

# Beginner's Guide To PIC18 Microcontrollers



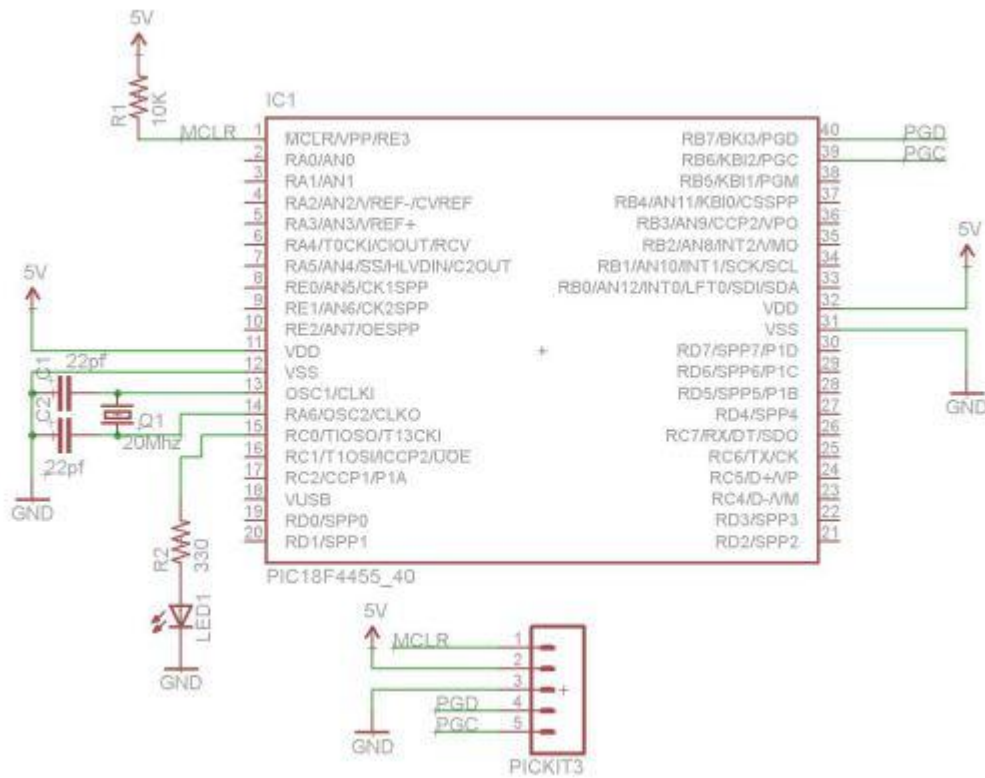
Starting today, I will focus on tutorials for Microchip's PIC18F series microcontrollers. Why am I shifting my focus from Atmega's to PIC18F microcontrollers? As I stated in the first post of the N64 emulator controller, which is now Pygames Controller, I spent a semester at RPI learning how to use these microcontrollers and completely forgot how to use Atmegas. Not to mention, a lot of PIC microcontrollers are used for more commercial applications than Atmegas. If I remember correctly, Atmegas are mostly used for automotive applications while PICs are used for more commercial devices (e.g. medical devices). But enough about that! Let's talk about how to start your first PIC18 microcontroller project.

## Necessary Parts

1. 1x PIC18F4553

2. 2x 22pf capacitors
3. 1 20Mhz crystal
4. 1 PICKIT 3 programmer
- A. It's highly recommended that you read this user guide for the programmer.
- B. Lots of male headers
- C. 1x Breadboard
- D. Lots of wire
- E. 1x 330ohm resistor
- F. 1x LED (any color)
- G. 1x 10k resistor

