

Creating Interactive Prototypes

Jason Forsyth

PhD Student

ECE Department

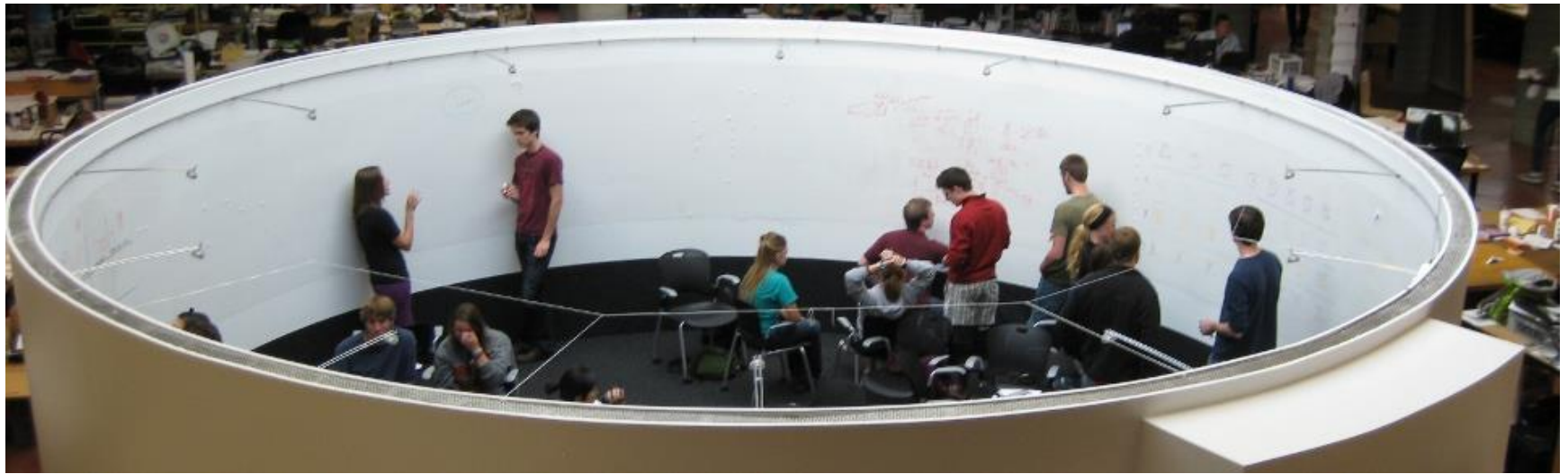
Virginia Tech

Ed Dorsa

Associate Professor

Industrial Design

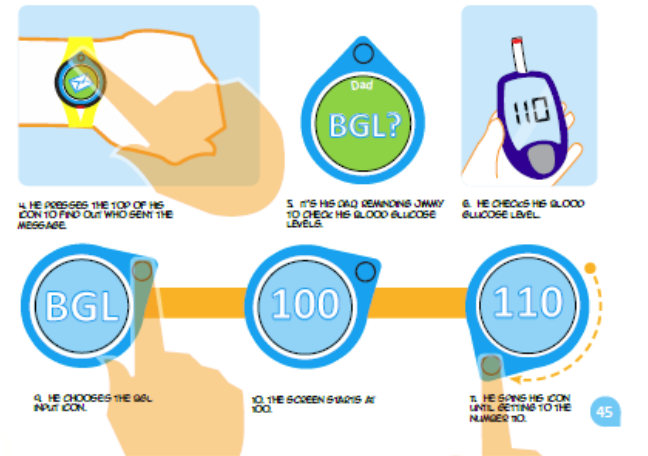
Virginia Tech



THE WATT



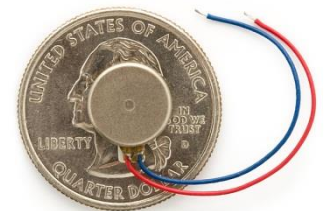
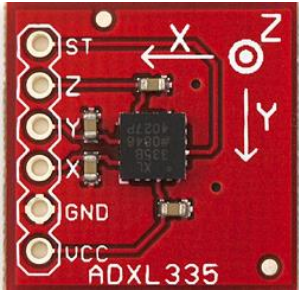
How our business works



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.

