# OnePIC Demo Board Quick-Start Guide

## System Requirements:

### Software:

- MPLABX  v1.00 *(Download)*
- .NET Framework v2.0 or later *(Provided - required for 8bit IR Demo GUI)*
- SimpleIO-M.dll *(Provided -required for 8bit IR Demo GUI)*

### Drivers:

- MCP2200  *(Provided)*
- ZENA Wireless Adapter (*Provided*)

### Compilers (use latest):

- PIC16 HI-TECH: v9.83 – (*Download*)
- PIC24 C30: v3.31  – *(Download)*
- PIC32 C32: v2.01  – *(Download)*

## Overview:

This demo board is populated with 8bit, 16bit, and 32bit PIC microcontrollers. Refer to the schematic of the board for more details.
**8 bit MCU : PIC16LF1939**
**16 bit MCU: PIC24FJ256GA106**
**32 bit MCU: PIC32MX795F512L**

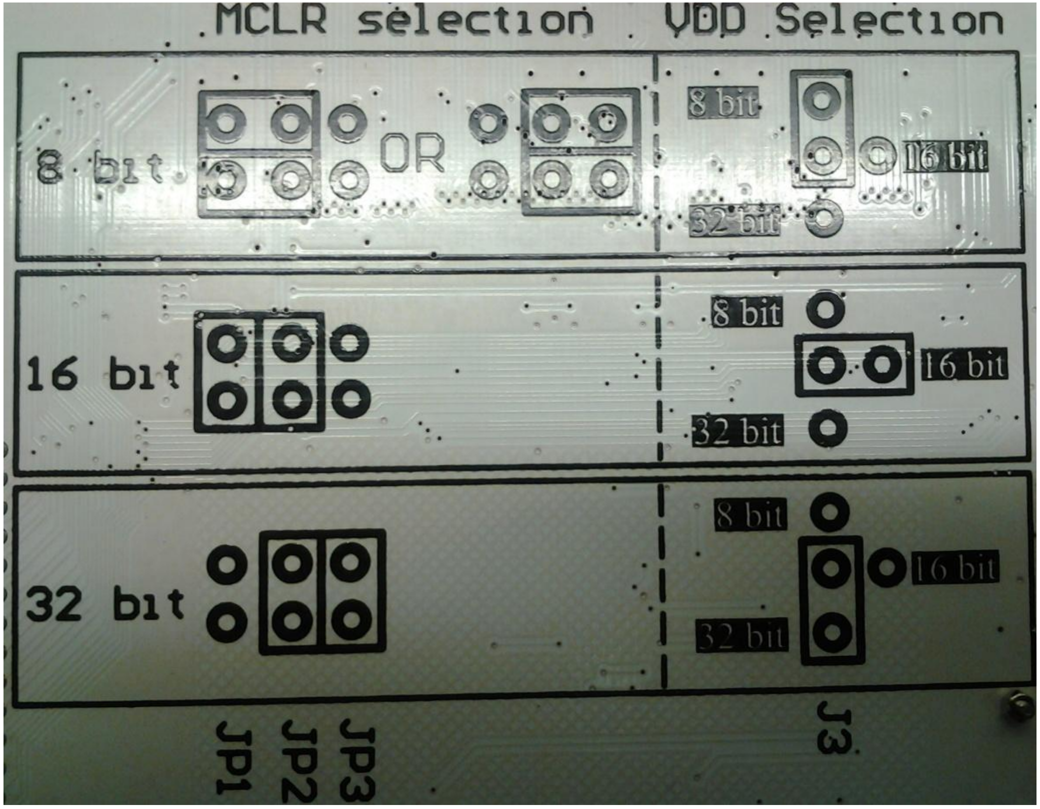
The demo code consists of:

1. A "base" code which demonstrates the
    a. LCD
    b. LEDS
    c. mTouch
    d. Potentiometer
    e. Switch
2. Three Wireless Demonstrations, one for each PIC
    a. 8bit – Infrared
    b. 16bit – RF4CE (Radio Frequency for Consumer Electronics)
    c. 32bit – WiFi

The following document is intended as a walk-through guide to using the provided software.  Items will be presented in the order they might be presented during a demonstration of the OnePIC board.

## Jumper selection for 8-bit, 16-bit and 32-bit MCUs

There are two sets of jumpers on the right top corner of the board, as shown in the photo below.



1) "MCLR Selection": This is double jumper with 2x3 positions. The jumper selection is shown on the bottom side of the board. If no shorting link on jumper, 8-bit MCU is selected by default.

2) "Vdd selection": This supplies power to the MCUs. By default, the jumper posts are not populated and all 4 jumpers are shorted on the PCB. There is no need to switch this jumper at all. The MCUs will be powered always. Only by selecting the MCLR jumper, two of the MCUs will be kept in reset and only one is kept active at any time.

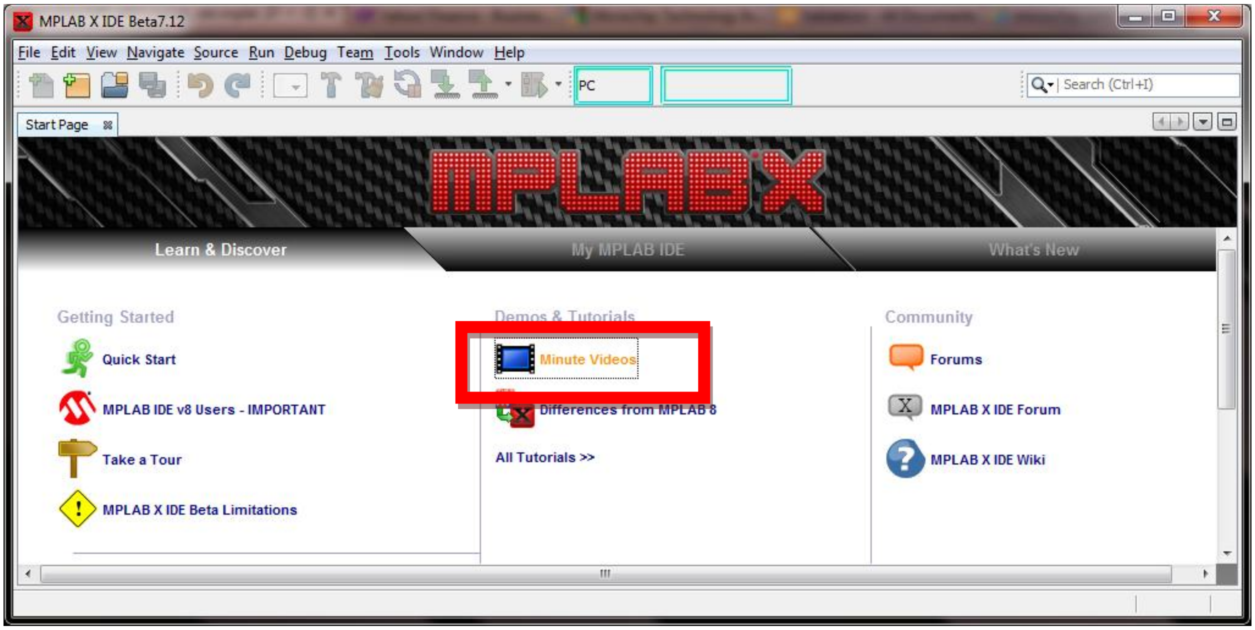
## 8bit – PIC16LF1939

### 1a) Base

A Quick Tour of MPLAB X. Hands-on #1 (Slide 50)

