EMERGING TECHNOLOGY

****GTosPMU on the Open Source PMU Project's CodePlex Site**

http://gtospmu.codeplex.com

***GridTrak Project Web Site**

http://www.gridtrak.com

by

Arnold Stadlin
121 W. Earleigh Hgts Rd
Severna Park, MD 21146, USA
ajstadlin@GridTrak.com

*** *GridTrak* is a personal project developed by the author.**
**** *GTosPMU* on the CodePlex *Open Source PMU* project site is *GridTrak's* PMU Sensor released under the Eclipse Public License (EPL).**

Table of Contents

Prerequisites

GridTrak Open Source PMU (GTosPMU) Software Requirements

GTosPMU Software Download and Installation

Simply download the latest version of the GTosPMU software and extract it to a folder of your choice. The GTosPMU software does not require an installation program. The latest GTosPMU software can be found at the following URL:

http://gtospmu.codeplex.com

Microsoft .NET Framework version 4

The GTosPMU software was developed and tested on Windows 7 and was built using the Microsoft .NET Framework version 4. Microsoft .NET 4 is available for download using Windows Update or by direct download at the following URL:

http://www.microsoft.com/downloads/en/details.aspx?FamilyID=0a391abd-25c1-4fc0-919f-b21f31ab88b7

Additional Fonts

The *CONSOLA* True Type Fonts are used to improve the displays. These fonts can be found in the Support folder under the with the GTosPMU software installation folder. There are several ways to install Fonts in Windows. One way is to double click on the font files in the GTosPMU Support folder with the ".ttf" extensions to preview the fonts in a dialog where you should see an "Install" button to click. Another method is to open the *Control Panel's Fonts* configuration applet and follow the instructions for selecting and installing fonts.

GridTrak PHzSensor Model 10 Hardware

The GridTrak PHzSensor Model 10 hardware is a modification of the PHzMonitor Model 9 sensor hardware. To convert the PHzMonitor Model 9 to a PHzSensor Model 10, perform the following steps:

1. Remove the EEPROM 24LC1024 chip

2. Replace the PHzMonitor programmed dsPIC30F3012 with a GridTrak programmed dsPIC30F3012

3. Optionally connect a GPS or other PPS signal to the dsPIC30F3012's INT0 pin. The requirements for the PPS interface circuit may vary. However, if modifying a PHzMonitor Model 9 sensor, the footprint for the 24LC1024 pins provided 5V, Ground, bypass capacitors, and resistors that could be wired in as needed for you GPS PPS interface. See the schematics at the end of this document.

GTosPMU Model A Hardware

The GTosPMU Model A hardware is an original design SynchroPhasor PMU.

1. Optionally connect a GPS or other PPS signal to the dsPIC30F3013's INT0 pin. Unlike the PHzSensor Model 10, the GTosPMU Model A's PPS connection is TTL/CMOS compatible only. Any additional circuit requirements must be supplied external to the Model A's circuit. See the schematics at the end of this document.

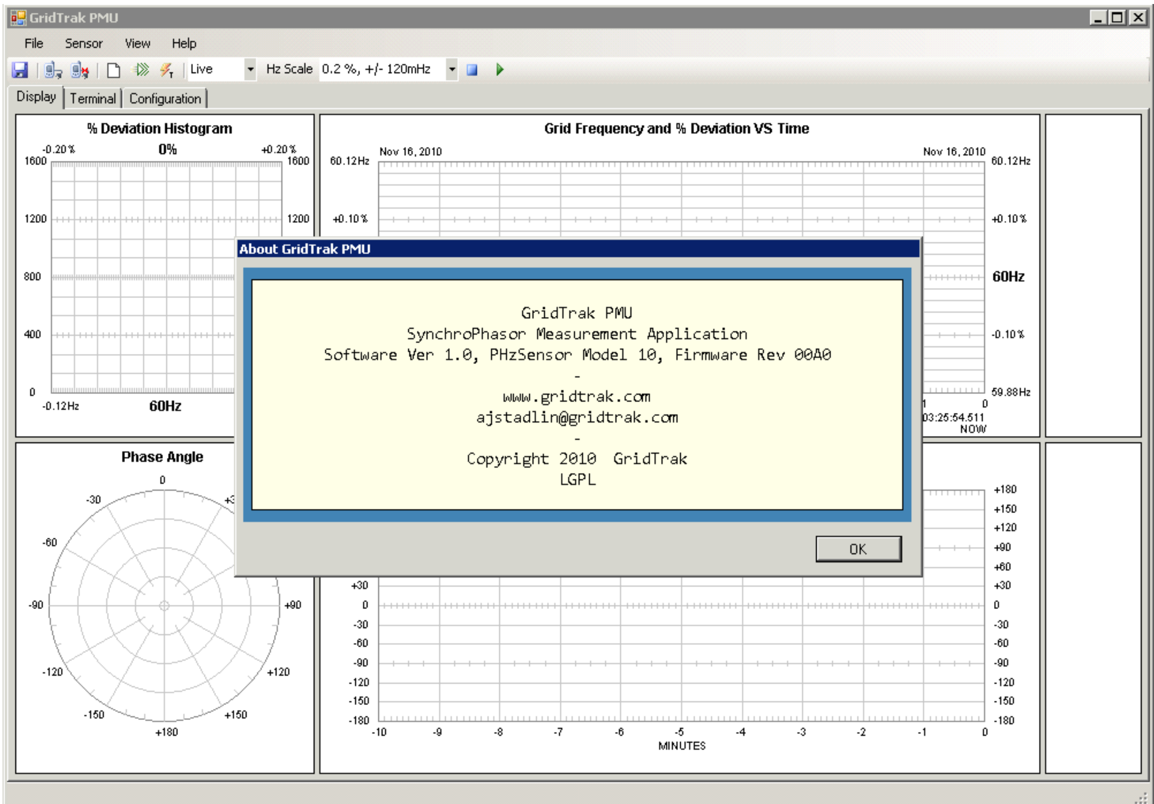
PHzMonitor Web Services Support

The PHzSensor Model 10 is being phased out. I am not making any more and is not part of the GTosPMU project. However, the Model 10 uses the same host PC software and is intended to be compatible with the openPDC software using IEEE C37.118-2005 protocol and the existing Model 9/10 sensors will be supported until a new openPDC or similar service replaces the PHzMonitor system.

A host relay application is planned to enable the new PMU sensors to transmit frequency data to the PHzMonitor project web services.