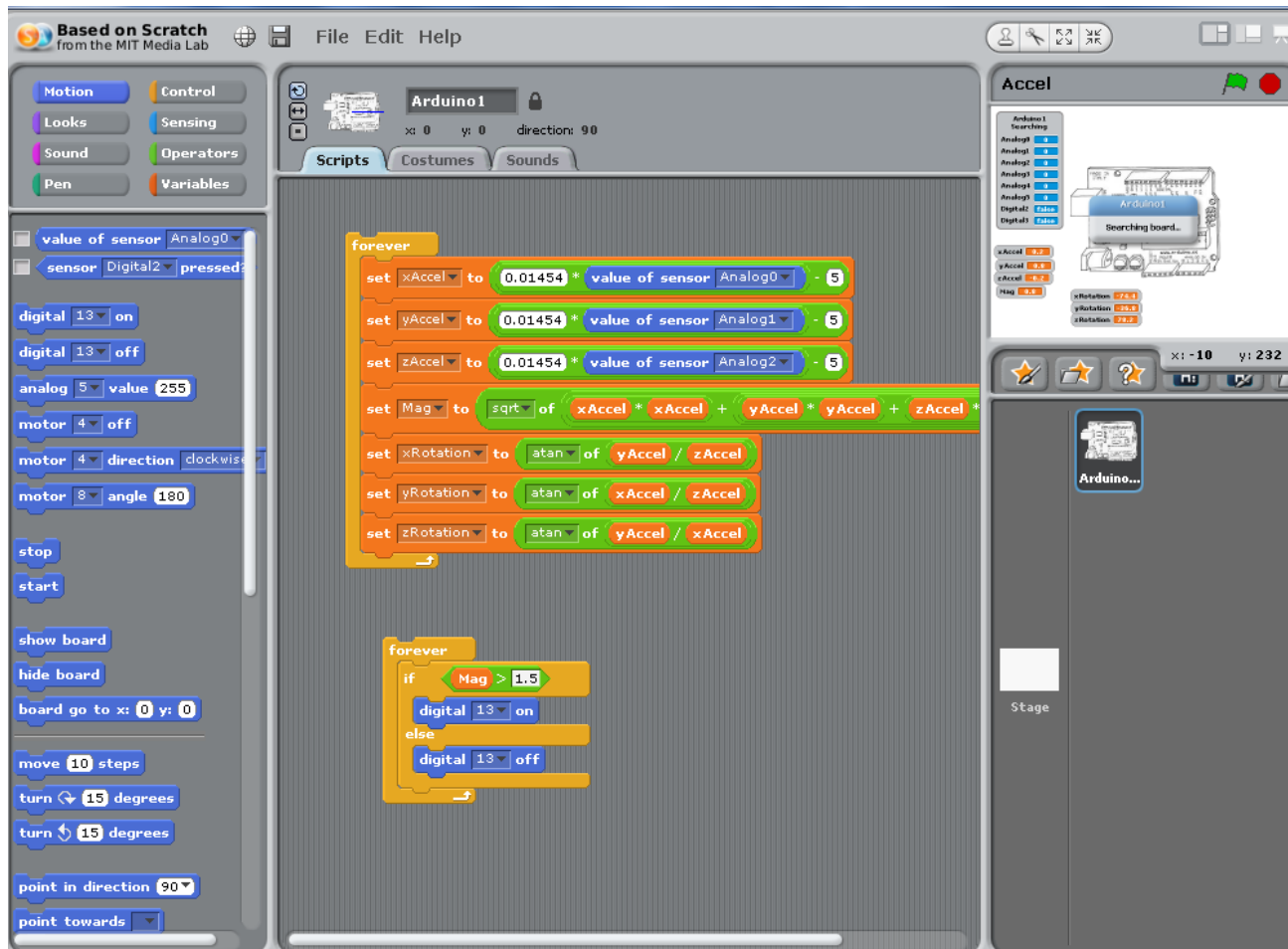
What It Tells You: The greater the value reported by the FSR, the harder it is being pressed. The sensor will report 0 if untouched. While it is possible to convert this value to a physical unit such as pounds, we have not done this in our exercise.