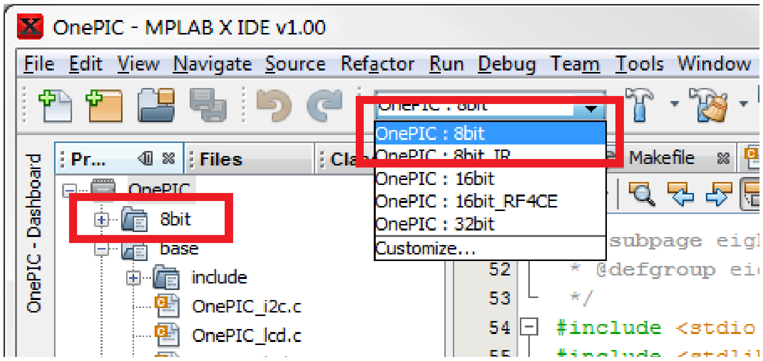
Steps to get running the base demo

1. Open MPLABX,
2. Go to the "Start Page" on the MPLAB X IDE.
3. Click the "Minute Videos"
4. Show the Video #1 to the attendees, "Creating a New Project".

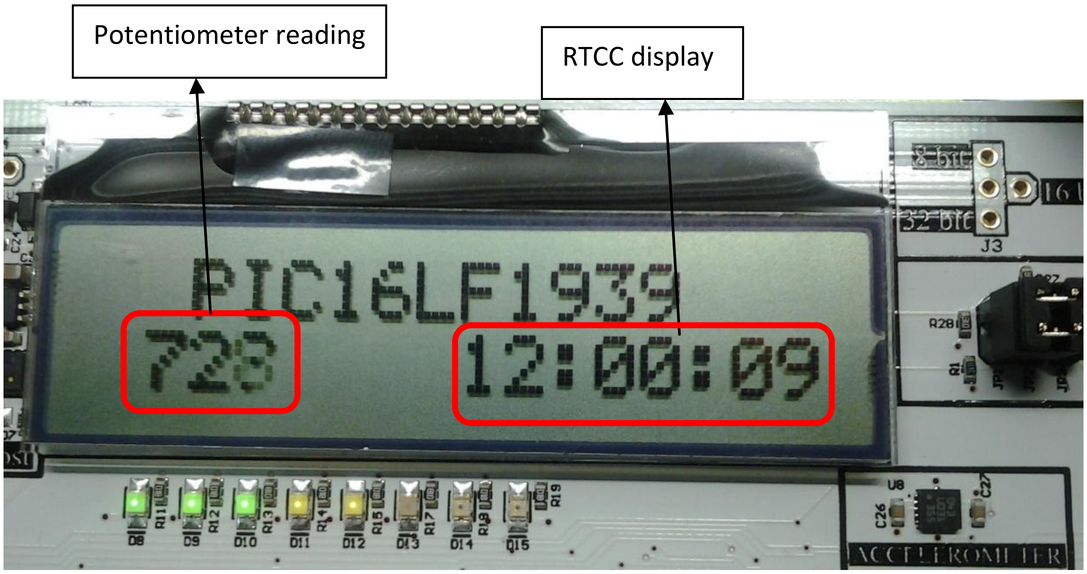


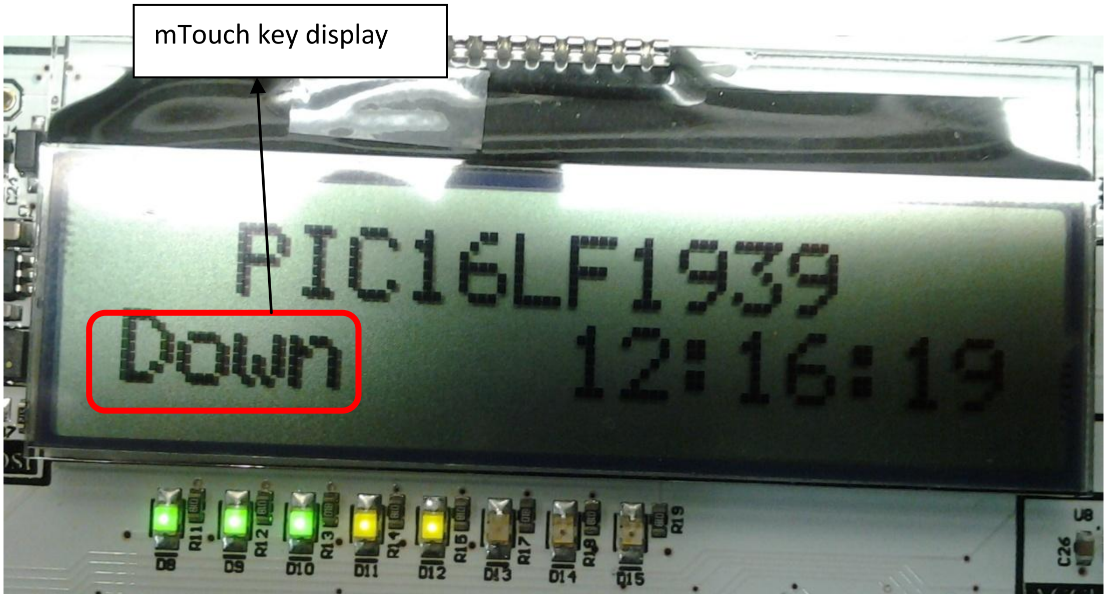
5. The second video, “Importing an MPLAB 8.xx Project” is optional to play in the class.
6. After the Video #1, the attendees should have the basic knowledge of the MPLAB X operation.

7. In the MPLAB X IDE, go to “File> Open Project>” and navigate to “OnePIC project”
8. Ensure that “OnePIC: 8bit” configuration is selected on the Configuration selection. A screen shot of the “Project Configuration” is shown in the figure below.
9. Change jumper setting on OnePIC board to 8bit. For the jumper selections, refer to the section “Jumper selection for 8-bit, 16-bit and 32-bit MCUs” above.
10. Project is now ready to build/program.