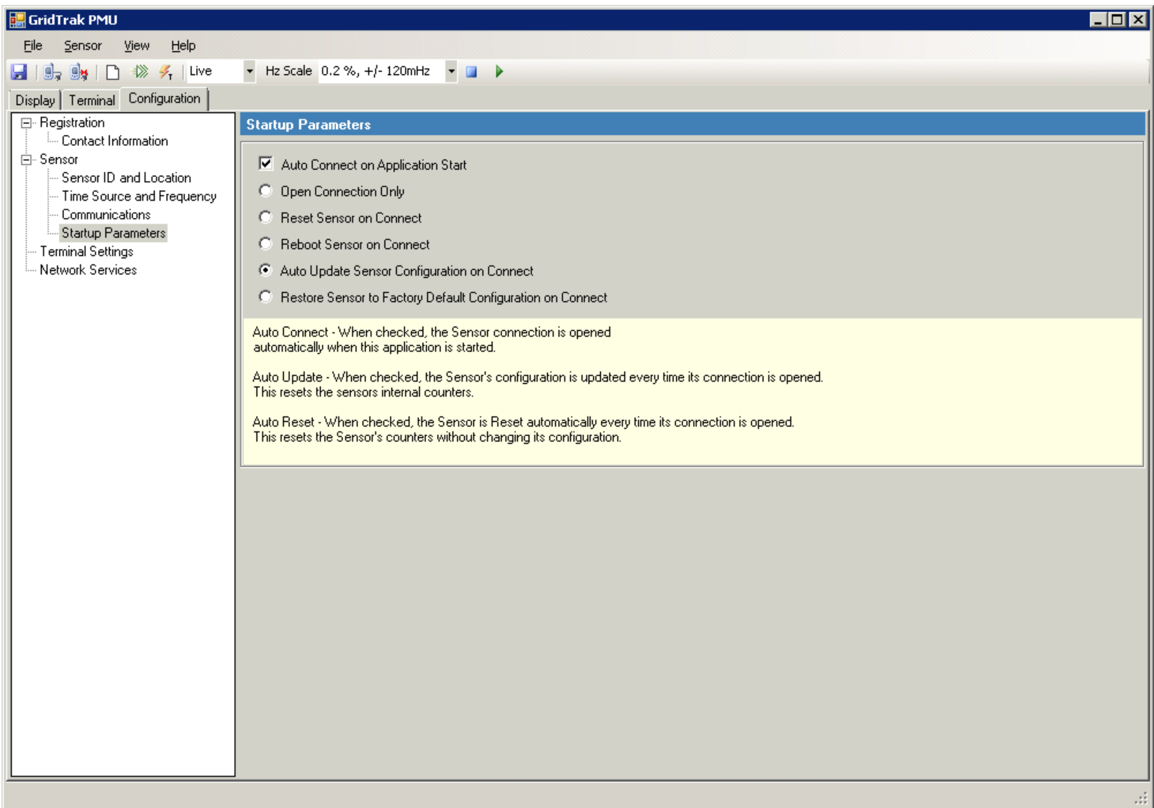
GTosPMU Application Screen Shots

When you first start the GTosPMU software, the About dialog, similar to the one pictured below, displays the credits.



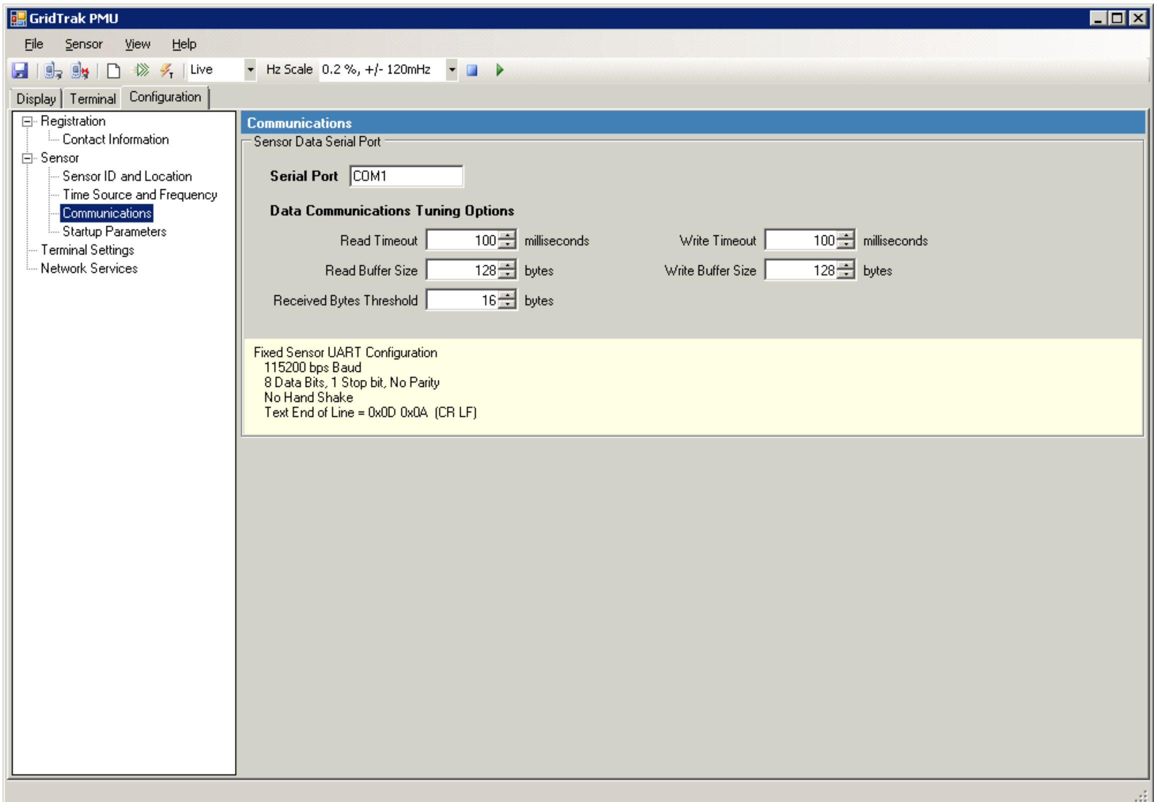
After you start the application, switch to the *Configuration* tab.

In the *Sensor \ Startup Parameters*, select the *Auto Connect on Application Start* if you want to make the program start measuring automatically when it runs. You can add the GTosPMU application to your *Windows Startup* menu folder to run automatically when the PC Host starts.



Select the Configuration *Communications* node and set the Serial Port name for the port that connects to the sensor. The Serial Port baud rate, data bits, stop bits, parity, and handshake parameters are optimized for the sensor and cannot be changed.

