From: **John Sullivan** qimikom@gmail.com
Subject: Battery Current Sensor Schematic
Date: September 9, 2019 at 10:57 PM
To: Peter Perkins 150mpg@gmail.com

JS

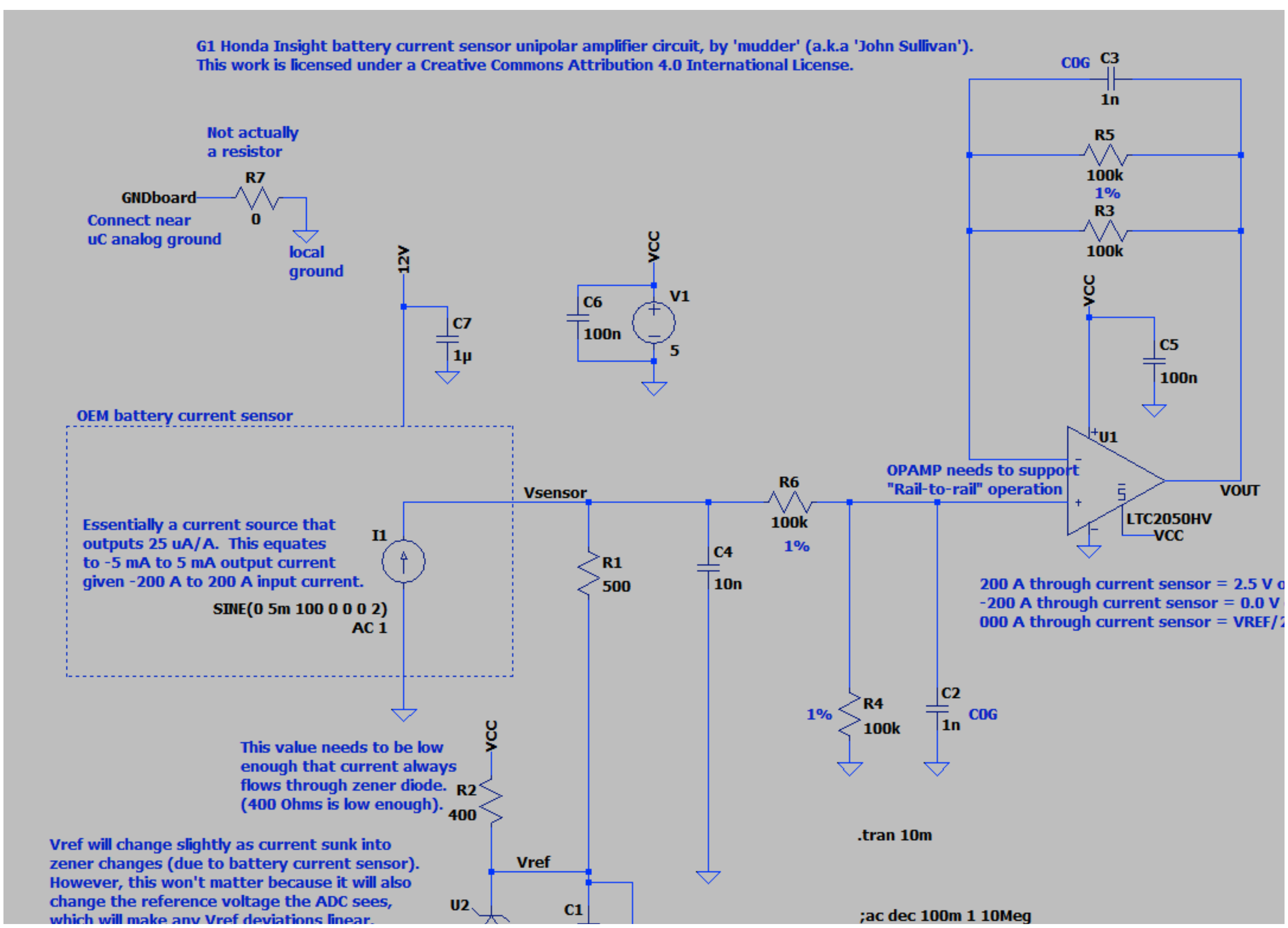
Hey Peter,
I used to have Isaac's email, but can't find it now... please forward to him (and read for yourself if you're interested).

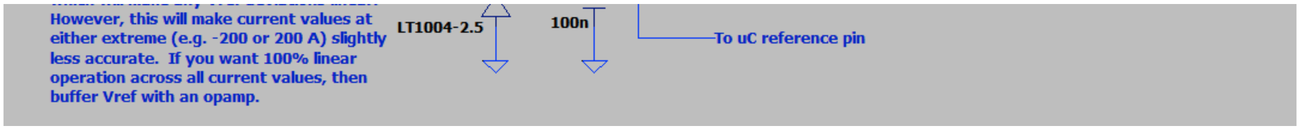
...

