

[\(/wiki\)](#)[\(/shop\)](#)

# PIC I/O Register Configuration

[Contents ▾](#)

In this tutorial we are going to discuss the port configuration of PIC controllers. In this tutorial we will be using PIC16F877A as reference, same will be applicable for other PIC series controllers.

At the end of this tutorial you will be familiar with the PIC GPIO's and the associated registers for configuring and accessing the GPIO's.

## GPIO Registers

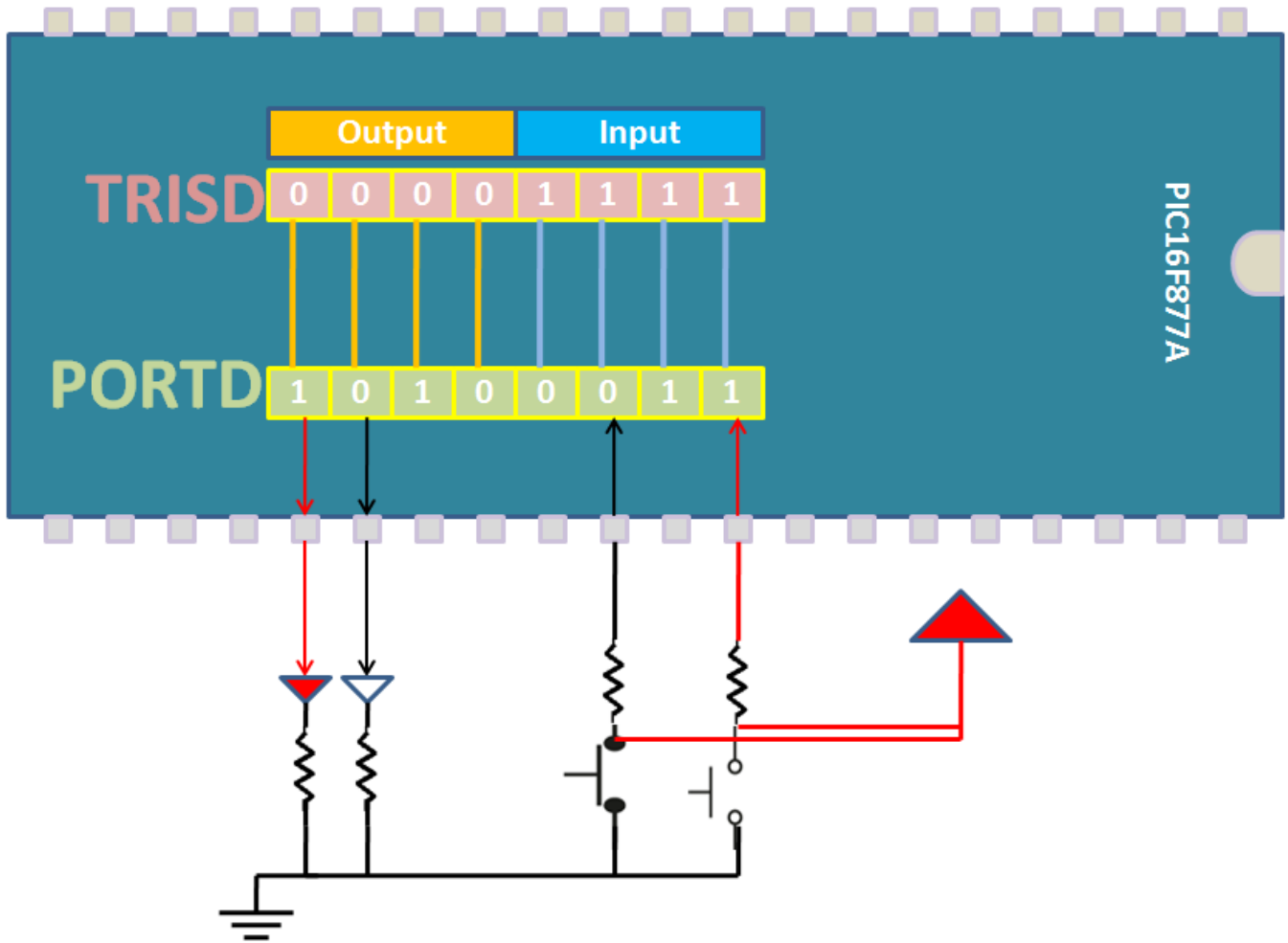
The basic and important feature of any controllers is the number of gpio's available for connecting the peripherals. PIC16F877A has 33-gpio's grouped into five ports namely PORTA-PORTE as shown in the below table.

