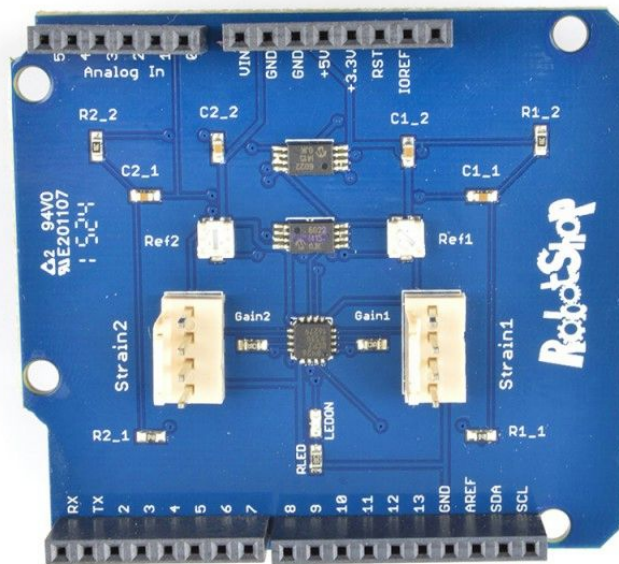


Strain Gauge / Instrument Amplifier Shield



Revision 1.2 January 2016

Table of Contents

[1. Overview](#)

[1.1 Pinout](#)

[2. Strain Gauges](#)

[2.1 Strain Gauge Characteristics](#)

[2.2 Wheatstone bridges](#)

[3. Circuit Building blocks](#)

[3.1 Strain Gauge Inputs via Wheatstone bridges](#)

[3.2 Instrumentation Amplifier](#)

[3.3 Filter](#)

[3.4 ADC](#)

[4. Arduino Sample Code / Calibration](#)

[5. Support / Warranty](#)

1. Overview

The Strain Gauge / Instrument Amplifier shield (SGS from here on further) is intended for precise amplification of measurements specifically for bridge amplifiers, medical instrumentation and industrial process control. The shield is intended to be used with normal shield-compatible Arduino-based microcontroller boards such as the Arduino Uno.

Instrumentation signals of small values, in the order of mV (millivolts) or less, need to be amplified for processing. Prototyping boards like the Arduino Uno normally have a 10 bit ADC which means that the resolution of input/outputs are 4.9mV for a 5V (will be different for 3.3V or other referencing voltages) supply. Therefore changes less than 4.9mV cannot be easily recognized by the process board without the necessary amplification and possible noise filtering.

The AD8426 was selected to be used as the amplifier for the shield. The gain produced by the AD8426 amplifier ranges from 1 to 1000 depending on the RGAIN resistor value. Another valuable feature is the voltage reference adjustment on board to adjust the output voltage reference in order to be used with a single power supply ADC, which the Arduino makes use of.

For filtering, a second order Bessel low pass filter for 1k Hz was designed to help smooth out unwanted interference.

This shield is known to work with the Arduino UNO/Duemilanove boards. For all other shield-compatible boards, please consult the pinout / schematic.

1.1 Pinout

The SGS uses the following pins:

- A0: Strain 1 input
- A1: Strain 2 input
- 5V, 3.3V and GND pins are used to power the shield

The white connectors on the board are 4 pin Molex connectors with 0.1" spacing. Note that a standard 0.1" spaced connector can be used to connect to this header.

2. Strain Gauges

Strain gauges are essentially sensors which produce a very small change in resistance when sensing strain. Strain gauges can be substituted with output characteristics of different sensors for specific applications.

Strain measurements relay the actual strain experienced by a material when a force is applied to it, whether it is a linear, axial, compressive or expansion force. A good educational reference is http://en.wikipedia.org/wiki/Stress%E2%80%93strain_curve which explains the science behind strain in materials.

2.1 Strain Gauge Characteristics

Good explanations on how strain gauges operate and the basic science behind them can be found on the following websites:

- <http://www.ni.com/white-paper/3092/en/>
- http://en.wikipedia.org/wiki/Strain_gauge
- http://en.wikipedia.org/wiki/Load_cell
- <http://www.sensorland.com/HowPage002.html>
- https://www.youtube.com/watch?v=D3J41HE_RMA
- <https://www.youtube.com/watch?v=cP5rs3Ylcbo>

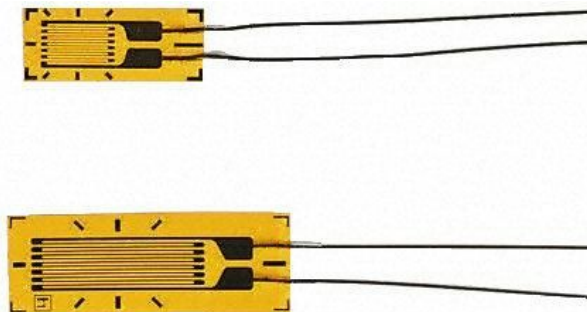


Figure 2: Strain Gauge.

