    printf("Socket create. ");
}
```

3

```
if(SocketBind(s,inet_addr(netmask),cport_int) == FALSE)
{
    printf("SocketBind() error\n");
    exit(1);//return 1;
}
else
{
    printf("Socket bind on port \n");
}
```

4

File : tc30/demo/test/UDP1.CPP

## ตัวอย่างการใช้งาน Dsock การส่ง/รับ ข้อมูลแบบ UDP(ที่ละตัวอักษร)

```

do
{
    if(kbhit())
    {
        data=getch();
        if (data!=0) {
            SocketSendTo(s,inset_addr(server),cport_int, &data, 1);
            cout<<"Send Data := "<<data<<"To server "<<endl;
        }
    }
    if(SocketDataReady(s))
    {
        SocketRecvFrom(s, &dwAddr, &wPort,&data1,1);
        inet_ntoa(szBuf,dwAddr);
        cout<<"From IP "<<szBuf<<" Port : "<<wPort<<" : ";
        cout<<"RX data := "<<data1<<endl;
    }
}while(data!='0');
SocketClose(s); SocketDestory(s); DSocket_Close();

```

File : tc30/demo/test/UDP1.CPP

Use UDP socket to send data.

Check incoming data of a socket

Receive data from UDP socket

Convert a network address into a string in dot notation

Close a socket  
Release a socket.  
Close DSocket socket library



## ตัวอย่างการใช้งาน Dsock การส่ง/รับ ข้อมูลแบบ UDP(String)

```

do
{
    if(kbhit())
    {
        data=getch();
        if (i==0) cout<<"Data := ";
        if (data!=0) {
            data2[i]=data;
            cout<<data;
            i++;
        }
        if (data==0x0d)
        {
            cout<<"\n Send Data to Server === "<<data2<<endl;
            SocketSendTo(s,inset_addr(server),cport_int, data2, i);
            i=0;
        }
    }
}while(data!='0');
SocketClose(s); SocketDestory(s); DSocket_Close();

```

File : tc/30/demo/test/ UDP2.CPP

Use UDP socket to send data.

Close a socket  
Release a socket.  
Close DSocket socket library



ตัวอย่างการใช้งาน Dsock

การส่ง/รับ ข้อมูลแบบ UDP(String)

File : tc/30/demo/test/ UDP2.CPP

```
do
{
    if(kbhit())
    {

    }

    if(SocketDataReady(s))
    {
        SocketRecvFrom(s, &dwAddr, &wPort, data3 ,5 );
        inet_ntoa(szBuf,dwAddr);
        cout<<"From IP "<<szBuf<<" Port : "<<wPort<<" : ";
        cout<<"RX data :="<<data3<<endl;
    }

}while(data!='0');
```

Check incoming data of a socket

Receive data from UDP socket

Convert a network address into a string in dot notation

SocketClose(s);      SocketDestory(s);      DSock\_Close();  
 Close a socket      Release a socket.      Close DSocket library



ตัวอย่างการใช้งาน Dsock

การส่งข้อมูลใน File แบบ UDP(char)

File : tc/30/demo/test/ timein2.CPP

```
1 #include "..\dsock.h"
2 #include <stdio.h>            #include <conio.h>            #include <stdlib.h>
3 #include <values.h>           #include <dos.h>            #include <iostream.h>
4 #include <bios.h>            #include <conio.h>           #include <string.h>
5 #include <time.h>            #include <dos.h>
6 void dsock_config();
7 char c;
8 .....//Copy From File : CONFIG1.CPP
9 .....
10 .....
11 .....
12 .....
13 .....
14 .....//Copy From File : CONFIG1.CPP
```



ตัวอย่างการใช้งาน Dsock

การส่งข้อมูลใน File แบบ UDP(char)