PORT	Direction Register	Number of Pins	Alternative Function
PORTA	TRISA	6 (PA0-PA5)	ADC
PORTB	TRISB	8 (PB0-PB7)	Interrupts
PORTC	TRISC	8 (PC0-PC7)	UART,I2C,PWM
PORTD	TRISD	8 (PD0-PD7)	Parallel Slave Port
PORTE	TRISE	3 (PE0-PB2)	ADC

As shown in the above table many I/O pins have 2-3 functions. If a pin is used for other function then it may not be used as a gpio.

Though the gpio pins are grouped into 8-bit ports they can still be configured and accessed individually.

Each Port is associated with 2 registers for direction configuration(Input/Output) and for Read/Write.



(/wiki/File:Pic16f877a\_register\_configuraion.png)

Register	Description
TRISx	Used to configure the respective PORT as output/input
PORTx	Used to Read/Write the data from/to the Port pins

note: Here 'x' could be A,B,C,D,E so on depending on the number of ports supported by the controller.

**TRISx:** TRI-State Register/ Data Direction Register

Before reading or writing the data from the ports, their direction needs to be set. Unless the PORT is configured as output, the data from the registers will not go to controller pins.

This register is used to configure the PORT pins as Input or Output. Writing 1's to TRISx will make the corresponding PORTx pins as Input. Similarly writing 0's to TRISx will make the corresponding PORTx pins as Output.

1. `TRISB = 0xff; // Configure PORTB as Input.`
- 2.
3. `TRISC = 0x00; // Configure PORTC as Output.`
- 4.
5. `TRISD = 0x0F; // Configure Lower nibble of PORTD as Input and higher nibble as Output`
- 6.
7. `TRISD = (1<<0) | (1<<3) | (1<<6); // Configure PD0,PD3,PD6 as Input and others as Output`

### PORTx:

This register is used to read/write the data from/to port pins. Writing 1's to PORTx will make the corresponding PORTx pins as HIGH. Similarly writing 0's to PORTx will make the corresponding PORTx pins as LOW.

**Before reading/writing the data, the port pins should be configured as InputOutput.**

1. `PORTB = 0xff; // Make all PORTB pins HIGH.`
- 2.
3. `PORTC = 0x00; // Make all PORTC pins LOW..`
- 4.
5. `PORTD = 0x0F; // Make Lower nibble of PORTD as HIGH and higher nibble as LOW`
- 6.
7. `PORTD = (1<<PD0) | (1<<PD3) | (1<<PD6); // Make PD0,PD3,PD6 HIGH,`
- 8.
- 9.
10. `TRISB = 0x00; // Configure PORTB as Output`
11. `TRISD = 0xff; // Configure PORTD as Input`
12. `PORTD = PORTB; // Read the data from PORTB and send it to PORTD.`

## Led Blinking Example

After knowing how to configure the GPIO ports, its time to write a simple program to blink the Leds.

1. Configure the PORTS as outputs using TRIS registers.
2. Turn ON all the LEDs and wait for some time.
3. Turn OFF all the LEDs and wait for some time.

