FluxPMU Makers Guide For OpenPMU V1

Written by Emmett Williamson
With Luigi Vanfretti and David Laverty

Updated Summer 2020

Table of Contents

General Information

Welcome! The purpose of this document is to provide more information to those looking to build FluxPMU, the open source and educationally purposed version of OpenPMU V1[1]. FluxPMU was initially created and developed by David Laverty at Queen's University, Belfast. More information on his work, and the research and development version, OpenPMU V2[2] can be found at https://sites.google.com/site/openpmu/. Discussed in these documents is information on what to do where, and How-To guides for specific processes, and to ameliorate your time while building FluxPMU. The documents in this repository are split into three groups, the original documents, the new documents, and the VI[3] documents.

The original OpenPMU documents written and developed by Dr. Laverty are as follows:
The PDFs

- OpenPMU - Enclosure Assembly - V1.1
- OpenPMU - PCB Assembly Instructions (USB-6009-OEM) - V1
- OpenPMU - PCB Component Placement (USB-6009-OEM) V1.1
- OpenPMU - PCB Connections (USB-6009-OEM) V1

And the power point:

- Component Placement

The second group of documents are those that were written by Emmett Williamson at Rensselaer Polytechnic Institute in Troy, New York. Each document begins with a description of what the document is about and a list of other documents that are relevant. Make sure to look at each pertaining document, as they will provide more specific information. Each of these documents relate either to a problem that was encountered or an area more information would have been useful. The documents are split into three categories:

1. Hardware: This group of documents pertains to most of the processes including hardware. The folder contains all documents for ordering the parts, how and where to assemble said parts, and several other key assembly procedures that must take place.
2. Simulations and Testing: This group of documents is useful for configuring the NI USB OEM 6009, and using signal generators to learn how the OEM 6009 works. Integration with the common collegiate instrument, Analog Discovery is documented.

---

[1] D. M. Laverty, R. J. Best, P. Brogan, I. Al Khatib, L. Vanfretti and D. J. Morrow, "The OpenPMU Platform for Open-Source Phasor Measurements,"
[2] X. Zhao, D. M. Laverty, A. McKernan, D. J. Morrow, K. McLaughlin and S. Sezer, "GPS-Disciplined Analog-to-Digital Converter for Phasor Measurement Applications,"
[3] LabVIEW programs-subroutines are termed virtual instruments (VIs).

3. Specific VI Documents: This folder contains information about each Virtual Instrument (VI) used in the LabVIEW software for FluxPMU. The documents act to further your understanding of each VI, and help solve problems. In each document, the Front Panel and Block Diagram are presented. The inputs and outputs, as well as operations of what the VI performs are all discussed.

_MACOSX: None of the Macintosh files have been modified or updated. If you are working on a Mac, consider downloading a virtual box, or some sort of virtual operating system to run the Windows VIs. You may run into other problems that are not in any of these documents.

Below shows the VIs that were written by Dr. Laverty and Emmett Williamson.

Dr. Laverty:

- 00 Main (single phase)
- 00 Main (three phase)
- 00 PMU Tester
- 10 DAQ Config (PFI0)
- 11 DAQ Config (continuous)
- 20 Limit 180 Degrees
- 21 Symmetrical Components
- 22 Frequency Average
- 23 Input Scaling
- 24 Global Front Panel
- 30 GPS Time Fetch
- 31 GPS Interrupt RS232 Read
- 32 GPS Time Code Parser
- 33 GPS Time to Labview Time
- 34 GPS Time from Config
- 40 Config File Path
- 41 Config File Write GPS Time
- 42 Config File Read GPS Time
- 43 Config File Get Input Scales
- 44 Config File Read Telecoms
- 50 NMEA Calc Checksum

VIs created by Emmett Williamson:

- 00 Main Simulated Signal
- 00 Main Simulated Triple Signal
- 01 Waveform Simulation
- 02 Triple Waveform Simulation
- 12 Simulated DAQmx Signal
- 13 Simulated Signals
- 14 Function Generator Signal
- 15 Triple Phase Signal Generation

Getting Started

This guide gives what I, as a student, felt are the appropriate ways to begin the entire process of building your FluxPMU.

1. Decide on a Version: Before starting anything, you will need to decide on which option of FluxPMU you would like. You can build FluxPMU with either the Hall Effect Transformers (V37) or AVB Isolation Transformers (V38). The differences are simple. The Hall Effect Transformers will give more accurate and precise results at higher voltages, but are more expensive.

2. Order Your Parts: This was by far the most time consuming process as orders were misplaced on an administrative level at my school. Make sure to get all of the parts ordered, INCLUDING the circuit boards, and triple check that your board and parts version are the same (such as Hall effect transformers and Hall effect circuit boards).

   One important aside: There are some parts on the Parts list that are not necessary to build FluxPMU. Refer to the Ordering Parts Section of this document for more info.

3. Downloading and Installing Programs\Code: FluxPMU is based on National Instruments (NI) software, LabVIEW, which is a wiring/diagram based programming language. I was able to download LabVIEW through my school's software licenses. If you don't have access to LabVIEW through your school, you will need to purchase it from NI's website.