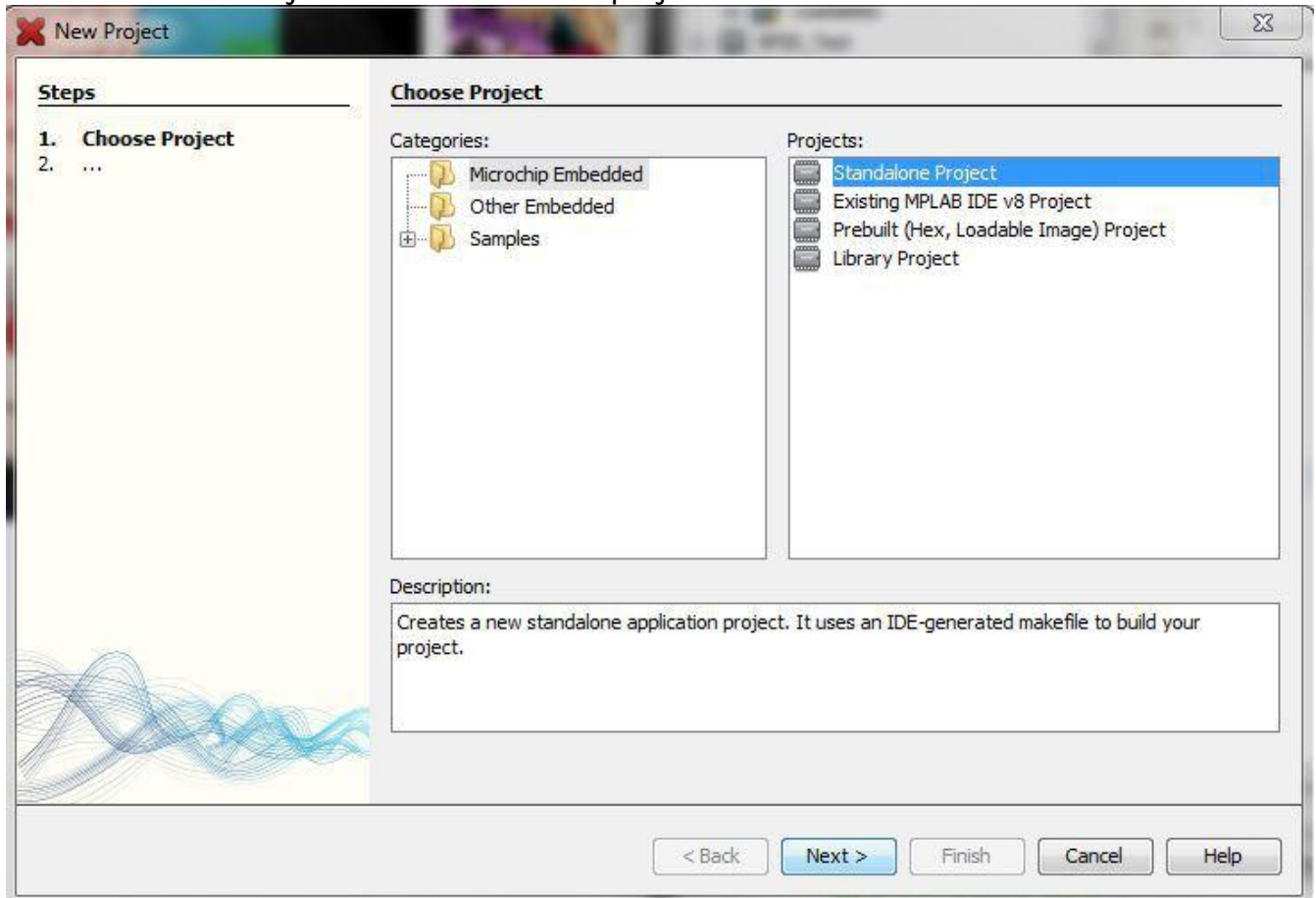
Here's the schematic of the circuit. Although the picture shows a PIC18F4455, they have the same pinout