11. After the PIC16LF part is programmed, the LCD should display a splash screen with the “Microchip Technology” and PIC16LF1939.
    a. Show the attendees, LCD display, LEDs , Cap touch and potentiometer.
    b. The LCD should display as shown in the picture below. The left half of the second line shows the Potentiometer level or the Cap touch status. The right side of the second line shows the Real time clock read from MCP79410.





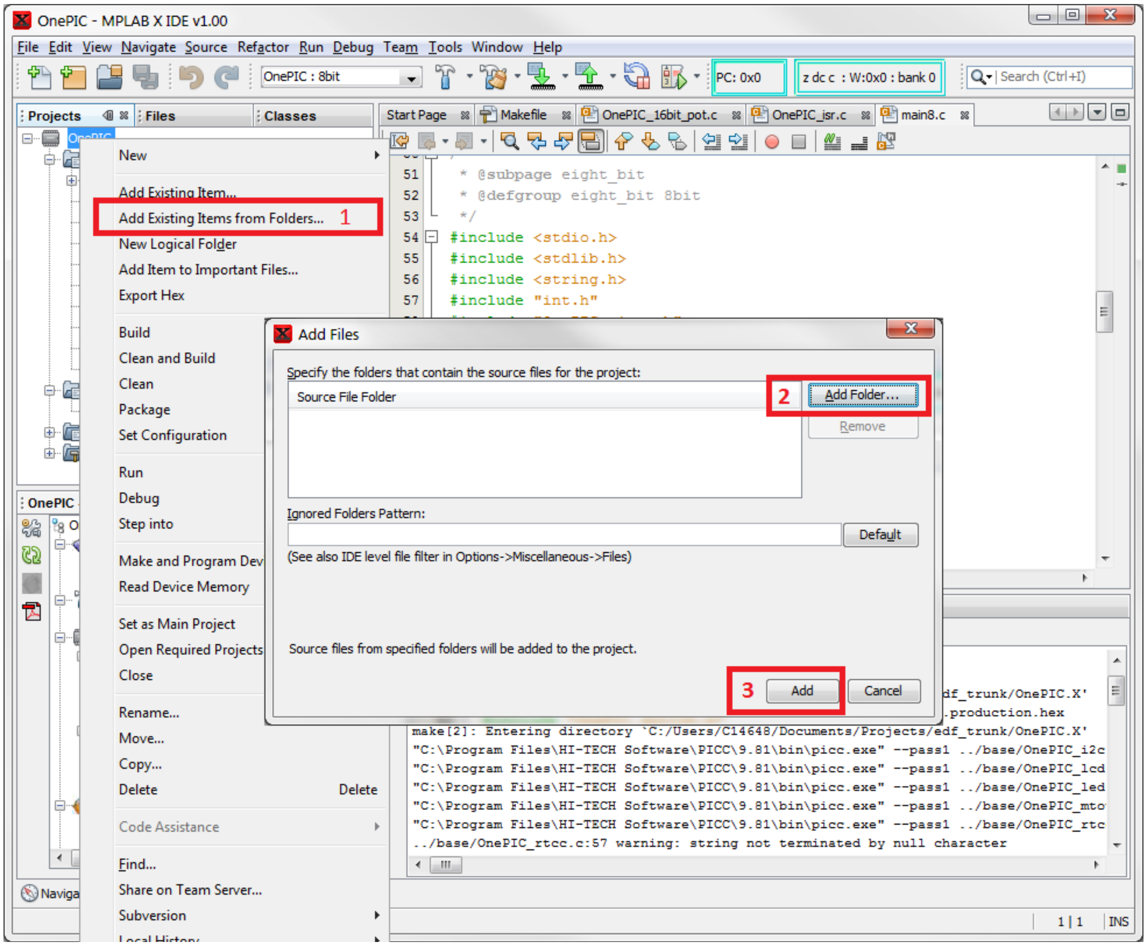
**Note: The base code in the "Base" folder remains the same for all three architectures. It includes the LCD display, LEDs and RTCC(I2C communication).**

## 1b) Infrared

Hands-on #2: IR Example (slide 88)

Steps to run demo:

1. Add 8bit IR folder by right clicking the "OnePIC" project header and follow the screenshot below.
2. Select OnePIC: 8bit_IR



```
 * @subpage eight_bit
 * @defgroup eight_bit 8bit
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "int.h"
```

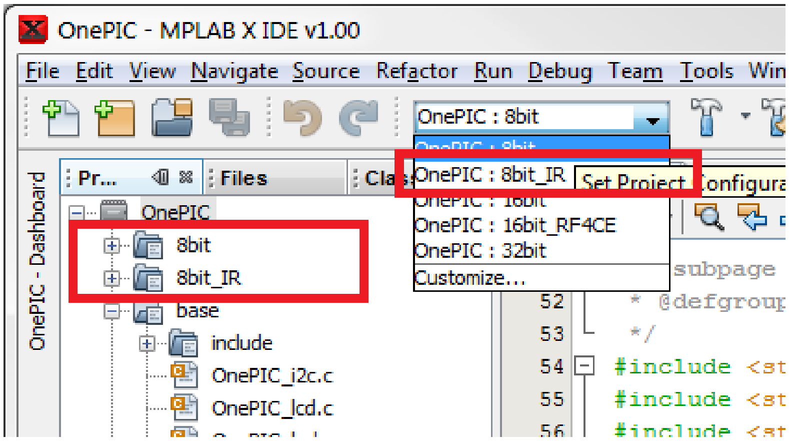
Specify the folders that contain the source files for the project:

Ignored Folders Pattern:

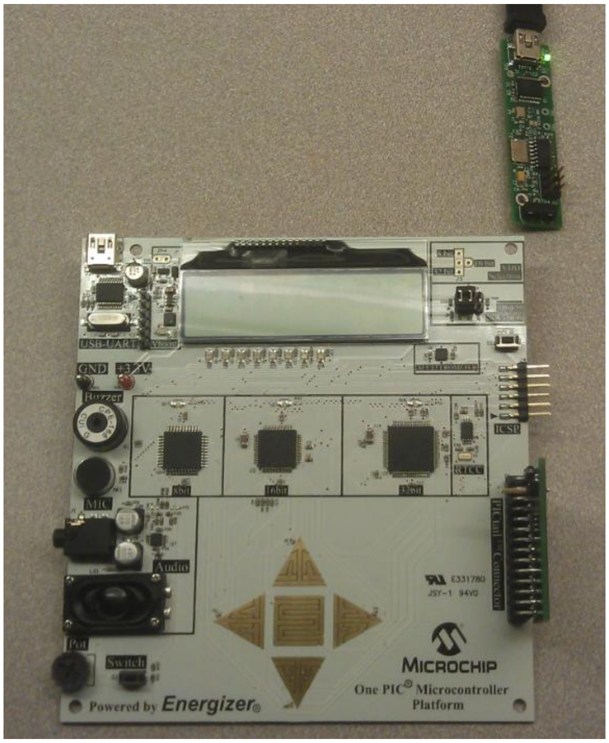
(See also IDE level file filter in Options->Miscellaneous->Files)

Source files from specified folders will be added to the project.