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



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

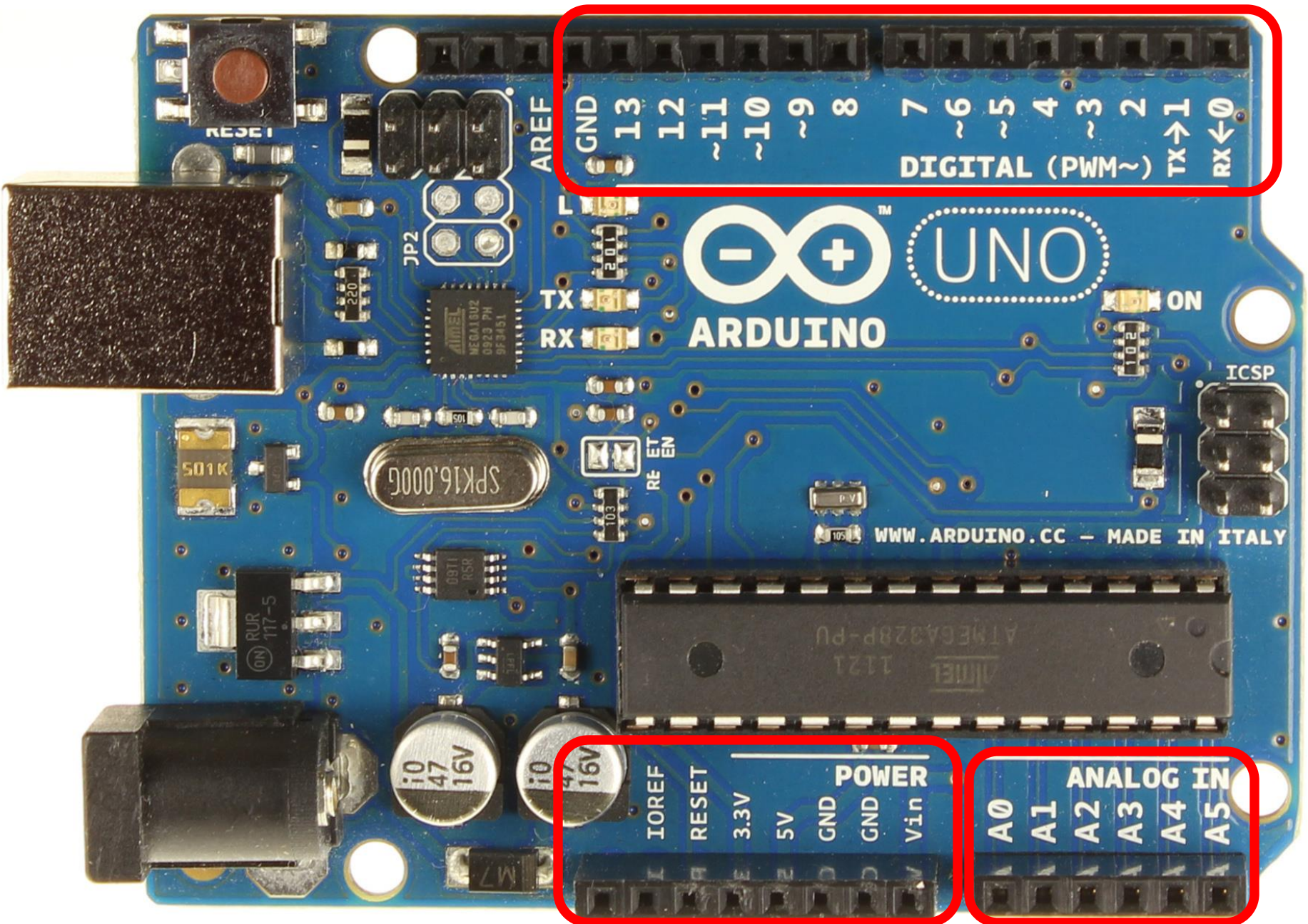
- Make the LED on your Arduino Blink
- Use *Digital On* and *Digital Off* blocks under the Motion Tab
- Key points:
 - Click to activate/deactivate blocks
 - Blocks are like messages to the Arduino



Moving On To Better Things....

