**Programmable Current Controller - Qwiic Compatible**

**By Virtual\_Ground**

A black circuit board with white text

Description automatically generated

What is it?

A simple way to precisely control current with Arduino/RPi/ESP32.

What is included?

* Current Controller PCB
* Phoenix male screw terminal – pairs with the terminal on the PCB.
* IRLP3034 – N-Channel MOSFET with a low gate threshold and high power TO-247 package. Comes unmounted for easier shipping - must be soldered to the PCB. It is recommended that this is connected to a heat sink (not included).

How It Works?

The current controller is composed of the following key parts:

* AD5693 - 16-Bit I2C DAC
* AD8655 - Low Noise, Precision CMOS Amplifier
* IRLP3034 – N-Channel MOSFET
* 1 Ohm Shunt Resistor

The DAC outputs a voltage from 0-950mV, as set by the master over i2c. The op amp compares this voltage to the voltage at the shunt (which reflects the actual current through the load). The op amp then controls the gate of the MOSFET in order to equalise these two voltages. The MOSFET is operated in it’s linear region. Thus, a feedback loop is created, where the current through the shunt is essentially determined by the output voltage of the DAC.

One advantage of this setup is that the load current is independent of VCC and the resistance of the load, which for some loads (such as LEDs/Laser diodes) changes as the device heats up leading to thermal runaway.

VCC

A diagram of a circuit

Description automatically generated

SHUNT

LOAD

Test Points

Some test points are included on the PCB. These can be used during calibration, or to measure actual current across the shunt resistor.

**DAC:** Raw DAC output. This can be anywhere from 0-950mV.

**GATE:** This is the voltage that the op amp is applying to the gate of the MOSFET. Ranges from 0-3.3V. Normally hovers around the 2V mark.

**SENSE:** Voltage across the 1 Ohm shunt. For 100mA, this voltage will read 100mV.

Wiring Diagram

A circuit board with wires connected to it

AI-generated content may be incorrect.

You can use either qwiic socket to connect to the master device. The other socket is optional and can used to daisy chain current controllers or another i2c device.

The green Phoenix terminal connects to the load and power supply as above. Note that since the device uses an N-channel MOSFET, it is sinking current to ground rather than sourcing it from VCC.

Important Notes

Since the MOSFET is essentially a variable resistor in this setup, it dissipates heat. Power dissipated depends on the load resistance and VCC.

Recommendation: For higher currents, mount the TO-247 MOSFET to a suitable heatsink to keep it cool. If driving a 1.8V LED, for example, using a VCC of 9V will cause more heat to be dissipated than if VCC was 5V. That is because the MOSFET has to burn-off/drop this excess voltage as heat. So try to somewhat match load to VCC. For small currents this is less important.

When the DAC is set to 0, some tiny current (<0.5mA ) still flows. This may cause some LEDs to glow very very dimly even when the current is set to 0.

Recommendation: There is space for a 0805 resistor to be soldered on the underside of the PCB (see R7, located under the green Phoenix terminal). Soldering a resistor here can divert some of this current and offset the 0 reading (Note you may need to recalibrate in software)

Imagine the DAC is set to 100mV (~100mA), but the load is not connected or there is no VCC power source. The feedback loop will see that no current is flowing and will fully turn the MOSFET on. Suddenly re-connecting the load or VCC in this instance will cause a rush of current as it takes time for the feedback loop to react (several milliseconds)

Recommendation: Try not to disconnect/reconnect the load or power supply to the load while the device is operational. The load may experience a short spike of current as a result of hot-plugging.