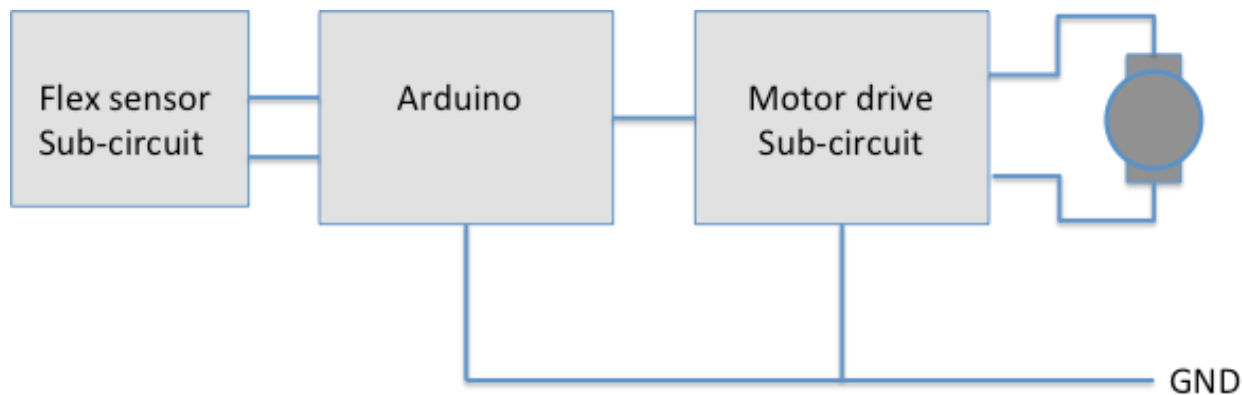


# Module: Wall-Following using Arduino

---

Now that you have a working wall-following vehicle enjoy a moment of satisfaction and then think how you might change the design to include more functionality. Maybe it follows a wall but if it drives over a green piece of paper where it spins around twice and then proceeds on its way. For more complex designs with more complex feedback paths a robust but flexible method of using input from multiple sensors to control wider range of responses is needed. Enter the Arduino.



To introduce you to using the software capability of the Arduino this module helps you integrate the Arduino to provide the same functionality as the wall-following circuit you constructed in experiment 8. The Arduino will **replace the PWM sub-circuit**. The two flex sensors still provides the feedback but now instead of tweaking the duty cycle of the PWM sub-circuit they become inputs to the Arduino. Based on the input voltage that changes as the flex sensor is bent an output signal is provided by the Arduino to each motor. With the right software the circuit should behave exactly as the one you built during experiment 8. In fact the signals feed to the motors are still square waves with changing duty cycles except now the Arduino board is generating the signals.

This module assumes that you have completed experiment 8 and have not taken apart your circuits. Do not dismantle the circuits on your protoboard yet, simply put the second-level on the vehicle and build the new circuits using the Arduino board and the other proto-board on top leaving the motor drive sub-circuits intact. A few connections will need to be made involving the motor drive circuitry but not many so this proto-board can remain on the lower level.

Notes:

---

## Powering the Arduino, Motors, and External Circuitry

Including the Arduino in the design necessitates a method of providing power to sub-circuits with different needs. The Arduino and the flex sub-circuit need 5V and draw little current. The motors need at least 5V and can operate at voltages up to at least 9V, but they need to draw more current than any of the Arduino output pins should source without damage. Luckily, the Arduino includes the necessary circuitry to provide a regulated 5V for the flex sensor sub-circuit and a non-regulated voltage drawn directly from the battery pack to power the motors.

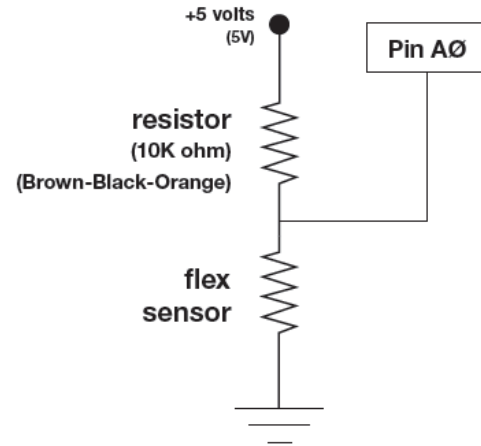
### Connecting the battery

- ✓ Connect either a 7.2V or 9V battery to the Arduino by plugging the barrel jack to the connector on the Arduino board.

### The Flex Sensor Circuit

- ✓ Gently remove the flex sensors from the other circuitry.
- ✓ Build the flex sensor sub-circuit illustrated in the figure on the new protoboard. The 5V needed to power the sub-circuit is available at the pin labeled 5V on the Arduino board and any of the pins labeled GND.
- ✓ Connect the flex sensor the analog input pin A0 as shown in the figure.

