PIC18F2525 Instructions

The document below gives instructions on how to program the PIC18F2525 Microprocessor. The document starts with the process on how to compile the code provided by OpenPMU. The method of uploading code is discussed afterwards.

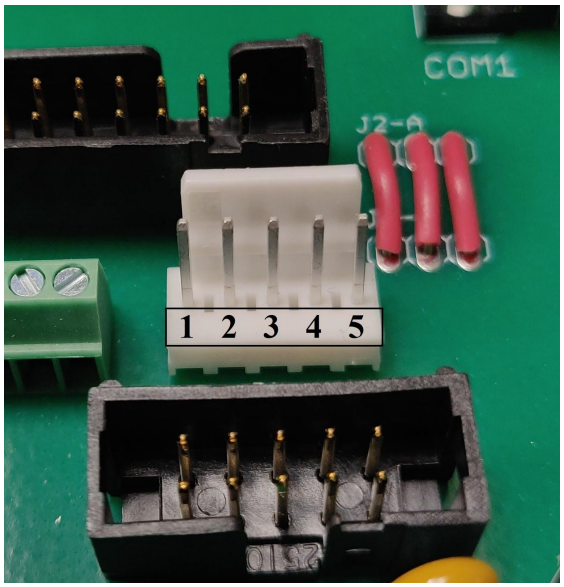
Compilation

1. Begin by downloading version 8.50 or later MPLAB IDE from http://www.microchip.com/development-tools/pic-and-dspic-downloads-archive You will have to create an account to do so, which is free. Make sure to download any version 8.50 or higher. Scroll to the MPLAB ARCHIVES section to find the version you want.
2. Download the compiler from https://www.microchip.com/Developmenttools/ProductDetails/SW006011 Choose the Standard Eval Version. When installing the compiler, when asking for the directory, install the compiler to C:\Program Files (x86)\Microchip\MCC18\ or C:\Program Files\Microchip\MCC18\ depending on what you downloaded.
3. Download the folder OpenPMU PIC - V5.1 if you haven't already or move the folder to an easily accessible location. (I saved it to my Desktop.)
4. Open MPLAB IDE. **File → Open Workspace… → testbenh.mcw** and open the workspace. Click OK to the missing file notification.
5. Make or Build the project once. A Tool-Location Discrepancy window will open. Click the **MPLAB IDE → Use This** option. The build should fail.
6. In the workspace, testbenh.mcw window, find the Linker Script folder. Right click on the file 18f2525.lkr and select Locate Missing File → Use a file of my own choosing → Follow the path:
   C: → Program Files (x86) → Microchip → MCC18 → bin → LKR → 18f2525_g.lkr
   and open the file 18f2525_g.lkr as your new file.
7. Go to **Project → Build Options… → Project → MPLINK Linker**. Check the box Use Alternate Settings and change the input from
   /m"$(BINDIR_)$(TARGETBASE).map" /w /o"$(BINDIR_)$(TARGETBASE).cof"
   to
   /m"$(BINDIR_)$(TARGETBASE).map" /w **/p18f2525**
   /o"$(BINDIR_)$(TARGETBASE).cof"
   Ensure there are spaces like:... /w /p18f2525 /0… not .../w/p18f2525/o...
8. Make or Build again. There will be a syntax error in the code dml_common.c Double click on the error line to open dml_common.c or go to File → Open → dml_common.c The error here is odd, and must be something specific to the compiler. The code must end with a blank line. To resolve the syntax error, simply add a blank line to the end of the code, after the comment and save it. Close dml_common.c.

9. Make or Build again, and when the Tool-Location Discrepancy windows open, click the MPLAB IDE Use This option again. The build should now succeed.
10. If the build still fails, go to Project → Build Options… → Project → Directories. In the drop down menu for Show directories for: under Include Search Path and Library Search Path, ensure that the program files is either C:\Program Files\... or C:\Program Files (x86)\... is consistent with your other settings.

