

The VI 00 Main (single phase) combines all the of the VIs to act as a PMU and output data. Note that everything on the Front Panel is output, and all inputs are found through the sub VIs. Make sure to note:

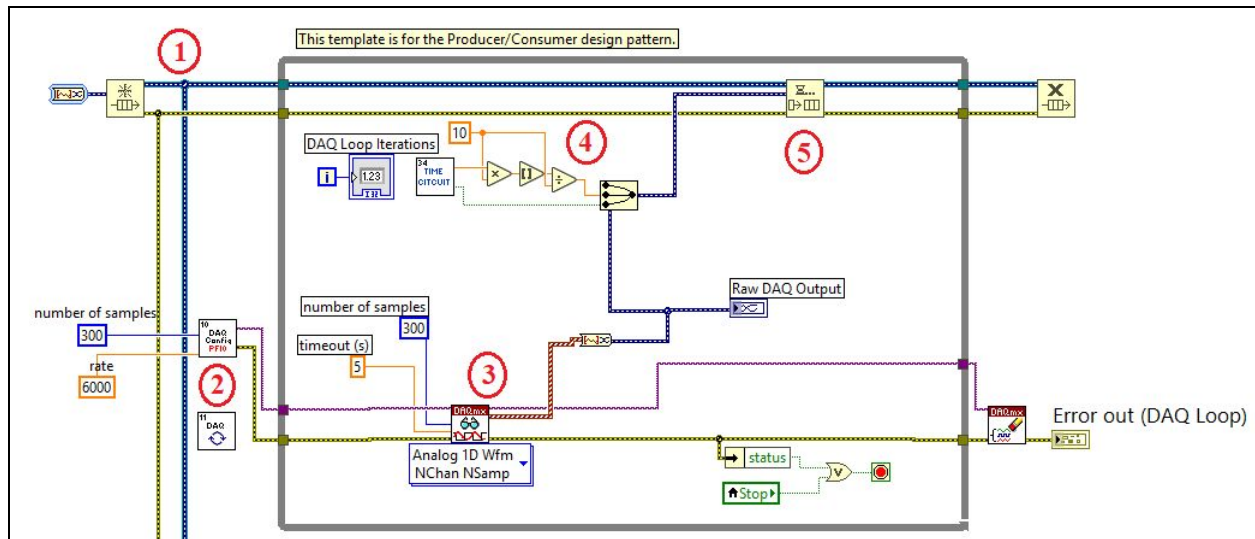
Configuration Information:

The ID and IP address to which the OpenPMU sends data is configured by 'C:\PMUconfig.ini'. If this doesn't exist, create it.

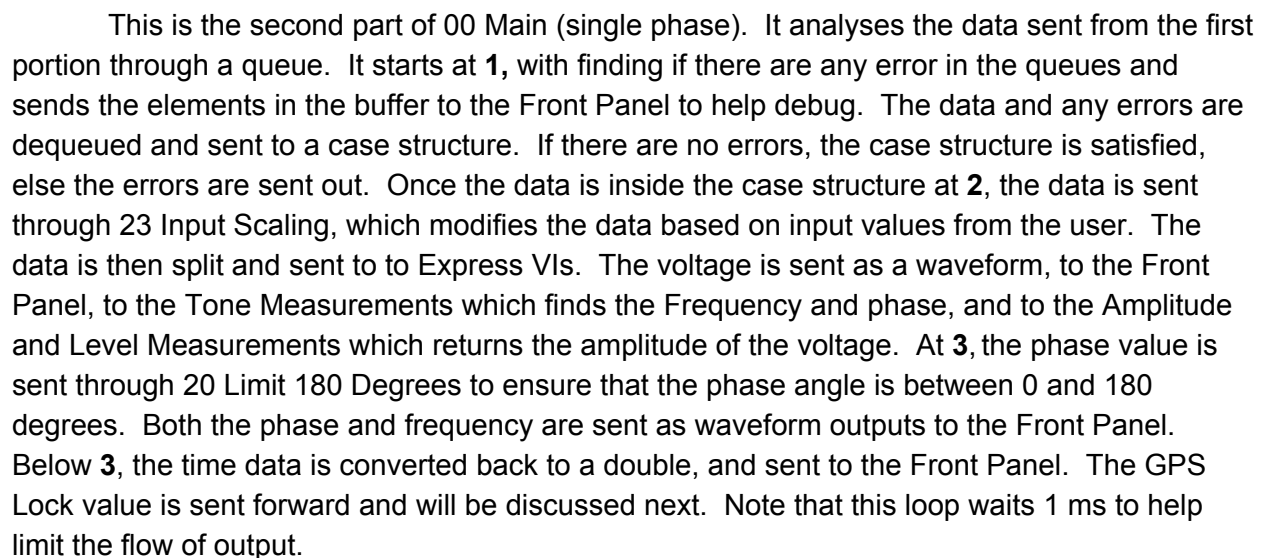
Copy and paste the blue text below without the dashed lines into PMUconfig.ini.