Notes:



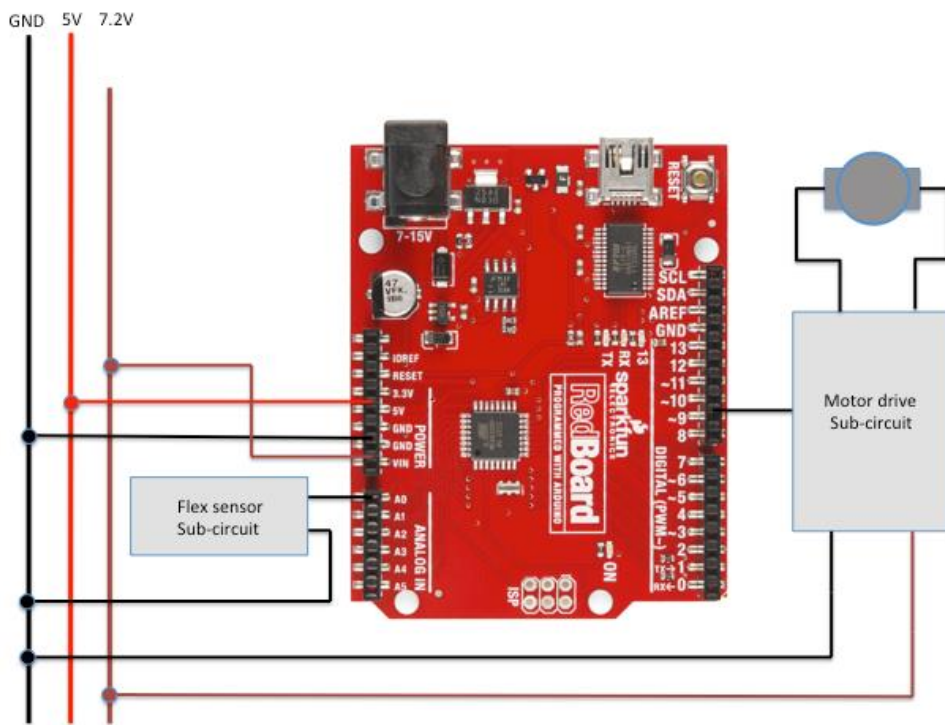
- ✓ Upload the AnalogReadSerialOut program from the Arduino IDE examples and run the program and bring up the console so that you can see the numbers output by the Arduino to the serial port.

**Question 1:** As you flex the flex sensor look at the values recorded in the console window. What are the maximum and minimum values?

## Putting It All Together

You have already connected one flex sensor sub-circuit to the board. The analog input pin A0 is connected across the flex sensor. As the flex sensor is flexed the resistance changes and therefore the voltage between pin A0 and GND. Let's add the motors. The figure below shows, in block-diagram form how one of the motor drive circuits is connected to the board. The I/O pin 9 provides a square wave output replacing the PWM sub-circuit.

- ✓ Build another flex sensor circuit connecting it to pin A1.
- ✓ Gently disconnect 7.2V battery if you have not do so already. Connect the 7.2V that shows up at the pin labeled – confusingly –  $V_{in}$  and GND provided by the Arduino to the protoboard as a replacement.
- ✓ Connect the drive signals to the motor drive sub-circuit replacing those from the PWM sub-circuit. The figure below shows one of the motors connected to pin 9, use pin 11 for the second motor.



Notes:

Now a program is needed to program the Arduino to emulate the wall-following circuit you put together in experiment 8. The following code illustrates how to connect 1 flex sensor to the Arduino and output a square wave with a duty cycle that changes as the flex sensor is flexed.

```
int sensorPin = 0;    // The potentiometer is connected to
                    // analog pin 0
int PWMPin = 9;      // The LED is connected to digital pin 13
int flexMax = 768;   // The maximum value read from A0 when attached
                    // to the flex sensor sub-circuit
int flexMin = 248;;  // The minimum read from A0 when attached
                    // to the flex sensor sub-circuit
int flexRange=flexMax-flexMin; //Maximum range found empirically

void setup()
{
    pinMode(PWMPin, OUTPUT);
}

void loop() // this function runs repeatedly after setup() finishes
{
    int sensorValue;

    sensorValue = analogRead(sensorPin);

    int speed=255*(sensorPin-flexMax)/flexRange;
    analogWrite(PWMPin,speed);
}
```

It reads an analog value from pin A0 and uses that value to compute the desired motor speed. The `analogWrite` command instructs the board to present a square wave signal on Pin 9 that is to be used to control the speed of the motor. You need to modify this code to navigate using 2 flex sensors connected appropriately to 2 motors. The `flexMin` and `flexMax` are the two numbers you obtained for question 1 to replace the `xxx` in the code.

Your job now is to expand this design to include two motors that are connected to the flex sensors properly so the vehicle follows the contours of a wall using the flex sensors.

**Question 3:** Draw a block diagram level schematic of the resulting circuitry – be sure to label which ports are connected to which sensors.

**Question 4:** Draw a schematic of the entire circuit. You can designate the wires connected to the Arduino with simple labels like the one designating A0 in the flex sub-circuit diagram if you wish.