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

Digital Input/Output



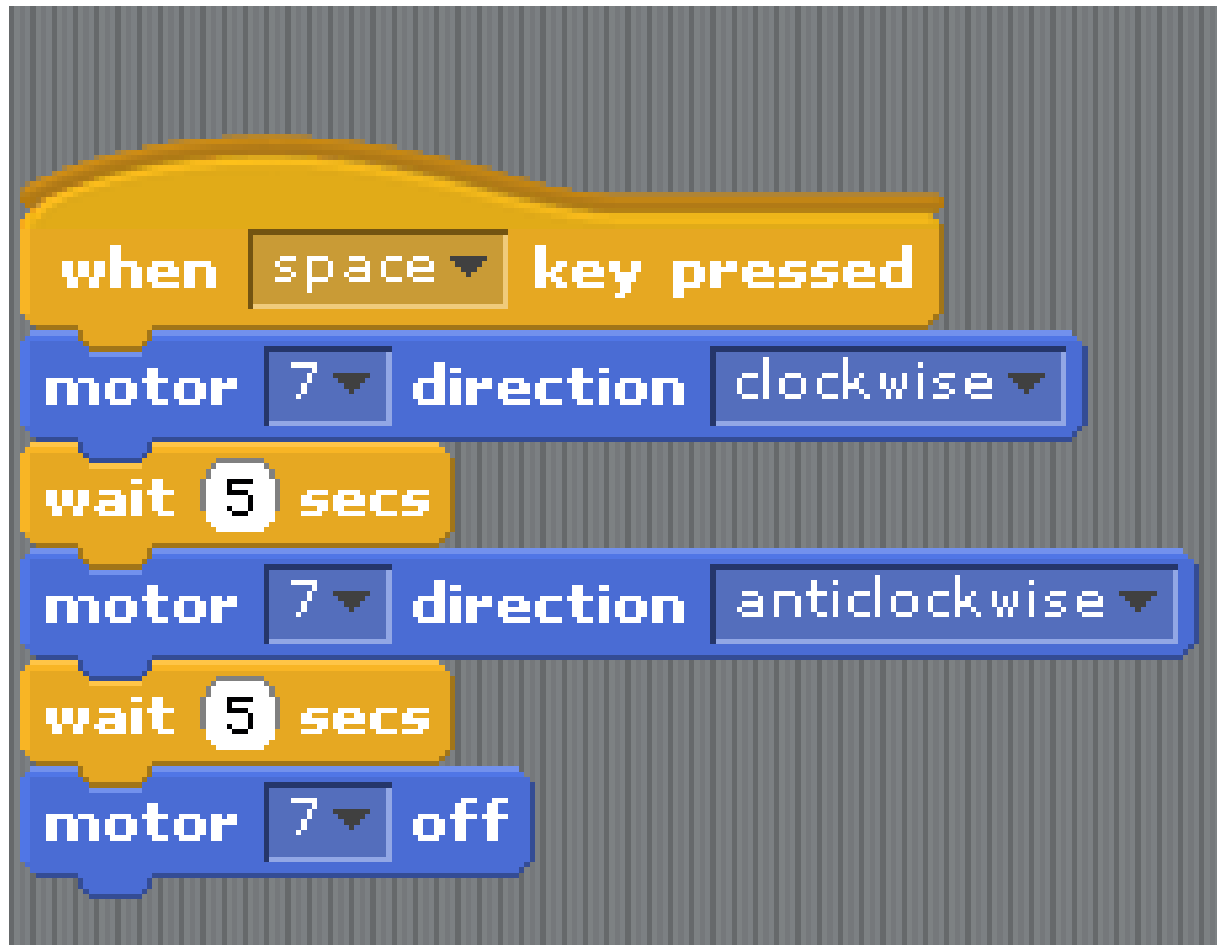
Power & Ground

Analog Input

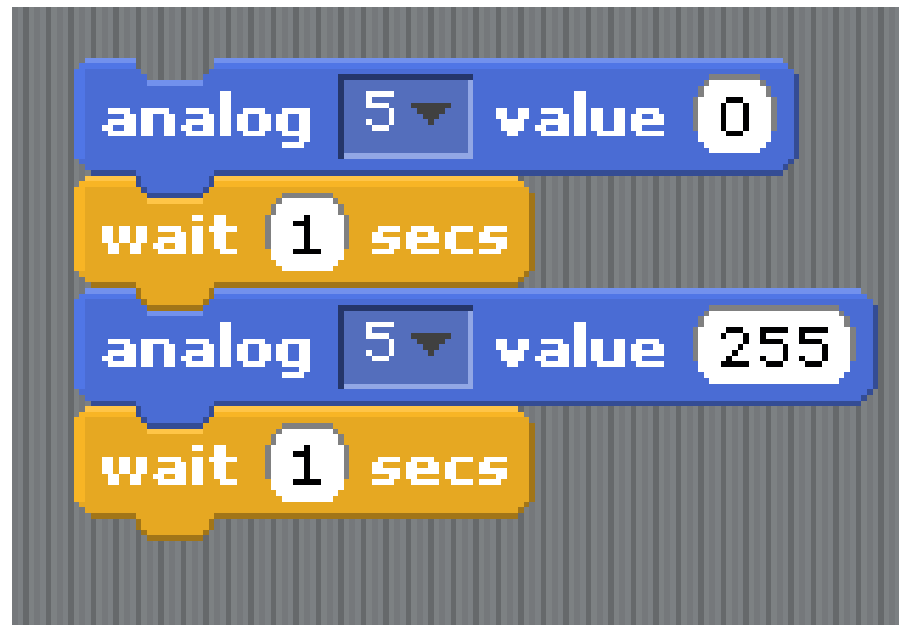