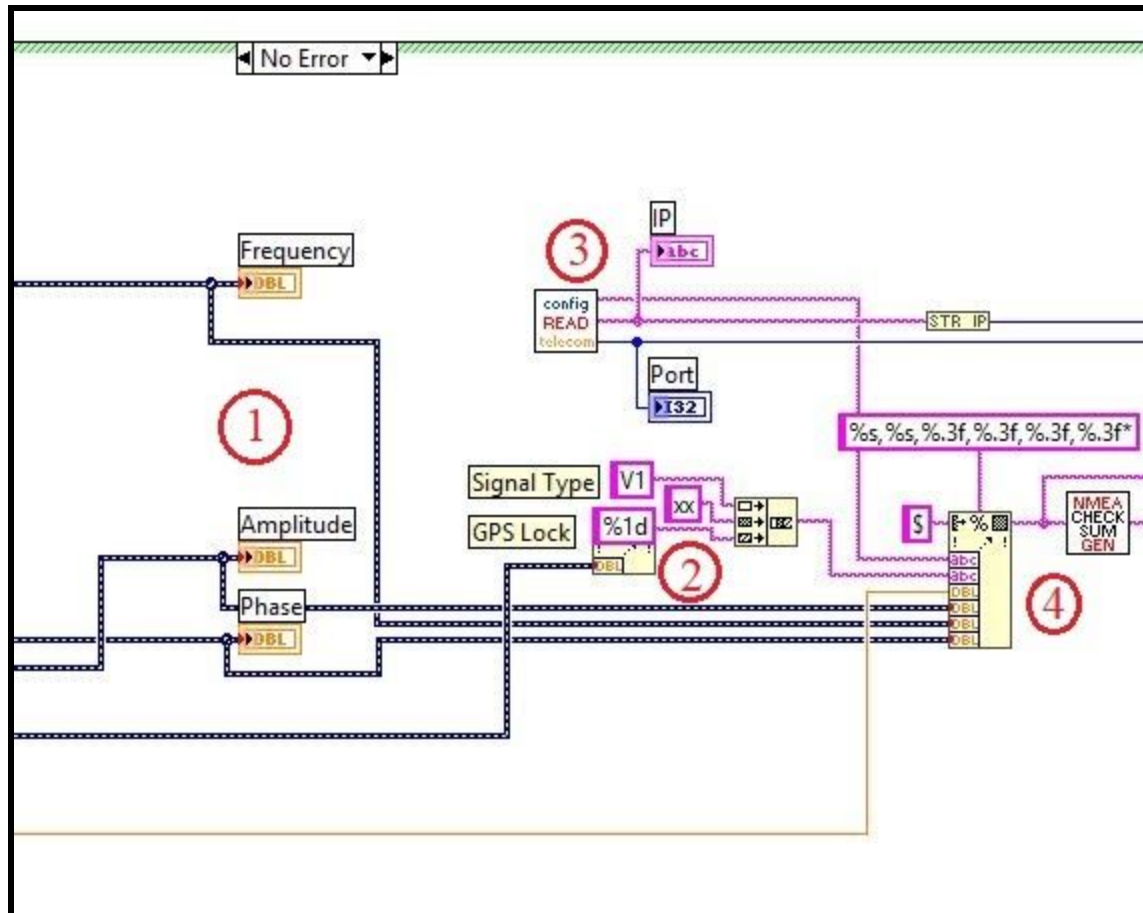
```
[Telecoms]  
ID=PMU123  
IP=192.168.1.100  
Port=40001
```

IP and Port may be any values within normal IPv4 range. ID should be six alpha-numeric characters.

