

---

# General Purpose Digital IO con PIC

GPIO( General Purpose IO) is the most basic method of communication between MCU and external world. These are done with what is called a PORT. Ports are nothing but a set of bits physically connected to PINs of Microcontroller and available outside the chip. As we are working on PIC micros and they are 8 bits so the maximum size of a PORT is 8 bits. Some PORTs have fewer than 8 bits. You can turn each bit in a PORT on and off under program control. The bits which you set as 1 becomes HIGH and the physical PIN of Micro is at Vcc(supply voltage, usually 3.3v or 5v). And the PINs which you clear (=0) becomes low and physical level is 0v(GND).

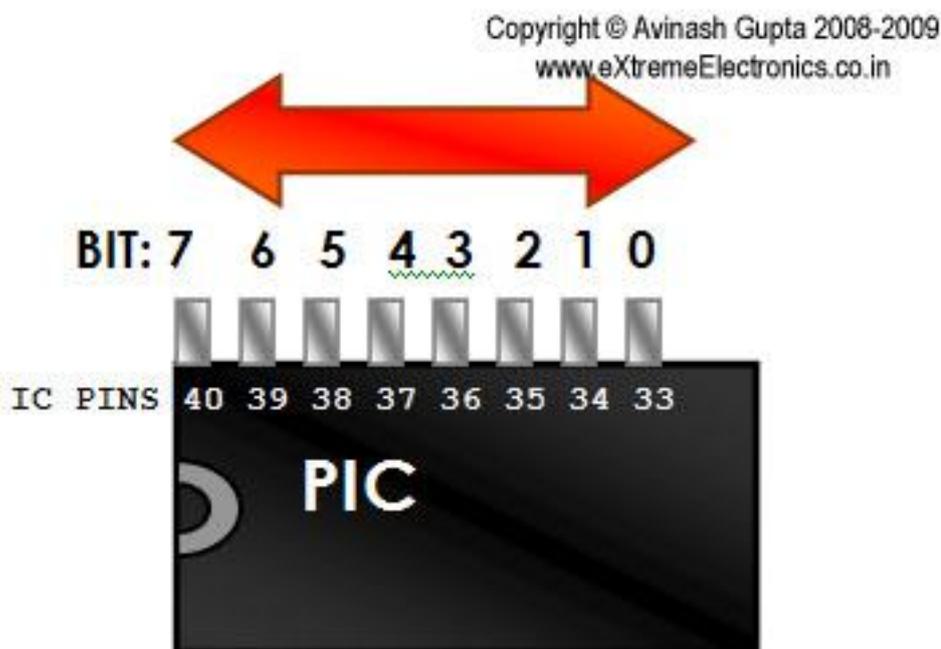
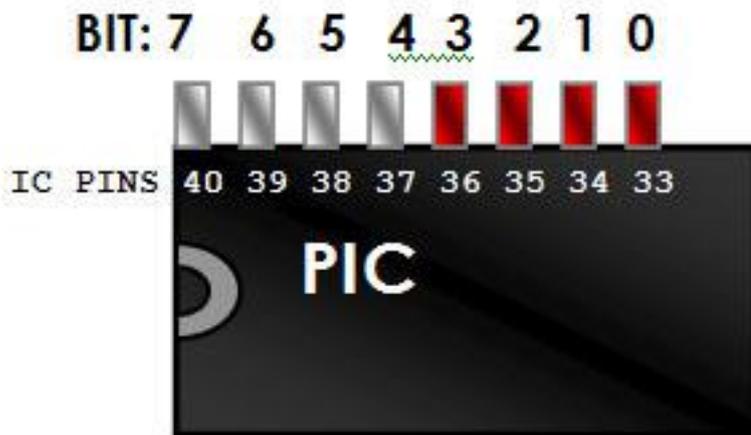


Fig.: PIC IO Port example, PORTB.



**PORTB = 0x0F (00001111)**  
**RED PINS are AT 5V Level**

Fig.: PIC IO Port example, Using PORTB, Turning BITS on and off.

You can test the operation of an IO port by using LEDs as shown below.