   To download the code that runs FluxPMU you will need to go to the github repository located at: https://github.com/ALSETLab/FluxPMU

   Here you will find all of the related documents needed, and all of the LabVIEW code. Make sure that once the code is downloaded, it is all located in the same folder, in an easy to access location. Also, make sure that the .dll files are included in the download and are in the same folder as all of the VIs.

4. LabVIEW: Once the parts are ordered, you will have some time to kill. I recommend that if you haven't already worked with LabVIEW, watch the online tutorials found at: http://www.ni.com/en-us.html → My Account → Online Training → Log In → Access Training. Once there I would recommend Cores 1, 2, and 3 along with Data Acquisition

Using NI-DAQmx and LabVIEW. If you don't have access to the training, and you have some computer programming experience, I would recommend trying to recreate past computer science projects/homeworks/labs in LabVIEW. Another option is to just YouTube "labview tutorials" and use those as a resource.

a. When working with any of the 00 Main VI's, you must create two files, PMUconfig.ini and GPS_time.ini. To do so, open a simple text editor like Notepad. Save the text as either PMUconfig.ini or GPS_time.ini in an easy to access location. Do this for both files. Once the files are created, you will have to set the Path Constants in the VI's 42 Config File Read GPS Time and 40 Config File Path. Make sure to copy the following lines into PMUconfig.ini
    i. [Telecoms]
       ID=PMU123
       IP=192.168.1.100
       Port=40001
b. Once labview is installed, there are several additional drivers you must download and install:
    i. DAQmx Driver: http://www.ni.com/nisearch/app/main/p/bot/no/ap/tech/lang/en/pg/1/sn/catnav:du,n8:3478.41.181.5495,ssnav:ndr/
    ii. VISA Driver: https://www.ni.com/en-us/support/downloads/drivers/download.ni-visa.html#305862
    iii. Real Time Module Driver: http://www.ni.com/download/labview-real-time-module-2017/6636/en/
    iv. (Optional) NI CompactRIO: http://www.ni.com/download/compactrio-device-drivers-december-2017/7087/en/

5. Once Parts Arrive: Once the parts arrive, there are several things I would recommend you do, but it depends on which part arrives first. Below are the specific actions I would recommend once that part arrives.

a. **NI USB-6009 OEM:** Once the data acquisition board arrives, I would strongly recommend you look at simulating signals with function generators using the Simulated Signals VIs and the instructions for how to do so. Doing so better helped me understand where FluxPMU received the data and how it was processed.

b. **GPS:** Once the Garmin GPS arrives make sure to follow the instructions on how to set it up, and make sure it works.

c. **PIC Microprocessor:** The PIC18F2525 is its own beast. It will require several other programs to download. Refer to the documents on a full disclosure on what will need to be done to set the PIC microprocessor up.

6. Build: Once all of the parts and the board has arrived, you can now begin the build process. The instructions are in the document and PowerPoint called Assembly Instructions. The build is separated into groups of parts. I would encourage you to get a soldering aid tool to help support the circuit while you solder, shown below. Once you start building, make sure to properly set up your soldering station with a fan, as you don't want to breathe in the solder smoke. I would also recommend using a lamp or light source, and any sort of visual amplification glasses, to help get a closer up view.

Once the FluxPMU circuit is actually built follow this checklist to ensure that everything else is set up to ensure everything is ready for recording data.

Procedure	**Related Document**
Programming the PIC Microcontroller	PIC Microprocessor
Setting up the GPS	Garmin 18x GPS
Setting up Power Supplies	Power Supplies Example
Testing the Device	Makers Guide-Testing Your Setup

7. Testing: Refer to the section of this document, the Makers Guide, Testing Your Setup.

Ordering Parts

Start with the document US Official Parts List (2018). This is a comprehensive list of all parts you will need along with how many parts per PMU, and estimated prices. All costs in this document are estimated for two reasons: 1) the market price of each component will change with time and 2) the cost of the board itself is at the discretion of the provider. I estimate that each board will cost around $60 to $90 depending on who you choose to order with.

Color Identification:

At the top of the document, the blue accent cells gives the estimated cost for the Hall Effect version of FluxPMU and the green accent cells gives the estimated cost for the Isolation Transformers version of FluxPMU. Note that the orange box below "Number of PMUs you want to build/ order" can be modified by you to get a price for multiple PMUs. The actual list begin starts with the following list, where the number on the list represents the column number:

1. The name of the supplier. Search in a web browser to find their website.
2. Description of the component. Used in other documentation like the assembly instructions.
3. Number of parts required for 1 PMU build.
4. Cost for each part
5. Cost for 1 PMU
6. Part Number, search this on the suppliers website to find each specific part quickly.

Important Considerations

Note: The DIN Rail order is for a 2 meter long piece of rail, and does not change in the parts order list. You only need 20 cm of rail per PMU, and I assumed that you probably won't be ordering more than 10 PMUs. The DIN Rail and the PICkit 3 are marked with purple, to grab your attention.

