

Creating Interactive Prototypes

Jason Forsyth

PhD Student

ECE Department

Virginia Tech

Ed Dorsa

Associate Professor

Industrial Design

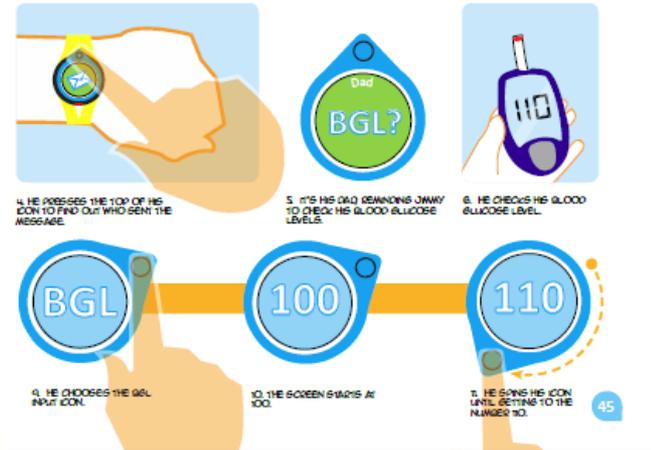
Virginia Tech



THE WATT



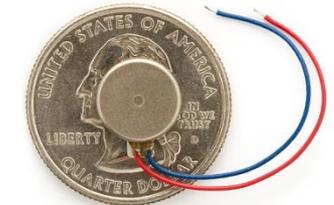
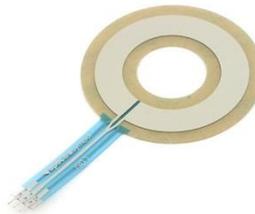
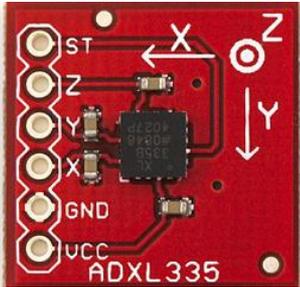
How our business works



Today

- Introduce Sensors/Actuators, Scratch, Arduino
- Walk through 4 examples
 - Input: Slide sensor, Force sensor
 - Output: Tri-color LED, Servo motor
- Create Simple Game/Children's Toy
- Free Time
 - Explore the sensors
 - Storyboard your ideas

Sensors & Actuators Available



Rotation – Force – Light – Magnet – Motor – Touch – Range – Vibration

User Friendly Data Sheets

Force Sensitive Resistor



What It Does: The Force Sensitive Resistor (FSR) is a sensor that changes its resistance when pressed. The change in resistance can be measured by the Arduino and can be used to determine the force applied to it.

What It Tells You: The greater the value reported by the FSR, the harder it is being pressed. The sensor will report 0 if untouched. While it is possible to convert this value to a physical unit such as pounds, we have not done this in our exercise.

Required Connections: For this workshop, the FSR should already be wired up, but you may need to attach the connections to your Arduino. The red wire is for power and should be connected to **5V**. The black wire is ground and should be connected to **Ground/GND**. A third wire (of another color) should be connected to an **Analog Input**.

Using the FSR in Scratch:

Use a *value of sensor* block to read in from the selected Analog Input. Adjust the pull down

Scratch Examples for Each Sensor

The screenshot displays the Scratch IDE interface with an Arduino script. The script is titled "Arduino1" and is located in the "Scripts" tab. The script consists of two "forever" loops. The first loop calculates acceleration and rotation values from three analog sensors. The second loop checks if the magnitude of acceleration is greater than 1.5 and toggles a digital pin (13) accordingly.

Script 1: Acceleration and Rotation Calculation

```
forever
  set xAccel to 0.01454 * value of sensor Analog0 - 5
  set yAccel to 0.01454 * value of sensor Analog1 - 5
  set zAccel to 0.01454 * value of sensor Analog2 - 5
  set Mag to sqrt of xAccel * xAccel + yAccel * yAccel + zAccel * zAccel
  set xRotation to atan of yAccel / zAccel
  set yRotation to atan of xAccel / zAccel
  set zRotation to atan of yAccel / xAccel
```

Script 2: Magnitude Check and Digital Pin Toggle

```
forever
  if Mag > 1.5
    digital 13 on
  else
    digital 13 off
```

The right-hand side of the interface shows the "Accel" sensor panel with a diagram of an Arduino board and a "Searching board..." button. The stage area is currently empty, showing a "Stage" label.

