

انتقال داده‌ها به پورت توسط زبان برنامه نویسی

در این قسمت کلا از دو زبان ++C تحت TC++ و پاسکال (امکان کامپایل راحت برنامه های نوشته شده تحت دلفی با استفاده از قسمت Console Application به راحتی وجود دارد) بهره گرفته شده است.

از سی ++ برای برنامه های تحت داس و از دلفی برای برنامه نویسی سخت افزار تحت ویندوز بهره خواهیم گرفت.

در هر زبان برنامه نویسی برای استفاده از رابطها به چهارنوع دستور نیاز خواهیم داشت.

- 1- دستوری برای باز و بسته کردن رابط
 - 2- دستوری برای تنظیم رابط
 - 3- دستوری برای ارسال داده به رابط
 - 4- دستوری برای دریافت یک داده از روی رابط و قرار دادن آن در یک متغیر متناسب با داده وارد شده به پورت .
- بهرتر است یک مثال را در نظر بگیریم و این روند 4 گانه را طی بررسی مراحل برنامه نویسی برنامه مفروض تحت ++C بررسی نمائیم.

توجه کنید که در این قسمت فرض بر این است خواننده آگاهی نسبی از روش برنامه نویسی تحت یک محیط ادیتور و تبدیل برنامه به کد شی را دارا می باشد (کامپایلر کد)

مثال:

می خواهیم برنامه ای بنویسیم به محض اینکه پین 13 یا همان بیت 4 رجیسترو وضعیت یک شد عدد 0xFFH روی پورت داده نوشته شود و در غیر اینصورت عدد 0x00H به پورت داده ارسال گردد. (پورت LPT1 را در نظر بگیرید)

1- پین 13 متعلق به کدام یک از رجیسترهای سه گانه است ؟ به جدول 1-6 مراجعه کنید؟

جواب: بیت S4 که در آدرس 379H یعنی رجیسترو وضعیت (Base+1) قرار دارد

2- به نظر شما بوسیله زبان برنامه نویسی چگونه قادر خواهیم بود این بیت را چک بکنیم؟

برای این کار در صورتیکه از زبان اسمبلی استفاده نمی کنید، از روش ماسک کردن استفاده می کنیم.

ماسک کردن:

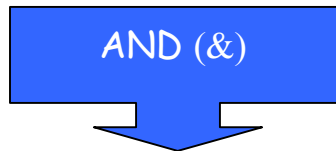
برای مثال فرض بکنید یک عدد مفروض 8 بیتی بنام A داریم حالا می خواهیم ببینیم بیت 5 ام آن 1 است یا صفر؟

برای این کار بهتر است کل آن عدد را با عدد $B = (10)H = (0001\ 0000)_{10}$ عمل یا منطقی بیت به بیت بکنیم

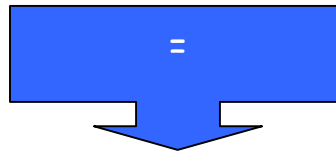
در این صورت اگر بیت 5 صفر باشد با AND شدن کل عدد باید صفر شود پس در این حالت می فهمیم که آن بیت صفر است

ولی اگر حاصل مساوی عددی بود که ما برای تست آن را ایجاد کرده بودیم خواهیم فهمید که بیت مورد نظر 1 است.

B7	B6	B5	B4	B3	B2	B1	B0
0	1	1	1	0	0	1	0



B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	1	0	0	0	0



B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	1	0	0	0	0

شکل 1-12: طریقه Mask بیتی

به این روش تکنیک اصلاح بیتی نیز گوئیم . البته روش های دیگری برای تکنیک بیتی (XOR) بکار گرفته می شود. براساس حاصل این بیت می توان یک دستور شرطی نوشت تا با توجه به 0 یا 1 بودن این بیت با توجه به حاصل ماسک کردن عدد های مفروض را به پورت ارسال نمائید.

برنامه نویسی مثال تحت C++:

- 1- ابتدا سرفایلی (Header) را که اجازه ارتباط با پورت موازی را خواهد داد، معرفی می کنیم
در اینجا : dos.h و bios.h
- 2- یک متغیر متناسب با نیازمان برای دریافت (a) و یکی برای ارسال (b) معرفی می کنیم

با تعریف سرفایل dos.h می توان گفت :

برای دریافت داده ها در زبان سی ++ از دستورات زیر می توان استفاده کرد:

```
a = inportb (0x379);
```

برای دریافت اعداد 1 بیتی

```
a = inport (0x379);
```

برای دریافت اعداد 2 بیتی

برای ارسال داده های در زبان سی ++ از دستورات زیر می توان استفاده کرد:

```
outportb (0x378 , Number);
```

برای ارسال اعداد 1 بیتی

```
outport (0x378 , Number);
```

برای ارسال اعداد 2 بیتی

با تعریف سر فایل Conio.h می توان گفت:

برای دریافت داده ها از پورت موازی در زبان سی ++ از دستورات زیر می توان استفاده کرد:

a = inp (0x379); برای دریافت اعداد 1 بایتی

a = inpw (0x379); برای دریافت اعداد 2 بایتی

برای ارسال داده های به پورت موازی در زبان سی ++ از دستورات زیر می توان استفاده کرد:

outp (0x378 , Number); برای ارسال اعداد 1 بایتی

outpw (0x378, Number); برای ارسال اعداد 2 بایتی

البته در این سر فایل توابع همه کاره ای بنام biosprint برای کار با پورت موازی وجود دارد.