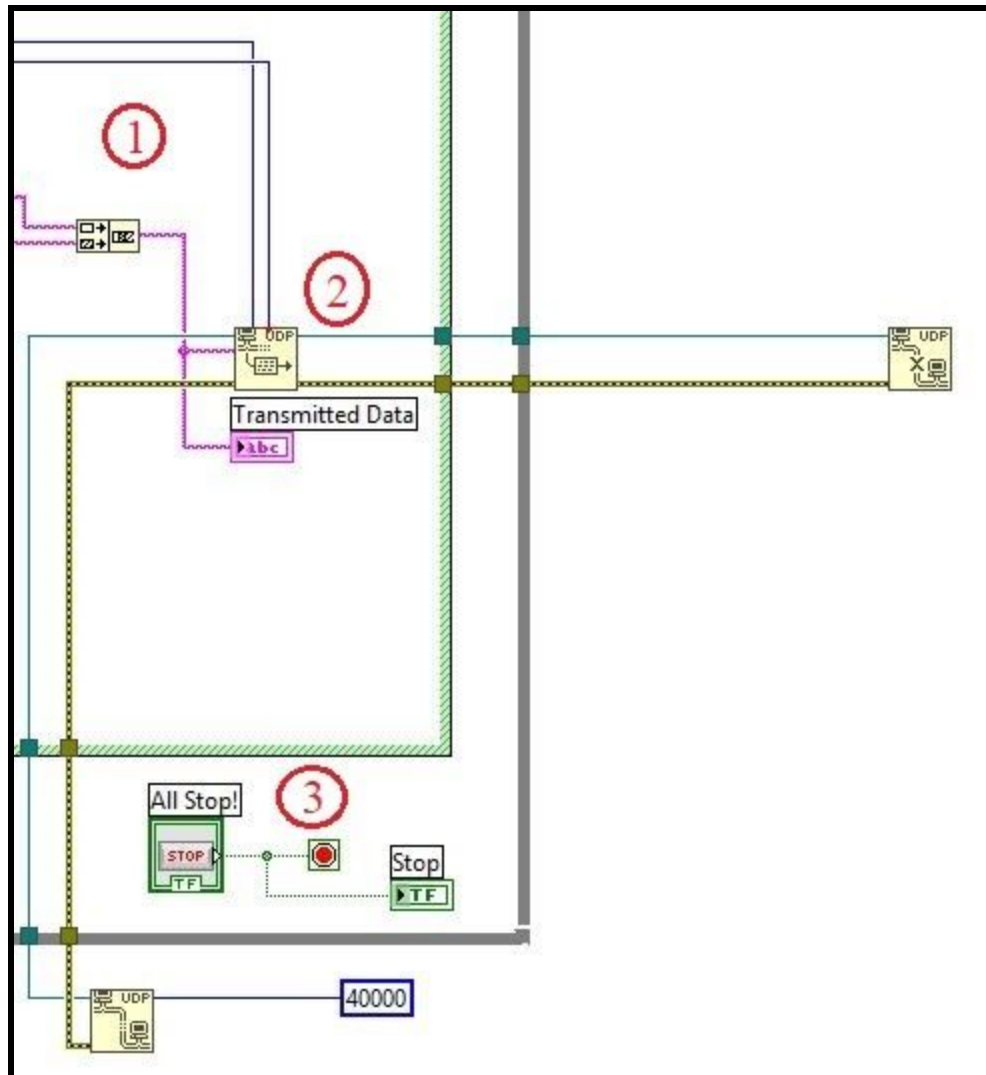


Above is the first While loop of 00 Main (single phase). The purpose of the loop is to continuously take an analog input and store it in a queue. The VI starts at **1** by creating a queue with a Dynamic Data Type. The VI 10 DAQ Config (PFI0) at **2**, sets the number of samples and the rate to sample, then creates the arbitrary task. For all intents and purposes, the task data line acts as our data acquisition line, or the data recorded for processing. Look at <http://zone.ni.com/reference/en-XX/help/370689M-01/daqmxtutorial/newconceptsinnidaqmx/> for more information on what a task is. Following the data line task out, the task is sent to the DAQmx Read (Analog 1d Wfm NChan NSamp) VI at **3**, which helps start collecting the voltage analog input. At **4**, the VI 34 GPS Time from Config reads the time code values from the GPS_time.ini file, that was written too in another portion of the code. The value is multiplied by ten, rounded, and divided by ten to ensure that the value sent to the merge signals is only to the tenths place. The merge signal function takes the time, the voltage and a boolean value if the GPS is on Lock. The merge signal is now enqueued. At **5**, the merged signal of voltage values, timing and GPS locations values are sent to the queue. All errors and tasks are sent to the DAQmx Clear task or Error out (DAQ Loop) outside of the loop. If there is an error, or the Stop button on the Front Panel is pressed, the loop stops and errors are outputted.