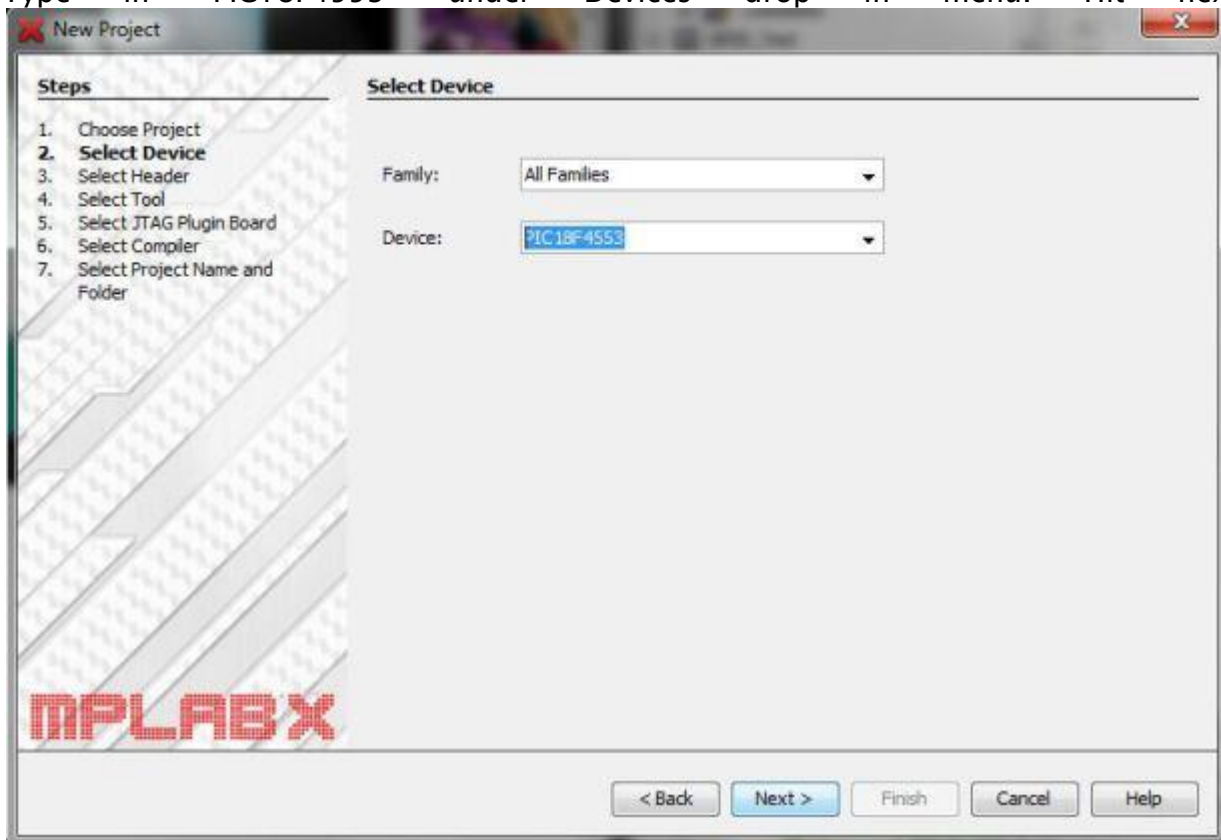
### Necessary Software

1. MPLab X
2. C18 Compiler

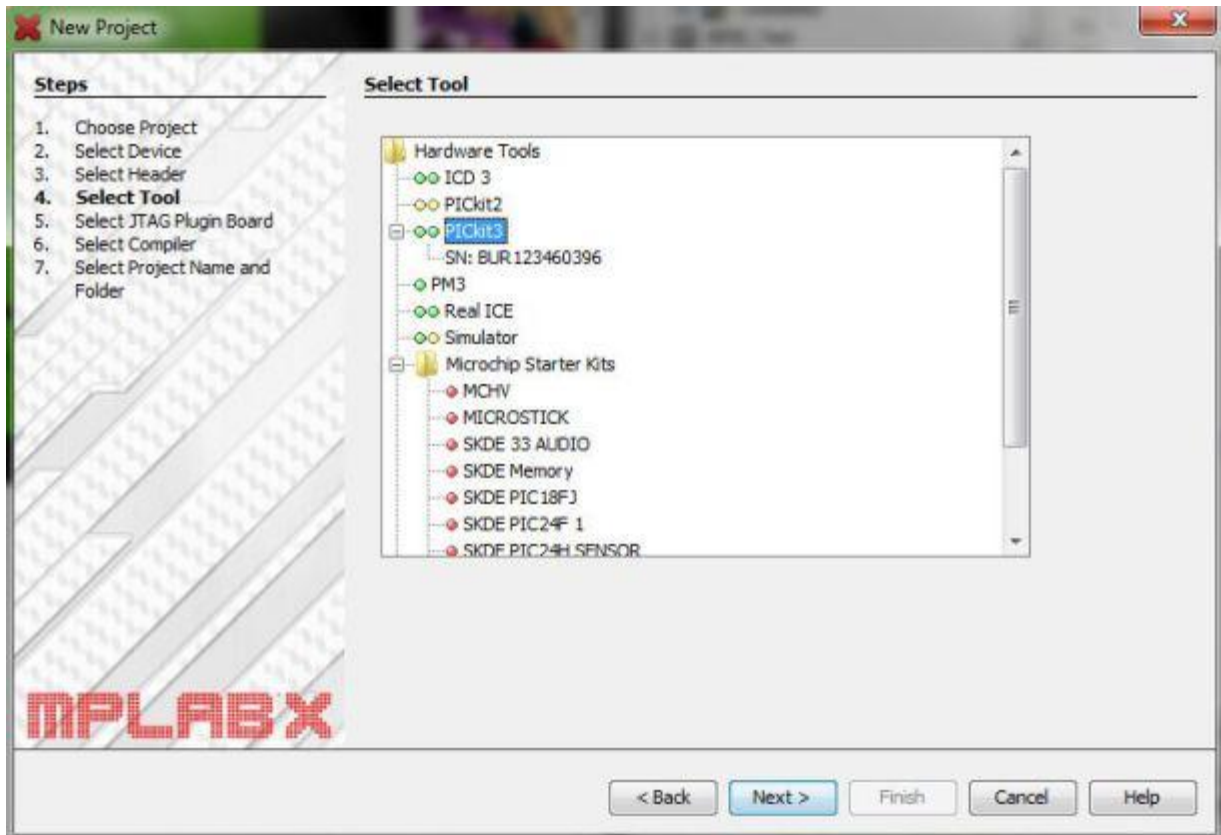
Under the C18 Compiler link, select MPLab C for PIC18. So once you guys install the necessary software, follow the next steps for setting up your project with MPLab X. First select file->New Project. Select Standalone project and hit next.