Note: Similarly to the DIN Rail, it is assumed that you will only need one PICkit 3 programmer. The parts list only assumes that you will need one PICkit 3.

Note: Some of the components are highlighted in yellow-orange. These components are a part of the entire FluxPMU and include components for connecting to a Front Panel, and Current Board. Both of these are not necessary for the FluxPMU to operate and are thus options to order.

As suggested before, some parts in the parts list are not necessary to construct FluxPMU. The other highlighted parts are those that use the DIN rail system. The DIN rail is useful for constructing FluxPMU for a 19" Rack. Refer to the Original Document OpenPMU - Enclosure Assembly - V1.1 for more information on what the Enclosure assembly looks like. The Enclosure Assembly however, is not necessary for the operation of FluxPMU, and therefore all of the DIN rail, and components, are not necessary to buy. The decision to buy them is therefore up to you and your research group, and depends on how/where the FluxPMU will be used. In my instance, the DIN components were ordered, and the final product is shown below.

Circuit Boards

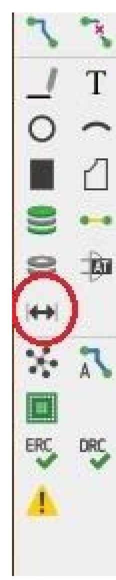
The first thing to know is that there are two boards, one for each version of the board. One is for the LV25-P Hall Effect Transformers and one is for the AVB 2,3/2/12 Isolation Transformers. Download the .brd files from the github repository. These files will need to be uploaded to a circuit board website, which each have their own procedure to upload files. We used the company, Sunstone Circuit, which worked fine. The company is located in Oregon, which is far if you are located on the East Coast. A recommendation is to use OSH Park, but using their company depends on how many boards you want, as when I tried to order, I could only get boards in groups of 3.

In general, you should be able to simply upload the .brd files to a circuit board provider's website to get a quote. If there is a case where the file isn't compatible to the circuit website, you will need to modify it. Below shows the issue that I had, and how it was fixed, however depending on the provider, you may have a different issue.

Modifying the Board Files

To create the files needed to order the protoboard you will need to download Autodesk Eagle, found at https://www.autodesk.com/products/eagle/free-download The file that contains the circuit board information is called: OpenPMU (LEM) Motherboard V38 (Transformers).brd or OpenPMU (LEM) Motherboard V37 (Hall Effect).brd Once Eagle is downloaded, open it. You may have to make an account to use it, but the program is free. To create the file to send to circuit board makes perform the following:

1. Go to **File → New → Board**
2. In the new board window **File → Open → OpenPMU.brd** file
3. The problem with the board is that the origin point that measures the dimensions of the circuit is off, which makes online circuit board websites think that the board is smaller than .25x.25 inches.
4. To solve the problem, in the Eagle Window, on the left there is a tool called "Dimension" shown to the right. Click on the icon.



5. In an upper toolbar, shown below, the Layer level is shown with a drop down menu.

Select the dropdown menu, and select **20 Dimension.**

6. Now on the circuit board image, you will need to select the length and width of the circuit board. The length should be around 6.65 inches or 168.91 mm and the width should be around 5.25 inches or 133.35 mm. Once the length and width markings are created, save the file.

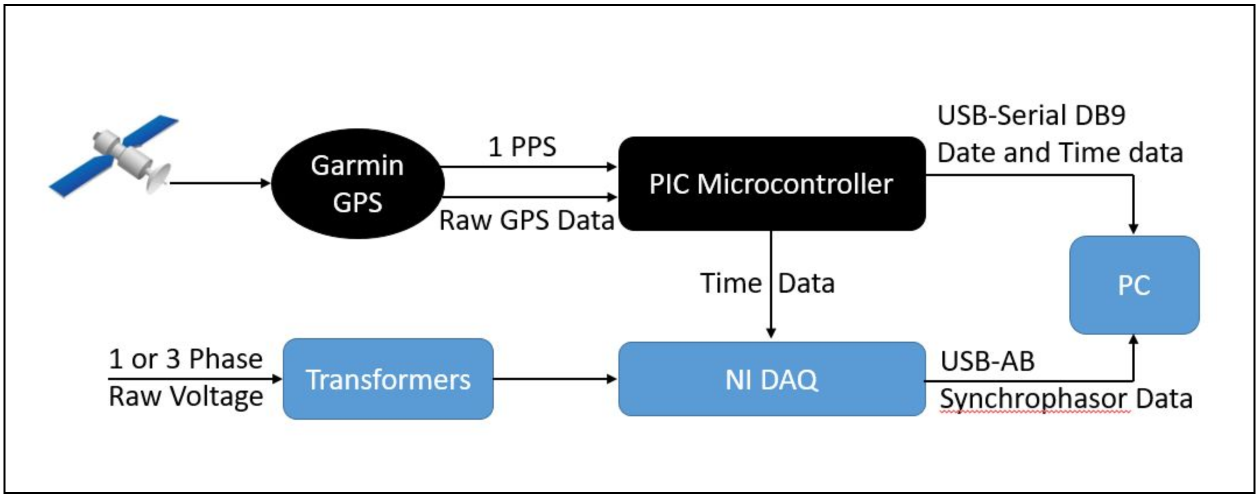
The files should now be ready to upload to circuit board providers. Some potential providers are:

- Sunstone Circuits: (https://www.sunstone.com/) The circuit board provider that we used.