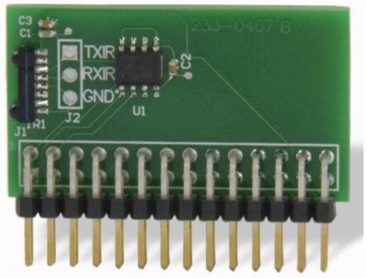
```
df_trunk/OnePIC.X'
.production.hex
make[2]: Entering directory `C:/Users/C14648/Documents/Projects/edf_trunk/OnePIC.X'
"C:\Program Files\HI-TECH Software\PICC\9.81\bin\picc.exe" --pass1 ../base/OnePIC_i2c
"C:\Program Files\HI-TECH Software\PICC\9.81\bin\picc.exe" --pass1 ../base/OnePIC_lcd
"C:\Program Files\HI-TECH Software\PICC\9.81\bin\picc.exe" --pass1 ../base/OnePIC_led
"C:\Program Files\HI-TECH Software\PICC\9.81\bin\picc.exe" --pass1 ../base/OnePIC_mto
"C:\Program Files\HI-TECH Software\PICC\9.81\bin\picc.exe" --pass1 ../base/OnePIC_rtc
../base/OnePIC_rtcc.c:57 warning: string not terminated by null character
```



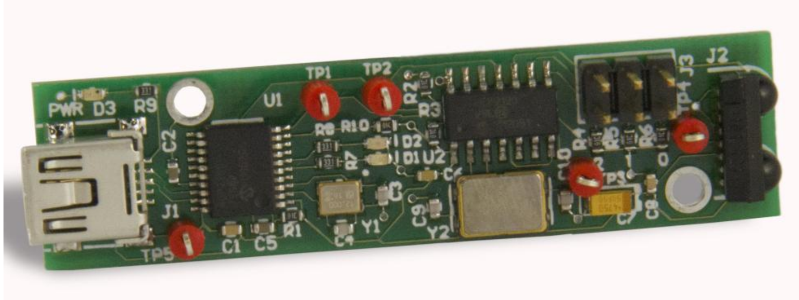
```
 * @subpage
 * @defgroup
 */
#include <st
#include <st
#include <st
```



Physical Setup for the 8bit IR demo.
The IR Link is directional, the IR Dongle and PICtail must face each other



IR PICtail



IR Dongle

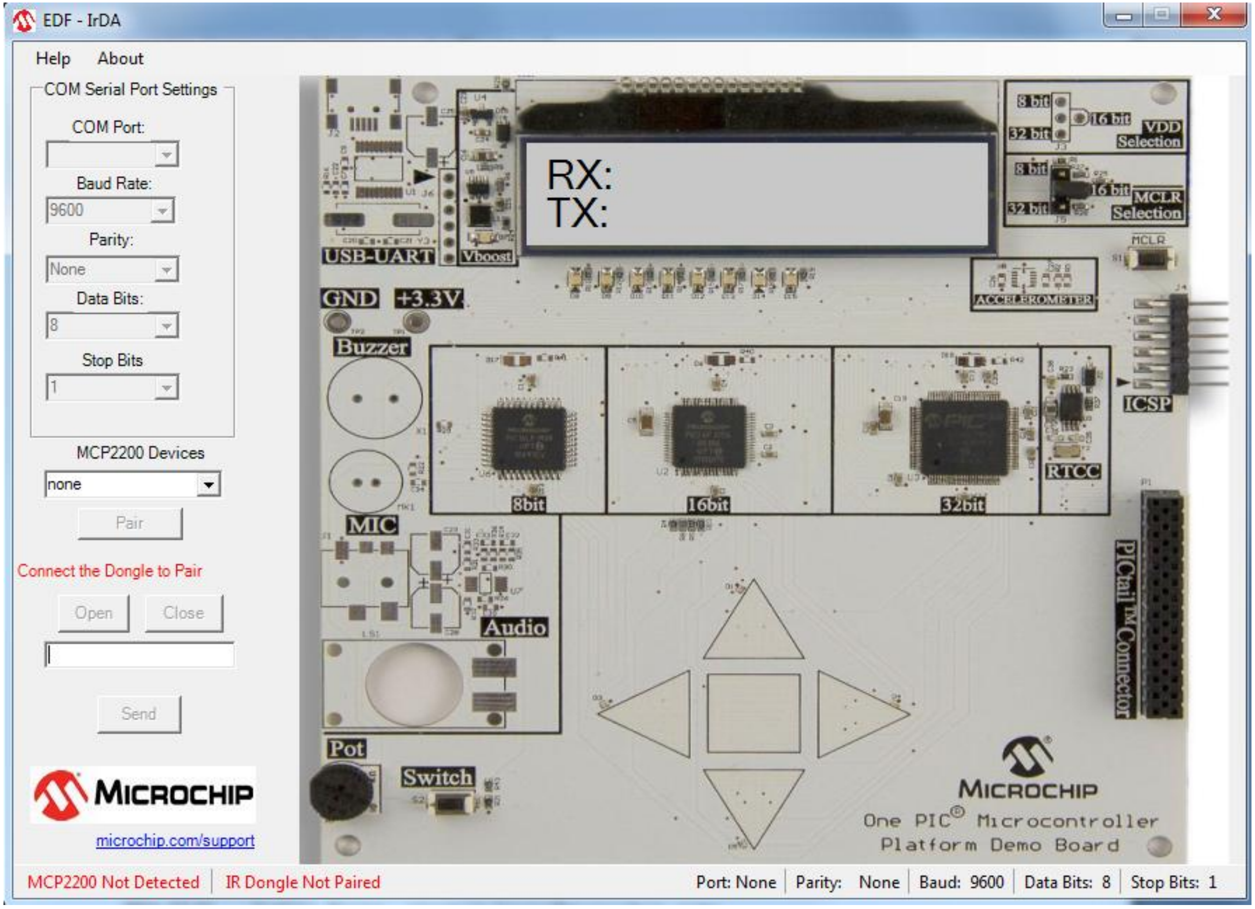
## 8bit IR GUI Setup:

For more in-depth documentation, please see the associated 8bit IR document.