```

16 FILE * fptr;
17 main()
18 { clrscr();
19   dsock_config();
20   fptr=fopen(".....", "....");
21   while((c=fgetc(fptr)) != EOF)
22     {
23       .....(s, inet_addr(server), cport_int, &c, 1);
24     }
25   fclose(fptr);
26   cout<<"==End Data==";
27
28   SocketClose(s);
29   SocketDestory(s);
30   DSock_Close();
31   cout<<"Exit Program";
32   getch();
33   return 0;
34 }
35
36 void dsock_config()
37 {
38   //Copy From File : CONFIG1.CPP
39 }

```

File : tc/30/demo/test/ timein2.CPP



ตัวอย่างการใช้งาน Dsock

การส่งข้อมูลใน File แบบ UDP(string)

```

1 #include "..\dsock.h"
2 #include <stdio.h>           #include <conio.h>           #include <stdlib.h>
3 #include <values.h>         #include <dos.h>           #include <iostream.h>
4 #include <bios.h>          #include <conio.h>         #include <string.h>
5 #include <time.h>          #include <dos.h>
6 void dsock_config();
7 char c;
8 .....//Copy From File : CONFIG1.CPP
9 .....
10 .....
11 .....
12 .....
13 .....
14 .....//Copy From File : CONFIG1.CPP
15 FILE * fptr;

```

File : tc/30/demo/test/ timein21.CPP

```

15 FILE * fptr;
16
17 main()
18 { char temp[500]={" "};
19   int line=0;
20   clrscr();
21   dsock_config();
22   fptr=fopen(".....","..");
23   while((c=fgetc(fptr)) != EOF)
24     { temp[line]=c;
25       line++;
26       if (line==500)
27         { .....(s,inet_addr(server),cport_int,.....);
28           strcpy(temp," ");
29           line=0;
30         }
31     }
32   if (line!=0)SocketSendTo(s,inet_addr(server),cport_int,temp,line);
33   fclose(fptr);
34   cout<<"==End Data==";
35   SocketClose(s);
36   SocketDestory(s);
37   DSock_Close();
38   cout<<"Exit Program";
39   getch();
40   return 0;
41 }
42
43 void dsock_config()
44 {
45   //Copy From File : CONFIG1.CPP
46 }

```

File : tc/30/demo/test/ timein21.CPP

```

1 #include "..\dsock.h"
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 reset();             void dsock_config();
7 #define COM1 0           #define DATA_READY 0x100
8 #define SETTINGS ( 0xE0| 0x03 | 0x00 | 0x00)
9 int data;
10 int tag_sn_state,read_id_name_state;
11 int sn_tag[8];
12 char sn_tag_char[40]={" "};
13 char name_tag_char[64]={" "};
14 char id_tag_char[15]={" "};
15 char c;
16 char ip[30]={" "},netmask[30]={" "},gateway[30]={" "};
17 char server[30]={" "},cport[30]={" "};
18 long netmask_int,cport_int;
19 SOCKET s;
20 DWORD dwAddr;           //typedef unsigned long  DWORD;
21 WORD wPort;             //typedef unsigned short WORD;
22 char szBuf[32];
23 FILE * fptr;

```

File : tc/30/demo/test/ timein3.CPP

```

25 void main()
26 { char key;
27 bioscom(0, SETTINGS, COM1);
28 config_gpio();          dsock_config();
29 clrscr();               reset();          clrscr();
30 cout<<"\n\t\t\tExit Program Key ESC"<<endl;
31 do
32 { do{
33     check_tag();
34     }while(!kbhit()&&data!=254&&!SocketDataReady(s));
35 if(kbhit()) key=getch();
36 if (data==254)
37     { get_sn_tag();
38       read_name();
39       if ((tag_sn_state==1)&&(read_id_name_state==1))
40         { cout<<"\n\t\t Tag SN ="<<sn_tag_char<<"==\n";
41           cout<<"\t\t Tag ID ="<<id_tag_char<<"==\n";
42           cout<<"\t\t Tag Name ="<<name_tag_char<<"==\n";
43           char send_data[100]={""};      int j;
44           strcpy(send_data,id_tag_char);
45           j=strlen(send_data);
46           .....(s,inet_addr(server),cport_int,.....);
47         }
48       else
49         { cout<<"\n\t\t Tag ID = Read Data Error==\n";
50           sound(500);delay(200);nosound();
51         }
52     }
53 do{
54     check_tag();
55     }while(!kbhit()&&data!=255&&!SocketDataReady(s));
56 if(kbhit()) key=getch();

```

