
COMP551 – Interfacing

Fall 2017

Lab 6

Programming the PIC-P40 Proto-board

Students:				
Student ID's:				
Section:	01	02	03	04

NOTE: Labs are due at the start of the next lab period. Only submit one lab per group of two students.

Lab 6 – Programming the PIC-P40 Proto-board

6. Introduction:

So far in the labs you have been doing, you have been using the MPLAB Simulator to test and run your programs. In this lab, you will be writing and compiling programs in MPLAB IDE, as you have before, but instead of running them in the simulator, you will be 'burning' them into the PIC-P40 and executing the programs directly from the PIC-P40 board.

6.1 - Connecting the Hardware:

1. Connect the USB cable from the computer to the micro USB port on the PICKkit3 programmer.
2. Connect the 6-wire cable from the programming connector of the PICKkit 3 (Figure 6-1a) to the ICSP/ICD connector on the PIC-P40 (Figure 6-1b).

NOTE: Be sure to connect pin 1 on the PICKkit 3 (indicated by the white arrow) to pin 1 on the ICSP/ICD connector on the PIC-P40.

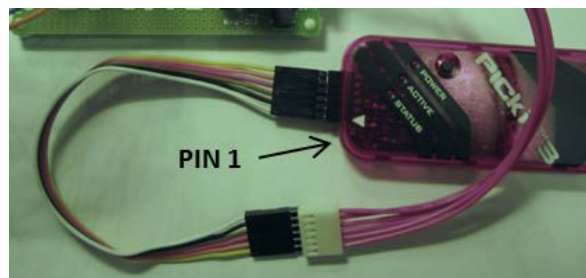


FIGURE 6-1a

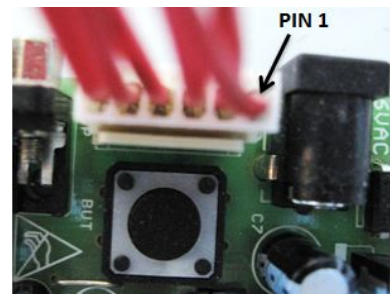


FIGURE 6-1b

3. Connect the barrel connector of the 9 VDC power supply to the barrel connector on the PIC-P40 board (Figure 6-1c) and plug the other end into an AC outlet.

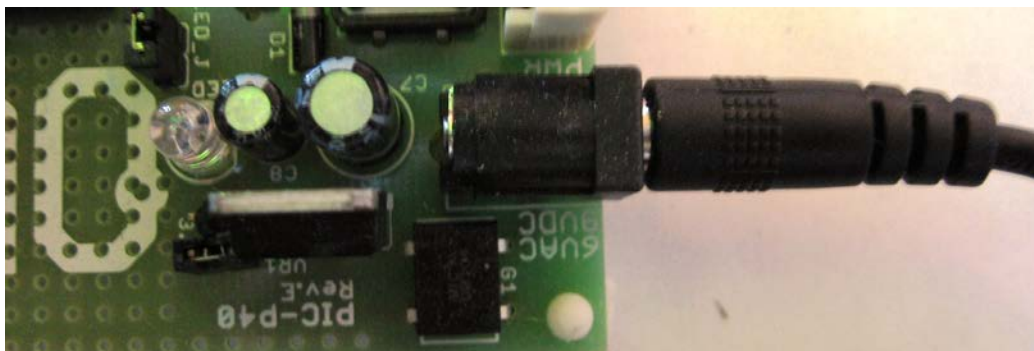


FIGURE 6-1b

6.2 - Configuring MPLAB IDE to work with PICKit 3 Programmer

- 6.2.1** In order to be able to use the PICKit 3 programmer, you will first have to unload the MPLAB SIM debugger, if it is loaded. MPLAB IDE only allows you to load the programmer or debugger, but not both at the same time. If you try to load both at the same time, you will get the following error message, Figure 6-2.

