N12AY Capacitance Fuel Gauge with Capacitance Fuel Sender by RED
A DIY Fuel Gauge Project for those
Not afraid to 3D print and hand wire electronic circuits!

Background:
I replaced my 2¼ inch Westach E-F fuel gauge with a volt and amp meter as the capacitance probe with the standard gauge was just not reliable or easily interpreted. Full would indicate full, ½ on the gauge was about 1.3 hours of fuel and a ¼ was down to 30 minutes or less. Any change in fuel type from auto fuel to 100LL was a cause for a recalibration as the capacitance of the fuel was significantly different. The Princeton Capacitance probe seemed to have difficulty keeping its calibration even if I used AvGas exclusively. A complete analysis and tear down of the Princeton probe revealed auto fuel had crept into the nylon to aluminum probe connection and had compromised the circuit board wiring causing a short.

After a bench check teardown and attempted repair, I decided the best course of action was to replace the capacitance unit with a newer RED capacitance probe. This probe is a two-part unit. It consists of the probe itself (it uses a rubber grommet seal just as the Princeton probe did, and the electronic circuit board and recalibration box easily fit in the headrest. A small separate box containing the push buttons, circuit board and wiring plugs is mounted to two AN3 bolts in the headrest. In my review of how to make the fuel quantity reading and display system more reliable, I decided that the Europa fuel tanks odd shape made a typical Empty to Full in ¼ increments analog gauge a poor choice as my Westach gauge worked flawlessly but was a metal gymnastics exercise to note how much fuel was available. As in my case a ¼ of a tank was under 4 gallons remaining yet half was about 7 gallons. Not a difficult memory task to interpret but today's typical commercial automotive fuel quantity is either a capacitance sender located in the tank or more commonly a fuel float resistance sender that displays in gallons or miles until empty. The output and the display is calibrated and reads out digitally in gallons or miles until empty. The typical capacitance 12 volt output reading from a 0-5 volt sender is read by a simple voltage regulator, oscillator circuit and output converted by a small CPU on gauge or digital displays on the instrument panel. Normally today, the driver's display is in gallons as well as fuel flows, average fuel milage, divided by fuel remaining to yield miles until empty. Why can't airplanes be like this?

As N12AY is an analogue gauge aircraft, it is equipped with an EI FP-5 fuel flow, which displays fuel used, flow, and time until empty. Since I still have the mental capacity to calculate distance and fuel remaining divided by fuel flow into distance as well as time, I would only need a more precise measurement of my fuel level remaining in the tank. Why? Sometimes the fuel quantity in the EI FP-5 doesn't get updated by the pilot! But I also have a sight gauge but…

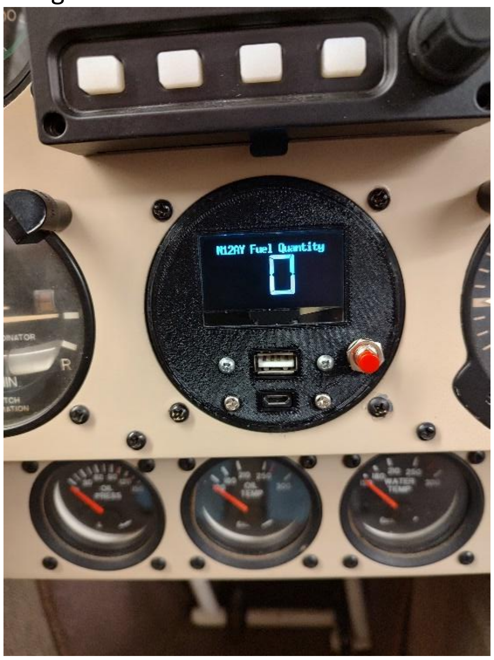
To be honest, sight gauges are fine unless fuel is trapped in the vent line, and those site tubes can get rather dreary to view through, therefore, I decided to fabricate an electronic display of my digital fuel quantity.

Solution:
I painfully measured my tank gallon by gallon and measured and marked up the side of the tank each amount. In the Europa when the wings are removed this is easy to do because the tank is opaque. I then made a calibration stand out of PVC and measured and marked it to reflect the fuel column height. This allowed a quick and repeatable full to empty and vice versa fuel level vs gallons remaining for probe calculations. I calibrated my probe as well as marking my tank through the wing spar opening and marked the fuel level for critical amounts on the tank as well as my sight gauge.