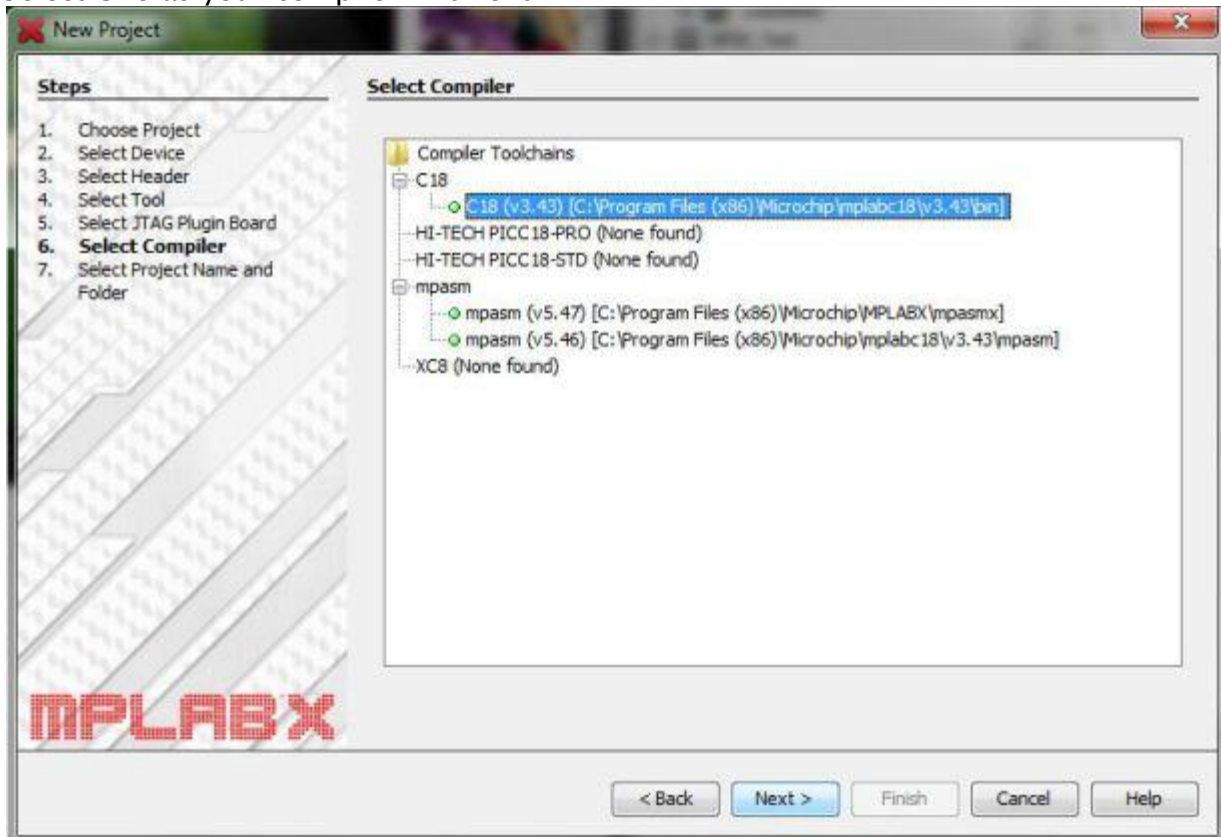
Type in "PIC18F4553" under Devices drop in menu. Hit next.