2.2 Wheatstone bridges

A “Wheatstone bridge” is the principal layout for certain sensors which use the electrical properties of the circuit to extract useful information. The SGS without the Wheatstone bridge would only be a differential amplifier with a high sensitivity for weak signals.

There are different configurations of Wheatstone bridges depending on the sensitivity and magnitude of the signal produced by the sensing element(s), which in this case is a strain gauge. The Wheatstone bridge can also adjust for certain initial offset imbalances or for outside factors that influence the sensing element like temperature as shown in the reference links in section 3.1.

The following references explains the setup of the Wheatstone bridge:

- <http://www.hbm.com.pl/pdf/w1569.pdf>
- <http://www.transducertechniques.com/wheatstone-bridge.aspx>

Pin layout and cable colours for many Wheatstone sensors:

Identification	Molex pin number	Normal Cable Colour
VCC	1	Red
S+	2	Green
S-	3	Blue
GND	4	Black

Table 1: Pin connection with cable colours of pigtails

3. Circuit Building blocks

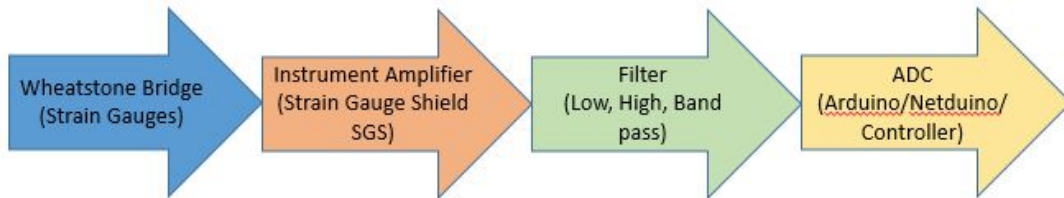


Figure 3: High-level block diagram

The following sub-paragraphs will explain more about the different building blocks.

3.1 Strain Gauge Inputs via Wheatstone bridges

See chapter 2 for more detail.

3.2 Instrumentation Amplifier

The instrumentation amplifier used, AD8426, is a differential amplifier. The datasheet will show all the electronic characteristics of this amplifier. There are two changes per channel that can be adjusted for the specific use of the SGS, which are:

Gain: The gain (amplification per channel) through the amplifier is dependent on the resistor value $R_{GAIN1/2}$ marked as Gain1/Gain2 on the board layout. The gain factor needs to be picked based on the specific application and sensitivity of the total Wheatstone bridge with the strain gauges incorporated. Take note that the gain achieved on the input from the Wheatstone bridge does not exceed the maximum voltage of 3.3V or 5V.

Figure 4 shows the values that $R_{GAIN1/2}$ should be made to get the required gain per application. The standard $R_{GAIN1/2}$ value selected, 100 Ohms for the SGS, is in the general selection range for applications with Wheatstone bridge Strain gauges. Figure 5 shows the actual position of where the $R_{GAIN1/2}$ resistors were placed. If a different gain might be required that is different than the setup gain of 495 with a 100 ohm resistor just replace it with the appropriate resistor.

1% Standard Table Value of R_G	Calculated Gain
49.9 k Ω	1.990
12.4 k Ω	4.984
5.49 k Ω	9.998
2.61 k Ω	19.93
1.00 k Ω	50.40
499 Ω	100.0
249 Ω	199.4
100 Ω	495.0
49.9 Ω	991.0

The AD8426 defaults to $G = 1$ when no gain resistor is used. The tolerance and gain drift of the R_G resistor should be added to the AD8426 specifications to determine the total gain accuracy of the system. When the gain resistor is not used, gain error and gain drift are minimal.

Figure 4. Extract from the AD8426 datasheet

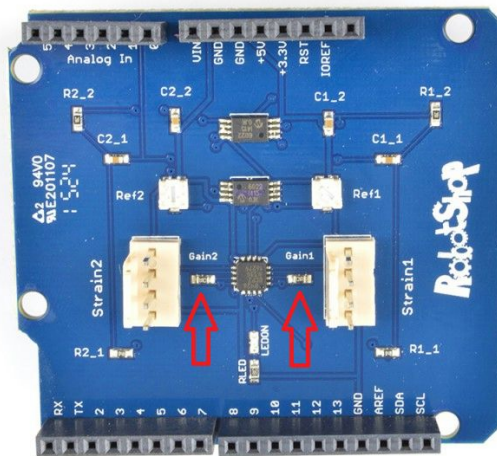


Figure 5: RGAIN positions on SGS

Reference voltage per channel: The reference voltage is used to offset the output signal to a mid-supply voltage in order to be used with a single power supply ADC which is part of the Duino boards in use. Each Strain channel has its own reference voltage to be set with the specific channel potentiometer as shown in Figure 6.

