Measuring battery voltage in real-time

To measure the battery voltage, the voltage must be split off from the main line, divided to a smaller range, then converted to digital and communicated to the RPI through I2C protocol. This requires a voltage divider (two resistors, R1 and R2, as depicted in the image below), and an ADC (analog to digital converter). For the ADC, I chose a common one that’s supposed to work well with RPI over I2C protocol: ADS1115 (just the chip alone is too difficult to solder, so I’ve purchased a breakout board). The ADS1115 has a programmable gain amplifier (PGA) that allows us to adjust the range up to +/-6.144V. However, it can only actually measure as high as the supply voltage, which from the buck converter is about 4.8 V. So we’ll need to set the PGA to the high value (as opposed default of +/-2.048V), and step down the battery voltage from 12.6V to 4.8V at max charge. The ADS1115 also requires a minimum current of 124 micro-Amps. The formula for a voltage divider in this formation (see figure below) is

V\_out = R2/(R1+R2)V\_battery

We’re looking for a ratio of about 0.38. Rearranging this, we find that we need R2 = 0.613\*R1 or R1 = 1.63\*R2. Now to pick R1, we need to hit three criteria:

1. Provide sufficient current to the ADC. I=V/R, and we need at least 124 micro-Amps, so R<12.6/124e-6 = 101.6 kOhms. So this is the maximum total resistance we’re allowed, where R = R1 + R2.
2. Minimize power loss. I = V/R, P = IV = V^2/R. Therefore, we want to maximize R1, so we should be looking somewhere close to R1 = 63kOhms and R2 = 38.8kOhms
3. Pick convenient standard resistors. We had 10kOhm, 13.3kOhm, 2kOhm and 1kOhm resistors on hand. Letting **R1 = 16.3kOhm and R2 = 10kOhm**, we get a ratio of 0.38, a current of 540 micro-Amps, and a power loss of 0.007 Watts, which is pretty small. These resistors are also rated to 1/4W, so they can handle the power. The actual resistance values are slightly different: 16.333 and 9.98. These were measured empirically and adjust in the code as needed.

The power calculations suggest that there is so little current running through this new branch of the circuit, that a fairly small wire gauge can be used. Initial circuit test confirms that with the thin wire I’ve been using for RPI power and PWM signals is sufficient for this circuit, and power loss appears to be minimal.

There are other ways to set this up that would drain less power, but this was the easiest to implement.

A paper with a diagram

Description automatically generated

The ADS1115 is configured in the Sensors.py init function, and its data is read during each call to the Sensors.py getState function.