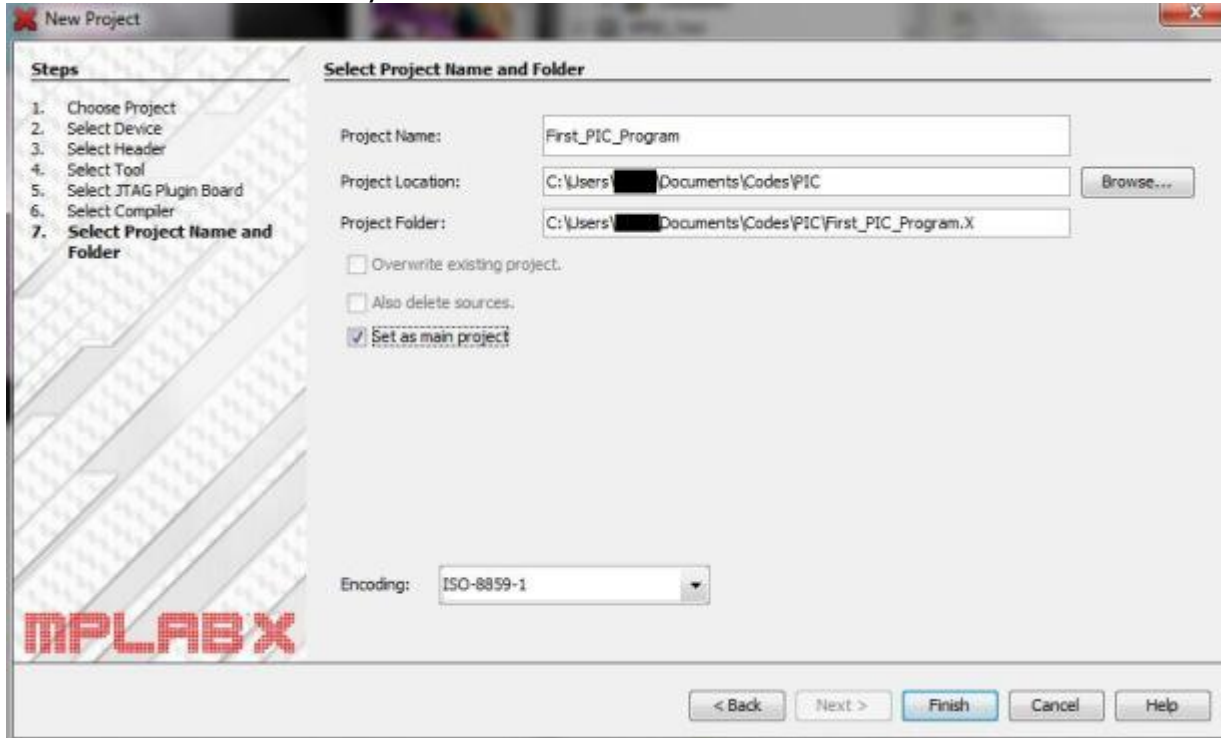
Select PICKit3 under tools. Press next.



Select C18 as your compiler. Hit next.



Now we have create a project name and directory. Normally the project location and folder will be placed under my documents. Before this tutorial, I set the project location under my PIC folder. Remember to select “Set as main project.” This will save you the frustration later on. Finally hit finish.



Now we have to create a main file. Right click on Source files and select New-> C Main file. Type in main.c and hit finish.



MPLAB X IDE v1.51 - First\_PIC\_Program : default

File Edit View Navigate Source Refactor Run Debug Team Tools Window Help

default PC: 0x0 nov z dc c : W:0x0 : bank 0

Pro... Files Classes ...in.c pwm\_test.c main.c main.c usart\_test\_code.c MAIN

Encoder\_Reading  
First\_PIC\_Program  
Header Files  
Important Files  
Linker Files  
Source Files  
Libraries  
Loadables

NS4\_test  
PWM\_Example  
Header Files  
Important Files  
Linker Files  
Source Files  
Libraries  
Loadables

RFID\_Test  
Header Files  
Important Files  
Linker Files  
Source Files  
Libraries  
Loadables

```
23 //  
24 //  
25 //  
26 //  
27 //  
28 //  
29 //  
30 // Main loop  
31 //  
32 void main(void)  
33 {  
34     // Set all of PORTC as outputs  
35     TRISC=0x00;  
36  
37     // Configure UART  
38     OpenUSART(USART_TX_INT_OFF & USART_RX_INT_OFF & USART  
39     Delay1KTCYx(4);  
40  
41     putcUSART(0x7C);  
42     Delay1KTCYx(4);  
43     putcUSART(0x04);  
44     Delay1KTCYx(4);  
45     while(1);  
46  
47 }  
48
```

First\_PIC\_Progra... : First\_PIC\_P...