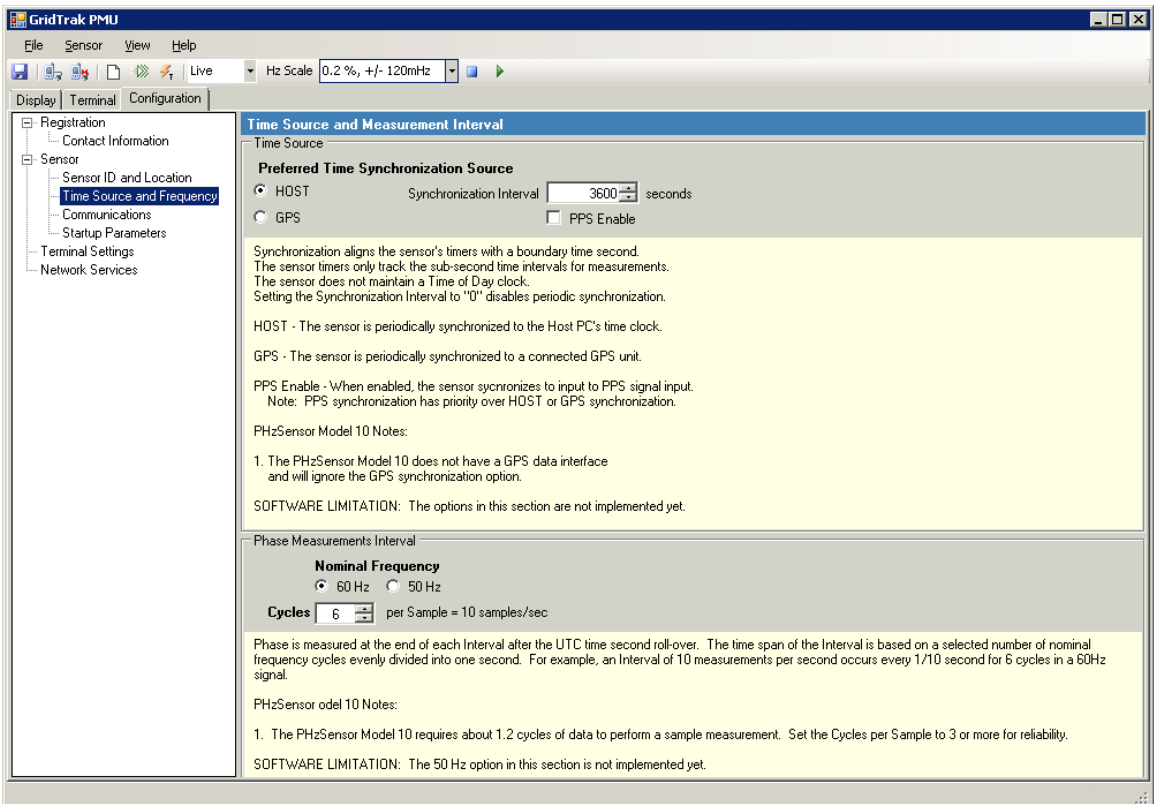
The *Data Communications Tuning Options* are still experimental and their optimal settings have not yet be determined. For best results, keep the buffer sizes larger than the Threshold and keep the timeouts less than the time per sample. Do not make the Threshold larger than the PING response string.



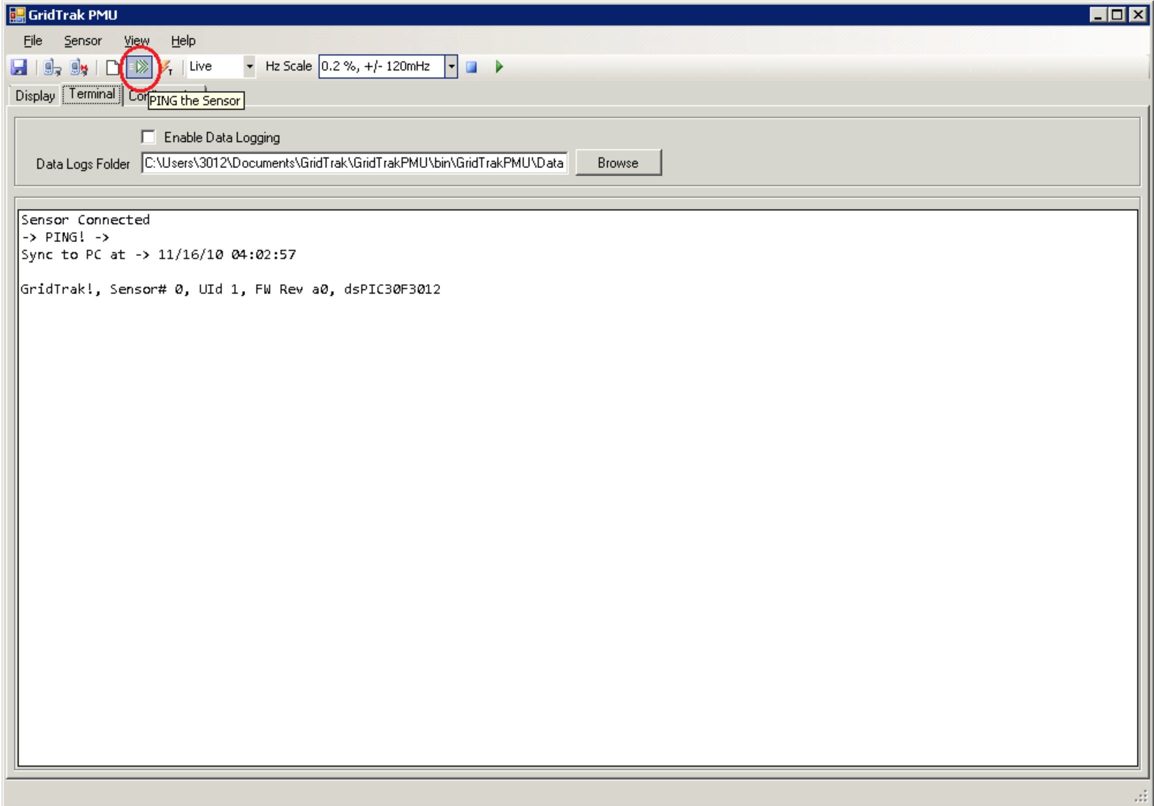
The final configuration setting of immediate interest is the number of Nominal Hz *Cycles per Sample*. This is really just a convoluted way of expressing the number of measured *Samples per Second*. The Samples per Second are computed and displayed. The *Cycles per Sample* terminology may be changed in future releases.

The GTosPMU Model A and PHzSensor Model 10 are technically capable of measuring 60 Samples per Second (or 1 Cycle per Sample). However, this release failed to perform reliably at more than 20 Samples per Second (3 Cycles per Sample). A reliable sample rate for this release is 10 Samples per Second (6 Cycles per Sample). You won't hurt anything if you try 30 or 60 Samples per Second. You may have to physically reset the sensor to get it to respond to commands. I think the issue with 30 Samples per Second or faster is PC Host communications performance.