```

61 if(.....(s))
62     { char data3;
63       do
64         { .....(s, &dwAddr, &wPort,&data3,1);
65           inet_ntoa(szBuf,dwAddr);
66           cout<<"From IP "<<szBuf<<" Port : "<<wPort<<" : ";
67           cout<<"RX data :="<<data3<<endl;
68           }while(!kbhit()&&SocketDataReady(s));
69
70         if (data3=='.....')
71           { sound(1000);delay(200);nosound();
72             sound(500);delay(200);nosound();
73           }
74         else sound(500);delay(200);nosound();
75         do{ check_tag();
76           }while(!kbhit()&&data!=255);
77     }
78
79 }while(key!=27);
80 }

```

```

84 void read_name()
85 {
86     //Copy From File : TC\BIN\MIFARE\ISP15693\READ_NA.CPP
87 }
88 void get_sn_tag()
89 {
90     //Copy From File : TC\BIN\MIFARE\ISP15693\READ_SN.CPP
91 }
92 void reset()
93 {
94     //Copy From File : TC\BIN\MIFARE\ISP15693\READ_SN.CPP
95 }
96 void config_gpio()
97 {
98     //Copy From File : TC\BIN\MIFARE\ISP15693\CH_TAG1.CPP
99 }
100 void check_tag()
101 {
102     //Copy From File : TC\BIN\MIFARE\ISP15693\CH_TAG1.CPP
103 }
104 void dsock_config()
105 {
106     //Copy From File : TC30\DEMO\TEST\CONFIG1.CPP
107 }

```



## RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

## ตัวอย่างการใช้งาน Dsock

## การอ่าน Config จากไฟล์ (TCP/IP)

```
FILE * fptr;
clrscr();
```

```
fptr=fopen("ip.cfg","r");
while(fgets(ip,30,fptr) != NULL)
{
    cout<<" ip = "<<ip<<endl;
}
fclose(fptr);
```

1

2

```
fptr=fopen("netmask.cfg","r");
while(fgets(netmask,30,fptr) != NULL)
{
    cout<<" netmask = "<<netmask<<endl;
}
fclose(fptr);
sscanf(netmask,"%d",&netmask_int);
```

```
fptr=fopen("gateway.cfg","r");
while(fgets(gateway,30,fptr) != NULL)
{
    cout<<" gateway = "<<gateway<<endl;
}
fclose(fptr);
```

3

```
fptr=fopen("server.cfg","r");
while(fgets(server,30,fptr) != NULL)
{
    cout<<" server = "<<server<<endl;
}
fclose(fptr);
```

4

```
fptr=fopen("port.cfg","r");
while(fgets(cport,30,fptr) != NULL)
{
    cout<<" port = "<<cport<<endl;
}
fclose(fptr);
sscanf(cport,"%d",&cport_int);
```

5





ตัวอย่างการใช้งาน Dsock      การกำหนดรูปแบบการสื่อสารแบบ TCP/IP

```
1
if(DSock_Open() == FALSE)
{
    printf("Unable to initialize socket library.\n");
    exit(1); // return 1;
}
Else
{
    printf("Open DSock library.\n");
}
```

```
2
DSock_LoadConfigFile("dsock.cfg");
printf("Load configuration dsock.cfg.\n");
```

```
3
s = SocketCreate(TCP_SOCKET);

if (SocketConnect(s,inet_addr(server),cport_int) ==
FALSE)
{
    cout<<" Socket Conext Error "<<endl;
    getch();
    exit(1);
}
else
{
    cout<<" Socket Conext OK "<<endl;
}
```

**Dsock.cfg**

```
ip=192.168.0.13
netmask=255.255.255.0
gateway=192.168.0.1
nameserver=192.168.0.1
```

File : tc30/demo/test/TCP1.CPP



ตัวอย่างการใช้งาน Dsock      การส่ง/รับ ข้อมูลแบบ TCP/IP(ทีละตัวอักษร)

File : tc30/demo/test/TCP1.CPP