- OshPark: (https://oshpark.com/) Recommended for me to use. Unfortunately, when I tried to order circuit boards, they only produce circuit boards in groups of 3. Probably the best offer, if you are ordering boards in groups of three.

Software Organization Overview

The following document gives insight into how the 00 Main LabVIEW VI's work together with the sub VIs. The descriptions below are split up into the main functionalities of the software: acquiring voltage signals, displaying output, timing/GPS acquisition, and the PIC code. Refer to any of the VI documents for further specific information



The image above shows a simplified network diagram of FluxPMU.

Acquiring Voltage Signals

FluxPMU software begins with the VI 10 DAQ Config(PFI0) which sends the voltage acquisition task to the DAQmx VI to acquire the voltage signals. Based on the task provided, all 3 voltage signals are found for both the single phase and three phase 00 Main VIs. The signals are consolidated to the Dynamic data type wire and combined with the time code and GPS Lock Status. Once combined, the signals are sent to a queue, so that no data is misplaced or analyzed out of order.

Displaying Output

To display the output waveforms and information, several VIs are used. The display process begins by unloading the queue mentioned above and using the LabVIEW functions Tone Measurement and Amplitude and Level Measurements to read the output variables. 20 Limit 180 Degrees is used to convert the phase angle to between -180 and 180 degrees. After some manipulation, all output variables are, for the most part, just sent to the Front Panel. 44 Config FIle Read Telecoms reads information from the other ini file, PMUconfig.ini to determine where information can be sent via a network connection.

Timing/GPS

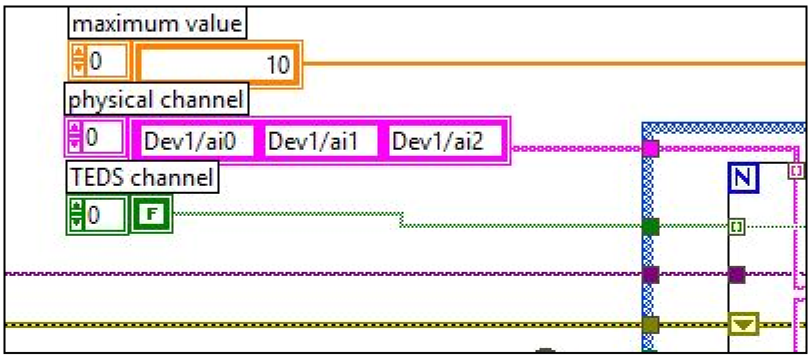
The timing software begins with 30 GPS Time Fetch, which uses the RS-232 GPS signal and the 1 PPS signal to write a timestamp used by FluxPMU. 30 GPS Time Fetch starts by recording the serial input time code sent from the board. 30 GPS Time Fetch then determines if the time code is valid, and writes the time code to the file GPS_time.ini. The VI 34 GPS Time from Config then reads the GPS_time.ini file for the time stamp information. The time stamp is then added to the dynamic data type and synchronizes with the acquired voltage measurements.

Important Configuration Settings

This portion of the document is for settings that you, as a builder of FluxPMU must be aware of.

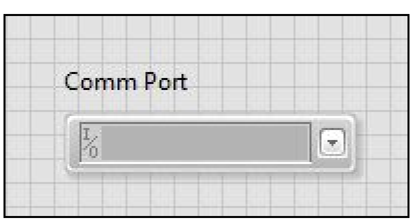
1. PMUconfig.ini and GPS_time.ini. These are two files that you need to create, as stated in the “Getting Started” section. Make sure they are in an easily found location on your computer.

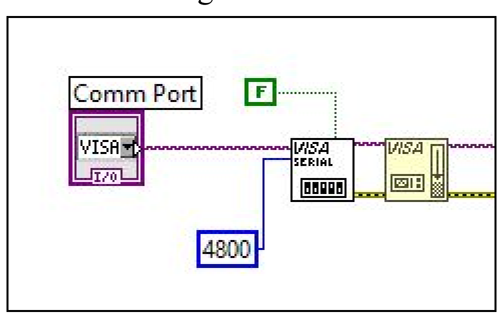
2. Device Address/Physical Channels: The physical channel settings look like:



This instance is located in 10 DAQ Config (PIF0). The setting states “DevX/aiY.” Dev stands for Device and X stands for that devices number. “Ai” stands for Analog Input and Y stands for the pin number input on the USB 6009. All of the values are set via NI MAX, when configuring your USB 6009. Refer to the NI MAX Document for more information. You will only have an issue with these settings if you use multiple NI-6009s. Each 6009 is set as a “DevX” so if you have 2 6009s, one will be “Dev1” and the other “Dev2” which can lead to an error.

3. Comm Port: The Comm port settings from the VI 31 GPS Interrupt RS232 Read, are shown below. Both the input from the front panel and the block diagram are shown.