Controlling Sensors with Scratch

- Programming through Blocks
 - Blocks are individual commands, appropriate blocks fit together like a jigsaw
 - Blocks stack to form a program
- Four Main Groups for this Workshop
 - Motion (blue): send/receive data from Arduino
 - Control (yellow): direct the flow of your program
 - Operators (green): add, subtract, multiply...
 - Variable (orange): create and store variables

Learning Through Examples

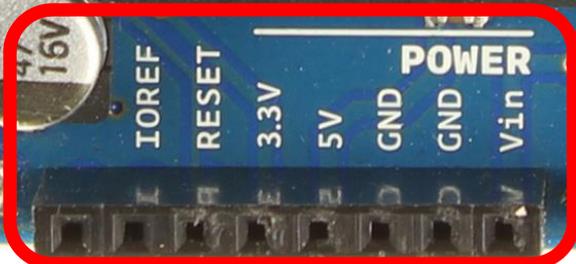
- Make the LED on your Arduino Blink
- Use *Digital On* and *Digital Off* blocks under the Motion Tab
- Key points:
 - Click to activate/deactivate blocks
 - Blocks are like messages to the Arduino



Moving On To Better Things....

- Need to understand our Arduino board
- Difference between Input / Output / Analog / Digital

Digital Input/Output



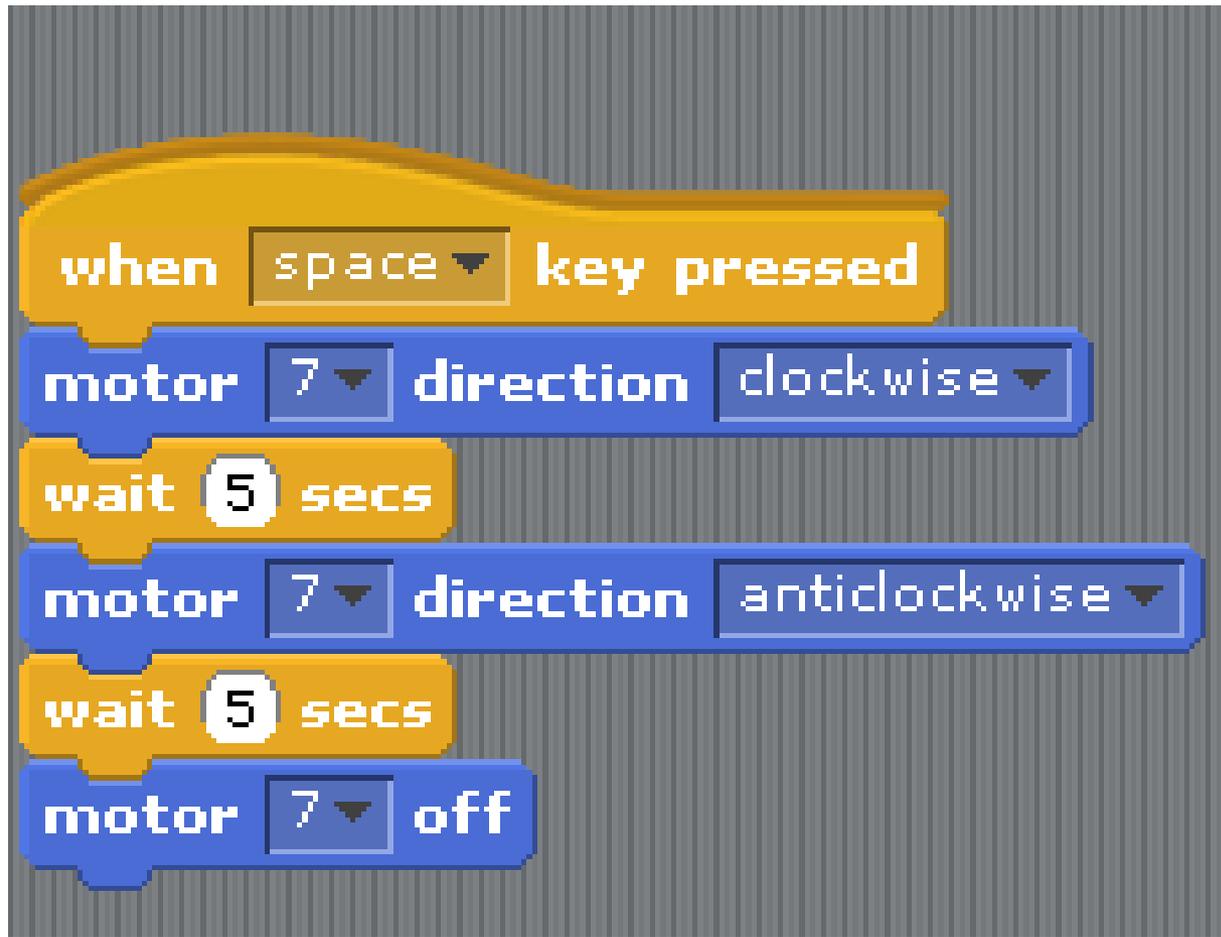
Power & Ground

Analog Input

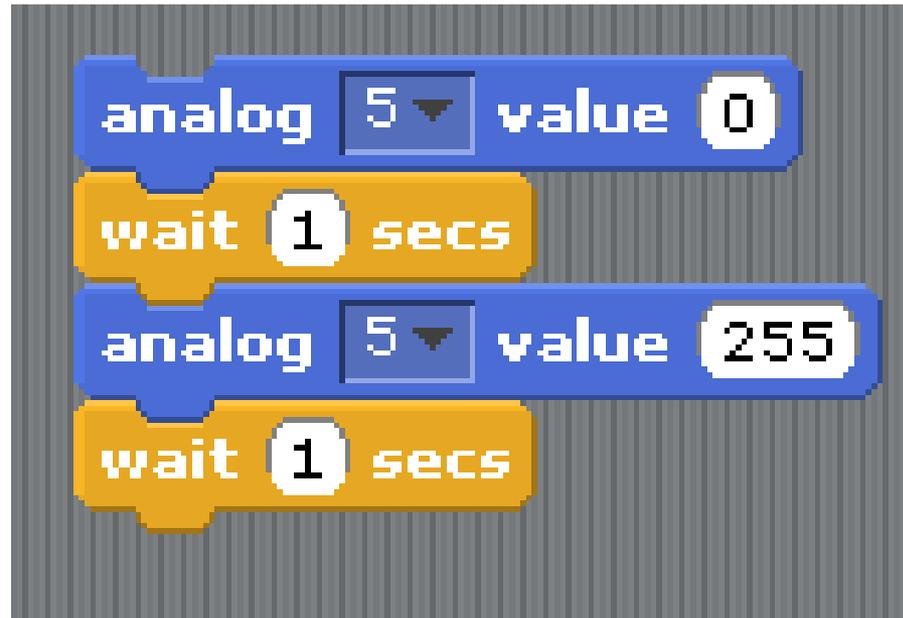
Servo Example

- Connect Servo to Arduino
 - Red => 5V
 - Black => Ground/GND
 - White => Digital Input 4 or 7
- Blocks
 - *Motor Direction* : rotate motor
 - *Motor Off*: stop motor

Servo Example



RBG LED



```
analog 5 value 0  
wait 1 secs  
analog 5 value 255  
wait 1 secs
```

The image shows a sequence of four Scratch code blocks on a grey background. The first block is a blue 'analog' block with a dropdown menu set to '5' and a value field set to '0'. This is followed by an orange 'wait' block with a value of '1' and the unit 'secs'. The third block is another blue 'analog' block with a dropdown menu set to '5' and a value field set to '255'. The final block is an orange 'wait' block with a value of '1' and the unit 'secs'.

RGB LED

```
analog 9 value 0

set Red to 0
repeat until Red > 255
  analog 5 value Red
  change Red by 10
  wait 1 secs

set Green to 0
repeat until Green > 255
  analog 9 value Green
  change Green by 10
  wait 1 secs

set Blue to 0
repeat until Blue > 255
  analog 6 value Blue
  change Blue by 10
  wait 1 secs
```

Sliders / Force