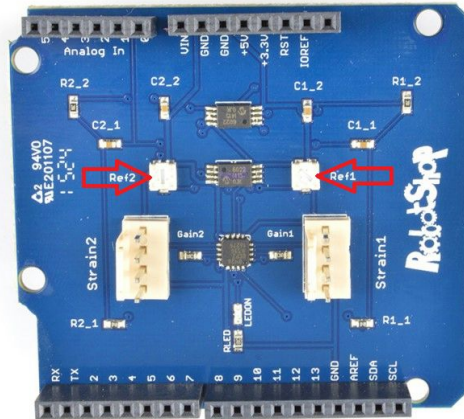


Figure 6: Reference potentiometers positions on SGS.

3.3 Filter

The SGS has been fitted with a low-pass 2nd order Bessel filter at 1000 Hz for both channels after the instrument amplifier before it is read by the Arduino Analog input pins A0 (Strain1) and A1 (Strain2). The filter can easily be adjusted by replacing the resistor values (R1_1, R1_2, R2_1 and R2_2) and capacitor values (C1_1, C1_2, C2_1 and C2_2) for the specific frequency that might be required. Component names R1_1, R1_2, C1_1 and C1_2 are related to channel one (First number of component indicates the channel number it is associated with). Component names R2_1, R2_2, C2_1 and C2_2 related to channel 2.

For ease of reference to adjust the filter one can use the Freeware program FilterLab (<http://filterlab.software.informer.com/2.0/>) to calculate the required values of the 2nd order Bessel filter to cater for the necessary frequencies if the setup 1000 Hz is not sufficient for the user.

It is also possible to change the low-pass filter to a high-pass filter if need be, because the resistors and capacitors can change places (resistors and capacitors are 0603 specified packaging) for easy placement on the PCB.

3.4 ADC

The Arduino board contains a 10 bit analog to digital converter (ADC). It will convert input voltages between 0 and 5 volts into integer values between 0 and 1023. The resolution of the ADC is

5V/1024 units = 4.9mV per unit. it takes 100 microseconds to read an analog input, so the sampling frequency of the ADC is 10kHz.

4. Arduino Sample Code / Calibration

The following Arduino code reads Analog pin 0 (Strain 1) and Analog pin 1 (Strain 2) so the load cells can be calibrated by linear interpolation. Note: These steps are the same for the two outputs.

1. Copy / paste the code below to the Arduino IDE.
2. Upload the code to your Arduino board.
3. Once done, open the Arduino serial monitor.
4. Apply the first known load and note reading.
5. Apply the second known load and note the reading.
6. Edit the values of ReadingA_Strain1 and ReadingB_Strain1 with the taken readings within the code.
7. Edit the values of LoadA_Strain1 and LoadB_Strain1 with the correct load measured in Kg, Lbs ...
8. Upload the code to the Arduino board and retest the known loads.
9. Try an unknown load.

```

// SGS Calibration by linear interpolation for Strain 1 and Strain 2
// Apply two known loads to the Strain Gauge sensor and record the values obtained below
// You can use Strain 1 or Strain 2 or the two Strains at the same time.

float ReadingA_Strain1 = 301.0;
float LoadA_Strain1 = 0.0;           // (Kg,lbs..)
float ReadingB_Strain1 = 302.0;
float LoadB_Strain1 = 80.0;        // (Kg,lbs..)
float ReadingA_Strain2 = 1.0;
float LoadA_Strain2 = 0.0;         // (Kg,lbs..)
float ReadingB_Strain2 = 60.0;
float LoadB_Strain2 = 80.0;        // (Kg,lbs..)
int time_step = 2500 ;             // reading every 2.5s
long time = 0;

void setup() {
  Serial.begin(9600);               // setup serial baudrate
}

void loop() {
  float newReading_Strain1 = analogRead(0); // analog in 0 for Strain 1
  float newReading_Strain2 = analogRead(1); // analog in 1 for Strain 2

  // Calculate load by interpolation
  float load_Strain1 = ((LoadB_Strain1 - LoadA_Strain1)/(ReadingB_Strain1 - ReadingA_Strain1)) *
(newReading_Strain1 - ReadingA_Strain1) + LoadA_Strain1;
  float load_Strain2 = ((LoadB_Strain2 - LoadA_Strain2)/(ReadingB_Strain2 - ReadingA_Strain2)) *
(newReading_Strain2 - ReadingA_Strain2) + LoadA_Strain2;

  // millis returns the number of milliseconds since the board started the current program
  if(millis() > time_step+time) {
    Serial.print("Reading_Strain1 : ");
    Serial.print(newReading_Strain1); // display strain 1 reading
    Serial.print(" Load_Strain1 : ");
    Serial.println(load_Strain1);     // display strain 1 load
    Serial.println('\n');
    Serial.print("Reading_Strain2 : ");
    Serial.print(newReading_Strain2); // display strain 2 reading
    Serial.print(" Load_Strain2 : ");
    Serial.println(load_Strain2);     // display strain 2 load
    Serial.println('\n');
    time = millis();
  }
}

```

5. Support / Warranty

A warranty of 30 days is provided for this product starting from the day the product is received by the customer. Any questions relating to the proper use of this product, or issues encountered should be addressed to supportcenter@robotshop.com