Display:

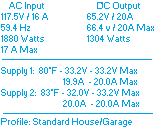
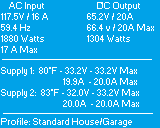
* AC voltage
* AC Amperage
* AC frequency
* Max Allowed AC Amps
* Max Allowed DC Amps
* Power supply 1 Temp
* Power supply 2 Temp
* Power supply 1 Voltage
* Power supply 2 Voltage
* Power supply 1 Max Voltage
* Power supply 2 Max Voltage
* Total DC voltage
* Power supply 1 Amperage
* Power supply 2 Amperage
* Power supply 1 MAX Amperage
* Power supply 2 MAX Amperage
* Total DC Amperage
* Default Setting name
* Any faults that occur:
  + Vehicle BMS stop signal
  + Input voltage drop too low
  + Input voltage too low
  + Input amperage draw too high – too high after trimming pot to min power.
  + Output voltage too low
  + Output voltage too high
  + Output amperage too high – too high after trimming pot to min power.
  + Temp too high – any PS temp too high
  + Unknown Power supply shutdown. – if one cuts off for whatever reason

Goals:

1. Monitor AC input and DC output.
2. Set AC amperage max.
3. Set DC output.
4. Have “profiles” for inputs. (house, chargepoint, etc) current in limit
5. Profile for output??
6. Accept BMS cutoff or min DC amp shutoff.

Sub goals:

1. Read AC voltage
2. Read AC Frequency
3. Read AC amperage
4. Isolate AC/USB/serial
5. Use graphical display
6. Read DC voltage to isolated serial
7. Use digital potentiometer
8. Have digital pot move amperage
9. Setup menu system/display



Arduino input/outputs

Analog inputs:

* AC voltage / Frequency
* AC Amperage
* Power supply 1 Temp
* Power supply 2 Temp
* Power supply 1 Voltage
* Power supply 2 Voltage
* Power supply 1 Amperage
* Power supply 2 Amperage
* Total DC output\*\*

Digital:

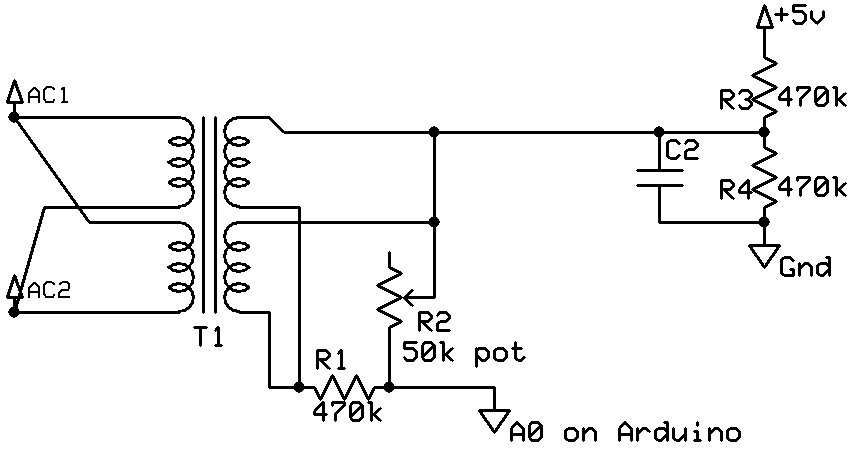
* Button
* Button
* Encoder
* Encoder

Outputs:

* LCD serial out
* Digital potentiometer output
* PS1 power
* PS2 power

Thoughts on Measuring AC voltage:  
coil will isolate and drop AC voltage from 115 to 10v  
10/115 = 0.0869565217391304  
multiply that ratio number by the AC RMS voltage to get the new RMS voltage  
multiply any RMS voltage by 1.414 to get the peak

@100vAC RMS – out 8.695v AC RMS – peak ~ 12.29v  
@260vAC RMS – out 22.60v AC RMS – peak ~31.95v

If worse case of 260v input:  
T1 changes to 31.95 peak voltage signal.  
Voltage divider pot gets tuned down until 32v input = 2.5v peak  
This makes a 2.5 to negative 2.5 wave.  
A second voltage divider circuit , raises Ground by half the supply voltage.  
So the signal is raised 2.5v… changing the final signal from a +2.5 to -2.5 signal into a 5v to 0v signal.  


Tuning the voltage/freq input:

When trimming R2 into range, we first make the Arduino run in a loop for about a second taking the highest ADC number it sees. Then output it from the serial onto the PC for display. Then repeat. This will give a peak voltage reading every second over and over on the screen.  
We can take the RMS voltage of the AC mains given by another meter. Multiply the AC RMS number by 0.0869565217391304 to get the output RMS voltage. Confirm with meter. This should confirm the transformer ratio.  
We then take the transformer RMS power and multiply by 1.414 to get peak voltage.  
The voltage converter should convert 32v peak into 2.5v peak + ground ref of 2.5v making 5v...

1023 ADC reading  
------- = ------------------  
 5 Analog Voltage Measured  
https://learn.sparkfun.com/tutorials/analog-to-digital-conversion/relating-adc-value-to-voltage

Measuring current:

From http://openenergymonitor.org/emon/buildingblocks/ct-sensors-interface

Max current to sense is 60A

Convert RMS current to Peak Current  
60\*1.414=84.84A -- 84.84 amps peak current.

Divide the peak-current by the number of turns in the CT to give the peak-current in the secondary coil.The CT has 2000 turns, so the secondary peak current will be:   
  
Secondary peak-current = Primary peak-current / no. of turns  
84.84 / 2000 = 0.04242amps

To maximize measurement resolution, the voltage across the burden resistor at peak-current should be equal to one-half of the Arduino analog reference voltage. (AREF / 2) Arduino running at 5V: AREF / 2 will be 2.5 Volts. So the ideal burden resistance will be:  
Ideal burden resistance = (AREF/2) / Secondary peak-current  
2.5/0.04242=58.93 ohms  
58 Ω is not a common resistor value. The nearest common value is 57.6 Ω . Always choose the smaller value, or the maximum load current will create a voltage higher than AREF. In some cases, using 2 resistors in series will be closer to the ideal burden value. The further from ideal the value is, the lower the accuracy will be.  
Here are the same calculations as above in a more compact form:  
Burden Resistor (ohms) = (AREF \* CT TURNS) / (2√2 \* max primary current)

Reading DC voltages:  
<https://learn.adafruit.com/adafruit-4-channel-adc-breakouts>  
  
We will use ADS1115 chips to read the differential signals of each power supply… and use them with internal amplifier for the DC shunt