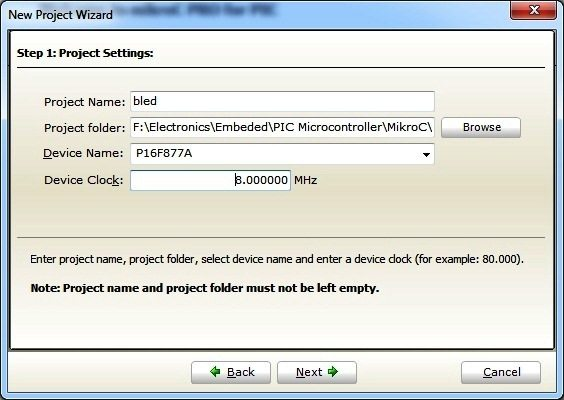
**LED Blinking mikroC**

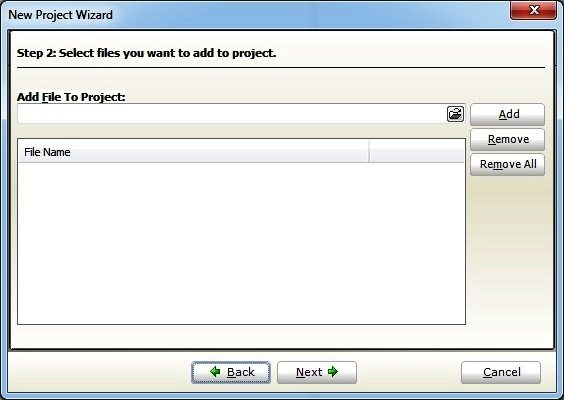
* Click on **Project >> New Project**



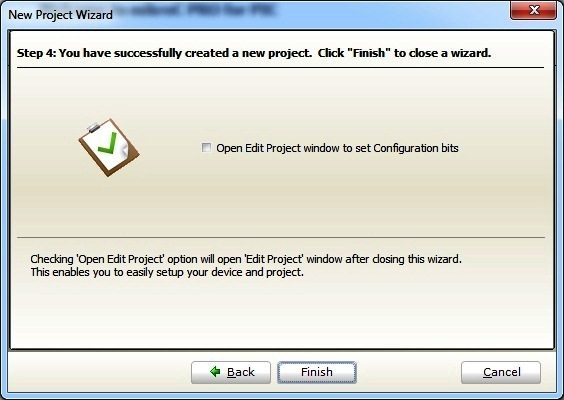
* Click **Next**



* Enter Project name, path (created folder path), clock frequency, microcontroller and click **Next**. Clock Frequency is the frequency of oscillator used with microcontroller. Here we use PIC 16F877A microcontroller with 8MHz crystal.



* Here you can add your subprogram files or user defined header files in large projects. Hence we are dealing with simple LED Blinking in this tutorial, ignore it and click **Next**.

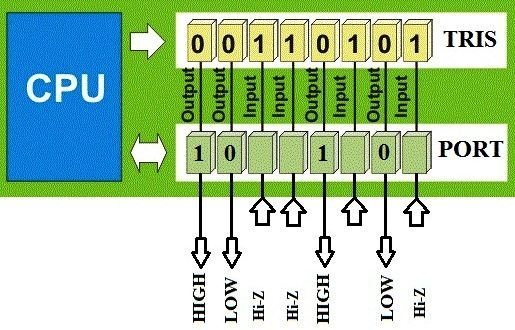


* Click **Finish**, to complete the New Project Wizard.
* Then you will see the editor, where you can enter the MikroC Code.

**MikroC Programming**

Before going to the programming you should understand the following things:

* Output pins of a PIC Microcontroller is divided in to different **PORTS** containing a group of **GPIO** (General Purpose Input Output Pins) pins.
* In 16F PIC Microcontrollers, there are two registers associated with a port, TRIS and PORT. eg: **TRISB**, **PORTB**, **TRISC**, **PORTC**
* **TRIS** stands for Tri-State, which determines the direction of each GPIO pin. Logic 1 at a particular bit of **TRIS** register makes the corresponding pin Input and Logic 0 at a particular bit of **TRIS** register makes the corresponding pin Output. An Input pin of PIC Microcontroller have very high input impedance, thus it may said to be in Hi-Impedance state.
* **PORT** register is used to read data from or write data to GPIO pins. Logic 1 at a particular bit of **PORT** register makes the corresponding pin at Logic High (VDD) and Logic 0 at a particular bit of **PORT** register makes the corresponding pin at Logic Low (VSS) if that pin is an Output pin (**TRIS** bit is 0).
* **PORT** register can be used to read digital data from an Input pin. Logic 1 indicates the pin is at Logic High and Logic 0 indicates that the pin is at Logic Low.



* You can write to PORT and TRIS register entirely or bit by bit.