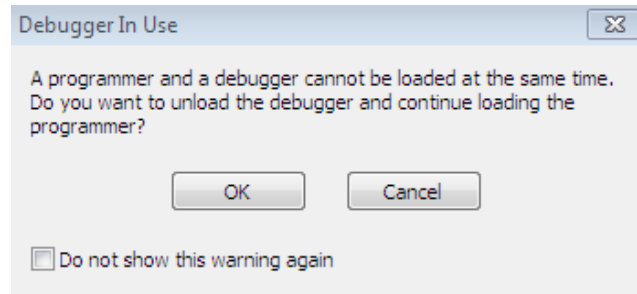


FIGURE 6-2

- 6.2.2** Start MPLAB IDE. If the MPLAB SIM debugger is loaded, select **Debugger -> Select Tool -> None** to unload it. See Figure 6-3.

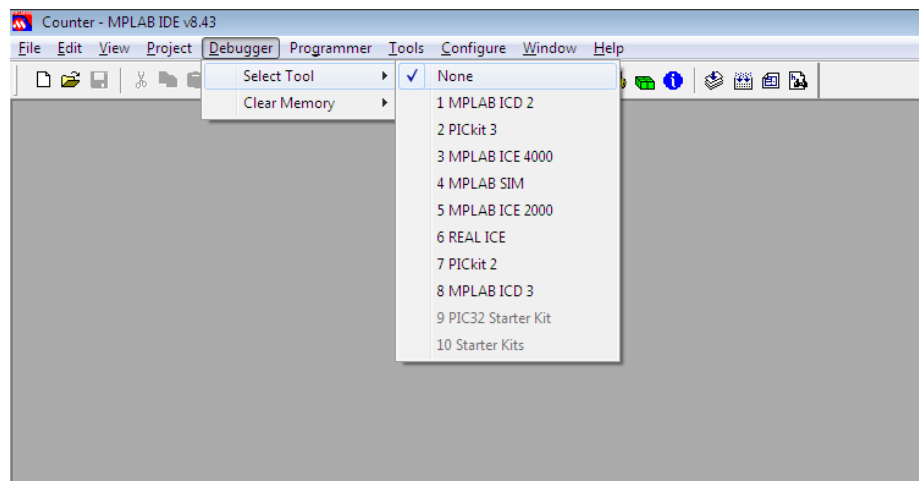


FIGURE 6-3

6.2.3 To load the PICkit 3 Programmer, select **Programmer -> Select Programmer -> PICKit 3**. See Figure 6-4.

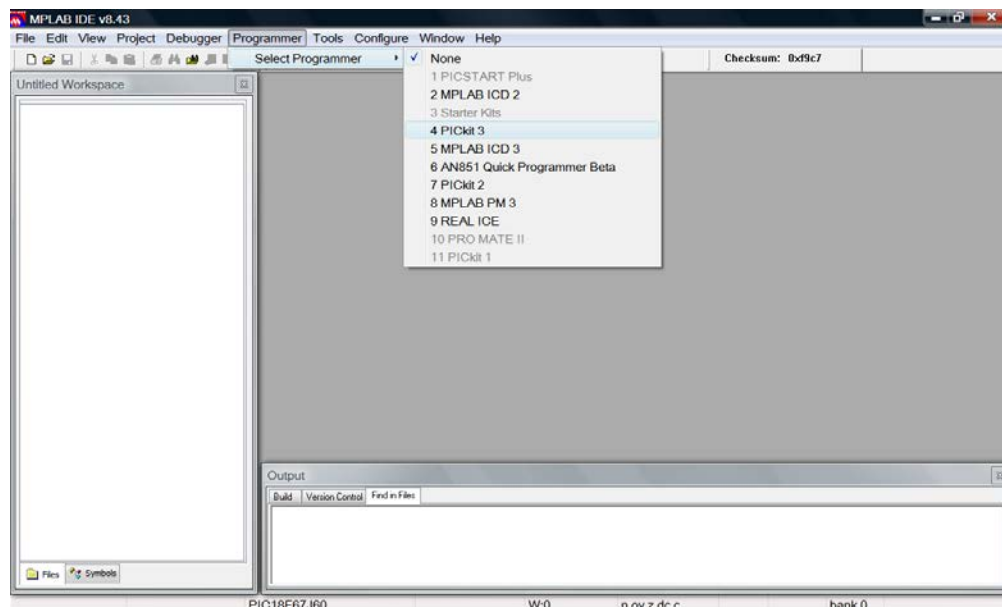



FIGURE 6-4

Once MPLAB IDE detects the PICkit 3, the following Voltage Caution window will appear. Click **OK**. See Figure 6-4a.



FIGURE 6-4a

You should now see the following in the output window (see Figure 6-5). Notice the new toolbar  available at the top.

