

Using Vernier Sensors with Arduino[®]

Introduction

Have you collected data with sensors for a science class? Most people collect data and accept without question that sensors and software work together to provide relatively accurate readings.

This activity is designed to remove some of the mystery of how sensors work.

Objectives

- Understand how sensors work
- Measure the output from a sensor using an Arduino[®] microcontroller
- Convert the output of the Arduino into a voltage

Preliminary Activity

The Arduino IDE comes with an extensive collection of Example programs, one of which will be used in this activity. View a complete list of available programs by choosing Examples from the File menu within the software.

Materials

SparkFun[®] RedBoard (or equivalent) with USB cable and power supply

Vernier Analog Protoboard Adapter or Vernier Interface Shield

Vernier Gas Pressure Sensor

Computer or Chromebook[™] with Arduino software

Background

There are two ways to connect Vernier sensors to an Arduino board. One method requires simple wiring and a protoboard adapter (Figure 1), while the other involves using a shield (Figure 2) that sits on top of the Arduino board and makes the necessary connections.

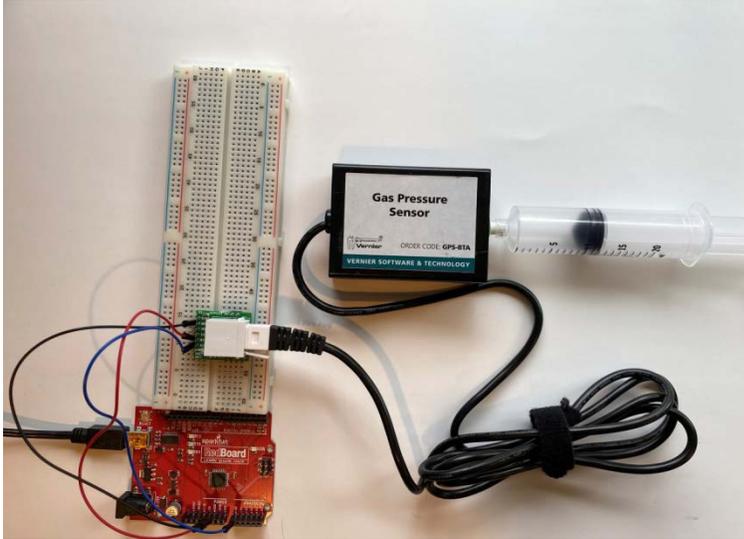


Figure 1 Displayed is a Vernier Protoboard Adapter on a protoboard, wired to an Arduino. The sensor cable plugs into the adapter.

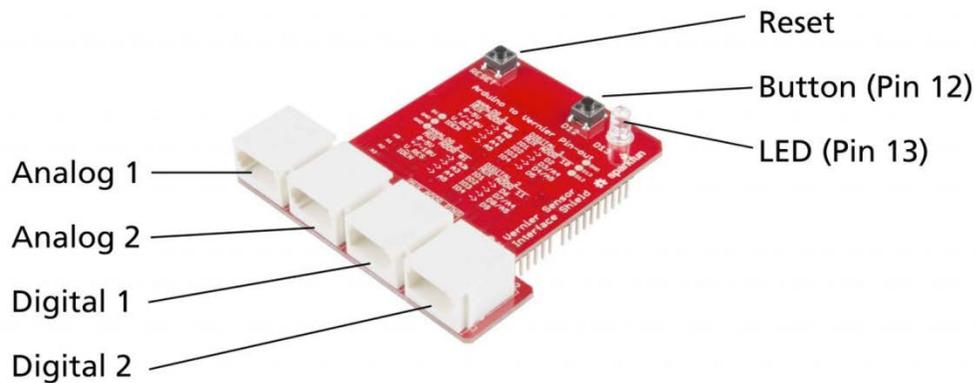


Figure 2 The Vernier Interface Shield pin headers connect to the RedBoard and make the connection between the sensor pins and the RedBoard.

The Vernier Interface Shield has ports to accept up to two analog and two digital devices, a Reset button, a general purpose button (tied to digital pin 12), and an LED. The Reset button and LED are linked to the same components on the RedBoard. When the shield is mounted on the Arduino, the header pins connect the two together.

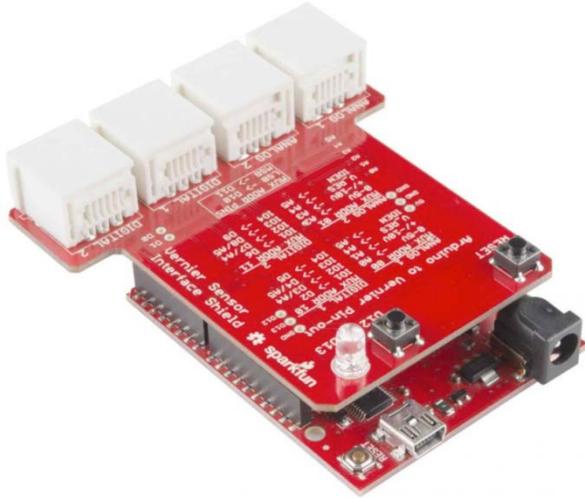


Figure 3 The shield has header pins that connect to the RedBoard.