TRISC.F0 = 1; //Makes 0th bit of PORTC Input

TRISC.F5 = 0; //Makes 5th bit of PORTC Output

PORTB.F3 = 1; //Makes 3ed bit of PORTB at Logic High

PORTB.F7 = 0; //Makes 7th bit of PORTB at Logic Low

* A number with a prefix ‘0b’ indicates a binary number.
* A number with a prefix ‘0’ indicates an octal number.
* A number with a prefix ‘0x’ indicates a hexadecimal number.
* A number without prefix is a decimal number.

Let’s see some examples…

|  |  |  |  |
| --- | --- | --- | --- |
| Decimal | Binary | Octal | Hexadecimal |
| 0 | 0b00000000 | 00 | 0x00 |
| 1 | 0b00000001 | 01 | 0x01 |
| 128 | 0b10000000 | 0200 | 0x80 |
| 255 | 0b11111111 | 0377 | 0xFF |

PORTB = 0xFF; //Makes all pins of PORTB Logic High

TRISC = 0x00; //Makes all pins of TRISC Output

PORTD = 128; //Makes 7th bit of PORTD Logic High

**MikroC Code – Blinking an LED**

The following program blinks an LED with a delay of 1 second.

void main()

{

TRISB.F0 = 0; //Makes PORTB0 or RB0 Output Pin

while(1) //Infinite Loop

{

PORTB.F0 = 1; //LED ON

Delay\_ms(1000); //1 Second Delay

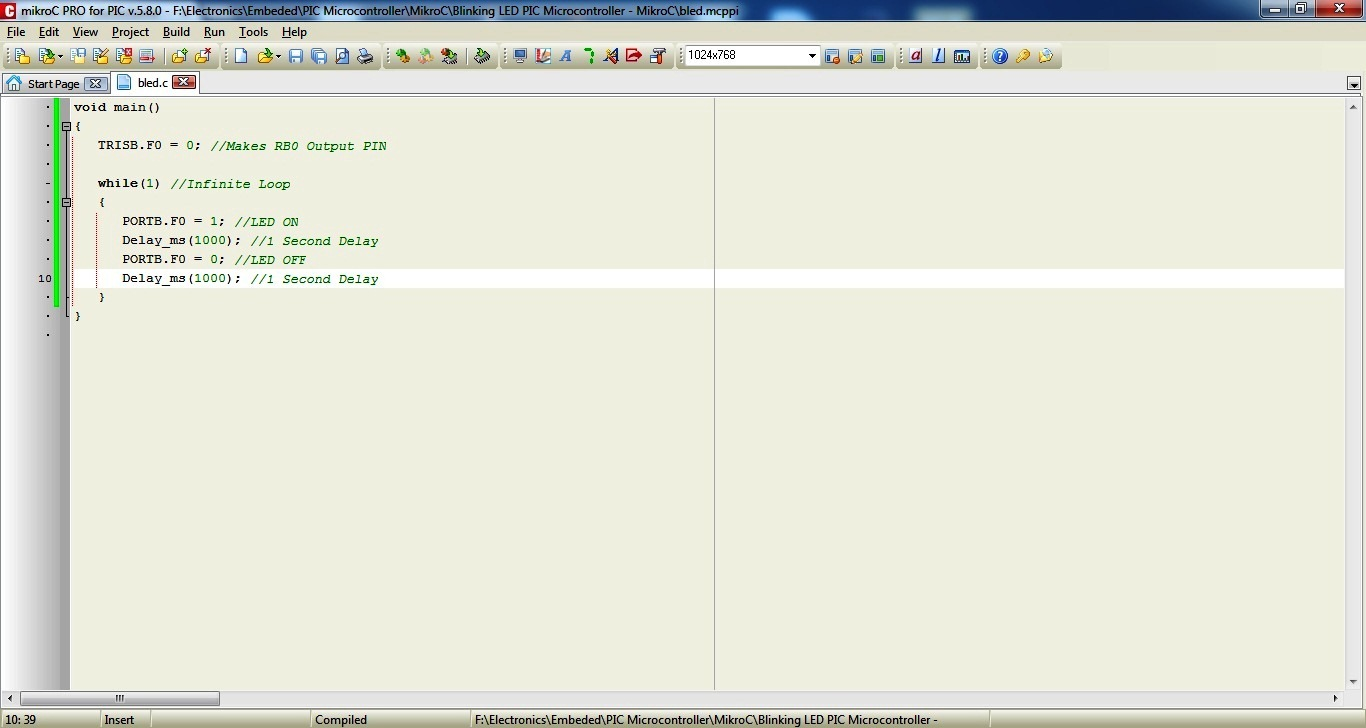
PORTB.F0 = 0; //LED OFF

Delay\_ms(1000); //1 Second Delay

}

}

* Enter the above MikroC code



Save it

Then Compile it. Click Build >> Build (or Ctrl+F9)

A hex file will be generated in your Project Folder. You need to write this file to microcontroller using a programmer.