The Comm port specified here reflects which port the GPS signal is being sent to. If you are unsure of which port that is, you can use Windows’ Device Manager to find the correct port. The Comm port should be named “Prolific USB-to-Serial Comm Port (COMX)” where X is the port number. Select this port number in from the front panel and save the VI when setting the port.

4. NI_timer.dll: The dynamic link library file NI_timer.dll is a part of the VI 31 GPS Interrupt RS232 Read. When this VI is first launched, you will get an error, as the path to the DLL file will be incorrect, and needs to be set manually. Double click or right click on the ___ to change the file path.

Testing Your Setup

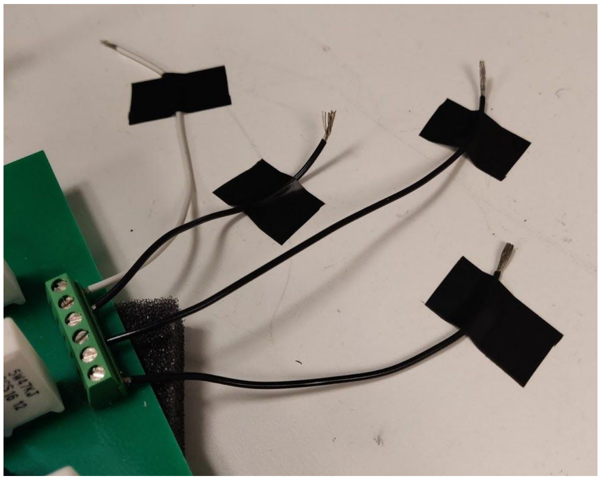
Once you have built FluxPMU, and completed the following checklist, you should be ready to power the device up and test the device with line voltage.

CAUTION: Don't be stupid here. You are (probably) an Electrical Engineering student, and we are now dealing with line voltages which are dangerous. Refer to your Labs safety regulations first. When testing the device, make sure you are not alone in your lab. Be extremely cautious.

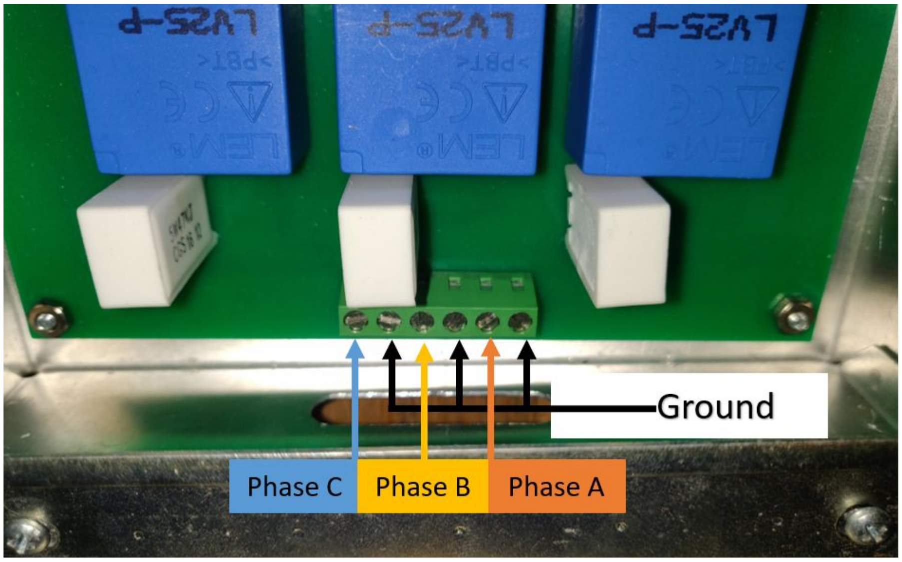
To actually observe the AC line voltage, you will need some sort of connection from the wall to the PMU input ports. I used a the cable connection shown below:

For Safety, make sure to plug into the wall outlet LAST.

It is good practice to tape down any wires that you are not using when working with high voltages with a simple electrical tape (or whatever sort of tape you have at hand. Don't damage the surface you are working on though!) If something were to get bumped or moved unintentionally, taping wires down reduces the risk of a short. I found that being generous with taping gave me the most secure feeling when finally plugging into the wall outlet. Below shows how I used the tape to secure each of the input wires.