Uploading Code

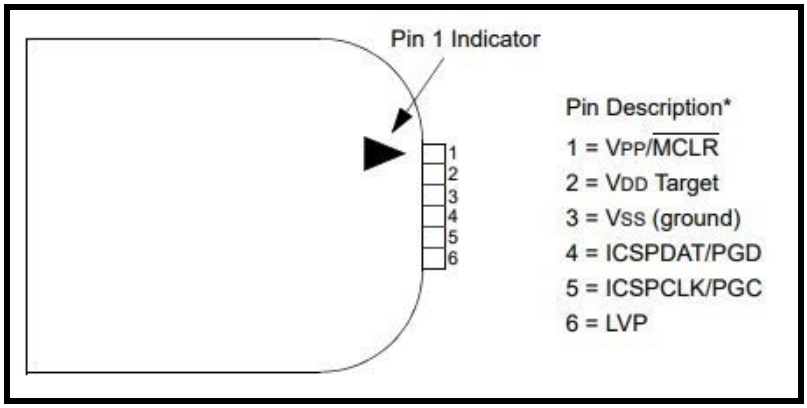
To start the process of programming the PIC you will need the completed, populated PCB Board. The board itself will be used to program the PIC, so you don't need to buy any sort of PIC programming base or kit. To program the PIC, begin by plugging the chip into the 28 Pin IC Socket. The 5 pin PCB header, shown below, will be connected to the PICkit 3.



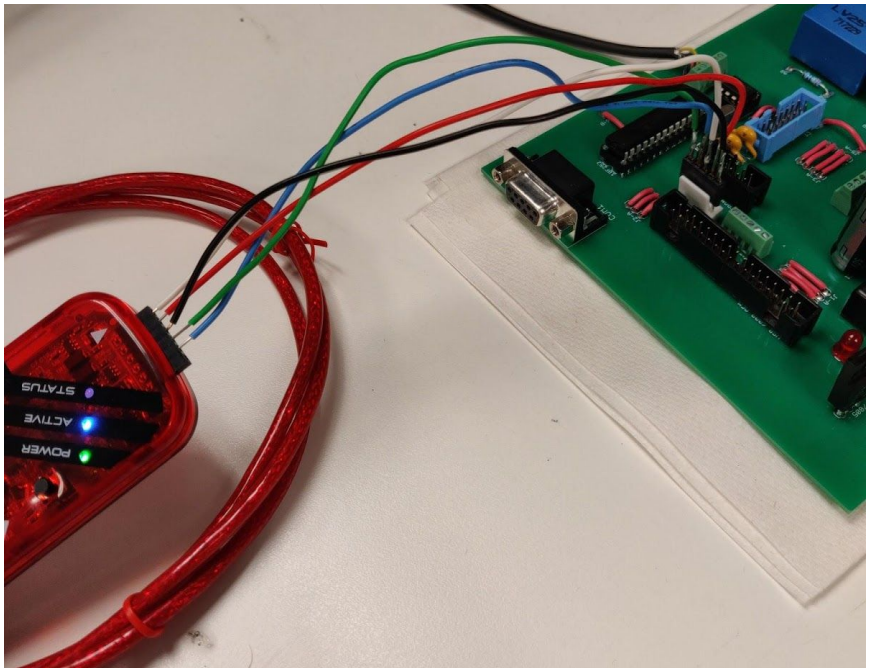
The 5 pin PCB header has the following pins:

1	2	3	4	5
Ground	Vcc (VDD Target)	MCLR	PGC	PGD

The PICkit 3 has the following pinout:



Because the pinouts are not in the same configuration/order, you will need to use the PCB programming cable to connect the correct input and outputs. You will need to cut the ribbon cable to connect each wire to the correct input. The set up should look like the following.



The image above shows a recreation of my connection to the board. Note that the GPS is connected to the board in the picture, however it should NOT be connected when programming the PIC. Also note, that the wired connection is not the same as the one in the parts list.

Connect the PICkit 3 to your computer.