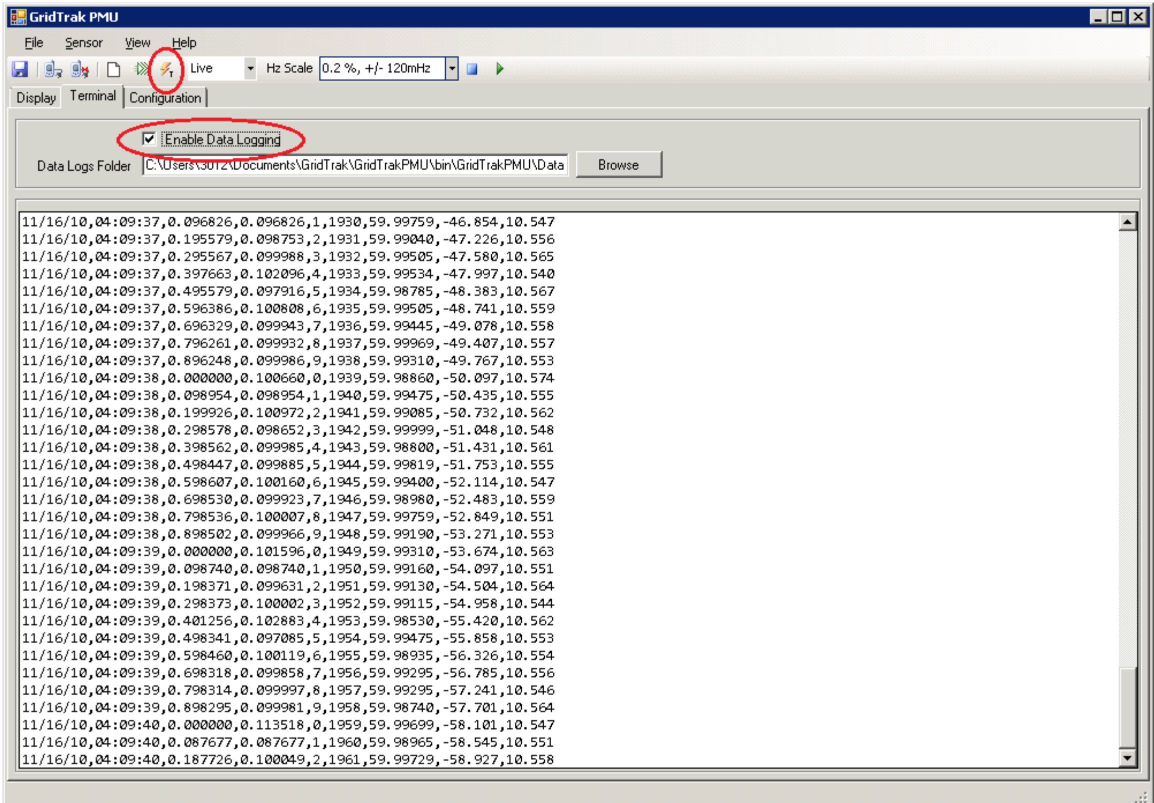
In GTosPMU 1.0.0.4, the Time Source options are not yet implemented. The Time Source *HOST* means that the PC Sends a synchronization command to the Sensor to enforce sensor synchronization to the rollover second. This is the currently implemented mode of operation. Completing this feature will become a high priority when GPS is integrated (est. March 2011).



Test the Connection to the Sensor by clicking the *PING* button on the Toolbar. The *PING* button will automatically switch you to the *Terminal* tab. You should get a response from the Sensor displaying the Firmware Revision and Micro Controller model.



While in the *Terminal* tab, check the *Enable Data Logging* checkbox to save the data to *CSV* files in the selected Data Logs Folder. Click the *Terminal Run* button.



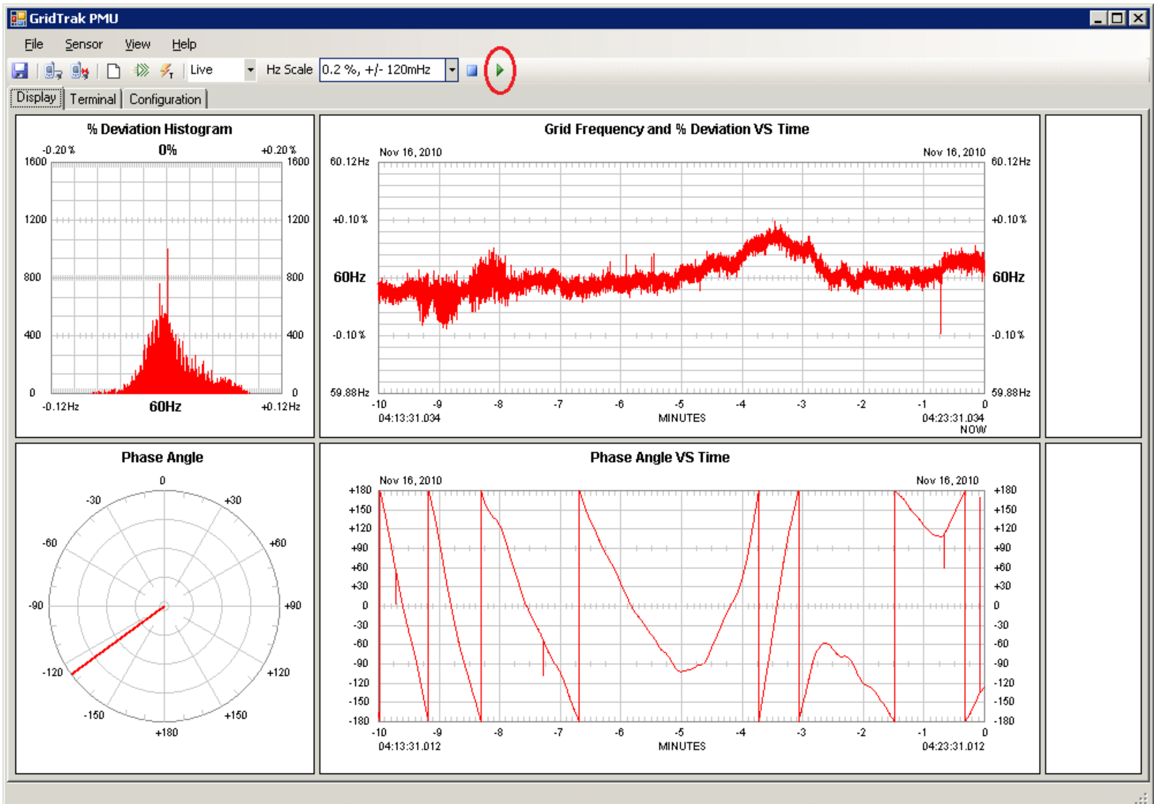
The Data Columns are:

1. UTC Time Second
2. Fraction of a Second after the UTC Second that the Sample is Received by PC
3. PC Measured Elapsed Time since previous sample
4. The Sample Interval after the UTC Second.

   The actual time of the sample = UTC Time Second + (Interval # * 0.001 ms)

5. Sequential Sample ID transmitted by the sensor (used to detect missed samples)
6. Frequency in Hz
7. Phase Angle
8. Amplitude of an ideal sine wave calculated using the frequency and time between the last 2 voltage references and zero crossing.

To watch the live chart, click the *Run "VCR"* button. The chart can be observed live while recording data.



SynchroPhasor Implementation

GridTrak Open Source PMU (GTosPMU) SynchroPhasor Measurement Concept

The SynchroPhasor measurement requires a precision timing pulse to establish a common time base for multiple sensors. The PMU receives a GPS PPS (Pulse Per Second) pulse on a micro controller external interrupt pin or uses its own internal timer if a PPS is not available. The following oscilloscope screen shows a typical PPS signal in purple and a synchronized 60Hz square wave in blue. The top most, green, signal is one of the AC rectified half waves. The AC signal at the PPS rising edge has a phase angle of 0 degrees. The IEEE C37.118-2005 convention measures the phase angle at the peak voltage rather than the zero crossing.