First\_PIC\_Program  
Device  
PIC18F4553  
Checksum: 0xB358  
Compiler Toolchain  
C18 (v3.43) [C:\Program Files  
Memory  
RAM 2048 (0x800) bytes  
0%  
RAM Used: 0 (0x0) Free: 2  
RAM Reserved: Production  
Flash 32768 (0x8000) bytes  
0%  
Flash Used: 0 (0x0) Free:  
Flash Reserved: Productio  
Resources  
Program BP Used: 0 Free: 3  
Data BP Used: 0 Free: 3  
Data Capture BP: No Support  
SW BP: No Support  
Debug Tool  
PICKIT3

Variables Output Tasks

TEST\_USART (Build, Load, ...) x PICKIT 3 x

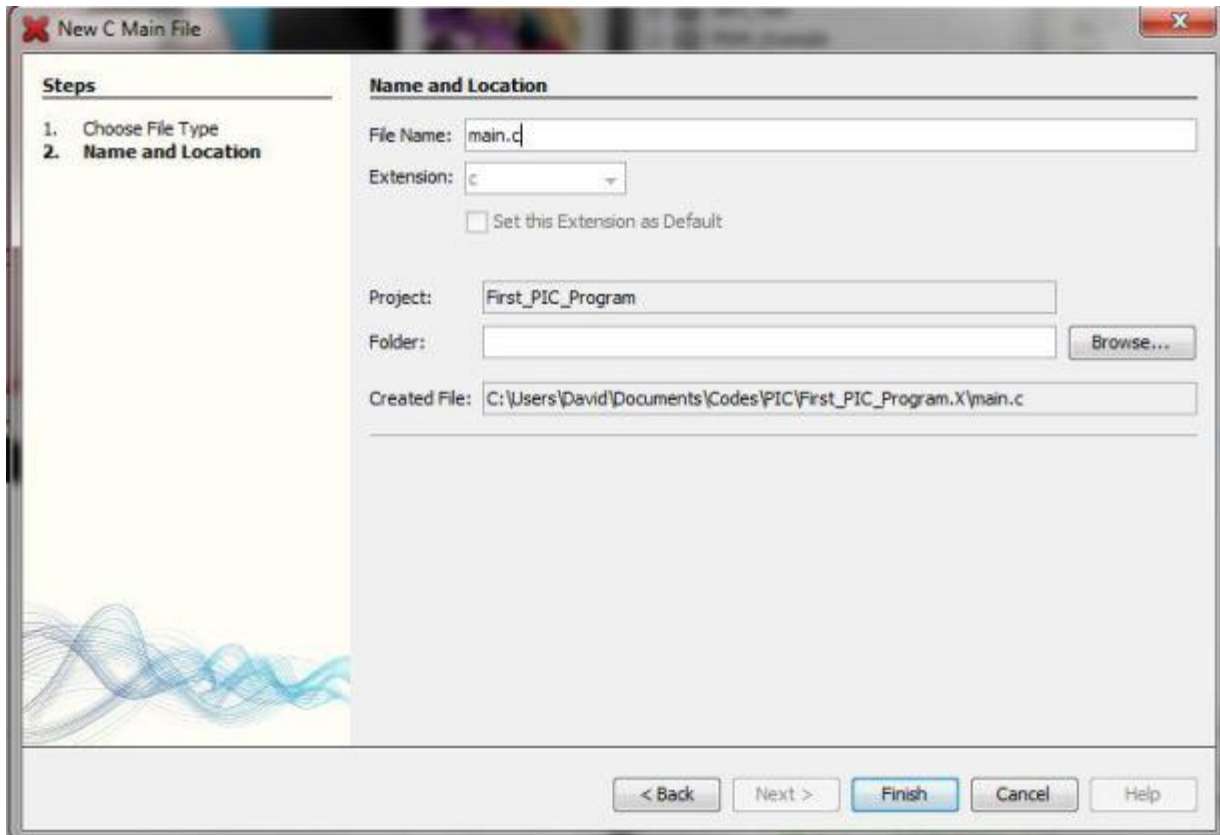
Firmware Suite Version.....01.28.18  
Firmware type.....PIC18F