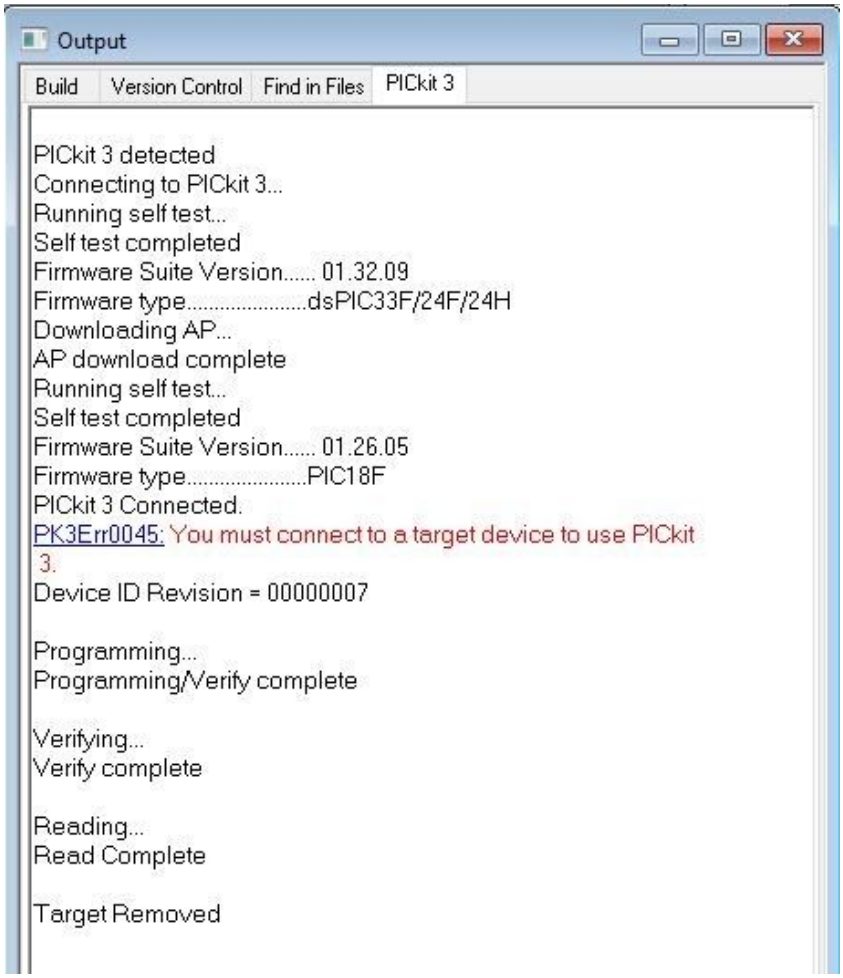
Before you start any programming, the USB-6009 OEM MUST BE DISCONNECTED. The voltages used by PICkit 3 may be able to damage the USB-6009 OEM.

Disconnect the GPS from the board as well if you have connected it. Better to be safe than sorry.

Open MPLABS IDE v8.50. Go to File → Open Workspace.... And open the compiled testbenh.mcw that you compiled earlier. In the upper toolbar, go to Programmer → Select

Programmer → PICkit 3. In the Output window, an output should print “PICkit 3 detected.” More lines will be outputted. Once “PICkit 3 Connected” is outputted, if not using a universal seat, you should get an error. To solve the error go to Programmer → Settings → Power → Check the box “Power target circuit from PICkit 3” and make sure the Voltage is set at 5V. Click Apply, and OK when the warning window pops up. (If you get any sort of detection errors, make sure all of the pins of the 18F2525 entered their respective port. I had several problems with pins not being lined up correctly, and bending, breaking the connection.)

A new group of icons in the toolbar should become available to the right. Click Program. Once programming is complete, click Verify. The output window should be something similar to the image below. Once you have finished uploading the code, make sure to go back to Programmer → Settings → Power → and uncheck the power button.

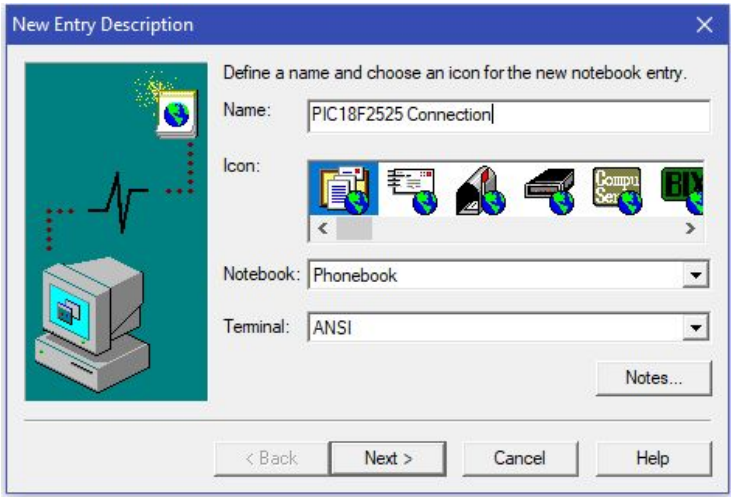


The PIC should now be ready to communicate data. To verify that the programming was completed correctly, if you haven’t already, disconnect all PIC communications including the PICkit 3. Plug in the RS-232 communications cable to COM1 of the board, and to your computer. You will now need access to the program HyperTerminal or HyperAccess. In this explanation I am using HyperAccess. There is a free demo for 30 days for HyperTerminal that

you can download here: https://www.hilgraeve.com/hyperterminal/ HyperTerminal should perform the same functionality for our purposes as HyperAccess.