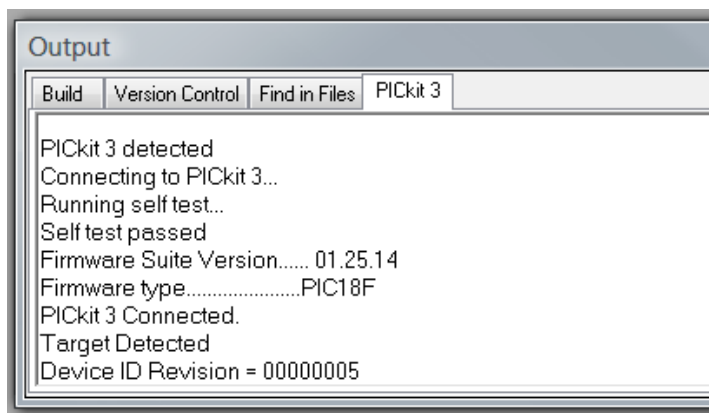


FIGURE 6-5

6.2.4 To be able to program the PIC-P40, you need to select the correct microcontroller. In MPLAB, select **Configure -> Select Device** and select the PIC18F458 from the drop down menu (Figure 6-6). Then click **OK**. If the Voltage Caution window appears again, click **OK**.

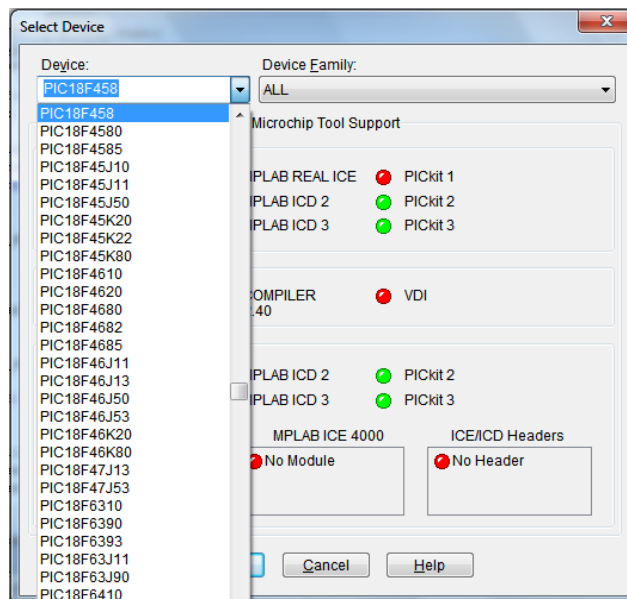


FIGURE 6-6

- 6.2.5** The first time you use the PICkit 3 programmer, it will need to download the proper firmware specific to the selected device, in this case, the PIC18F458 microcontroller. If you've selected the device for the first time, you will see a message similar to Figure 6-7.