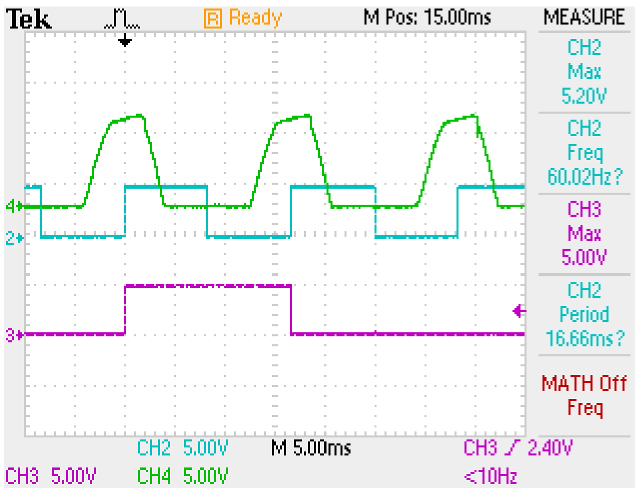


Figure - PPS (Purple), Synchronized 60Hz Signal (Blue), and Grid Signal Half Wave (Green)

On the oscilloscope, we can observe the AC signal half wave gradually shift left when the frequency is greater than 60Hz and shift right when the frequency is less than 60Hz. When the signal is at 60Hz, the half wave signal will maintain a fixed position with respect to the PPS and synchronized 60Hz signal. The greater the deviation from the nominal 60Hz frequency, the faster the shift.

The PMU converts the Grid's Power Signal to complimentary square waves using half wave rectification and a dual channel OpAmp with a +2.5V reference. We select one of the channels as the positive voltage channel by interpreting the duty cycle and square wave edges. Since our reference voltage and rectified half wave are positive, it reduces the duty cycle for the time that the voltage is greater than our reference voltage. Our OpAmp is configured as an *Inverting Amplifier* - thus the shorter portion of the duty cycle is the low part of the signal. In the following figure, the transition where the blue channel 2 signal is leading the yellow channel 1 signal is the "rising edge" of the channel 2 square wave and is the "falling slope" of the channel 2 half sine wave.

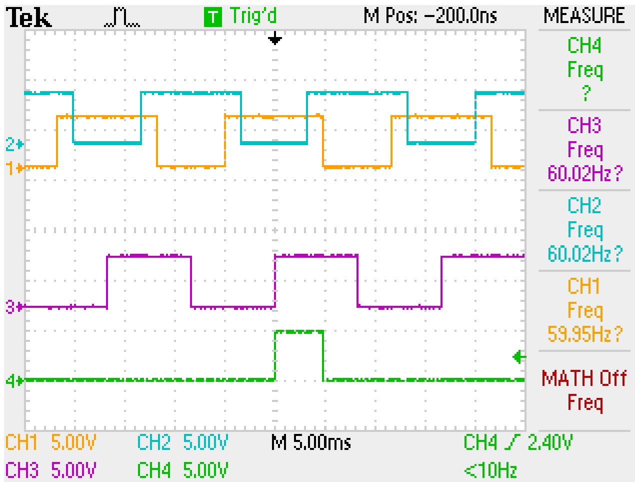


Figure - PPS (Green), Synchronized 60Hz Signal (Purple), and Grid Signal Converted to Square Waves by the PHzSensor OpAmp (Blue and Yellow)

The SynchroPhasor measurement of the signal Phase is computed at intervals that are multiples of the nominal frequency. The PMU will use the last measured frequency and the time from the PPS to compute the phase angle and rate of phase change at each interval.

Table - Interval Multiples for the Nominal Frequencies

Note: Interval Values are in Seconds

60Hz Multiple	Interval	50Hz Multiple	Interval
1	0.01667	1	0.02000
2	0.03333	2	0.04000
3	0.05000		
4	0.06667		
5	0.08333	5	0.10000
6	0.10000		
10	0.16667	10	0.20000
15	0.25000		
20	0.33333		
30	0.50000	25	0.50000
60	1.00000	50	1.00000

The following figure shows a PPS pulse (bottom signal in green), Synchronized 60Hz signal (second up from the bottom in purple), Synchronized Interval Pulse ever 6 cycles (third up from the bottom in yellow), and the line frequency half-wave signal converted to square wave (top in blue).

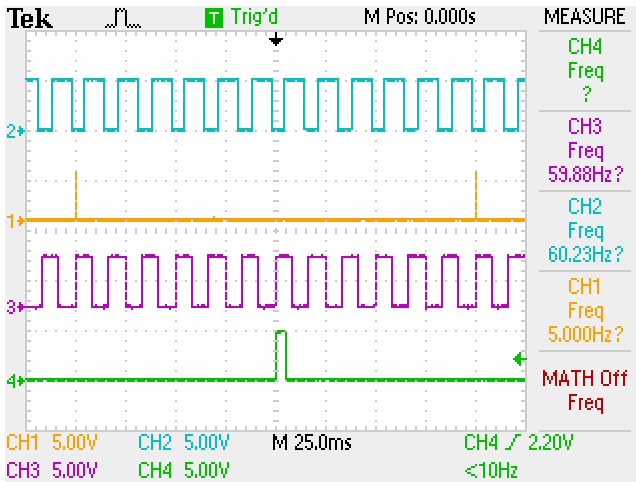


Figure - PPS (Green), Synchronized 60Hz Signal (Purple), Synchronized Interval (Yellow), and Grid Signal Converted to Square Wave by the PHzSensor OpAmp (Blue)

The next figure shows a Synchronized Interval Pulse one cycle after the PPS. Note the phase for the top line frequency signal is shifted. We want to measure the phase shift and how rapidly it changes, with respect to the synchronized PPS and interval pulses. In the oscilloscope screen configured as in the figure below, when the line frequency is greater than 60Hz, the top signal will appear to be shifting to the left. If the frequency is less than 60Hz, the top signal will appear to be shifting to the right. The speed of the line signal shift increases as the difference between the line frequency and the nominal frequency increases.

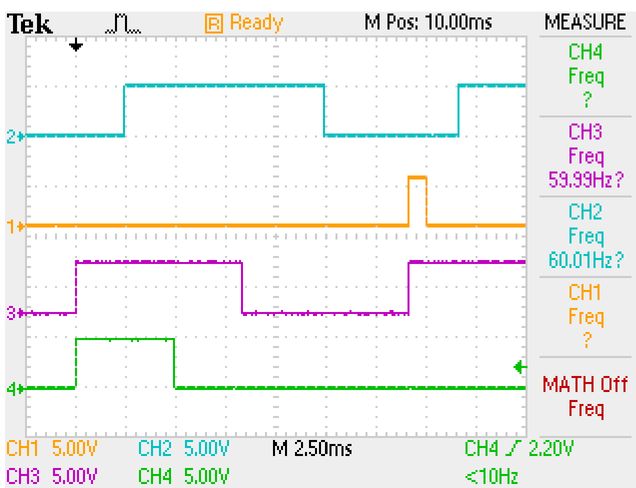


Figure - Synchronized Interval with 1 Cycle Duration

SynchroPhasor Measurement

Description

There are 3 components used by the GridTrak sensor to measure a *SynchroPhasor.* These are the Line Frequency, GPS Synchronized UTC Time Second Roll-over (PPS) Start Time, and Time Intervals that are a multiple of the Nominal Frequency's cycle time. **The IEEE C37.118-2005 standard specifies that the Phase Angle is 0 degrees at the peak of a Nominal Frequency's cycle.** If describing the Nominal Frequency cycles as a Sine Wave, then the zero crossing on the rising slope of the sine wave is -90 degrees or -*pi*/2 radians.