Please install the MCP2200 driver and then double-click the *8bitIR.exe.* The Simple-IO.dll should also be in the same directory as the executable.

The EDF board should also have its IR PICtail inserted into the board with the transceiver facing the LCD & programming header as seen in the above picture.

Once the driver is installed and a board or dongle (or any MCP2200 device) is plugged in, Windows will create a virtual COM port. Below shows the splash screen of the GUI.



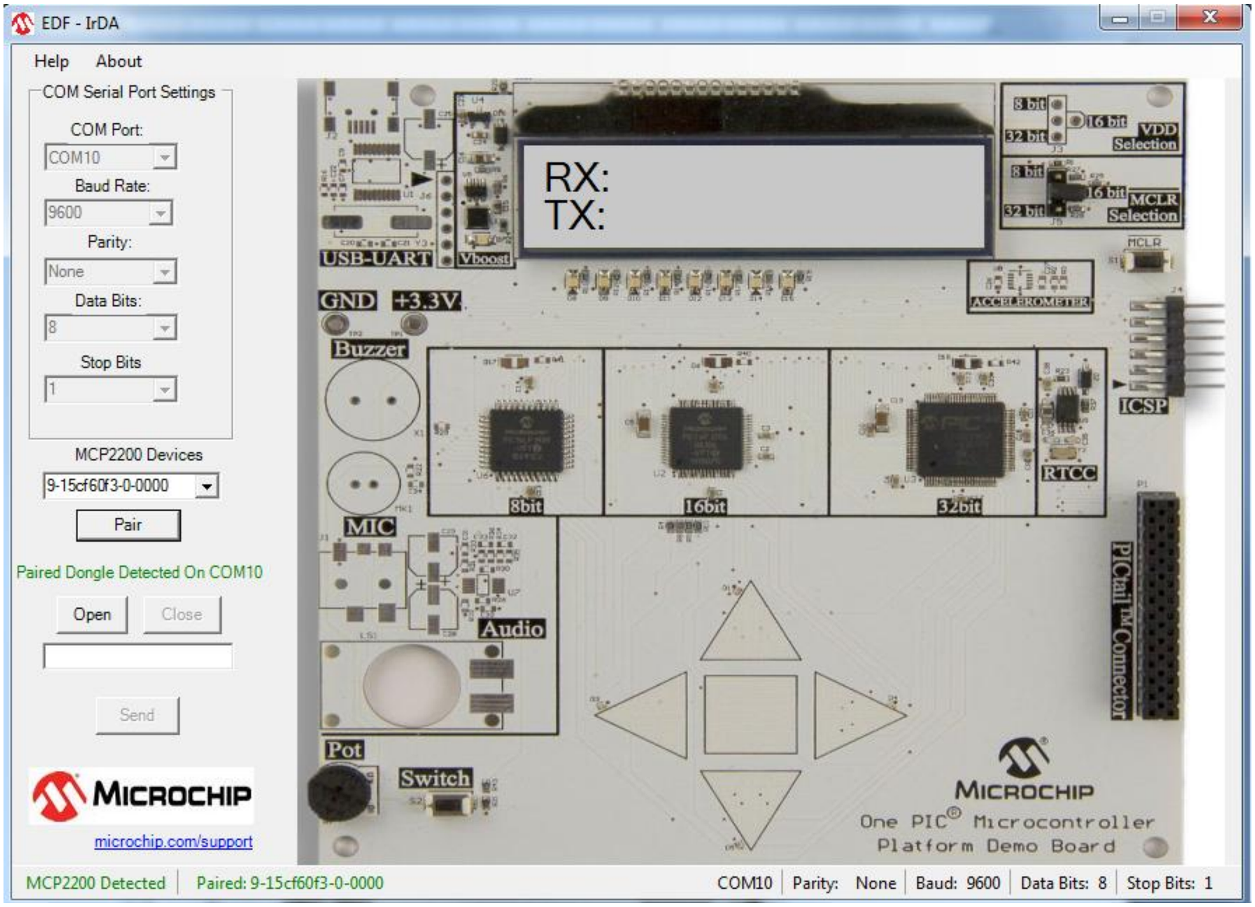
COM Port:
Baud Rate: 9600
Parity: None
Data Bits: 8
Stop Bits: 1
MCP2200 Devices: none

Connect the Dongle to Pair

MCP2200 Not Detected | IR Dongle Not Paired | Port: None | Parity: None | Baud: 9600 | Data Bits: 8 | Stop Bits: 1

Main appearance of the GUI

This is the screen that will be presented when the GUI is first started. The Dongle must be paired first before any communication is to occur. Only plug in one MCP2200 device (the dongle) to pair. Once paired successfully, you should see something similar below.



COM Port: COM10
Baud Rate: 9600
Parity: None
Data Bits: 8
Stop Bits 1
MCP2200 Devices 9-15cf60f3-0-0000
Paired Dongle Detected On COM10