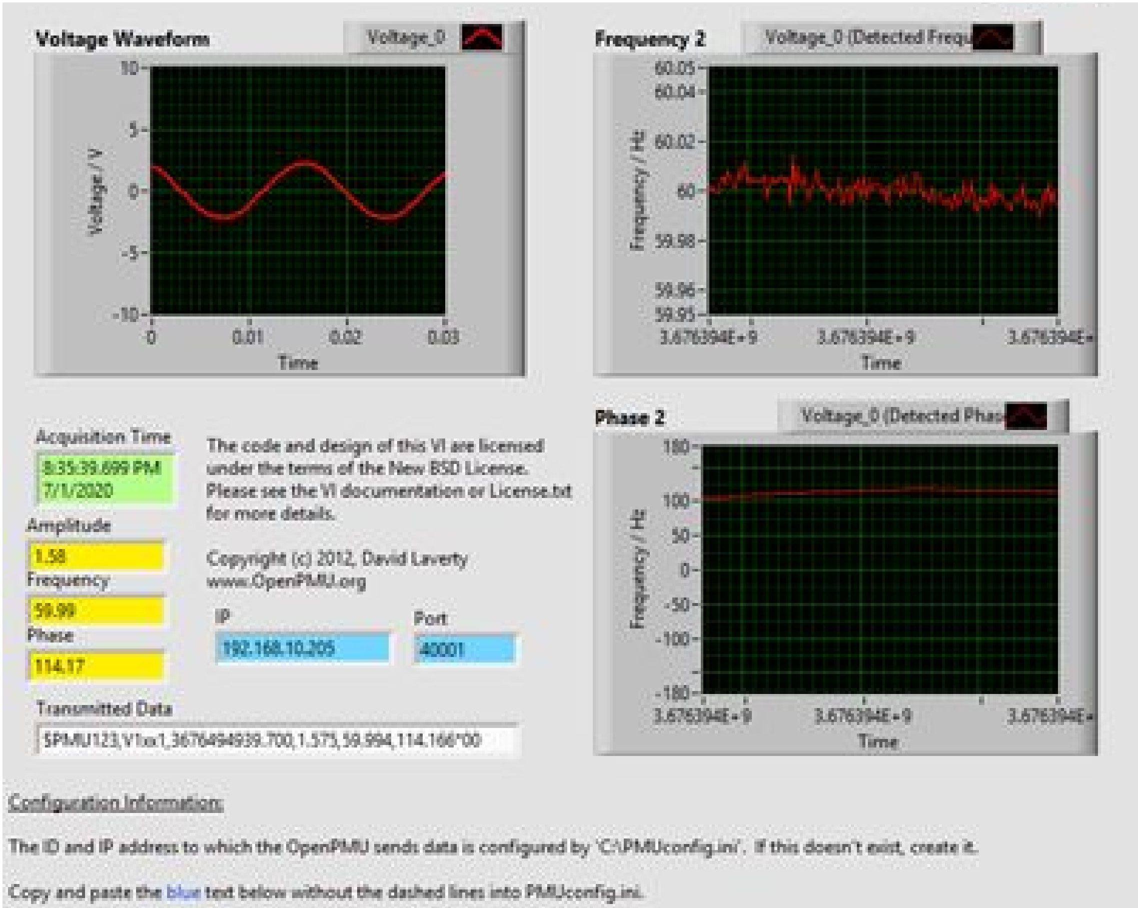


Note that the phase and ground connection are the following:



1. Connect Your USB 6009 via the USB A to USB B to your computer, and set up the device via NI MAX. If you have already tested the device, the device may already be saved.
2. Power up the FluxPMU circuit board via plugging in the power supply. You might still have to wait a little while (up to 10 minutes) if the GPS hasn't been configured before. It still takes time to locate the signal.
3. Connect your GPS to computer via a USB-to-RS232 cable.
4. Open the LabVIEW 00 Main program, either single or triple phase versions depending on your setup.
5. Make sure all of the settings as discussed in the section "Important Configuration Settings" have been dealt with.

6. Once everything is connected, run the 00 Main program. Once running, and no errors are returned, plug in the connection to the wall outlet. You should get output that looks like the following:



Note that you may have to change the axis labels for each of the graphs shown above, especially with regards to the frequency. To do so, simply double click on the number on the top or bottom of each axis while the VI is not running. Enter a new number to act as the new max/min scale for the graph.

Links

HyperTerminal - Used for testing.

NI-DAQmx Driver - Used to communicate with the USB-6009.

NI VISA Driver - NI VISA driver, used to read from the Garmin GPS.

LabVIEW Real Time - Real time drivers

MPLAB IDE - Used to program the PIC18F2525

MicroChip Compiler - Compiler for the MPLAB IDE.

Sunstone Circuits - Circuit Board provider.

OSH Park - Circuit Board provider.

NI Website - To navigate to the online training tools for LabVIEW.

NI LabVIEW Download - Where to purchase and download LabVIEW.

NI Errors - A list of common errors encountered in LabVIEW.

Garmin GPS - Technical Information about the Garmin GPS, such as what time codes mean.