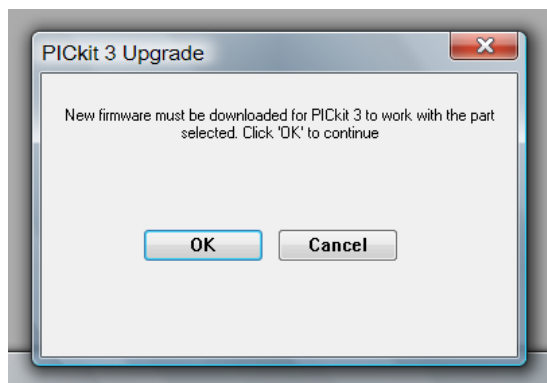


FIGURE 6-7

- 6.2.6** Click **OK** to continue. You should now see the following information in the output window (Figure 6-8).

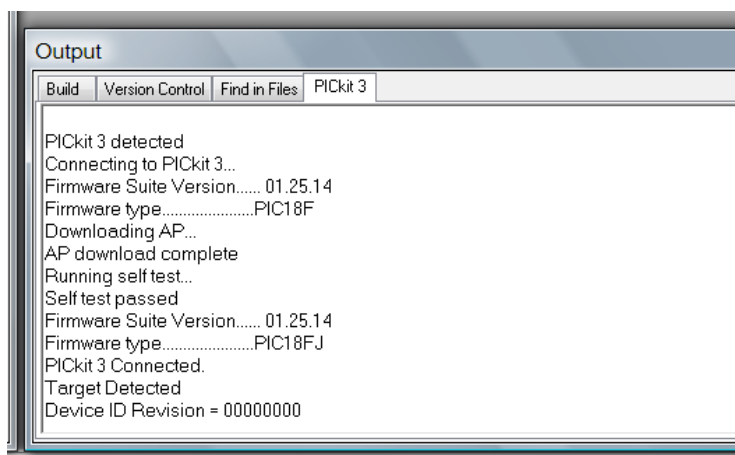


FIGURE 6-8

NOTE:

If you are using a clone PICkit3 programmer (ie. there's no Microchip logo on it) and you get an error message, please refer to the PICkit3 Clone handout (attached to this lab) for instructions on how to manually download a more recent version of the firmware.

The firmware only needs to be downloaded once for each specific microcontroller you select. Since you will only be working with the PIC-P40, which uses the PIC18F458 microcontroller, you will see the following message (Figure 6-9) each subsequent time you connect the PICkit 3 to program the PIC-P40.

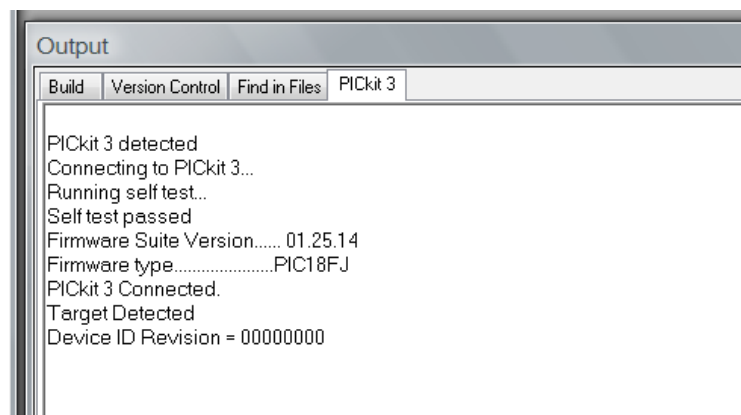


FIGURE 6-9

At this point, the PIC-P40 is ready to be programmed.

6.3 - Writing a Program to the PIC-P40

- 6.3.1** To write a program to the PIC-P40, you first create a new project and write the code the same way you would if you were using the simulator. Once you've finished coding, build the project. If there are no errors, you can then write the program to the PIC-P40 by clicking on the Program target device icon on the programmer toolbar. See Figure 6-10.