I have an unused 3.125" instrument hole beneath my EFIS and decided to use this hole to combine some USB ports for the GRT, a stall warning system test button needed a home , and I felt the 3 1/8 inch instrument had the extra space for all the above as well as house my CPU and display for the display of digital fuel quantity.

I have a couple of 3D printers and elected to design a gauge in 4 parts. A face, a receiver for the face with screw inserts, a CPU holding fixture, a rear cap and finally a cover for the electronics assembly. What good is an electronic sender unless one also has a fuel sender test set to check if the readings are correct for a specific fuel sender output. Example: My fuel gauge bench test set is designed as follows: Since the RED sender reads at or near 5 volts when the tank is full. At "Empty it reads zero volts. To dim the unit for night operations, it is a simple matter of feeding the dimmer pot voltage to the microprocessor and program the unit display to automatically dim when the dimmer pot is turned on. Both the gauge unit and the tester are shown below:

Gauge:



The gauge is unlabled in this photo:

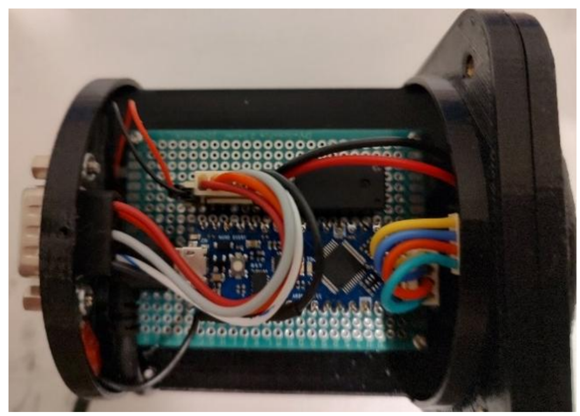
The Red button is a momentary stall warning test. That warning test is documented in another paper. Push the button and the stall warning light, horn and voice warning all fire up.

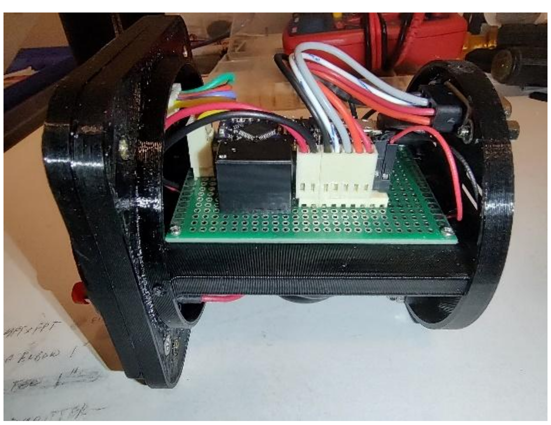
The upper USB A is for GRT updates.

The Micro USB is for the Arduino Nano Every Programable processor. If my calibration changes, I can simply reprogram my micro processor. It is an easy way to adjust your calibration, make the display flash or alert a low fuel situation etc.

The interior of the gauge is shown at right:

Looking from the top:
Note there are plugs making install and upgrade easy. That black box on the upper left is a 12 volt SPST relay above the Nano Micro Processor. My dimmer puts out 12 volts when turned on which activates the relay. The relay then makes connections on the circuit board to activate the dimming programming for the OLED display.





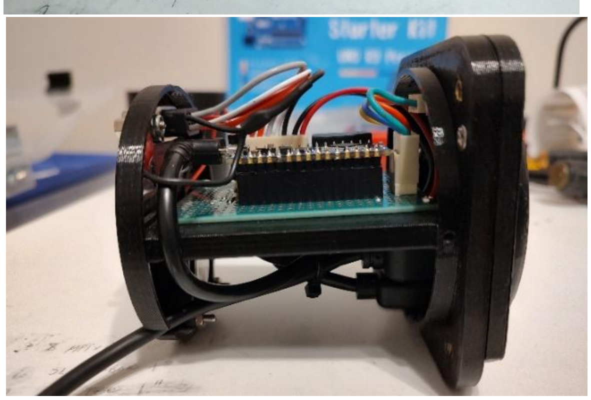
A look from the other side shows the original wiring block. I have now found some shorcuts to reduce pins and wiring connections.

The relay and the 9 pins soldered to the board take input from the DSub and output to the built in fan (although the unit does not get hot, the cockpit does so added ventilation is always prudent for electronics for those of us in hot climates.

