EM Arduino 4-20mA Shield Documentation

Version 1.5.0

Erdos Miller

October 22, 2014

1 Contents

1 Power

The Arduino 4-20mA board can either be powered externally by 24V or from the Arduino. When powering the board from 24V externally, the board has the capability of supplying the 4-20mA sensors with 24V. The shield has an on-board voltage regulator to supply its own chips with 5V. In addition, the shield can also supply the Arduino board with 5V, if the **EPWR** jumper is in place. If the **EPWR** jumper is not on the board, the Arduino must still be powered (from some other power source). This allows for some flexibility in powering the shield, Arduino, and sensors.

EPWR: this jumper is used to connect the Arduino's 5V supply with the shield's 5V supply. You can have the Arduino powered externally with this jumper in place, it won't hurt anything
VIN: 10V-26V recommended. You can use this to power the shield, the Arduino, and the sensors without extra wiring.
GND: Ground. This is shared with the Arduino's ground

2 Connecting Sensors

The following descriptions assume that the shield is supplied with 24V on the VIN terminal.

Type 2 Sensor (2-Wire) Hookup:

V: sensor power, the 4-20mA shield will provide VIN to the sensor here
+: sensor return, make sure that the ground jumper (Gn) is in place

Type 3 Sensor (3-Wire) Hookup:

V: sensor power, the 4-20mA shield will provide VIN to the sensor here
+: sensor signal, make sure that the ground jumper (Gn) is in place
G: sensor ground

Type 4 Sensor (4-Wire) Hookup:

V: sensor power, the 4-20mA shield will provide VIN to the sensor here
+: sensor signal +, make sure that the ground jumper (Gn) is REMOVED
-: sensor signal -
G: sensor ground

3 Scaling ADC Readings to Current in mA

The Arduino 4-20mA shield will provide a raw ADC reading to the Arduino. This raw value then has to be scaled in software to provide a current reading in milliamps. The following equations demonstrate how to do this in software. Note that all of the equations have a multiply by 1000 at the end, this is to scale the reading from amps to milliamps. The resistor sense constant will variety depends if the shields is connected to a 5.0V or 3.3V Arduino. Make sure to remove jumpers (3M1, 3M2, 3M3 and 3M4) if the 5.0V Arduino is desired to use.

Internal and External (ADS1115) ADC Transfer Function

$$I_{4-20mA} = (ADC_{raw}) \frac{(ADC_{fullscale}) * (1000)}{(ADC_{resolution}) * (AMP_{gain}) * (R_{sense})}$$

$AMP_{gain} = 2$

Internal ADC Transfer Function for 5.0V Arduino

$ADC_{resolution} = 1023\ bits$

$ADC_{fullscale} = 5.0\ V$

$R_{sense} = 120\ \Omega$ remove (3M1, 3M2, 3M3 and 3M4) jumpers

$$I_{4-20mA} = \frac{(ADC_{raw})}{49.152}$$

Internal Constant = 49.152

Internal ADC Transfer Function for 3.3V Arduino

$ADC_{resolution} = 1023\ bits$

$ADC_{fullscale} = 3.3\ V$

$R_{sense} = 80\ \Omega$ place (3M1, 3M2, 3M3 and 3M4) jumpers

$$I_{4-20mA} = \frac{(ADC_{raw})}{49.6}$$

Internal Constant= 49.6

External ADC (ADS1115) Transfer Function

$ADC_{resolution} = 32768\ bits$

$ADC_{fullscale} = 6.144\ V$

$R_{sense} = 120\ \Omega$ using 5.0V Arduino Board

$R_{sense} = 80\ \Omega$ using 3.3V Arduino Board

$$I_{4-20mA} = \frac{(ADC_{raw})}{853.33} \quad \text{for } R_{sense} = 80\ \Omega$$

$$I_{4-20mA} = \frac{(ADC_{raw})}{1280} \quad \text{for } R_{sense} = 120\ \Omega$$

External Constant = 853.33 using 3.3V Arduino Board

External Constant = 1280 using 5.0V Arduino Board

Note: ADC $_{\text{full scale}}$ can be changed in software, it is +/-6.144 volts by default.

4 Using with a 3.3V Arduino

The Arduino 4-20mA shield was originally designed for a 5V Arduino. There are a couple different ways that it's possible to use it with a 3.3V Arduino. There are two factors that affect this type of operation. First, the shield's on-board regulator outputs 5V to supply the Arduino (optional) and the shield's op-amps. Second, the op-amps have been configured to output 4.8V at a full-scale reading of 20mA. This voltage would be too high to feed into the Arduino's ADCs but work just fine when the shield's on-board ADC is used. Below we describe two ways in which you can use your 3.3V Arduino with the shield. But first, this is why we need to make some modifications to the shield.

Conditions to Avoid (these can damage your Arduino)

1. The shield must not supply 5V to the Arduino. If the EPWR jumper is on the shield, do not apply power to the VIN terminal. If you want to power the shield from VIN, remove the EPWR jumper.
2. Do not allow the shields analog outputs to exceed 3.3V. This means that the shield input (sensor output) current must not exceed 13.75mA.

The Recommended Way

We need to reduce the gain on the shield's op-amps so that they do not output more than 3.3V at a full-scale reading. To do this you have to replace the sense resistors (R5, R11, R17, R23) with 80.6 ohm resistors (e.g. CRCW080580R6FKEA available from Digikey). This will give you 3.224V output with a 20mA input giving you just enough head-room to keep the Arduino's ADC from saturating. Next you need to change the output of the shield's regulator to 3.3V, this can be accomplished by replacing R26 with a 1.58k resistor (e.g. CRCW08051K58FKEA available from Digikey). The resistors should have a 0805 footprint.

The Simple Way (No soldering)

We have to be careful to avoid two situations, supplying the Arduino’s 3.3V power supply with 5V from the shield, and applying more than 3.3V to any of the Arduino’s ADC pin. The first condition is easy to avoid by simply removing the EPWR jumper, the second is a little harder. The Arduino’s input pins are all diode protected, unfortunately the shield’s op-amps are capable of supplying 25mA. This is unacceptably high to allow. It is safer to cut all analog pins (A0, A1, A2, A3) from the shield. This will disconnect the analog outputs from the shield. This will force you to use the shields I2C ADC instead of the Arduino’s ADC. This still leaves two 4.7k pull-up resistors to 5V on the shield. This is quite alright since the Arduino’s internal protection diodes clamp the SDA and SCL voltages to a little more than 3.3V, the resistors limit the clamping current to about 276uA.