



ارتباط با دستگاه‌های جانبی

یکی از ویژگی‌های زبان C این است که به راحتی می‌تواند با دستگاه‌های جانبی ارتباط برقرار کند، به طوری که اطلاعاتی را از پورتی بخواند و اطلاعاتی را در پورتی بنویسد. در این پیوست، توابعی را مطالعه خواهیم کرد که این امکانات را فراهم می‌کنند. یکی از ضروریات کارکردن با این توابع این است که آدرس پورتهای دستگاه خود را بدانید. مثال‌های این فصل بر روی دیسک، در شاخه P قرار دارند.

توابع خواندن از پورت

تعدادی از توابع وجود دارند که بایت یا کلماتی را از پورت سخت‌افزاری می‌خوانند. در این بخش به بررسی این توابع می‌پردازیم.

تابع inp()

این تابع یک بایت را از پورت سخت‌افزاری می‌خواند. این تابع به صورت یک ماکرو پیاده‌سازی شده است و در فایل conio.h قرار دارد:

```
int inp(unsigned portid);
```

پارامتر این تابع یک مقدار صحیح بدون علامت است که آدرس پورت را مشخص می‌کند. مثال ۱ را ببینید.

تابع inportb()

این تابع بایتی را از پورت سخت‌افزاری می‌خواند. این تابع به صورت ماکرو پیاده‌سازی شده است و در فایل conio.h قرار دارد.

```
unsigned char inportb(unsigned portid);
```

پارامتر این تابع مقدار صحیح بدون علامت است که آدرس پورت را مشخص می‌کند. مثال ۱ را ببینید.

تابع inpw()

این تابع یک کلمه را از پورت سخت‌افزاری می‌خواند. این تابع در فایل conio.h قرار دارد و به صورت زیر به کار می‌رود:

```
unsigned inpw(unsigned portid);
```

پارامتر این تابع شماره پورتی است که اطلاعات باید از آنجا خوانده شود. مثال ۱ را ببینید.

تابع (inport)

این تابع یک کلمه را از پورت سخت‌افزاری می‌خواند. این تابع در فایل conio.h قرار دارد و به صورت زیر به کار می‌رود:

unsigned inport(unsigned portid);

پارامتر این تابع شماره پورتهی است که اطلاعات باید از آنجا خوانده شود. مثال ۱ را ببینید.

مثال ۱

برنامه‌ای که یک بایت و سپس یک کلمه را از پورت سریال شماره ۰ می‌خواند (خروجی برنامه را ببینید).

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int port = 0;
    int result;
    clrscr();
    result = inp(port);
    printf("The value 0x%X read from port %d\n", result, port);
    result = inportb(port);
    printf("The value 0x%X read from port %d\n", result, port);
    result = inpw(port);
    printf("The value 0x%X read from port %d\n", result, port);
    result = inport(port);
    printf("The value 0x%X read from port %d\n", result, port);
    getch();
    return 0;
}
```

خروجی

```
The value 0x8 read from port 0
The value 0x8 read from port 0
The value 0x808 read from port 0
The value 0x808 read from port 0
```

توابع نوشتن در پورت

برای نوشتن اطلاعات در پورت سخت‌افزاری از چهار تابع استفاده می‌شود. این توابع را در این بخش بررسی خواهیم کرد.

ارتباط با دستگاه‌های جانبی ۶۶۱

تابع (`outp()`)

این تابع یک بایت را در پورت سخت‌افزاری می‌نویسد، در فایل `conio.h` قرار دارد و به صورت زیر به کار می‌رود:

```
int outp(unsigned portid, int value);
```

`portid` شماره پورتی است که مقدار `value` باید در آن نوشته شود (بایت کم‌ارزش `value` نوشته می‌شود). این تابع مقدار `value` را نیز برمی‌گرداند.

تابع (`outpw()`)

این تابع یک کلمه را در پورت سخت‌افزاری می‌نویسد، در فایل `conio.h` قرار دارد و به صورت زیر به کار می‌رود:

```
unsigned outpw(unsigned portid, unsigned value);
```

`value` مقداری است که باید در پورت `portid` نوشته شود. `value` به عنوان نتیجه تابع نیز برگردانده می‌شود.

تابع (`outportb()`)

این تابع یک بایت را در پورت سخت‌افزاری می‌نویسد، در فایل `conio.h` قرار دارد و به صورت زیر به کار می‌رود:

```
void outportb(unsigned portid, unsigned char value);
```

این تابع مقدار `value` را در پورت `portid` می‌نویسد.

تابع (`outport()`)

این تابع یک کلمه را در پورت سخت‌افزاری می‌نویسد، در فایل `conio.h` قرار دارد و به صورت زیر به کار می‌رود:

```
void outport(unsigned portid, unsigned value);
```

این تابع مقدار `value` را در پورت `portid` می‌نویسد.

مثال ۲

برنامه‌ای که بایت و کلمه‌ای را در پورت سخت‌افزاری می‌نویسد.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int port = 0;
    int value = 'C';
    int value1 = 30231;
    clrscr();
    outp(port, value);
```

```
printf("Value %d sent to port number %d\n", value, port);
outportb(port, value);
printf("Value %d sent to port number %d\n", value, port);
outport(port, value1);
printf("Value %d sent to port number %d\n", value1, port);
outpw(port, value1);
printf("Value %d sent to port number %d\n", value1, port);
getch();
return 0;
}
```

خروجی

```
Value 67 sent to port number 0
Value 67 sent to port number 0
Value 30231 sent to port number 0
Value 30231 sent to port number 0
```

منابع و مأخذ

1. Herbert schildt, C the complete reference(C++, ANCI)
2. Herbert schildt, Turbo C the complete reference
3. Herbert schildt, C power user's guide
4. Kent A. Barclay, C problem solving and programming
5. Robert Lafor, Turbo C programming for IBM
6. Herbert schildt, Advanced Turbo C
7. Abacus, PC system programming
8. Herbert schildt, Teach yourself C
9. Kris Jamsa, Turbo C programmer's Library
10. Eliot B. Koffman & Jeri R. Hanly, problem solving and program design in C
11. Herbert schildt, Advanced C
12. Peter Aitken & Bradley Jones, Teas yourself C in 21 days