Hey Isaac,
I just worked out this current sensor schematic, which let's us use the OEM battery current sensor without having to generate any negative rails... instead of sinking to the ground reference, we're now sinking to the voltage reference itself, which means 0 A flowing from the battery will output exactly half whatever the voltage reference is. This circuit is setup to handle battery currents from -200 A to 200 A. If you want to lower the current range, just increase R1. If you do that, you can also increase R2 (to reduce current wasted as heat through the zener diode), but note R2 must always be less than R1 (to ensure the zener diode is always reverse biased and sinking current). It's very important that current always flows through the zener diode! You'll want to use a zener that can handle 15 mA (e.g. the LT1004-2.5 in SOIC-8).

Note that the voltage across a zener diode varies slightly based on the current sunk through it. However, in this case the zener reference voltage will only change 0.08% across the entire battery current sensor range (-200 A to 200A), which effectively means there's no change at all... particularly due to the much greater change caused over temperature changes. If for some reason you don't want the zener reference to change at all, you could buffer it with an opamp (but that would be overkill).

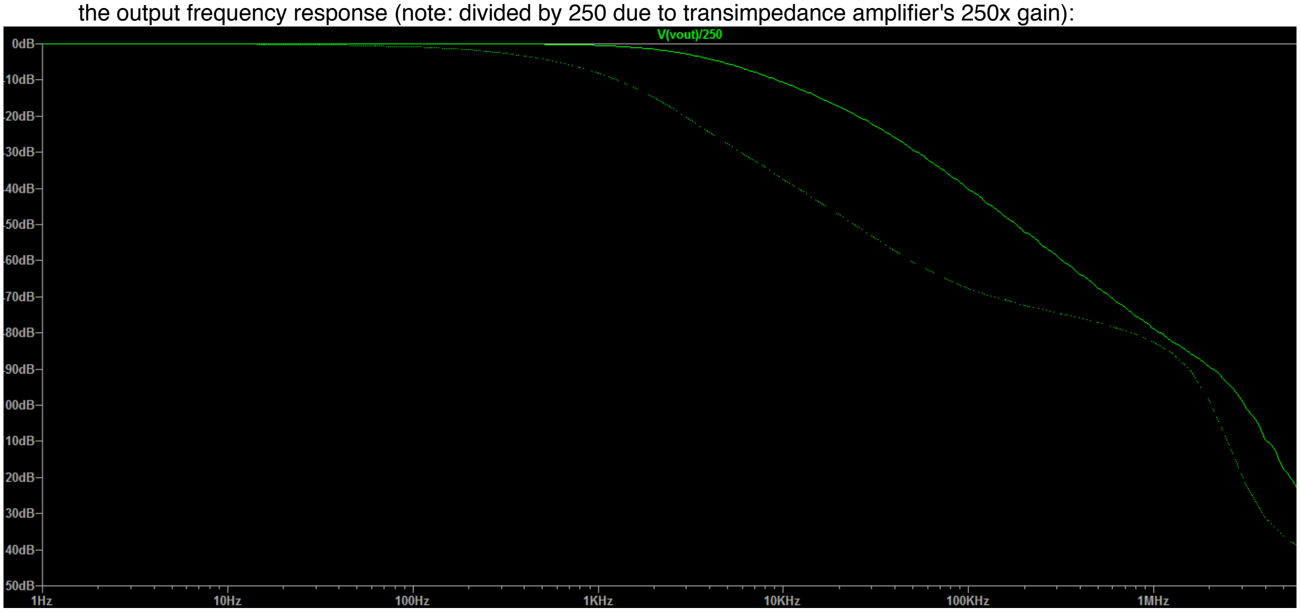
You'll connect the current sensor's positive lead to 12 V, and the negative lead to ground. I recommend creating a local ground on the PCB for all these grounds (see R7). Since the microcontroller's ADC also uses the zener reference voltage, I recommend connecting the two ground pins close to the analog ground pins on that IC (or, if there aren't dedicated analog ground reference pins, then to any ground pin at the chip. Note that the correct schematic symbol to use to create a dedicated ground net is called an ILTC... it's basically a solid bit of copper that has two different nets... that way when you flood the board ground, it won't flood into the analog ground shown in the schematic.





Here's a representative output showing how well the opamp's output voltage matches the current sensor's current output (converted to a voltage across R1). Note that the phase shift is due to lowpass filter... these waveforms are basically identical, which just a slight phase shift.

Here's the frequency response, which shows that we have ~50 degrees phase margin, and ~10 dB gain margin. C4 creates a primary LPF with a ~30 kHz cutoff frequency, and R6/R4/R3/R5 create a buffered LPF with ~3 kHz cutoff frequency. If that's too low, you can use smaller R6/R4/R3/R5 values (all four values should remain identical). Here's the output frequency response (note: divided by 250 due to transimpedance amplifier's 250x gain):



Note the opamp MUST support rail-to-rail operation if you want to be able to measure full scale current values... otherwise, the opamp output will trail off near the maximum battery current values (particularly when approaching 0V). The LTC2050HV is an excellent opamp for this application because it supports rail-to-rail and has very low input offset voltage.

Note that I haven't tested this circuit, but I did model it a whole bunch and it seems generally well behaved.

...

I strongly encourage you to build this circuit on a breadboard, as I haven't tested it in the real world. However, I did model it a whole bunch and it seems extremely well behaved.

-John