Servo Example

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

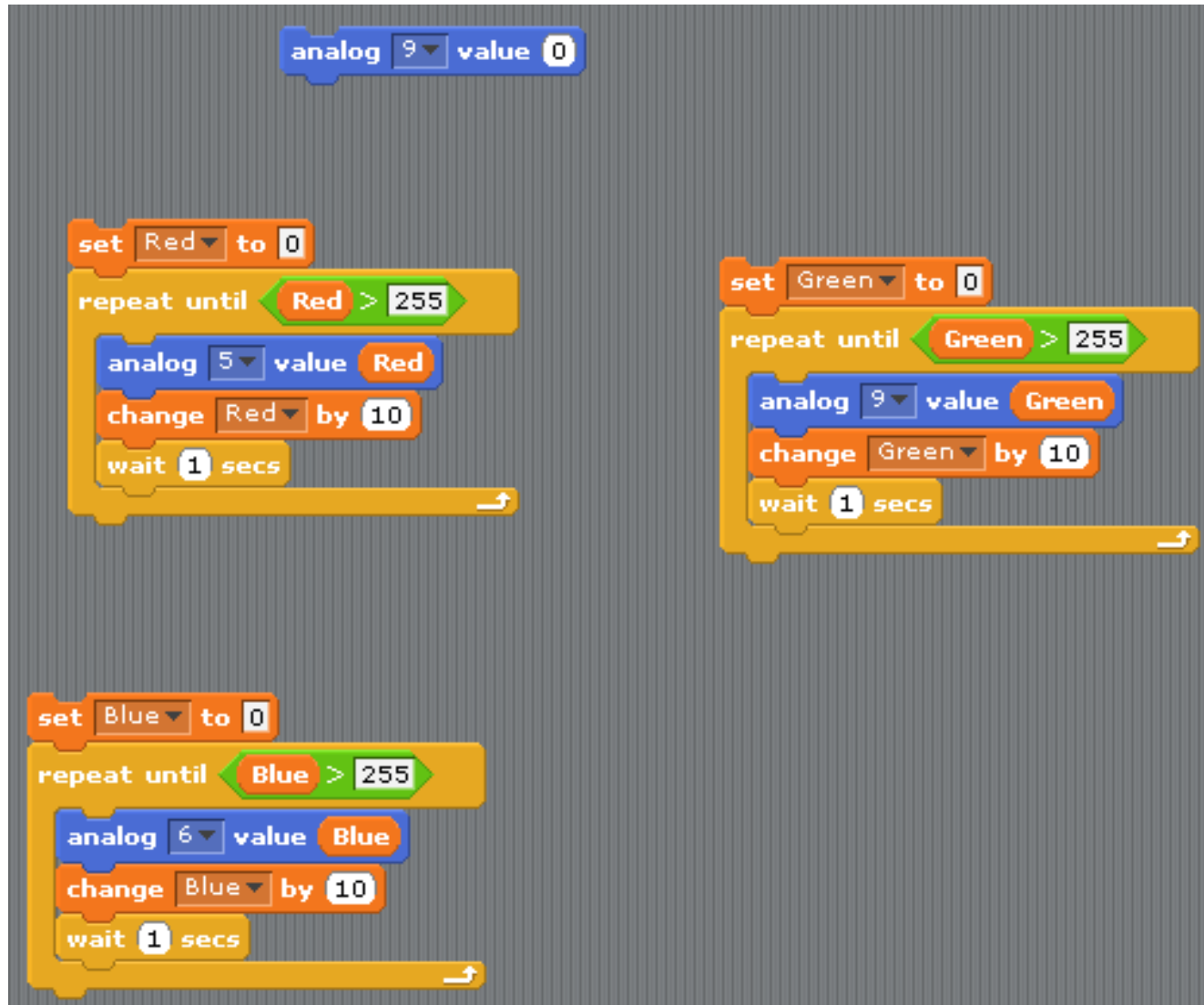
Servo Example



RBG LED



RGB LED



Sliders / Force

