

21

“Money that comes easily disappears quickly.”

Programming Ports

Ports can be thought of as hardware connection ports where devices with input/output lines connect to a bus. The CPU has ports for each of its bus: at least ISA (Industry Standard Architecture) and memory, for the simplest CPU. So using the port addresses we can access hardware devices. For example CMOS is accessed via port 70h and 71h. The port can be Read & Write (R/W), or Read only, or Write only.

21.1 Why use ports?

Direct port access is much faster in many situations than interrupt code. I already pointed out that interrupts are the kind of subroutines and these subroutines also use ports to access hardware devices whenever it is necessary. So invoking interrupts some times mean indirect port access.

One of the important advantages of using port address is that it's the only possible way of accessing the plug-in cards and some built-in hardware.

21.2 Port vs. memory

Usually people get confused between port and memory. Actually I/O ports are addressable devices which are not in memory space. From hardware perspective, memory is usually accessed by decoding addresses and Memory-Read & Memory-Write symbols, while I/O ports are decoded using addresses and I/O-Read & I/O-Write symbols.

21.3 Usual Problems

One of the usual problems we find with I/O ports is that every plugged-in device can attempt to claim the same I/O address.

21.4 Programming ports with Turbo C

For programming ports we can use `inportb()`, `inport()`, `outportb()` and `outport()` functions. In this book, you have many programs that use ports.

21.5 Example

Here I am giving an example program to find the scan code of a key using port 60h.

```
#define ESC (1)

int main( void )
{
    int key;
    while( (key=inportb( 0x60 ))!=ESC )
    {
        printf( "%x ", key );
        /* To see the values on monitor, add appropriate delay
           to reduce flickering (for faster machines only) */
        delay(15);
    }
    return(0);
}
```

Exercises

1. Find out the ports used by different peripherals. (Hint: Look into Ralf Brown's Interrupt List)
2. Find out the port used by your mouse. Use the details to write a mouse driver program.