```
do
{
    if(kbhit())
    {
        data=getch();
        SocketPutChar(s,data);
    }

    if(SocketDataReady(s))
    {
        SocketGetChar(s,&data1);
        cout<<"RX data := "<<data1<<endl;
    }

}while(data!='0');

SocketClose(s);    SocketDestory(s);    DSock_Close();
```

Write a character to a socket 😊

Check incoming data of a socket

Read a character from a socket 😊

Close a socket  
 Release a socket.  
 Close DSock socket library



ตัวอย่างการใช้งาน Dsock การส่ง/รับ ข้อมูลแบบ TCP/IP(String)

File : tc30/demo/test/TCP1-1.CPP

```
do
{
  if(kbhit())
  {
    data=getch();
    if (i==0) cout<<"Data := ";
    if ((data!='0')&&(data!=0x0d))
    {
      data2[i]=data;
      cout<<data;
      i++;
    }
    if (data==0x0d)
    {cout<<"\nsend data to server === " <<data2<<endl;
      SocketPutString(s,data2);
      strcpy(data2," ");
      i=0;
    }
  }
}while( (data!='0') );
SocketClose(s); SocketDestory(s); DSock_Close();
```

Write a string to a socket

Close a socket  
 Release a socket.  
 Close DSocket library



ตัวอย่างการใช้งาน Dsock การส่ง/รับ ข้อมูลแบบ TCP/IP(String)

File : tc30/demo/test/TCP1-1.CPP

```
do
{
  if(SocketDataReady(s))
  {
    SocketGetString(s,data1,Buf);
    cout<<"RX data :=" <<data1<<endl;
  }
}while( (data!='0') );
SocketClose(s); SocketDestory(s); DSock_Close();
```

Check incoming data of a socket

Read a character from a socket

Note :Data=String  
 Buf = int

Close a socket  
 Release a socket.  
 Close DSocket library



ตัวอย่าง การส่ง File ระบบศูนย์อาหารข้อมูลแบบ TCP/IP(String)

```

1  #include "..\dsock.h"
2  #include <stdio.h>           #include <conio.h>           #include <stdlib.h>
3  #include <values.h>         #include <dos.h>           #include <iostream.h>
4  #include <bios.h>          #include <conio.h>         #include <string.h>
5  #include <time.h>          #include <dos.h>
6  void dsock_config();
7  char c;
8  char ip[30]={"  "},netmask[30]={"  "},gateway[30]={"  "};
9  char server[30]={"  "},cport[30]={"  "};
10 long netmask_int,cport_int;
11 SOCKET s;
12 DWORD dwAddr;           //typedef unsigned long  DWORD;
13 WORD wPort;             //typedef unsigned short WORD;
14 char szBuf[32];
15 FILE *fptr;
    
```

File : tc30/demo/test/food1.CPP

ตัวอย่าง การส่ง File ระบบศูนย์อาหารข้อมูลแบบ TCP/IP(String)

```

17 main()
18 {  int line=0;
19     char temp[500]={"  "};
20     clrscr();
21     dsock_config();
22     fptr=fopen(".....",".....");
23     while((c=fgetc(fptr)) != EOF)
24     {  temp[line]=c;
25         line++;
26         if(c=='\n') {  temp[line]=0x0d;
27                         line++;
28                     }
29         if (line==500)
30             {  .....(s,.....);
31                 strcpy(temp,"  ");
32                 line=0;
33             }
34     }
35     if (line!=0)SocketPutString(s,temp);
36     fclose(fptr);
37     cout<<"==End Data==" ;
38     SocketClose(s);
39     SocketDestory(s);
40     DSocket_Close();
41     cout<<"Exit Program";
42     getch();
43     return 0;
44 }
46 void dsock_config()
47 {
48     //Copy From File : CONFIG1.CPP
49 }
    
```

File : tc30/demo/test/food1.CPP