Appendix

Figure 1: NI USB-6009 OEM Pinout Diagram

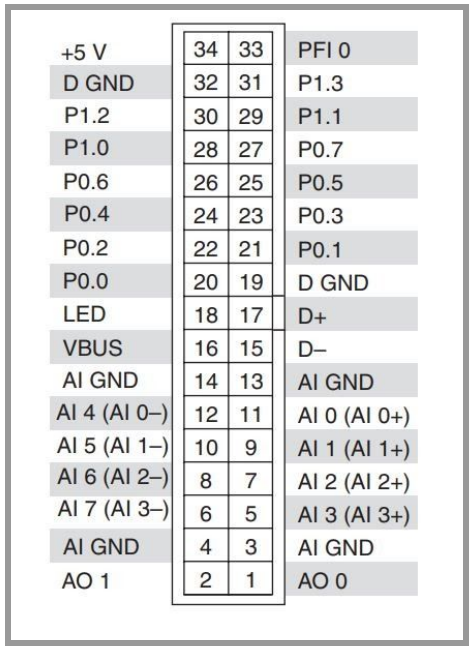


Figure 2: NI USB-6009 OEM Physical Pin Layout Diagram w/r to USB-6009 OEM

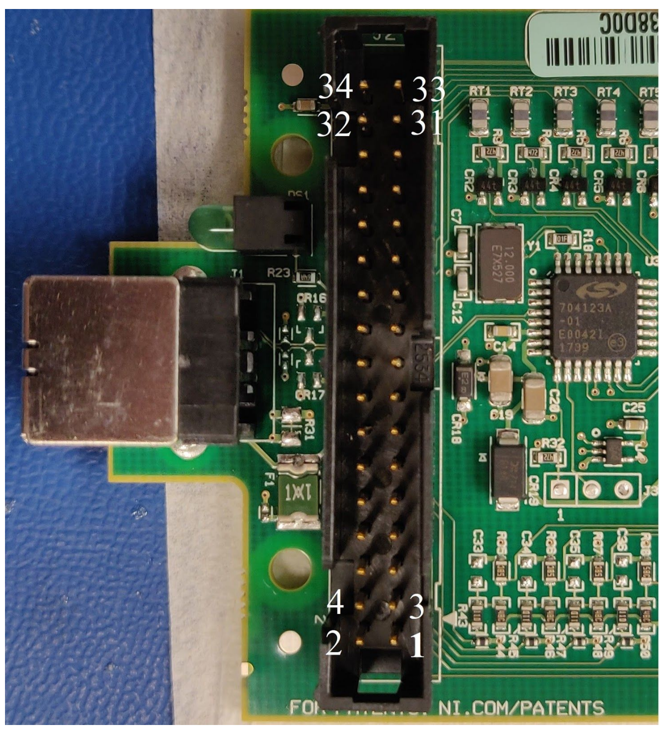


Figure 3: NI USB-6009 OEM Physical Pin Layout Diagram w/r to FluxPMU Board

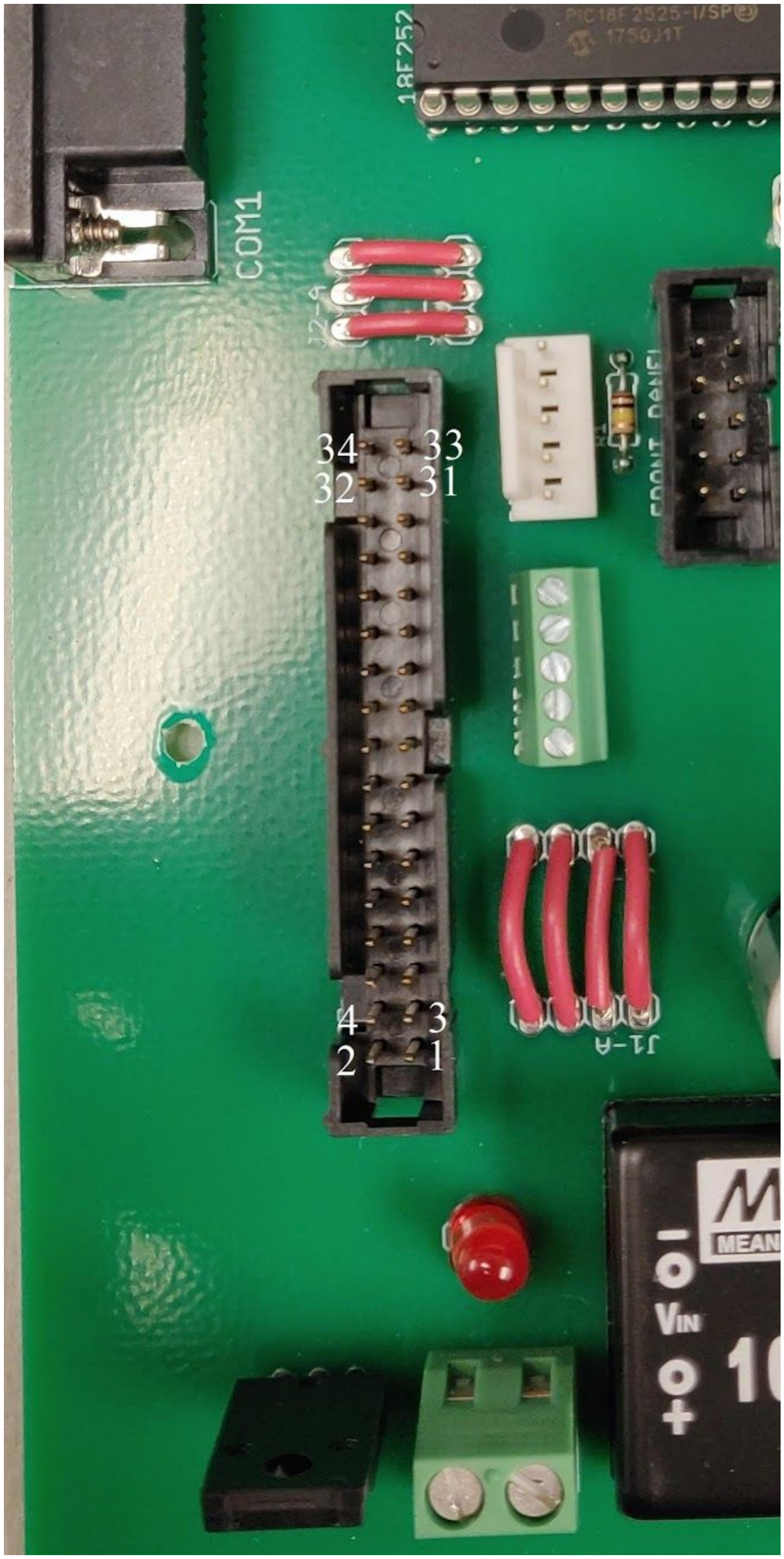


Figure 4: 18F2525 Pinout Form

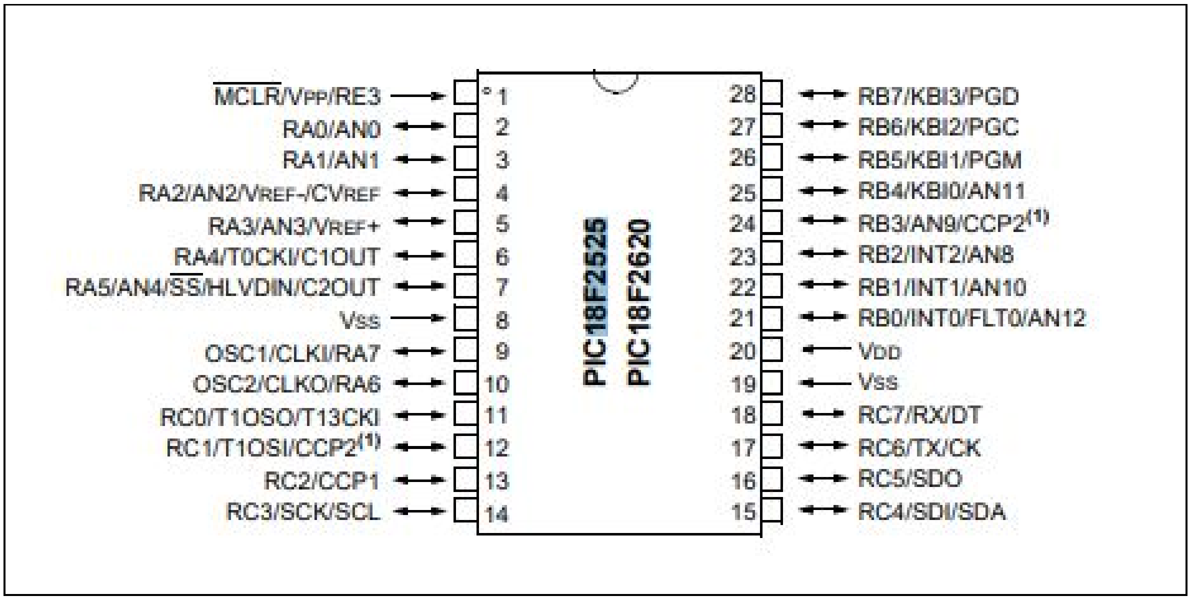
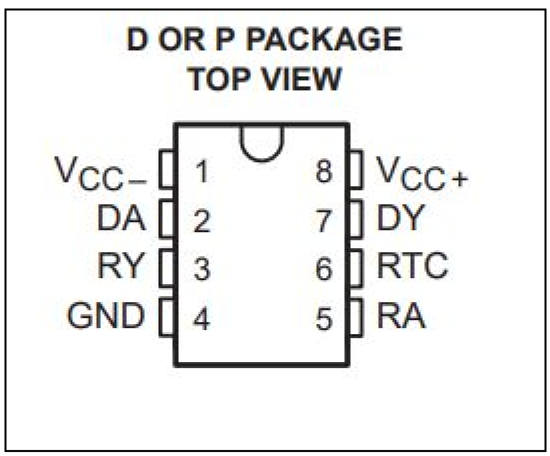


Figure 5: RS-232 Line Driver and Receiver Pinout



References

1. X. Zhao, D. M. Laverty, A. McKernan, D. J. Morrow, K. McLaughlin and S. Sezer, "GPS-Disciplined Analog-to-Digital Converter for Phasor Measurement Applications," in IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 9, pp. 2349-2357, Sept. 2017, doi: 10.1109/TIM.2017.2700158.

2. D. M. Laverty, R. J. Best, P. Brogan, I. Al Khatib, L. Vanfretti and D. J. Morrow, "The OpenPMU Platform for Open-Source Phasor Measurements," in IEEE Transactions on Instrumentation and Measurement, vol. 62, no. 4, pp. 701-709, April 2013, doi: 10.1109/TIM.2013.2240920.