Target detected  
Device ID Revision = 6

The following memory area(s) will be programmed:  
program memory: start address = 0x0, end address = 0x27f  
configuration memory

Programming...  
Programming/Verify complete

45 | 18 | INS



Finally copy and paste this code into your main.c

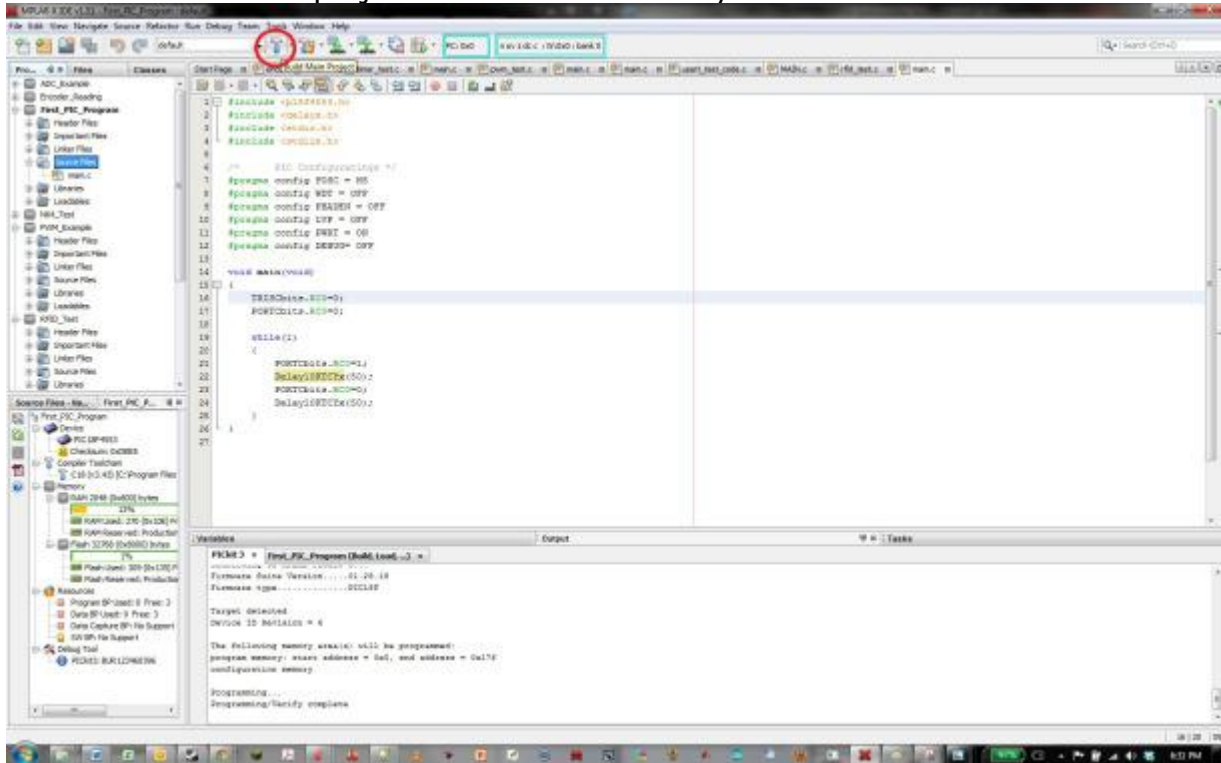
```
1  #include <pic18f4553.h>;
2  #include <delays.h>;
3  #include <stdio.h>;
4  #include <stdlib.h>;
5
6  /* PIC Configuratings */
7  #pragma config FOSC = HS
8  #pragma config WDT = OFF
```