Procedure

Open the Arduino IDE software, connect your Arduino, and configure your COM port (refer to Introduction to Arduino Programming activity for specific steps). Using either of the two connection methods (as described in the Background section), connect a sensor to the Arduino. If you are using the Protoboard Adapter, refer to Figure 4.

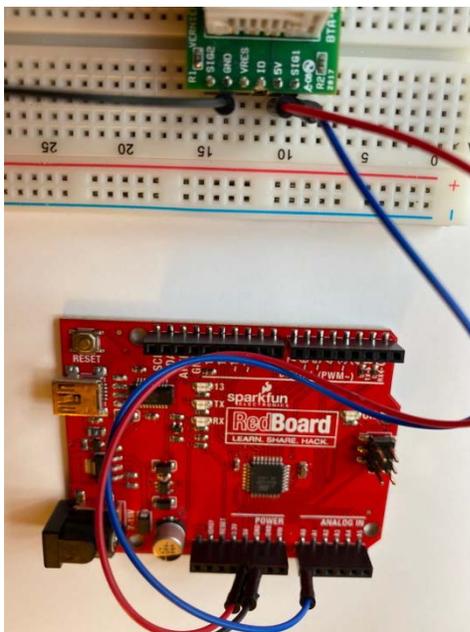


Figure 4 Connecting the Protoboard Adapter pins to the Arduino: 5V, ground, and Signal 1

The following connections are required:

- GND (Vernier BTA pin 2) to Arduino pin GND (ground)
 - 5V (Vernier BTA pin 5) to Arduino pin 5V (power)
 - SIG1 (Vernier BTA pin 6) to Arduino pin A0
1. Set the plunger on the syringe to about 10 mL and attach it to the Gas Pressure Sensor, as shown in Figure 1.
 2. Plug your sensor into the connector and open the sketch AnalogReadSerial.
 3. Open it from the File menu by choosing Examples > Basic > AnalogReadSerial. The entire sketch is listed below:

```

/*
AnalogReadSerial
  Reads an analog input on pin 0, prints the result to the serial monitor.
  Graphical representation is available using serial plotter (Tools > Serial
  Plotter menu)
  Attach the center pin of a potentiometer to pin A0, and the outside pins to
  +5V and ground.

  This example code is in the public domain.
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}

```

This sketch is not specific to Vernier analog sensors. The sensor signal is in the form of a voltage. The Arduino microcontroller converts that (analog) voltage reading to a digital reading between 0 and 1023, and prints it in the Serial Monitor. This is referred to as an analog-to-digital converter (ADC).

Activity 2

The value read by the Arduino board is often referred to as the “count” from the ADC. (Calculate the value of 2^{10} to see where the range of values has its “roots.”) The count is proportional to the voltage signal from the sensor (Vernier sensors typically range from 0 to 5 volts).

Verify and upload the AnalogReadSerial sketch by clicking on the Upload icon (right facing arrow) on the toolbar at the top of the screen. A message at the bottom of the screen will indicate the sketch has compiled and has been uploaded, and the program should run.

Click on the magnifying glass icon at the top right of the toolbar, or select Serial Monitor from the Tools menu, in order to monitor the output from the sensor. The Serial Monitor window will open and display the sensorValue. Notice how the sensorValue changes when you push or pull on the syringe plunger.

Follow these instructions to modify the sketch to display the signal voltage from the sensor instead of the count.

1. Modify the sketch to slow the sampling rate to two times per second. **Note:** The sampling rate is controlled by the delay statement at the end of the sketch. The number in the delay statement is measured in milliseconds.
2. Modify the sketch to calculate the voltage read by the Arduino.
 - a. Declare a variable “sensorVoltage” by typing this code at the top of the loop() function:

```
float sensorVoltage; // declare a variable to store the voltage
```

- b. Insert a line of code after your analogRead(A0) that converts the “count” to a voltage. Type the following line of code:
- ```
sensorVoltage = sensorValue *5.0 / 1023;
```
3. Add comments after this line to explain what the code accomplishes.
  4. Change your sketch to print the voltage instead of the sensorValue in the Serial Monitor.
    - a. Confirm that the sketch operates as expected.
    - b. Save this program with a unique name. **Note:** If you are using the Arduino IDE (instead of the online editor) the software will save it in a folder with the same name.

The line that calculates the sensorVoltage takes the count and divides it by 1023 and then multiplies by 5 volts. The order of operations and the type of data (integer and floating point) cause the calculation to occur in specific ways. Conduct the following investigations, try to understand the differences, and summarize your findings below:

- Multiply the sensorValue by 5 (vs 5.0).
- Change the order of operations in the calculation by first dividing by 1023 and then multiplying by 5.0.

Findings:

---

## Optional Extensions

These will require independent investigation.

1. Dynamically slow the sampling rate.
  - a. Modify the AnalogReadSerial sketch to sample at slower rates as it runs.
2. Change the Serial Monitor.
  - a. Print both sensorValue and Voltage in the Serial Monitor on separate lines.
  - b. Print the sensorValue and Voltage on the same line with a tab in between them.
    - i. `Serial.print("\t");` // inserts a tab character
    - ii. `Serial.print ();` // prints the parameter and stays on the same line in the serial monitor,
    - iii. `Serial.println ();` // prints the parameter and inserts a carriage return (CR) which places the cursor at the beginning of the next line in the serial monitor.
3. Search online to determine the origins of “carriage return”.