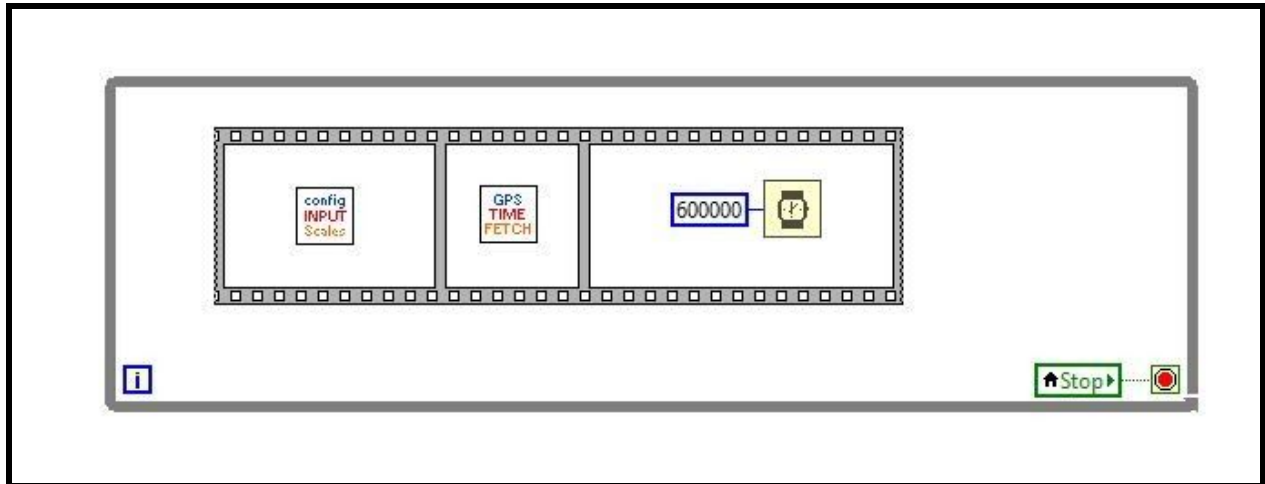


This section of the VI is still in the case structure that ensures that there are no errors before analysing the data. This section starts at **1** where the Frequency, Amplitude, and Phase are all sent to the Front Panel as output double values. **2** takes the GPS Lock boolean value and begins by converting it to a 1D array of the string, either T or F. The T or F is then concatenated with V1 and xx to produce a string sent to format a readable string at **4**. Before this, at **3**, the VI 44 Config File Read Telecoms, which returns the ID, IP, and Port value. The IP and Port are sent to the Front Panel as output. The ID is sent to the Format into String at **4** along with the Frequency, Amplitude, Phase, Time and concatenated string value. These are converted to the string as shown in the picture above of one string with values of two strings four, three decimal place floating point values all separated by a comma. Note the * on the end. The string value is sent to the VI 50 NMEA Calc Checksum, which returns a checksum value.



This portion of the VI finishes the analysis portion of the loop. It starts at **1**, where the GPS string and the Checksum string are concatenated. Once the string is concatenated, it is sent to the UDP (User Datagram Protocol) write, along with the port and IP as net addresses. More can be found out about what a UDP is at:

<https://cs.nyu.edu/bacon/phd-thesis/diss/node32.html> The GPS code is sent to the UDP, as well as output to the Front Panel as "Transmitted Data." At **3**, the UDP is opened for writing, with any errors and the port data sent to the write. The number is the timeout indicator. After 40 seconds, the write will timeout. Also note the all stop, which stops data collection, but does not exit the VI. Once the entire case structure is complete, the UDP is closed.



This portion of the VI is separate from the other two while loops. Each loop iteration starts by getting the user scale values from 43 Config File Get Input Scales. The second portion of the flat sequence is 30 GPS Time Fetch, which finds the time data from the GPS, and outputs it to the file GPS_time.ini for the rest of the VI to read. Once complete, the final part of the flat sequence waits for 10 minutes. Because the flat structure is in a while loop, this process is repeated while the VI is running.