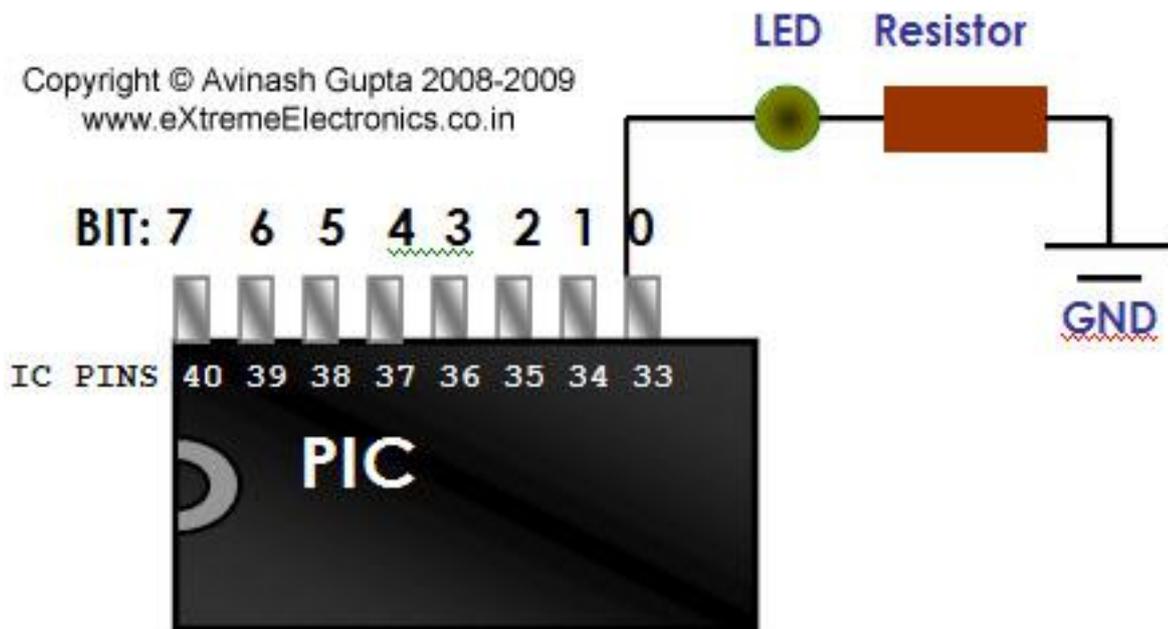


Fig.: When the bit is set to 0 the LED remains off.

When you write a value 1 to corresponding bit the LED starts glowing.

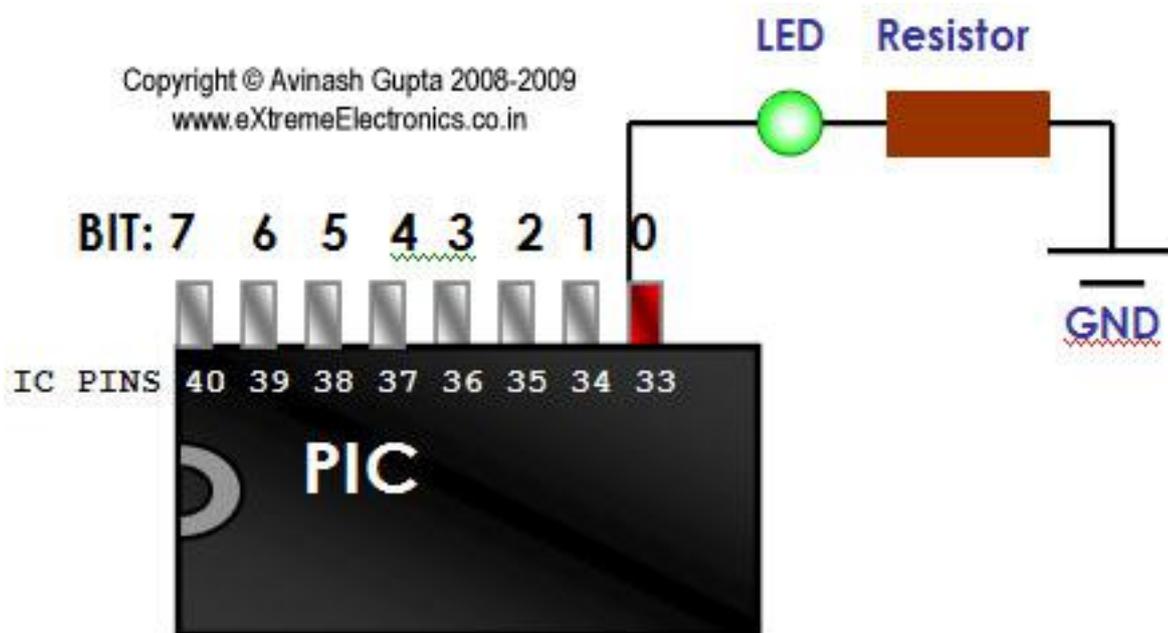


Fig.: When the bit is set to 1 the LED starts glowing.