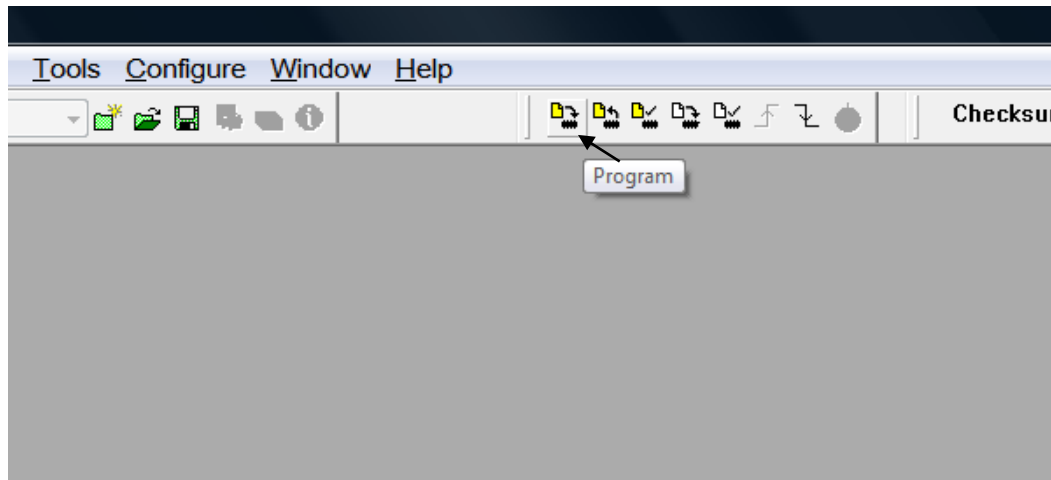


FIGURE 6-10

If all goes well, the MPLAB PICKit 3 output window should look similar to Figure 6-11.

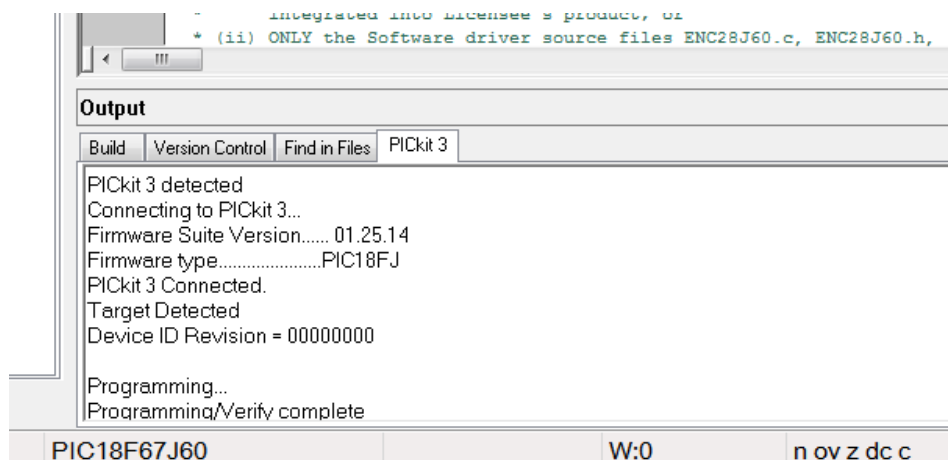


FIGURE 6-11

6.4 - Configuration Bits

Configuration bits can be programmed (read as '0'), or left un-programmed (read as '1'), to select various device configurations. Figure 6-12 lists some of the configuration bits that can be configured for the PIC18F458. The ones you need to set have been highlighted and marked with an arrow.