The following figure shows a nominal 60 Hz frequency signal in blue and an input 61 Hz signal in red. In the figure, both signals start at time = 0 seconds and phase angle = 0 degrees. At the end of the first 60 Hz cycle, the 61 Hz phase angle has shifted by 6 degrees. Because the angle's point of measurement is at the end of the 60 Hz cycle, and the 61 Hz peak has shifted *left* by 6 degrees, the IEEE C37.118 standard defines this *Phase Difference* as +6 degrees.

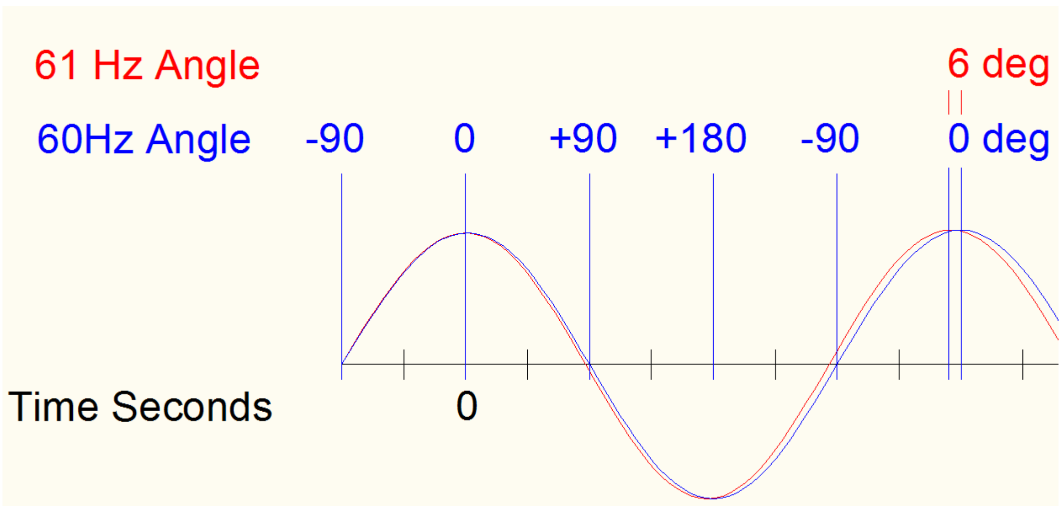


Figure - 60 Hz and 61 Hz Phase Angles for First Cycle with the Same Starting Time

In the next figure, the same signals are shown at the first 1/10 second interval (the end of the 6th nominal cycle). By this time, the 61 Hz signal has shifted +36 degrees with respect to the nominal 60 Hz signal.

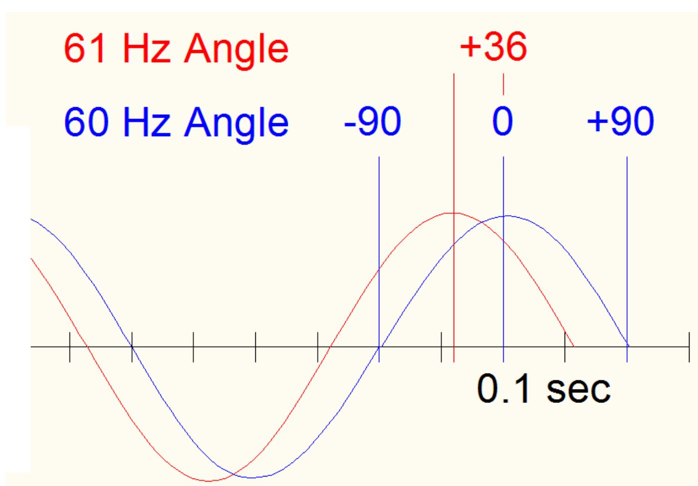


Figure - 60 Hz and 61 Hz Phase Angles after a 1/10 sec Time Interval

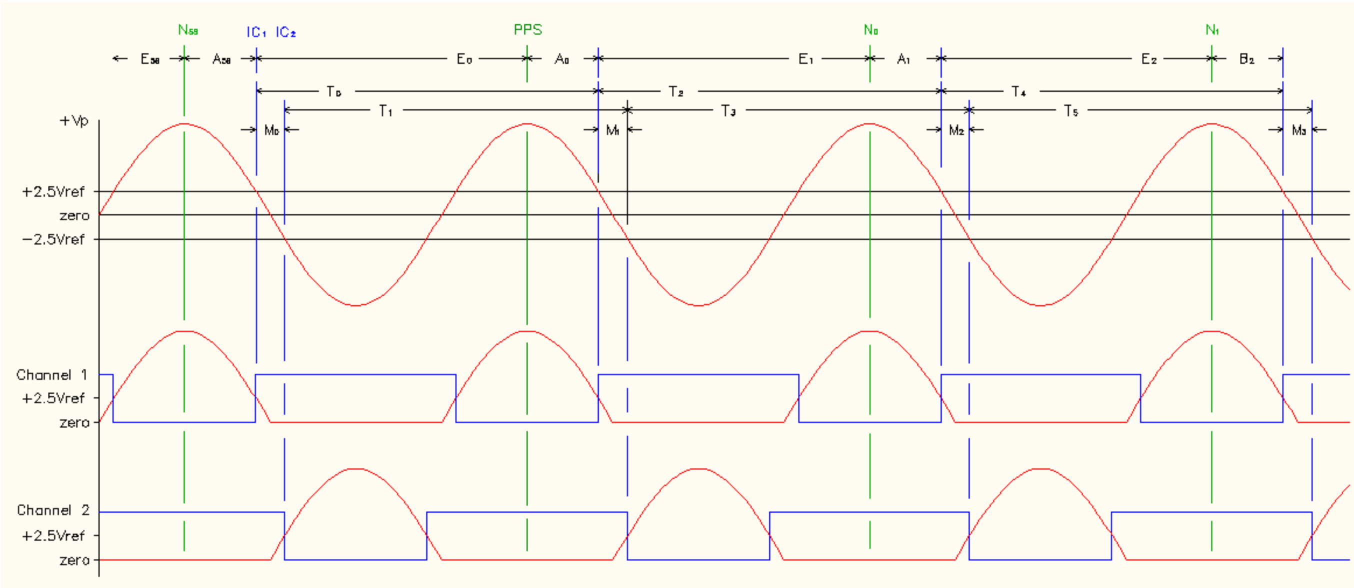


Figure - Synchronized Phasor Measurement

Algorithm

1. On Input Channel 1 Interrupt, Reset T(0)
2. On Input Channel 2 Interrupt, Record Z(0) and Reset T(1)
3. On Input PPS Interrupt, Record E(0) Elapsed Portion of T(0), Reset Interval Timer
4. On Input Channel 1 Interrupt, Record Total T(0)
5. On Input Channel 2 Interrupt, Record Total T(1), Z(1), and A(0)
6. On Interval Timer Interrupt, Record N(i) and E(i+1)
7. On Input Channel 1 Interrupt, Record Total T(i+2)
8. On Input Channel 2 Interrupt, Record Total T(i+3), Z(i+2), and A(i+1)
9. Calculate Phase Angle
   A. Phase Angle = 360 * [(E - M/2) / T]
10. Translate our Sine Wave Phase Angle by -90 degrees to convert it to IEEE C37.118-2005 Cosine convention
    A. Phase Angle = Phase Angle - 90
11. Normalize our Phase Angle to *Compass* (0 deg = "North") convention:
    Clockwise 0, +45, +90, +135, +180.000, -179.999, -135, -90, -45, 0
    A. If (Phase Angle > 180) Then Phase Angle = Phase Angle - 360