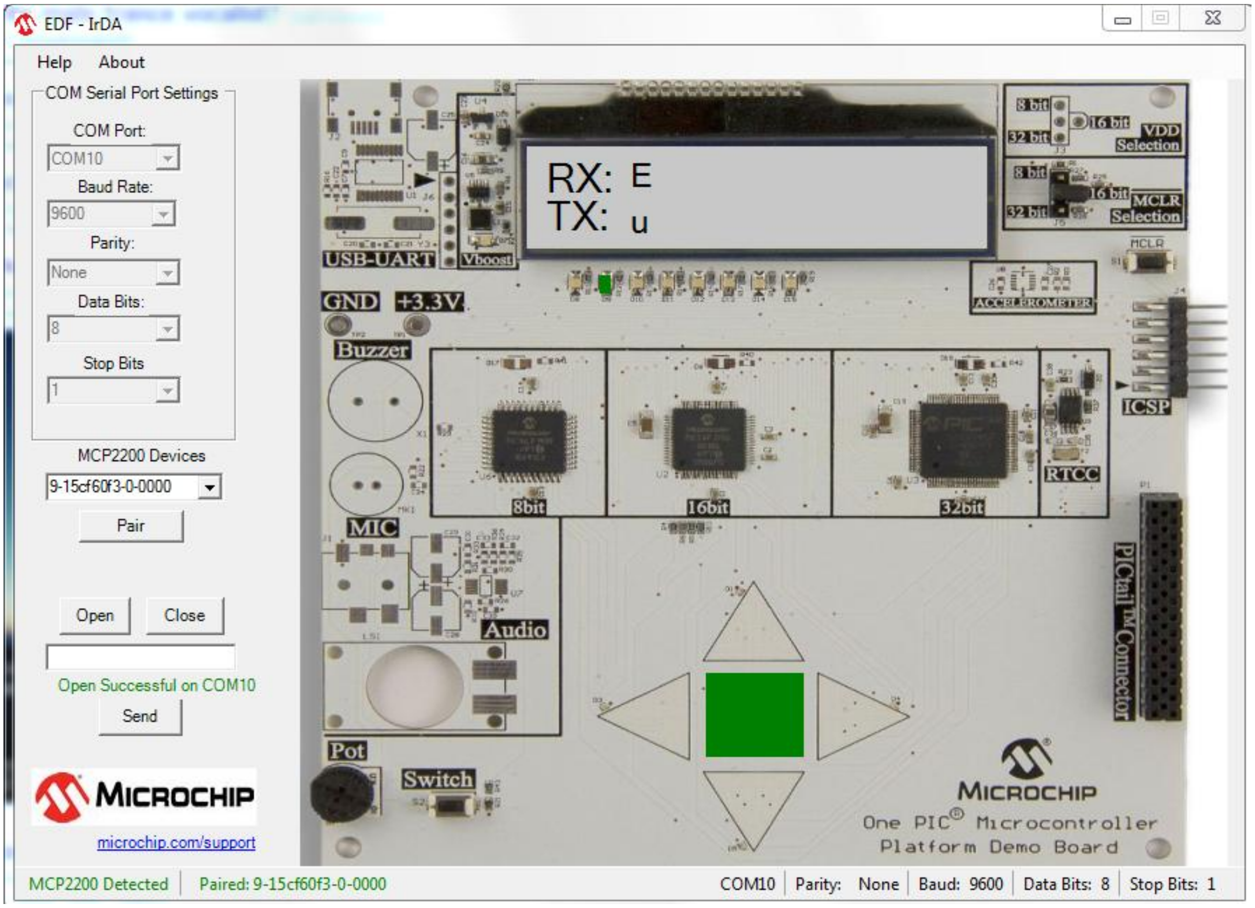
MCP2200 Detected | Paired: 9-15cf60f3-0-0000 | COM10 | Parity: None | Baud: 9600 | Data Bits: 8 | Stop Bits: 1

Display after the dongle is connected and paired.

The MCP2200 serial number as well as the COM port in which it is connected to will now be saved and automatically loaded on each time the GUI is run. These settings are saved in on the PC in the user's local AppData folder.

Press 'Open' to open the port and start communicating. When this is pressed, the Serial Port will be assigned the settings that are currently selected in their respective drop box. This must be performed! A "SUCCESS" will appear if windows could open the port. *This does not test if the board is successfully communicating with the GUI.*

Only the default settings work as of V1.0, which is why some boxes are grayed out. If any of these settings are changed from their default, a: "Warning. Unsupported Settings Used" will be issued.
The EDF board should now be programmed with the 8bit_IR project. It can be powered by either a battery or USB. The GUI should now display what button was pressed as well as TX/RX LEDs. The RX/TX status labels are with respect to the Dongle as seen below.



COM Port: COM10
Baud Rate: 9600
Parity: None
Data Bits: 8
Stop Bits: 1
MCP2200 Devices: 9-15cf60f3-0-0000
Open Successful on COM10

RX: E
TX: u

MCP2200 Detected | Paired: 9-15cf60f3-0-0000 | COM10 | Parity: None | Baud: 9600 | Data Bits: 8 | Stop Bits: 1

This is what is displayed if the middle button is pressed on the attached board.

To send a single character, type the character in the textbox and then either press enter or click ‘send’. The character will now be erased and displayed on the LCD next to ‘TX: ‘. Only the first character can be sent at a time. If a string is entered into the box, it will be discarded with the exception of the first byte. Board-to-Board communication is also possible without the use of the IR dongle. Simply connect the EDF board to the PC and pair it. You can now communicate to the EDF board directly.

*Note:*

Please do not disconnect any MCP2200 devices while the GUI is running. The virtual serial driver and protocol are antiquated and will break if this happens. To be safe, if any error messages occur such as “COMX does not exist”, simply unplug the device and restart the GUI.

# 16bit – PIC24FJ256GA106

## 2a) Base - MPLABX Feature to Highlight: File Exclusion

Hands-on #3: Migration from 8bit to 16bit (slide 125)

Steps to Transition from 8bit IR demo to 16bit Base Demo:

1. Remove the 8bit and 8bit_IR Folders:
    a. Right Click 8bit folder, select “Remove”. (Note: Highlighting the folder and pressing delete will permanently delete the folder from your harddrive)
    b. Right Click 8bit_IR foler, select “Remove”.
2. Add “16bit” source folder by using same procedure as in section 1A
3. Select configuration “OnePIC: 16bit”

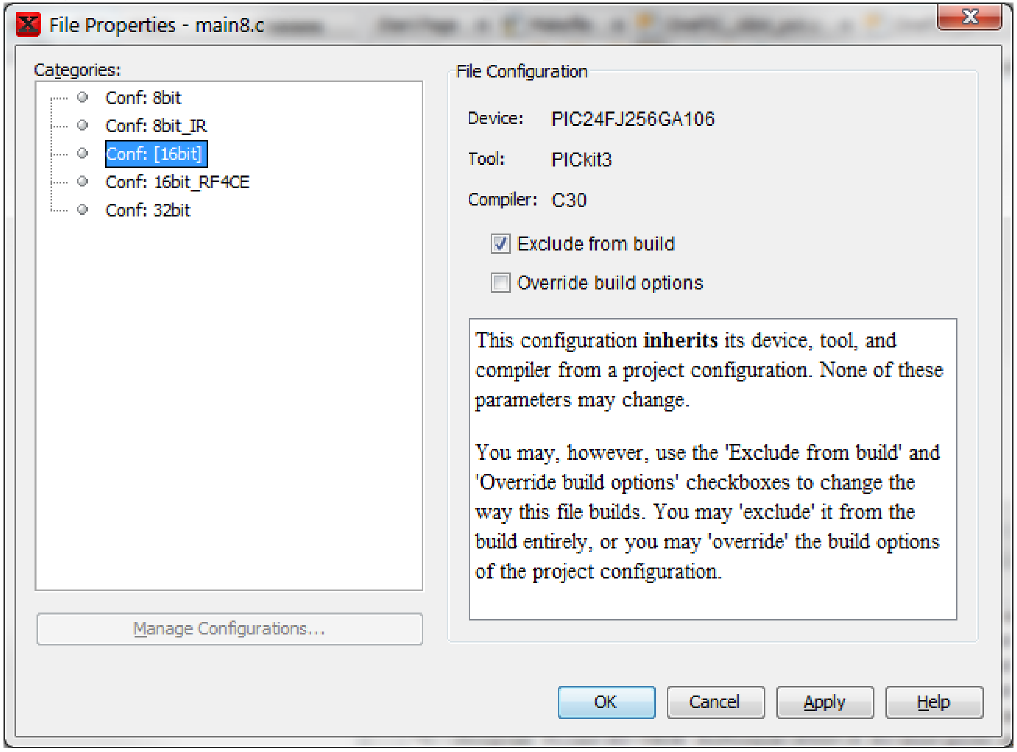
### MPLABX Feature: File Exclusion

File exclusion allows you to select which files are compiled based upon which configuration is selected. In this project, there are three main files – one for each part.

