



SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution



Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECB211 – Microcontroller Programming & Interfacing

II YEAR/ IV SEMESTER

UNIT 3 – PIC PROGRAMMING IN C

TOPIC 2 – I/O Programming in C



I/O Programming in C



- Input / Output (I/O) programming in C for PIC microcontrollers involves controlling the digital and analog input/output pins of the microcontroller to interact with external devices such as sensors, actuators, and other integrated circuits
- Here's a basic overview of how you can perform I/O programming in C for PIC microcontrollers

1. Include PIC specific header file: Before you start writing your code, include the appropriate header file for the specific PIC microcontroller you are using

eg

for a PIC16F877A microcontroller, you would include the pic16f877a.h header file

```
#include <pic16f877a.h>
```



I/O Programming in C



2. Define Configuration Bits

PIC microcontrollers require configuration bits to be set to configure various settings like oscillator type, watchdog timer, etc. These bits are usually defined using preprocessor directives

```
__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON & BOREN_ON & LVP_OFF);
```

3. Initialize I/O Ports

Configure the direction of each pin as either input or output using the TRIS register. Set the pin high or low using the PORT register

```
// Configure PORTB pins as output
TRISB = 0x00;
// Set PORTB pins to logic high PORTB = 0xFF;
```

4. Read Input from a Pin

To read the digital input from a pin, use the PORT register. For example, to read the input from pin RB0

```
if (PORTBbits.RB0 == 1) {
// RB0 is high
} else {
// RB0 is low
}
```



I/O Programming in C



5. Write Output to a Pin

To write a digital output to a pin, use the PORT register. For example, to set pin RB1 high
`PORTBbits.RB1 = 1;`

6. Analog Input/Output: If your PIC microcontroller supports analog-to-digital (ADC) or digital-to-analog (DAC) conversion, you can use these features to read analog inputs from sensors or output analog signals to actuators. Consult your microcontroller's datasheet and use the appropriate ADC or DAC modules and associated registers.

7. Interrupts: PIC microcontrollers support interrupts, which allow the microcontroller to respond to external events in real-time. You can configure interrupts to trigger on changes in I/O pins, timer overflows, etc.

```
// Example of configuring external interrupt on RB0 pin  
INTCONbits.INTE = 1; // Enable RB0/INT external interrupt  
INTCONbits.GIE = 1; // Enable global interrupts
```



THANK YOU

19ECT312/Emb.Sys /
Dr.B.Sivasankari/Professor/ECE/SNSCT