Amplitude Measurement

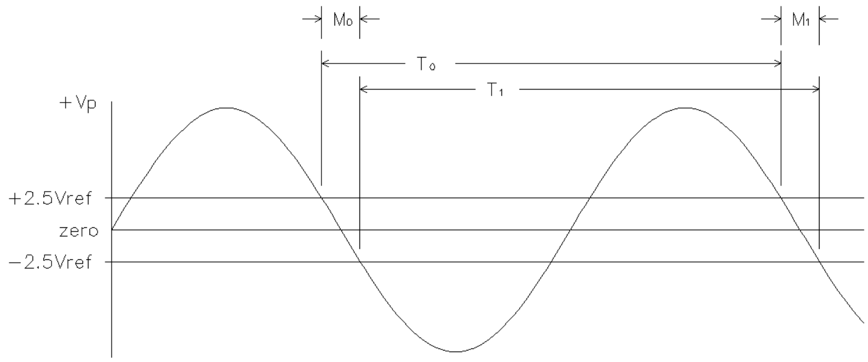


Figure - Sine Zero Crossing Detection

Algorithm for Ideal Sine Wave Amplitude Calculation

1. Measure M(0) and T(0)
   - 1.A. Vref = 2.5V
   - 1.B. M(0) is the time from +Vref to -Vref
   - 1.C. T(0) is the cycle time from +Vref on the falling slope to the next +Vref on the falling slope
   - 1.B. Z(0) = M(0) / 2 is the approximate time from +Vref to Zero Crossing
2. Derive a formula for calculating Vp Peak Voltage using the known values
   - 2.A. A full cycle is 2**pi* radians
   - 2.B. The portion of the cycle from 2.5 Vref to Zero Crossing is 2**pi* * [Z(0) / T(0)] radians
   - 2.C. The formula for determining the Voltage Axis Scale Factor *Vp* is:
     - 2.C.1) 2.5Vref = *Vp* * sine(2*pi* * [Z(0) / T(0)]) = *Vp* * sine(pi * [M(0) / T(0)])
     - 2.C.2) *Vp* = 2.5*Vref* / sine(2*pi* * [Z(0) / T(0)]) = 2.5*Vref* / sine(pi * [M(0) / T(0)])

The general formula for calculating the Amplitude *Vp* for a sine wave from the Zero Crossing slope time between equal voltage references is:

Vp = Vref / sine(pi * M / T)

where *M* is the time between *Vref* and *T* is the total time for the cycle.

We are actually sensing the *Vref* points in the half wave rectified signal. In the half wave signal, the half wave's bottom is 0 Volts DC in our micro controller's electronics circuit. However, with respect to the original sine wave, the half wave's bottom is actually approximately 0.7 V off the zero crossing after passing through the signal diodes. Due to this inaccuracy and the inaccuracy of the transformer voltage, our calculated amplitude is really just an estimate. What makes this estimate valuable is that it is consistent, and if a calibrated signal source is available, we can calibrate our calculation.

Amplitude Formula Example

1. Example Values
   - 1.A. *Vp* = +10V on a sine wave starting at *t* = 0
   - 1.B. *Hz* = Nominal Frequency = 60Hz
   - 1.C. *T* = cycle period in seconds
   - 1.D. *Z* = slope time from Zero Crossing to +*Vref* to Zero Crossing
     - 1.D.1) *Zrad* = radians to the first +*Vref*
     - 1.D.2) *Zsec* = seconds from Zero to +*Vref*
   - 1.E. *M* = slope time from +*Vref* to -*Vref*, *Z* = *M* / 2
2. *Vp* occurs at the first quarter of the cycle
   - 2.A. sine(*pi* / 2) = 1, therefore: *Vp* = sine(*pi* / 2) * *Vp* = sine(*T* / 4) * *Vp* = +10V
3. Solve for *Z* to get the *Vref* Slope Time in Seconds from Zero Crossing
   - 3.A. 2.5*VRef* = *Vp* * sine(*Zrad*)
   - 3.B. *Zrad* = asine(2.5*Vref* / *Vp*)
   - *3.C.* *Zsec* = (asine(2.5*Vref* / *Vp*) / 2 *pi*) * *T* sec = *M* / 2 seconds
4. If *Vp* = 10V then *Zrad* = asine(2.5*V* / *10V*) = 0.25268 radians
   - 4.A. *Zsec* = *Zrad* / (2 * *pi* * *Hz*) = 0.25268 / (2 * *pi* * 60 cycles/second) = 670 micro seconds

Concept Check

1. Example Values for a sine wave with the Amplitude equal to *Vref*.
   - 1.A. **Vp = Vref** = 2.5V
2. *Zrad* = asine(2.5*Vref* / 2.5*Vp*) = *pi* / 2 radians
3. *Zsec* = (*pi* / 2) / (2 * *pi* * 60 cycles/second) = 4.1667 ms
4. *Zsec* * 4 = 1 cycle's time = 16.667 ms or the period for a 60Hz cycle

GTosPMU Communications Protocol

The GTosPMU supports two communications protocols: IEEE C37.118 and its own custom protocol. The IEEE C37.118 messages start with the byte 0xAA. To distinguish between GTosPMU and IEEE protocols, the GTosPMU messages will start with 0x40 or ASCII "@".