However, using this file exclusion feature of MPLAB X, only one file is active at any time.

To choose which file is excluded from which configuration, select the file and Right-Click -> Properties.

Example: this is the Property box when main8.c was selected. Main8.c is excluded from every configuration except for “8bit” and “8bit_IR”.



Device: PIC24FJ256GA106

Tool: PICkit3

Compiler: C30

☑ Exclude from build

☐ Override build options

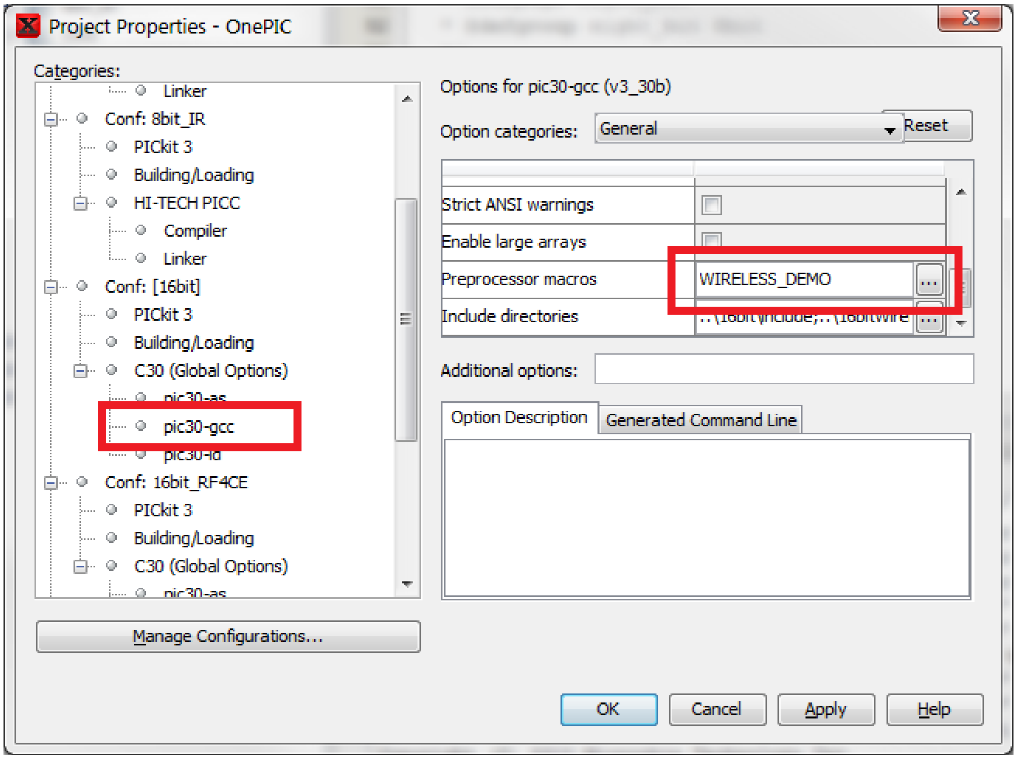
This configuration **inherits** its device, tool, and compiler from a project configuration. None of these parameters may change.

You may, however, use the 'Exclude from build' and 'Override build options' checkboxes to change the way this file builds. You may 'exclude' it from the build entirely, or you may 'override' the build options of the project configuration.

## 2b) RF4CE – MPLABX Feature to Highlight - Configuration based macro definition

Steps to run RF4CE Demo:

1. Add 16bit_RF4CE source file folder using same procedure as in section 1A
2. Select configuration "OnePIC: 16bit_RF4CE"



Options for pic30-gcc (v3_30b)

Option categories: General

Strict ANSI warnings ☐

Enable large arrays ☐

Preprocessor macros: WIRELESS_DEMO

Include directories: ..\16bit\include;..\16bitWire

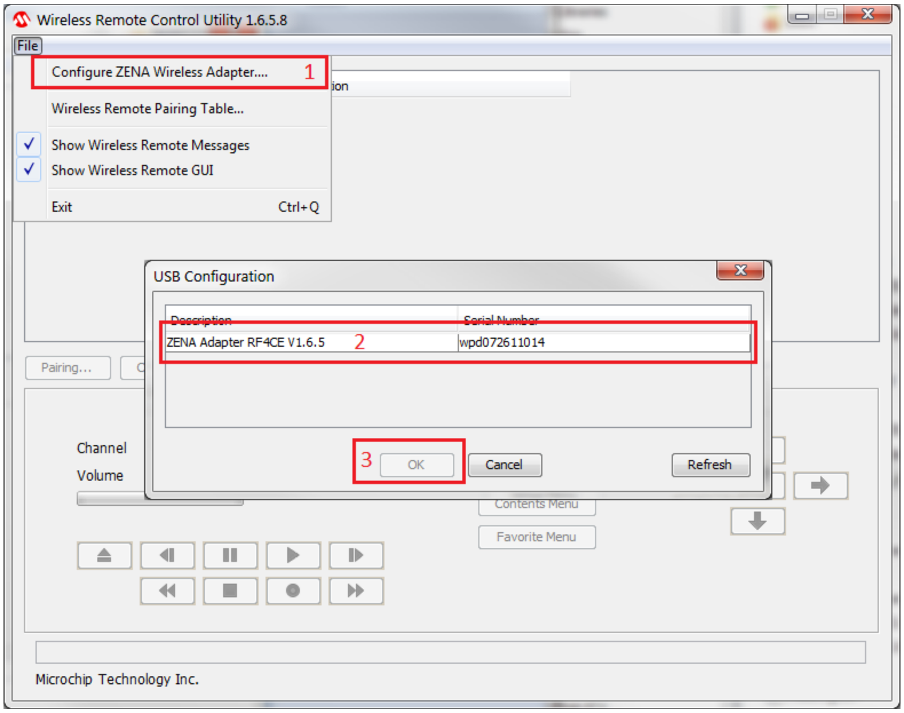
Additional options:

## RF4CE GUI- wireless_rc_utility_v1.6.5.8.jar

**The Driver must be installed first!** Once the GUI is started,

Setup:

1. File-> Configure Zena Wireless Adapter. "USB Configuration" window will appear.
2. Select ZENA Adapter RF4CE
3. Select OK
4. Ensure the OnePIC board is programmed with the RF4CE demo. The LCD line 1 should read "Pairing…." If the demo is running.
5. Click "Pairing….". Within 10-15 seconds the pairing should complete.
6. Pressed cap-touch buttons on the OnePIC board, see the arrow buttons light in the GUI.