1	<code>#include &lt;pic16f877a.h&gt;</code>
2	

3	<code>void DELAY_ms(unsigned int ms_Count)</code>
4	<code>{</code>
5	<code>    unsigned int i,j;</code>
6	<code>    for(i=0;i&lt;ms_Count;i++)</code>
7	<code>    {</code>
8	<code>        for(j=0;j&lt;1000;j++);</code>
9	<code>    }</code>
10	<code>}</code>
11	
12	<code>int main()</code>
13	<code>{</code>
14	<code>    /* Configure all the ports as Output */</code>
15	<code>    TRISA = 0x00;</code>
16	<code>    TRISB = 0x00;</code>
17	<code>    TRISC = 0x00;</code>
18	<code>    TRISD = 0x00;</code>
19	
20	<code>    while(1)</code>
21	<code>    {</code>
22	<code>        PORTA = 0xff; /* Turn ON all the leds connected to Ports */</code>
23	<code>        PORTB = 0xff;</code>
24	<code>        PORTC = 0xff;</code>
25	<code>        PORTD = 0xff;</code>
26	<code>        DELAY_ms(100);</code>
27	
28	<code>        PORTA = 0x00; /* Turn OFF all the leds connected to Ports */</code>
29	<code>        PORTB = 0x00;</code>
30	<code>        PORTC = 0x00;</code>
31	<code>        PORTD = 0x00;</code>
32	<code>        DELAY_ms(100);</code>
33	<code>    }</code>
34	
35	<code>    return (0);</code>
36	<code>}</code>

sharanago/6561921de354fc92f5254845dcc16063/raw/00104cb34c20d405a32517e551930a468757ab1e/pic16f877a\_ledBlinking.c)  
pic16f877a\_ledBlinking.c ([https://gist.github.com/sharanago/6561921de354fc92f5254845dcc16063#file-pic16f877a\\_ledblinking-c](https://gist.github.com/sharanago/6561921de354fc92f5254845dcc16063#file-pic16f877a_ledblinking-c)) hosted with ❤ by GitHub (<https://github.com>)

# Led and Switches

```
1  #include <pic16f877a.h>
2
3  int main()
4  {
5      TRISB = 0x00;          // Configure PORTB as output to connect Leds
6      TRISD = 0xff;          // Configure PORTD as INput to connect switches
7
8      while(1)
9      {
10         PORTB = PORTD;      // Read the switch status and display it on Leds
11     }
12     return 0;
13 }
```

bagwan/614103c3fa59fa99864bc71a03b99302/raw/ab1e62814433a0f539779151a323cf1900f4d96a/pic16f877a\_SwitchLed.c)  
pic16f877a\_SwitchLed.c ([https://gist.github.com/SaheblalBagwan/614103c3fa59fa99864bc71a03b99302#file-pic16f877a\\_switchled-c](https://gist.github.com/SaheblalBagwan/614103c3fa59fa99864bc71a03b99302#file-pic16f877a_switchled-c)) hosted with ❤ by GitHub (<https://github.com>)

## Downloads

Download the complete project folder from the below link:

Hardware design Files and Code Library

([https://github.com/ExploreEmbedded/Pic16f877a\\_ExploreUltraPicDevKit/archive/master.zip](https://github.com/ExploreEmbedded/Pic16f877a_ExploreUltraPicDevKit/archive/master.zip))

Have a opinion, suggestion , question or feedback about the article let it out here!

ALSO ON EXPLOREEMBEDDED.COM/WIKI

4 years ago · 13 comments

**Setting up Eclipse for ESP-IDF**

3 year

**Bin Ser No.**



What food delivery serv

Category (/wiki/Special:Categories): PIC Tutorials (/wiki/Category:PIC\_Tutorials)

**Subscribe to hear about our latest Explorations!**

name@example.com

SUBSCRIBE

[Contact \(/contact\)](/contact) [About \(/about\)](/about) [Warranty \(/refund\)](/refund) [Terms & Conditions \(/terms\)](/terms) [Reward points \(/rewards\)](/rewards)

<https://twitter.com/exploreembedded> <https://www.facebook.com/ExploreEmbedded/>

<https://www.youtube.com/channel/UCvXGpvPuosEI-ALxvCrSbaA> <https://github.com/ExploreEmbedded>



Now shipping worldwide from India with