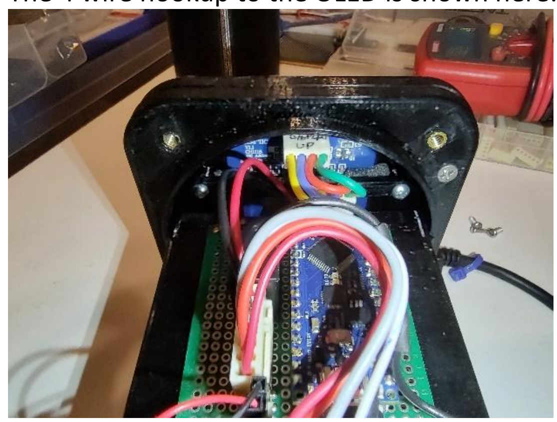
The interior bottom of the instrument is shown below:
Looking from the bottom side the USB cables are not cut nor is the heat shrink insulation removed but are coiled so they are taking up a large amount of space. Luckily the 3 1/8 inch gauge is large enough to house this. There is a fan blowing directly on the base of the Drok voltage regulator to ensure cooling of the components. Later versions eliminated this voltage regulator as the Arduino Nano Every can accept 9-30 volts and stay accurate in its operations. I had room so I put the stall warning test switch as an experiment. That works extremely well.

The fan is easily seen at the left.

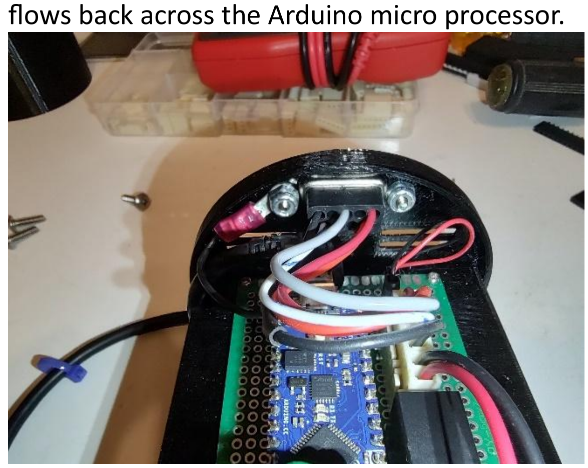
The USBs are quite tight to each other and just clear the stall warning test push button.



The 4 wire hookup to the OLED is shown here.

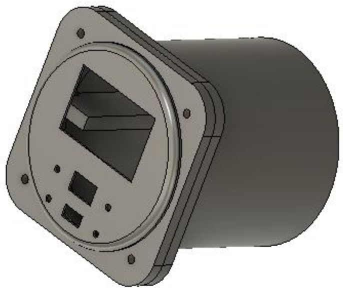


The D Sub shown below is flanked by two vents for air exit and it keeps the processor cool also. The fan air blows along the bottom of the circuit board then hits the face to cool the OLED then flows back across the Arduino micro processor.



The finished instrument has a sleeve that slides over the unit making a near air tight seal allowing excellent fan performance.

My 3D printed assembly includes holes for the OLED display, two USB slots for a micro and an A, plus a button hole for the stall warning test switch. The back cover has a fan hole, exit slots and a D sub holder.



The first installation in the panel without any labels is shown below. Since I carefully calibrated the capacitance probe vs the fuel level, vs the voltage output from the RED sender the unit has proved on the ground and in flight at level flight cruise to be extremely accurate.

The smoothing allows for nice slow fuel digital changes and works well enough during climbs and descents.

Originally, I chose to power the unit through a voltage regulator such as the Drok Buck converter. It keeps the unit voltage measurement much more accurate and as the voltage of the aircraft changes with RPM, the Buck converter will continue to supply a steady 12 volts. The unit fuel quantity doesn't change from 9-30 volts. However, the Nano Every board does this internally and is more compact and is less wiring.

The unit can be updated very easily from a laptop using the Arduino platform. Shown above, I have the USB micro cable installed from my laptop for an update tweaking of my dimmer, brightness and display wording.

To update the fuel gauge simply turn the master off to un-power the aircraft and all of the panel components. Simply insert the USB micro into the update USB connector slot with the other end in the laptop PC and open the Arduino Program, select the updated sketch and install. After updating your processor program, close the Arduino program. Unplug the USB micro and unplug the computer. Once the computer is removed, turn on the master and review the updated quantity, or your dimmer operation. Easy.