توجه :

ممکن است صورت دستورات در بعضی از کامپایلرها متفاوت باشد پس حتما قبل از برنامه نویسی به مستندات کامپایلر تان مراجعه فرمائید.

برنامه ارسال و دریافت به پورت موازی:

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

main( )
{
    int a,Number;
    do
    {
        a =inportb(0x379);           //input from port
        number=a&10;                 // Mask
        outportb(0x378,0x00);        //Send to port(D0-D7) =00000000
    } while(b!=0);
    outportb(0x378,0xFF);           //Send to port(D0-D7) = 11111111
}
```

برای آشنائی بهتر و سریعتر با زبان سی ++ توصیه می شود به کتاب سی ++ نوشته هربرت شیلد ترجمه شادمان پور مراجعه فرمائید.

در برنامه زیر از تابع biosprint استفاده شده است.

```
a = biosprint (2,0,0);           //input from 379 port (LPT1)
a = biosprint (2,1,0);           //input from 379 port (LPT2)
```

برنامه ارسال و دریافت از پورت موازی

```
#include bios.h>
#include <conio.h>
#include <stdio.h>

main( )
{
    #define STATUS 2           /* printer status command*/
    #define PORTNUM 0          /* port number for LPT1*/
```

```

int status , abyte=0;

printf("Please turn off your printer. Press any key to continue\n");
getch( );
status = biosprint (STATUS , abyte , PORTNUM );

if (status &0x01);
    printf("Device time out.\n");
if (status &0x08);
    printf("I/O error.\n");
if (status &0x10);
    printf("Selected.\n");
if (status &0x20);
    printf("Out of paper.\n");
if (status &0x40);
    printf("Acknowledge.\n");
if (status &0x80);
    printf("Not busy.\n");
return 0;
}

```

بوسیله این تابع می توان برنامه هائی برای خطایابی دستگاه جانبی نوشت.

برنامه نویسی برای مثال 1 تحت اسمبلی:

دراین زبان برای این پورت بسیار ساده ترمی توان برنامه نوشت . همانطوریکه می دانید سی ++ و پاسکال دلفی هر دو از اسمبلی حمایت می کنند پس می توان براحتی از آن در برناه بهره گرفت.

دستوراتیکه می توان توسط آن دیتا را از LPT خواند یا نوشت بترتیب دستورات in و out می باشد

توجه کنید که این دو دستور نمی تواند با آدرس های پورت که 2 بایتی هستند کار بکند

برای همین آدرس ها را در یک رجیستر 16 بیتی ذخیره می کنیم و بعدتوسط آن باپورت ارتباط برقرار می کنیم.

```

; Lpt Port Program
Start:
    MOV DX , 378H
    MOV AL , 0
    OUT DX , AL
    MOV DX , 379H
    IN AL , DX
    AND AL , 10H
JNZ Start
    MOV DX , 378H
    MOV AL,FFH
    OUT DX , AL
END

```

نوشتن همان برنامه تحت پاسکال :

توجه فرمائید که این برنامه را می توان در قسمت Console دلفی بدون هیچ مشکلی نوشت کامپایل نمود

A = Port [\$379]

برای دریافت اعداد یک بایتی

A = Portw [\$379]

برای دریافت اعداد دوبایتی

برای ارسال دریافت کافی است در محل [\$Address] به جای Address آدرس 378 را بنویسید.

```
Program Prt;
Uses crt;
Var
    A,B:byte;
Begin
    Clrscr;
    Repeat
        A:=Port[$379];           {input from Port}
        B:=A and $10;             {Mask}
        Port[$378]:= $00;         {Send to port}
    Until B=0;
    Port[$378]:= $FF;             {Send to port}
End.
```

مثال دوم: در این برنامه می خواهیم در صورتیکه بیت ACK یعنی همان بیت 6 از رجیستر وضعیت که به پین 10 پورت LPT متصل است هر وقت 1 شد برنامه اطلاعاتی را به دیتا پورت دریافت نمائید در غیر اینصورت عدد AFH را به دیتا پورت ارسال نمائید.

برنامه تحت سی ++ :

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

Main()
{
    int a,b;
    do
    {
        a = inportb (0x379);
        b = a & 0x40;
        if (b == 0)
        {
            Outport (0x37a,0x20);
            a = inport (0x378);
        }
        else
        {
            outport(0x27a,0x00);
            outport (0x378,0xAF);
        }
    } while(!kbhit( ));
    Outport (0x37a,0x00);
}
```

Start:

```
Mov DX,379H  
IN AL,DX  
AND AL,40H  
JZ PART1  
MOV DX,37AH
```

```
MOV AL,20H  
OUT DX,AL  
MOV DX,378H  
IN AL,DX
```

PART1:

```
MOV DX,37AH  
MOV AL,00H  
OUT DX,AL  
MOV DX,378H  
MOV AL,AFH  
OUT DX,AL  
JMP Start
```

END

پایان قسمت دوم

مرجع : اصول کامل راه اندازی و کنترل دستگاههای جانبی

مؤلف : مهندس محسن شکیبافر