PIC18F458

Oscillator Selection:

OSC = LP	LP
OSC = XT	XT
OSC = HS	HS
OSC = RC	RC
OSC = EC	EC-OSC2 as Clock Out
OSC = ECIO	EC-OSC2 as RA6
OSC = HSPLL	HS-PLL Enabled
OSC = RCIO	RC-OSC2 as RA6

Osc. Switch Enable:

OSCS = ON	Enabled
OSCS = OFF	Disabled

Power-up Timer:

PWRT = ON	Enabled
PWRT = OFF	Disabled

Brown-out Reset:

BOR = OFF	Disabled
BOR = ON	Enabled

FIGURE 6-12

→

BORV = 45	4.5V
BORV = 42	4.2V
BORV = 27	2.7V
BORV = 20	2.0V

→

Watchdog Timer:

WDT = OFF	Disabled
WDT = ON	Enabled

Watchdog Postscaler:

WDTPS = 1	1:1
WDTPS = 2	1:2
WDTPS = 4	1:4
WDTPS = 8	1:8
WDTPS = 16	1:16
WDTPS = 32	1:32
WDTPS = 64	1:64
WDTPS = 128	1:128

→

Stack Overflow Reset:

STVR = OFF	Disabled
STVR = ON	Enabled

FIGURE 6-12 (Continued)

→

Low Voltage ICSP:

LVP = OFFDisabled

LVP = ONEnabled

→

Background Debugger Enable:

DEBUG = ONEnabled

DEBUG = OFFDisabled

Code Protection Block 0:

CP0 = ONEnabled

CP0 = OFFDisabled

Code Protection Block 1:

CP1 = ONEnabled

CP1 = OFFDisabled

Code Protection Block 2:

CP2 = ONEnabled

CP2 = OFFDisabled

FIGURE 6-12 (Continued)

To set the configuration bits highlighted earlier, in code, add the following statements to each program you write that will run on the PIC-P40;

```
#pragma config OSC = HS, OSCS = OFF
#pragma config PWRT = OFF, BOR = ON, BORV = 45
#pragma config WDT = OFF
#pragma config DEBUG = OFF, LVP = OFF, STVR = OFF
```

By default, the watchdog timer (WDT) is enabled on most devices. This could interfere with program execution. The **#pragma config WDT = OFF** command will disable the WDT.

Program Template

```
#include <P18F458.h>

#pragma config OSC = HS, OSCS = OFF
#pragma config PWRT = OFF, BOR = ON, BORV = 45
#pragma config WDT = OFF
#pragma config DEBUG = OFF, LVP = OFF, STVR = OFF

void main(void)
{
    ADCON1 = 0x06;           //Sets RA0 to digital mode
    CMCON = 0x07;           //Disable Comparators on RD0-RD3

    //Your code goes here.

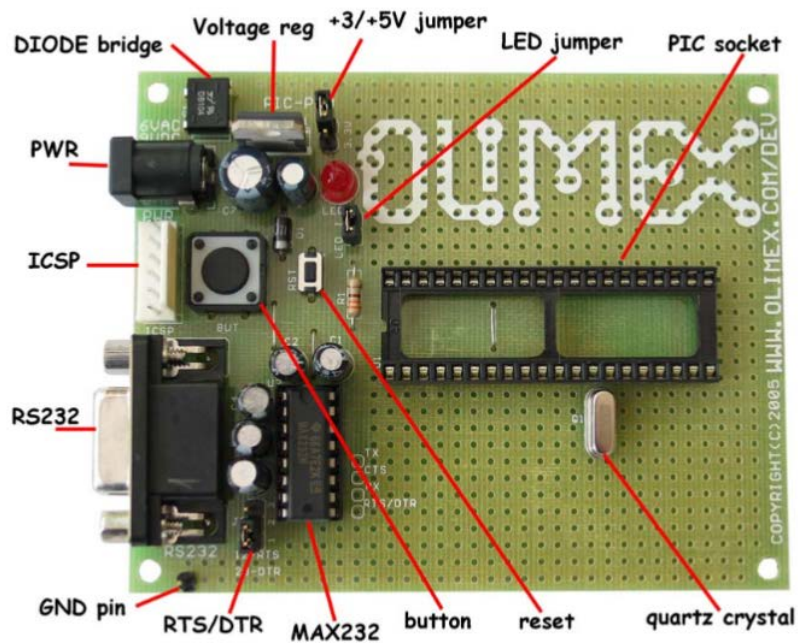
}
```

Exercises:

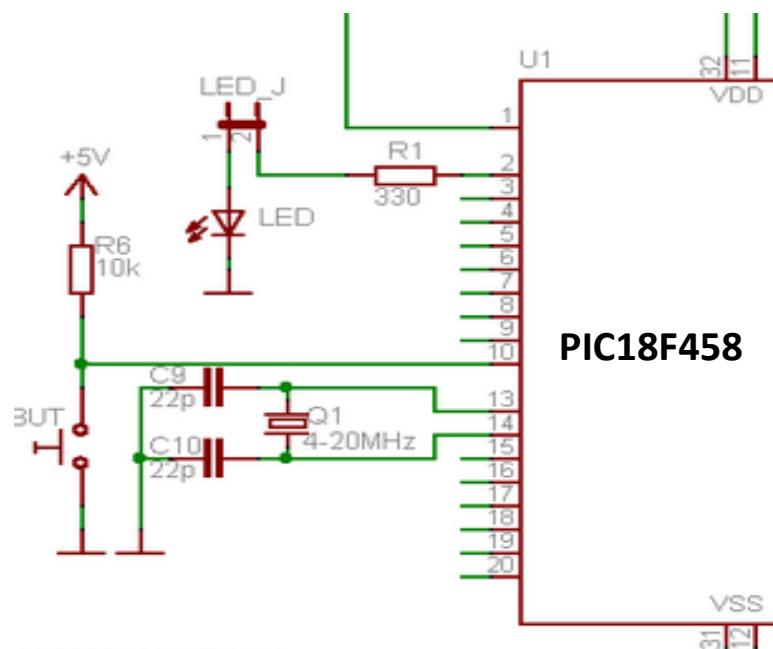
1. Write a program to flash the PIC-P40 on-board LED on and off, every $\frac{1}{2}$ second (see attached schematics for details). Use one of the Timer modules to create the delay. Be sure to use the MPLAB simulator to verify the proper operation of the program. Once you have fully debugged the program, write the code to the PIC-P40 and demonstrate its operation to the instructor.
2. Modify the program in Exercise 1, so the LED flashes on and off every 1.6 seconds and then when the on-board button (see attached diagrams and schematic) is pressed, the LED flashes on and off every 0.6 seconds. Use the Timer modules to create the delays. Be sure to use the MPLAB simulator to verify the proper operation of the program. Once you have fully debugged the program, write the code to the PIC-P40 and demonstrate its operation to the instructor.

Submit a hardcopy of the source code, the simulations, including the Stopwatch output and the calculations for the two programs.