Flight test has determined the accuracy in flight of the fuel gauge to be quite remarkable. Since the RED capacitance probe has smoothing built in its programming, the Arduino sketch does not require a smoothing program added to the sketch. Less programming was the result.

Connection to the aircraft is via a 9 pin D sub which most builders have learned to use and abuse and soldering and or crimping techniques must be learned. The OLED display presents a learning curve as well as the Arduino IDE or programming. Videos and tutorials by Arduino and other builders and programmers make it a long weekend learning event to learn the various programming techniques and procedures. The calibration of the fuel tank is important, don't skip it. To be honest, one can simply start from empty and record the RED sender voltage with a volt ohm meter and note the gallons of fuel from empty to full. Then adjust the sketch values for your tank and aircraft. Any aircraft having a fuselage tank such as a Rans S-6 or similar can use this system.

An example of the voltage to fuel quantity relationship for N12AY with the newest Europa fuel tank, is shown below in the spreadsheet.

Output of Sender vs. Fuel Elevation vs. Priceton Probe Output Voltage

Fuel Indications and Levels				Votage output per Level					
Fuel Gauge Indicator Level / Inches from tip	Probe Level from tip in inches	Europa Approx. Tank Level	Approx. Europa Fuel Quantity of Main Side	Princeton Probe Voltage Output vs Battery Voltage 12.35	15	Linear Voltage per Gallon	Curve	Approximate Flight Time	
Full	15	Full	15	4.98			LINEAR	Three Hours	
	14		13.3	4.91		0.27 v/gal			
	13		11.7	4.45					
	12.75	3/4	11.25					Two Hours	
	12		10	4.00					
3/4	11.25		9	3.85					
	11		8.25	3.60					
	10.25	1/2	7.5						
	10		7	3.19					
	9		5.75	2.92		0.32 v/gal	LINEAR	One Hour	
	8.5		5	2.56					
	8		4.5	2.38					
1/2	7.5		4	2.19					
	7.25	1/4	3.75						
	7		3.25	1.86					
	6		2.5	1.49					
	5		2	1.56					
	4		1.75	1.02					
1/4	3.75		1.5			0.74 v/gal	LINEAR		
	3		1.25	0.77					
	2		0.8	0.46					
	1		0.4	0.14					
Empty	0	Empty	0	0.03					2.5 Gal/30 min Reserve on Starboard Side.

Note:
This chart refers only to the physical tank and not the amount in the fill neck for the old tank.
New Europa tank contains about 1 gallon more fuel than original Classic tank.
All approximate levels are based on the new tank and is rounded to 15 gallons.

Note: 2 on New Tank:
Filled to the overflow drain plummbed into the cobra neck with the new tank and in a trigear the total fuel from zero to all lines, pumps, filters, and the fill neck is approximately 20 US gallons.
Useable fuel level flight main and reserve is estimated at 19 gallons without the fuel neck full.
Main side useable filled to the fuel neck may be as high as 16.5 gallons for the new tank. (19 Total)
Main side useable filled to the fuel neck may be as high as 15.5 gallons for the old tank. (18 Total)

The Pros and Cons are obvious for this unit.
Pro: It is a very accurate fuel gauge.
Con: One must have access to a 3D printer. One must know or at least be familiar with the Arduino programming of the Nano Every circuit board. The process is quite easy even for a beginner but is a bit time consuming to get to know the programming languages for the OLED vs the LED displays that are available. I chose to keep a monochrome OLED in this project for programming and wiring simplicity, but color OLEDs are available and many more interesting graphics can be included rather than my bland simple display. However, the program and number of wires is significantly different for some of these displays. I intentionally kept the number of contacts as low as possible for ease of construction.

Below is the sketch I have created and used successfully:

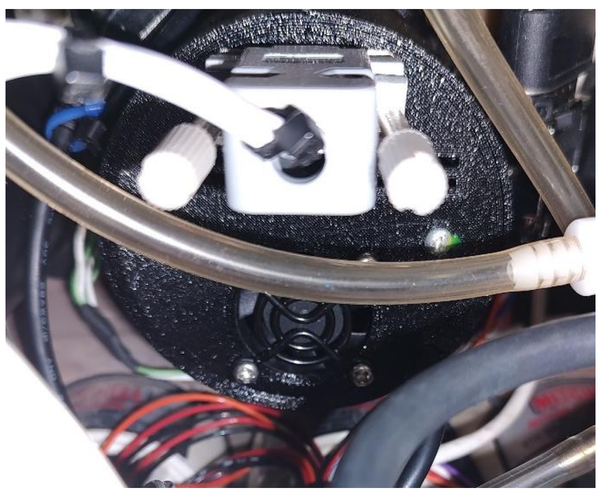
```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <splash.h>
#include <Arduino.h>
#include <U8g2lib.h>

float PotValue = 0;  // Changed to float from int
float PotPin = A1;  //  Changed to float from int
float Data = 0;    // This must be a float value because of potential size and variances.

int DimmerValue = 0;  //This is just an int as it is either high or low as far as it is needed only to set contrast of the display.
int DimmerPin = A0;  // A0 will be high for bright and low for dim
//  Display 1.5 inch uses a 1306 library and the items are defined below:
U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ SCL, /* data=*/ SDA, /* reset=*/ U8X8_PIN_NONE);   // All Boards
without Reset of the Display

#ifdef U8X8_HAVE_HW_I2C
#include <Wire.h>
#endif

void setup() {
  u8g2.begin();
  u8g2.enableUTF8Print();  // enable UTF8 support for the Arduino print() function
}

void loop() {

  u8g2.setFont(u8g2_font_6x13B_tr);  //  This is all print stuff to the 1.5 inch display of a header.
  u8g2.setFontDirection(0);
  u8g2.clearBuffer();
  u8g2.setCursor(6, 10);
  u8g2.print("N12AY Fuel Quantity");  //Header

  u8g2.setFont(u8g2_font_7Segments_26x42_mn);
  u8g2.setCursor(30, 55);
  DimmerValue = analogRead(DimmerPin); //This takes the relay voltage of 3.3 volts to define A0 as high or low a pull up resistor may be
necessary.
  if (DimmerValue >= 310 && DimmerValue <= 1023) {  //change due to use of 3.3 volt input. 1023/3.3=310
  u8g2.setContrast(0);  //Dims display
  }
  if (DimmerValue >= 0 && DimmerValue <= 309) {  //  change to 309 as 0-309 should be low enough and display as bright.

  PotValue = analogRead(PotPin);
  Data = (PotValue * (5.00 / 1023.));


if (Data >= 4.18 && Data <= 5.0) {
    u8g2.print("15");  //this can be reconfigured to say FULL  See 1306 print directions
  } else if (Data >= 4.07 && Data <= 4.17) {
    u8g2.print("14 ");
  } else if (Data >= 3.90 && Data <= 4.06) {
    u8g2.print("13 ");
  } else if (Data >= 3.69 && Data <= 3.89) {
    u8g2.print("12 ");
  } else if (Data >= 3.46 && Data <= 3.68) {
    u8g2.print("11 ");
  } else if (Data >= 3.25 && Data <= 3.45) {
    u8g2.print("10 ");
  } else if (Data >= 2.93 && Data <= 3.24) {
    u8g2.print(" 9");   // Space added pior to digit for centering with double digits
  } else if (Data >= 2.50 && Data <= 2.92) {
    u8g2.print(" 8");
```

```
 } else if (Data >= 2.15 && Data <= 2.49) {
   u8g2.print(" 7");
 } else if (Data >= 1.70 && Data <= 2.14) {
   u8g2.print(" 6");
 } else if (Data >= 1.25 && Data <= 1.69) {
   u8g2.print(" 5");
 } else if (Data >= 1.01 && Data <= 1.24) {
   u8g2.print(" 4");
 } else if (Data >= 0.96 && Data <= 1.00) {
   u8g2.print(" 3");
 } else if (Data >= 0.74 && Data <= 0.95) {
   u8g2.print(" 2");
 } else if (Data >= 0.44 && Data <= 0.73) {
   u8g2.print("1");  //The 1 digit is too far left with space so none added.
 } else if (Data >= 0.00 && Data <= 0.43) {
   u8g2.print(" 0");  //This can be reconfigured to print EMPTY also
 }
 delay(1000);
 u8g2.sendBuffer();
}
```

The final installation photos are below:
The gauge and sketch above makes the display unit accurate within ½ gallon as measured by both the totalizer and the markings on the tank as well as quantity filled using AVGAS 100LL.