PORTs are Named like PORTA,PORTB,PORTC etc. The PIC we are using PIC4550 has the following PORTs

- PORTA(8bit wide)
- PORTB(8bit wide)
- PORTC(7bit wide)
- PORTD(8bit wide)
- PORTE(4bit wide)

## Inputs using PORTs

You have seen how ports can be used to control the outside world. In that we have used the PORTs as Output. They can also be used to get input from outside. In input mode we can apply either logic 0 (GND) or 1 ( $V_{cc} = 5v$ ) and we can read the status of PORT and get what input we have on ports. This can be used to interface switches and sensor with MCU. Any value below  $V_{cc}/2$  is taken as low and any voltage above  $V_{cc}/2$  is high.

A simple switch interface is shown below.

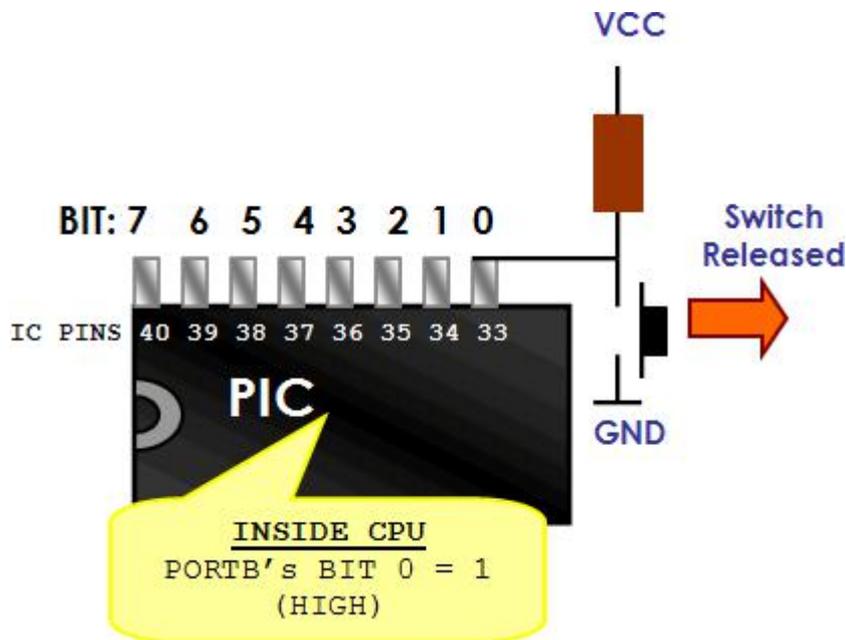


Fig.: A Simple switch as Input to PIC Micro.

The CPU can read the required port and take decisions accordingly.

# Practical Stuff.

Having the knowledge about digital IO Ports on PIC micro we can begin some practical experimentations. So how can we actually access the ports in C? Well the answer follows.

Each PORT in PIC micro has 3 associated registers.

- **TRIS:** This is the data direction register – As I have told you that port can be used as both input and output purpose, the direction control (input or output) is done using this register. Simply make the bit as 1 when you want its as input or 0 for output.

Example:

```
TRISB=0xFF; //1111-1111 All Inputs

TRISB=0x00; //0000-0000 All Outputs

TRISB=0x0F;//0000-1111 Bit 7,6,5,4, as Output and Bit 3,2,1,0
as Input
```

Note: On power up all PORTs are Input.

To Remember this note that 1 looks like 'i' therefore Input, 0 looks like 'o' so Output

- **LAT: (LATCH):** This register is used to set output values. Anything you want output on related pins must be put on this.

Example

```
LATB=0xFF; //All PINs becomes high

LATB=0x00; //All pins becomes Low

LATB=0x03; //0000-0011 BIT 0 and 1 becomes high.
```

- **PORT:** This is used to get input from the PORTs. Reading its value gives the actual levels applied on the PINs. It must be noted that PORT is first configured as input using the TRIS register. Suppose you connected a switch as shown above on PORTB-0 then if you read PORTB it will be 0000-0000 if switch is pressed and 0000-0001 if switch is released.