Open Windows Device Manager. In the device manager, find out which com port the board is communicating with by looking through the **Ports (COM & LPT)** section. The connection is called “Prolific USB-to-Serial Comm Port” followed by the port used. In this example, I connect to COM6.

Open your Hyper program (Terminal or Access.) Go to File → New… and create a “New Entry Description”. Enter a name for the program, don’t change the “Notebook:” from Phonebook or “Terminal:” from ANSI as their inputs.

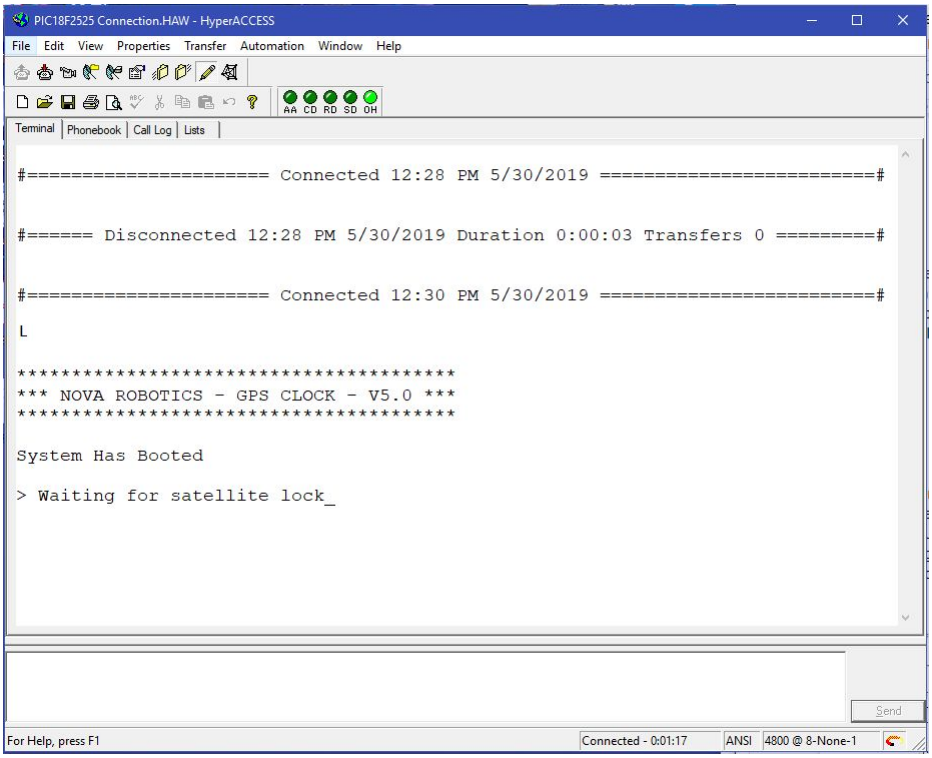


Click Next. Now, click the “Change…” box to connect to the COMM port found via the device manager. If not already selected, click on Direct Cabled COMX → Modify Connection… In the new window click “Change…” again, and in the final window “Connect Through” select the correct port. Once selected, click OK until you are back to the “New Entry Connection” window. Click Next. Change the connection settings to:

Bits per second: 4800
Data bits: 8
Partiy: None
Stop Bits: 1
Flow Control: None

Once set, Click OK → Next → Finish. In the main menu of HyperAccess, your new connection settings will be created. Double-click your settings to open it. HyperAccess will now automatically connect, so disconnect the program. The Telephone buttons in the upper left-hand corner can be used to connect and disconnect. If you haven’t already set up the power supplies, do so now via the document “Power Supplies Example”.

Once the power supplies is set up, but not plugged in, connect with HyperAccess. Plug in the power supplies, and you should get an output similar to the following:



If you get this output, congrats, the PIC18F2525 has been programmed correctly.

Note: In order to get the output shown above, you must first Connect with HyperAccess, and then turn on the power supplies. Turning the power to the board off and on does not seem to output a message. If the board is powered first, and then you Connect with HyperAccess, no output message will be shown.

Whenever you would like to reconnect to the GPS, you can change the port selection from HyperAccess via **Properties → Communication.** You may have to do this whenever you connect to the GPS, as Comm ports can change even when using the same physical port on your computer.