Note the addition of a red button. This button has nothing to do with the gauge, it is a warning test for the stall warning gauge and is yet to be labeled. The 3 1/8 inch gauge allows enough room for a voltage regulator as well as a Carbon Monoxide detector circuit. The two USBs take up some room but are convenient for the GRT Mini GA (soon to be upgraded hopefully to the new GRT Mini).

The gauge face had sufficient room for both the upper USB A for the Mini GA, and the Arduino circuit board update plug. The back of the instrument fits quite nicely in an overcrowded panel and the relatively short length allows easy access even if removal is needed. A single 9 pin D Sub is the only connection



Of course, one must have made back side of the panel access available to easily remove anything or you must remove the entire panel. Frankly once the unit is in after bench testing, it has been quite reliable needing no maintenance.

In Summary:
I found this to be a challenging project as I had to learn how to do CAD (Computer Aided Design) which was weeks of short videos and learning the tricks of the trade of Fusion 360 by Autodesk but I enjoyed the task and now make many things for my shop, car, aircraft and home.

I don't consider myself an expert in electronics or wiring, but I have excellent soldering skills and am quite quick at learning new things (or at least the things I want or have to learn such as programming). Designing circuits is not difficult and one can see I used simple cheap off the shelf circuit perf boards and simply wired from point to point soldering each connection as neatly as I could. It all worked flawlessly.

I found that using my 3D printer was boringly slow, so again, I took the time to learn how to speed up my prints. Printing all 4 components now only takes about 4 hours to print. I started with an inexpensive small printer and before long I bought another larger FDM printer and enclosed it for more rugged parts using nylon, carbon fiber and flexible filaments needed for other projects. None of which is necessary for this project.

For increasing reliability and maintainability, nearly every screw is backed by a threaded heat insert. So, I had to learn and procure these heat threaded inserts and then I also built myself a heat insert press. Frankly, a soldering iron works as well to insert these, but one can never leave it simple. I just can't help making new tools. It was fun and only $20 to now have a rugged and accurate heat insert tool. Heat inserts, wire, and plug connectors cost was another $20 for each . I figure the final cost was $60 for all hardware but that does not include the learning curve.

The actual cost for my unit and all the hardware was about $200, of course I have $600 in 3D printers, and my CAD program was freeware courtesy of Autodesk Fusion 360. New hardware like heat inserts, D subs, pins, sockets etc. cost about another $100 if you have no stock.

If you have electrical D subs from your panel build, any CAD experience, and access to a 3D printer you can of course do it cheaper than anyone starting from scratch learning all things cold and purchasing items new. Email me and I'll send you .STL files if you are going to do the 3 1/8 inch instrument.

Of course, during my experimentation after finishing my first unit, my old Princeton Capacitance probe began to fail. Fuel was leaking into the actual unit. The Princeton Fuel Probe company was sold off to RED which made a two part sender that was more reliable for about the same cost. It cost another $300 for a new RED fuel sender. I recalibrated it and now it is all working fine again. But more importantly, it has proven to be reliable, easy to read and more importantly accurate.

Installation of the RED Capacitance Sender proceeded well. The gauge and sender unit itself seems to be accurate to within 1 gallon of the totalizer and visually. If I filled the tank all the way into the neck with fuel, the fuel in the tank stays obviously stays above the capacitance probe. The extra fuel in the neck about 1.3 gallons limits the fuel sender to read full (15 gallons on my gauge). The sender only begins reading accurately when the level of fuel at the sender drops down about ½ inch. Therefore the level must drop down to about the 14 gallon mark before the sender begins sending accurately. If filled to just below the top of the sender, it begins and stays accurate. And if saturated with my tank filled well into the neck, accuracy below 14 gallons is preserved.

The only teething issue was the Dimming Circuit of the gauge. The unit was erratic dimming at times. The issue is the floating pin to the A0 lead would pick up spurious voltage. All that was needed was a pull-down resistor to force the input to be steady as the noise in the aircraft and Arduino circuit is causing the pin value to float.

In testing I removed to monitor internal circuit tests and I upgraded the fan to a 12 volt fan to draw current from the aircraft rather than the Arduino Nano Every board. Now the board only powers the OLED display and that is only millivolts/amps. With the 12 volt fan the components will last much longer in the Florida heat if I keep them cool.