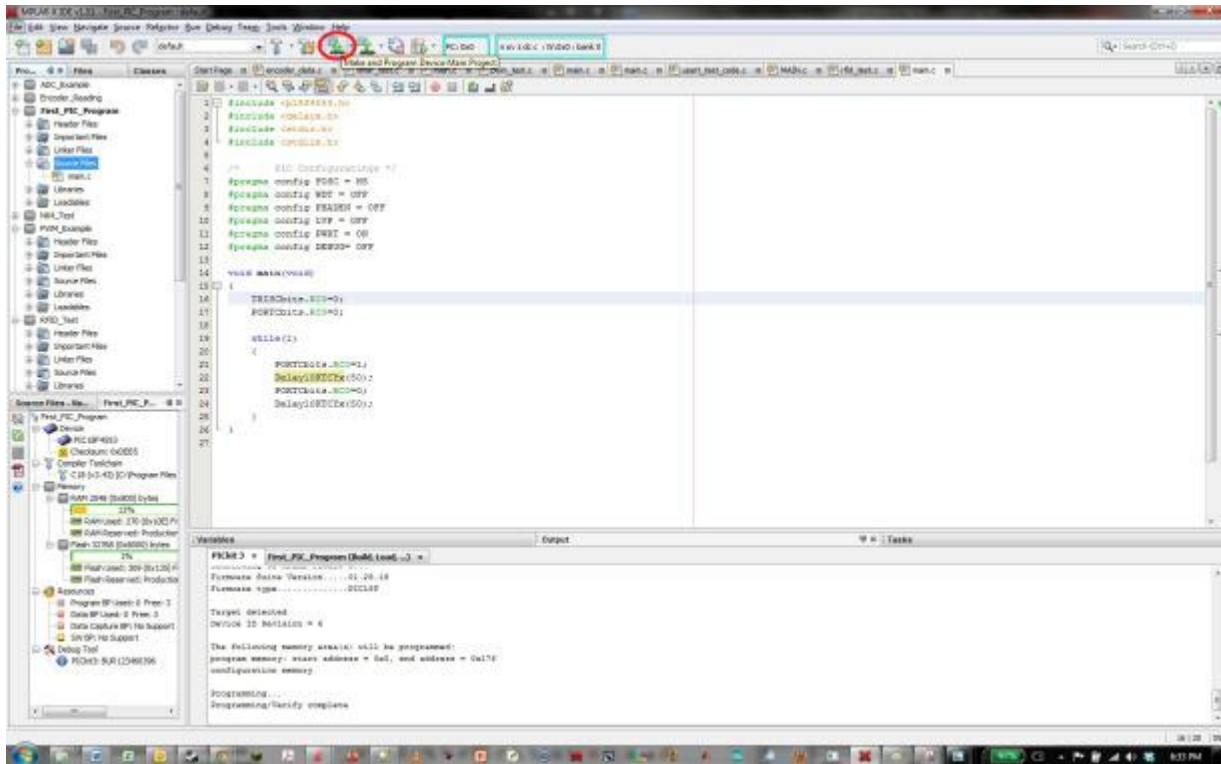
```
9   #pragma config PBADEN = OFF
10  #pragma config LVP = OFF
11  #pragma config PWRT = ON
12  #pragma config DEBUG= OFF
13
14  void main(void)
15  {
16      TRISCbits.RC0=0;
17      PORTCbits.RC0=0;
18
19      while(1)
20      {
21          PORTCbits.RC0=1;
22          Delay10KTCYx(50);
23          PORTCbits.RC0=0;
24          Delay10KTCYx(50);
25      }
```

26 } }

Now we're going to build and program our mcu. First select build. You should receive a confirmation that the project was built successfully.



Once it finishes building, hit the program button.



Congrats! Your LED is flashing at 10hz!

Here how the code works. Out al of the #pragma in the code, pay close attention to #pragma config FOSC =HS. This tells the PIC microcontroller to accept the external crystal we connect. Please note that if we were using the 28pin PIC18 series microcontroller, this would most likely be changed to #pragma config OSC =HS.

Now for the code. Whenever we want to set a particular pin as an output, we have to access the TRISx, where the x stands for the letter of the port we want (in this case, TRISC for PORTC), and write a 0 to that pin. Thus we write TRISbits.RC0=0 to set PORTC0 as an output. Finally, to either turn on or turn off this particular pin, we just have to write a 0 (off) or 1 (on) to PORTC. Thus PORTCbits.RC0=0 is setting PORTC0 off.

Finally, the rest of the code is just flashing the LED on and off every 100ms. The while(1) is there to make sure it runs forever.We, that's all I got for you guys today! Thank you guys for reading! If you have any comments or suggestions, please leave a comment!

Source: <http://coolcapengineer.wordpress.com/2012/12/20/tutorials-beginners-guide-to-pic18-microcontrollers/>