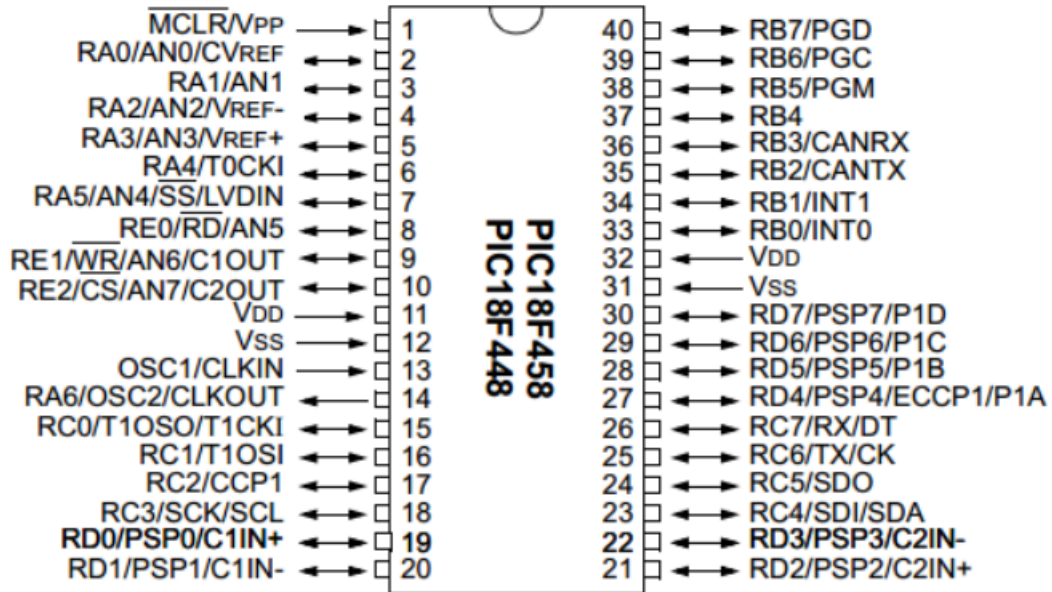
PIC-P40 DIAGRAM



PIC-P40 SCHEMATIC



PIN DIAGRAM – PDIP (Plastic Dual In-line Package)



NOTE: PORTE is only 3 bits wide.

PICkit3 Clone – Firmware Update

The problem is that the firmware on it is out of date. So you have to install new firmware manually, from the **Programmer - > Settings -> Configuration** Tab. Click on manual download, then select the JAM file. For the version of MPLAB we're using, it should be this file; PK3FW_012514.jam. Click Open and you should get these messages;

Downloading Firmware...

Downloading bootloader...

Bootloader download complete

At this point, if it looks like it's hung and still running. Unplug the USB cable from the programmer, then plug it back in. Then you will see these messages;

Running self test...

Self test passed

Downloading RS...

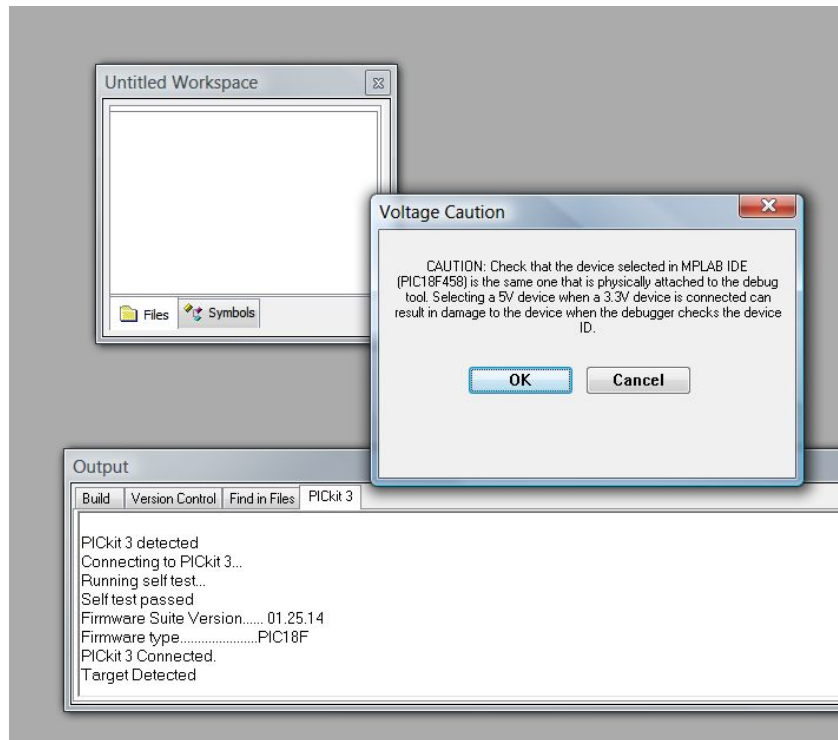
RS download complete

Again, if it looks like its hung and still running. Unplug the USB cable from the programmer, then plug it back in. Then you will see these messages;

Downloading AP...

AP download complete

Again, if it looks like its hung and still running. Unplug the USB cable from the programmer, then plug it back in. You should see this;



Click **OK**. Now you're good to go. Test it by clicking on **Programmer -> Reconnect**