To use a normal Arduino pull up command the dimmer programming can be changed thusly:

```
  DimmerValue = analogRead(DimmerPin); //This takes the relay voltage of 3.3 volts to define A0 as high or low a pull up resistor may be necessary.
  if (DimmerValue >= 310 && DimmerValue <= 1023) {  //change due to use of 3.3 volt input. 1023/3.3=310
  u8g2.setContrast(0);  //Dims display
  }
  if (DimmerValue >= 0 && DimmerValue <= 309) {  //  change to 309 as 0-309 should be low enough and display as bright.
U8g2.setContrast(255);  // Full bright display contrast
}
```

Now keep in mind this programming change is to use the Arduino built in pullup circuit however in my opinion it is best suited to use a pull down resistor rather than rework the sketch to allow automatic pull up, thereby making troubleshooting easier to see and troubleshoot with a volt meter. A programming expert would simplify the dimmer program to with a pull down to:

```
  DimmerValue = analogRead(DimmerPin); //This takes the relay voltage of 3.3 volts to define A0 as high while the pull down resistor simply holds the DimmerPin LO or to ground.
   if (DimmerValue >= 300 && DimmerValue <= 310) {  //change due to use of 3.3 volt input. 1023/3.3=310
  u8g2.setContrast(0);  //Dims display
 }
 if (DimmerValue <= 0 && DimmerValue <= 100) {
u8g2.setContrast(255);  //  Display is bright
}
```

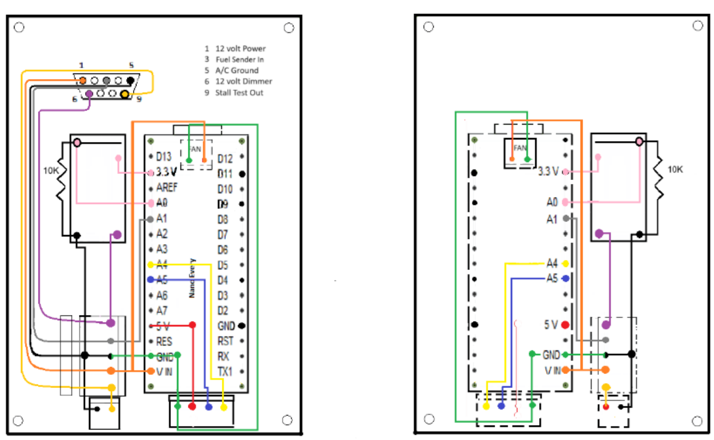
Bench testing showed the above worked well.

Final board wiring is shown below. Ops checked OK March and April 2025.

Simplified not regulated Arduino

Pull Down Resistor Added 3/2025

12volt fan added



This can be easily put onto a printed circuit board layout as the runs are now linear and for the most part do not cross.

Summary:
Rarely do I find electronic experiments work the first time. However, other than my dimming float issues, the unit has proven to be very accurate.

I have a habit of spending a minimum of 2 hours a day on learning a new skill. Arduino programming was quite simple. Yes I did feel like a monkey with a typewriter doing programming. However, there are many videos, and documents on how to do all the programming that is necessary. I cut and pasted bits of programming from others and left the actual programming to others which have made macros that took care of all the OLED display learning curve. Many thanks to the Arduino community for their time and valued input.

I will eventually put the STL files for 3D printing the 3 1/8 inch gauge. This will allow a DIY project to simply find someone with a 3D printer and a few hours of time to fabricate the parts. Heat inserts aren't necessary but make the unit solid and durable. One day I will make a 2 ¼ inch instrument. The 2 ¼ inch instrument will be different in the OLED will obviously be smaller and the USB will be only for the Arduino updates if required. The board will be smaller so I expect I will wire directly from the Arduino board to the D sub and then OLED. The USB may not be on the face but accessed through the rear of the instrument or a hole in the front directly to the board. Much measuring and tinkering will be required to get it to fit with off the shelf components.

Luckily the Arduino Nano Every is well documented and simple plug in "Perf Boards" can be used with simple hook up wire to make the circuit board. I had an operational circuit in less than two hours. I expect the 2 ¼ inch board to take no longer. Could this be made as a rectangular fuel gauge hobbs meter as well? Yes it can. Adding a time logging circuit to the Arduino or pressure sensor is just as easy.

Time is not on my side as I have more projects than life expectancy, but I love a challenge. I found this to be a fun and rewarding project.

I highly recommend this DIY project as a learning curve. For those of you with boats, it will work as well using Diesel or Mogas. The gauge will work with any capacitance or float sender, so it will even work in your kit car.

Bud Yerly
Custom Flight Creations, Inc.