The GTosPMU is designed to process the entire message

Command Message Prefix Bytes

Prefix Byte	Description
AA	First Byte of an IEEE C37.118 Command Message
40	First Byte of a GTosPMU Command Message, ASCII "@"

GTosPMU Command Set

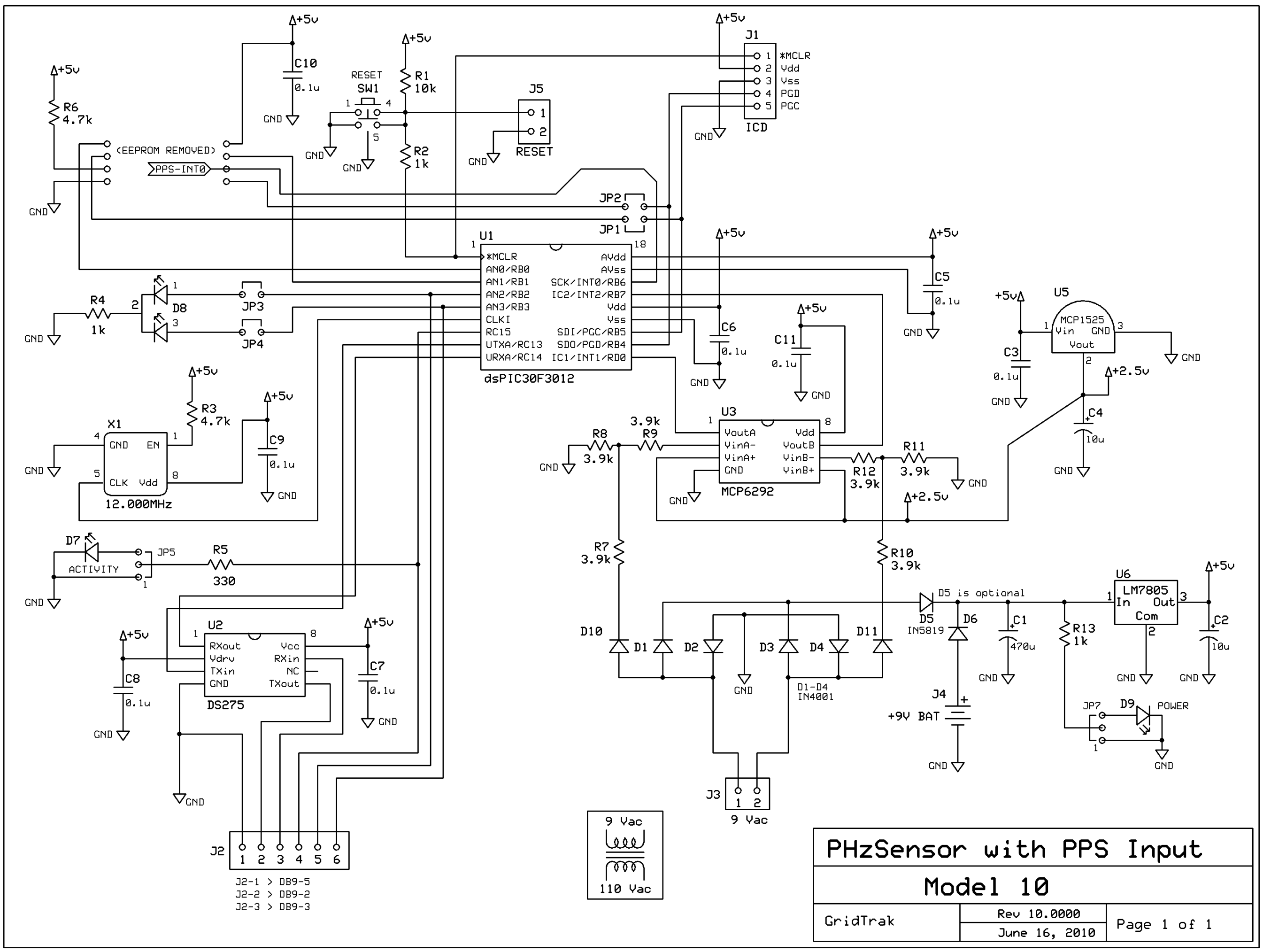
Command	Interpretation	Description
40 40	@@	PING, Send Device ID and Firmware Version
40 21	@!	Reset EEPROM Configuration with Factory Defaults and Reboot
40 23	@#	Reboot: Reload the EEPROM Saved Configuration and Reset
40 2A	@*	Soft Reset: Resets the process loop without reloading configuration. Resets Timers, Counters, clears buffers, etc.
40 30	@0	Idle Mode: Run without transmitting data
40 31	@1	Streaming Text Transmission
40 32	@2	SynchroPhasor PMU Style IEEE C37.118 Data Transmission
40 33 ##	@3 Intervals	Set the Synchronized Intervals = number of nominal cycles between measurements For example: Intervals = 6 cycles @ 60Hz makes a measurement every 1/10 second
40 41 ## ## ## ##	@A Sensor ID Password	Configure the Sensor ID and Password (16 bits each).
40 42 ## [32 bytes]	@B EEPROM Page# [32 bytes]	Configure 32 bytes configuration page# in the EEPROM
40 43 ##	@C EEPROM Page#	Transmit a 32 bytes configuration page# from the EEPROM
40 44 ##	@D Nominal Freq	Set the Nominal Frequency
40 53	@S	Save Running Configuration to EEPROM
40 57 ##	@W any byte	Pause: Hold Main Loop until next byte received.
40 58	@X	Quit the Main Loop; Stop All Processing (no reset and no reboot).

Command	Interpretation	Description
Diagnostics Modes		LED on Pin #10 (RC15) set high at the start and set low at the end of the procedure being observed. Except for the @a - None, the modes can be combined to measure overlap duration.
40 61 61	@aa	None, clear all diagnostics mode and run optimized
40 61 62	@ab	Main Loop
40 61 63	@ac	Command Execution
40 61 64	@ad	Data Packet Transmit
40 61 65	@ae	Channel 1 Input Capture, Frequency Half Wave Interrupt
40 61 66	@af	Channel 2 Input Capture, Frequency Half Wave Interrupt
40 61 67	@ag	Timer1, Synchronized Interval Interrupt
40 61 68	@ah	Timer3, Input Capture Time Base Interrupt
40 61 69	@ai	EEPROM Procedures
40 61 6A	@aj	UART1 Transmit
40 61 6B	@ak	UART1 Receive
40 61 6C	@al	UART2 Transmit
40 61 6D	@am	UART2 Receive
40 61 6E	@an	Zero Crossing, Time from +Vref to -Vref
40 61 6F	@ao	INT0, GPS PPS Interrupt
40 61 70	@ap	INTx, Other Interrupts (not yet implemented)
40 61 71	@aq	On GPS PPS Interrupt, Transmit Elapsed Time E[0] since Input Capture on Channel 1
40 61 72	@ar	On Input Capture Channel 1 Interrupt, Transmit T[1] and Z[0]
40 61 72	@as	On Input Capture Channel 2 Interrupt, Transmit T[0] and Z[1]
40 61 73	@at	On Timer1 Interval Interrupt, Transmit Data

APPENDIX A - GRIDTRAK PHZSENSOR MODEL 10 SCHEMATIC

The GridTrak PHzSensor Model 10 schematic shows the changes required to convert the PHzMonitor Model 9B to enable a GPS PPS Input signal on the micro controller's external interrupt pin RB6/INT0. Removal of the 24LC1024 EEPROM from its socket permits a connection to the RB0, RB1, and RB6/INT0 pins. The socket can be used as a makeshift interface header for RB6/INT0, RB0, and RB1 micro controller pins.

Technically, no changes need to be made to the Model 9 circuit board. Only the 24LC1024 EEPROM needs to be removed from its socket.



Appendix B - PHzMonitor PHzSensor Model 9B Schematic

Last PHzMonitor Sensor Model for Frequency Measurement.