Wireless Remote Control Utility 1.6.5.8
File
Configure ZENA Wireless Adapter....
1
Wireless Remote Pairing Table...
✓ Show Wireless Remote Messages
✓ Show Wireless Remote GUI
Exit
Ctrl+Q
USB Configuration
Description
Serial Number
ZENA Adapter RF4CE V1.6.5
wpd072611014
2
3
OK
Cancel
Refresh
Pairing...
Channel
Volume
Contents Menu
Favorite Menu
Microchip Technology Inc.

## MPLABX Feature: Configuration based Macro Definitions

MPLABX allows you to define configuration specific macros, which can be used to conditionally compile source code.

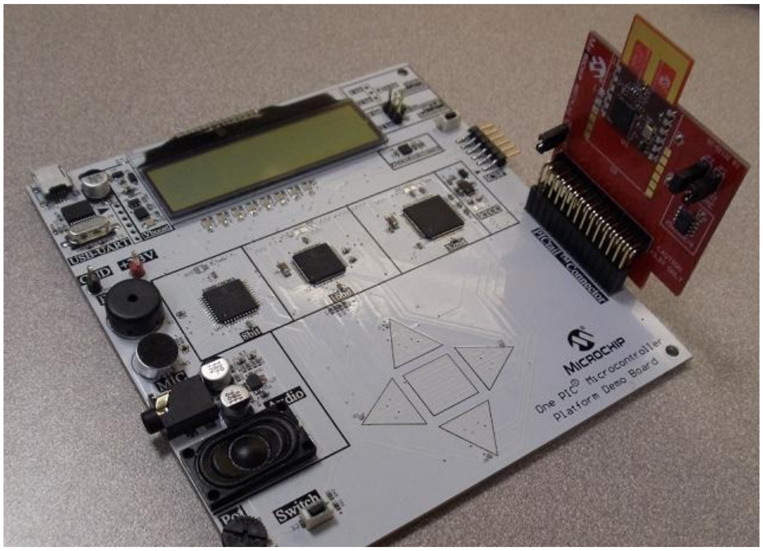
This project defines “WIRELESS_DEMO” when the respective wireless configurations are selected.

Example of use in main.c:

```
#ifdef WIRELESS_DEMO
    demo_main_loop(); //enter demo if building the demo
#endif
```

Using this feature, the function “demo_main_loop()” is only called when the current configuration defines the macro named “WIRELESS_DEMO”.

To define such a macro, open Project Properties, select the compiler of the configuration in which you’d like to define a macro, and enter the macro name.



16bit RF4CE setup

ZENA Wireless Adapter 2.4 GHz MRF24J40

## 32bit – PIC32MX795F512L

### 3a) Base

Hands-on #4 (slide 170)

Steps:

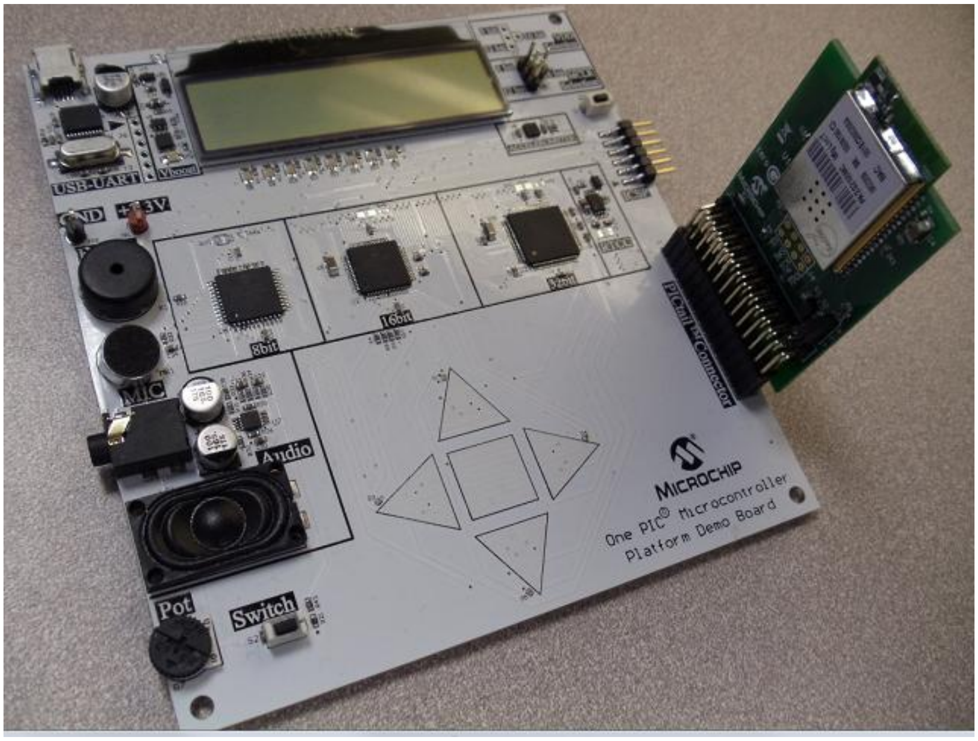
1. Remove folders “16bit” and “16bit_RF4CE”
2. Add folder “32bit”

### 3b) Wi-Fi – MPLABX Feature to Highlight: multiple projects

Steps:

1. Open project 32bit_WiFi/C32-WiFi Easy Config EDF.X
2. Right click on PROJECT, select “Set as Main”
3. Build/Program

4. If the red LED near D14 is slowly blinking red, then the board is functioning as a WiFi device.
5. Upon power up the device, it broadcasts an adhoc network with SSID "EasyConfig".
6. A client device (laptop, iPod Touch/iPhone/iPad) can then connect to the EasyConfig network.
7. Before trying to connect have all browsers closed, if you are having trouble connecting try repairing your wireless connection first.
8. The user can then use a standard web browser to go to the IP address of the demo (**http://169.254.1.1**).
9. The user will then be presented with some web pages from the web server. The index.htm web page has some additional information on
10. EasyConfig, and also shows the continually updating status of the LEDs on the development board. The LEDs on page can be changed by clicking them with the mouse. This in turn will get updated on the EDF board. The potentiometer also works. The buttons are not functional in this demo. The *configure.htm* page will allow the user to scan for networks, and connect to a network of their choosing



32-bit wireless demo