

Electronics Lesson 5

Overview

In this lesson students will learn how to create a controllable robotic arm.

Plan

1. Analog vs Digital
2. Servo
3. Potentiometer
4. Robotic Arm
5. Extra Credit: Train Crossing Gate

Big Concept #1: Analog vs Digital

Instructor: Analog is measured. Digital is counted. An analog example is a clock with a hand dial, the hand dial moves to the location that represents the approximate time. A digital example is your phone's clock it represents that exact time.

Instructor: In electronics, analog is when you change the voltage or current in a continuous rate.

To students: If I'm measuring two cups of *flour* for a cake recipe, is that analog or digital?

Expected Answer: Analog because you aren't getting exactly 2 cups.

To students: If I'm measuring two eggs for a cake recipe, is that analog or digital?

Expected Answer: Digital because you are getting exactly 2 cups.

Big Concept #2: Servo

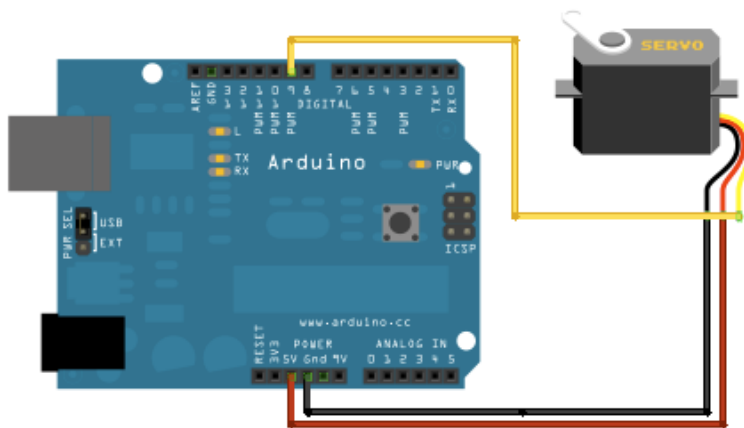
Instructor: Take out the servo and show it to the class. Show that it moves.

To students: What are servos or motors used for? Have several students give examples.

Expected Answer: To move wheels on a car, to control robotic legs to land on the moon, to control automatic doors.

Instructor: Servos typically move 180 degrees.

Instructor: Servos has three inputs. Show the wires of the servo. The red input takes electricity (5v). The black wire is for wiring to ground to complete the circuit within the servo. The yellow wire is for controlling the servo; this is done by the arduino sending electrical pulses to the servo.

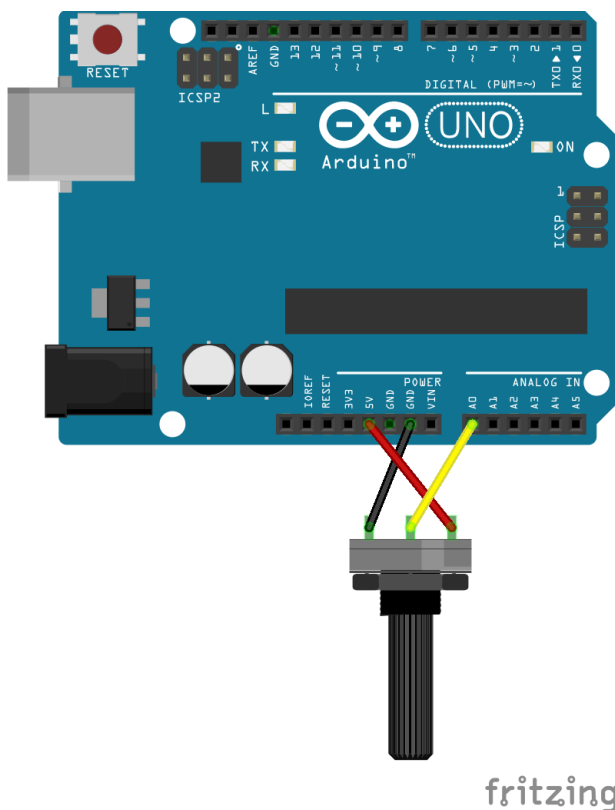


Big Concept #3: Potentiometer

Instructor: Take out the potentiometer and show it to the class.

Instructor: Potentiometer is an adjustable resistor. As you turn the knob it will increase or decrease the resistance when then can be measured by the arduino.

Instructor: The Potentiometer also takes three inputs. The left input wires to ground. The right input wires to 5v. These two inputs complete the circuit within the potentiometer. The middle input is for sending an electrical signal to the analog pin on the arduino that then can be read.



Big Concept #4: Robotic Arm

To students: Using a servo and potentiometer, anyone tell how can we create a robotic arm? Hint: remember we have to read the value of the button to control the light and buzzer.

Expected Answer: We can read the values from the potentiometer to control the servo.

Activity 1: Code a robotic arm.

Setup computers. Distribute worksheets, if no worksheets write the code on the board.

Instructor: Walk through the code with the students (see comments in the code).

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

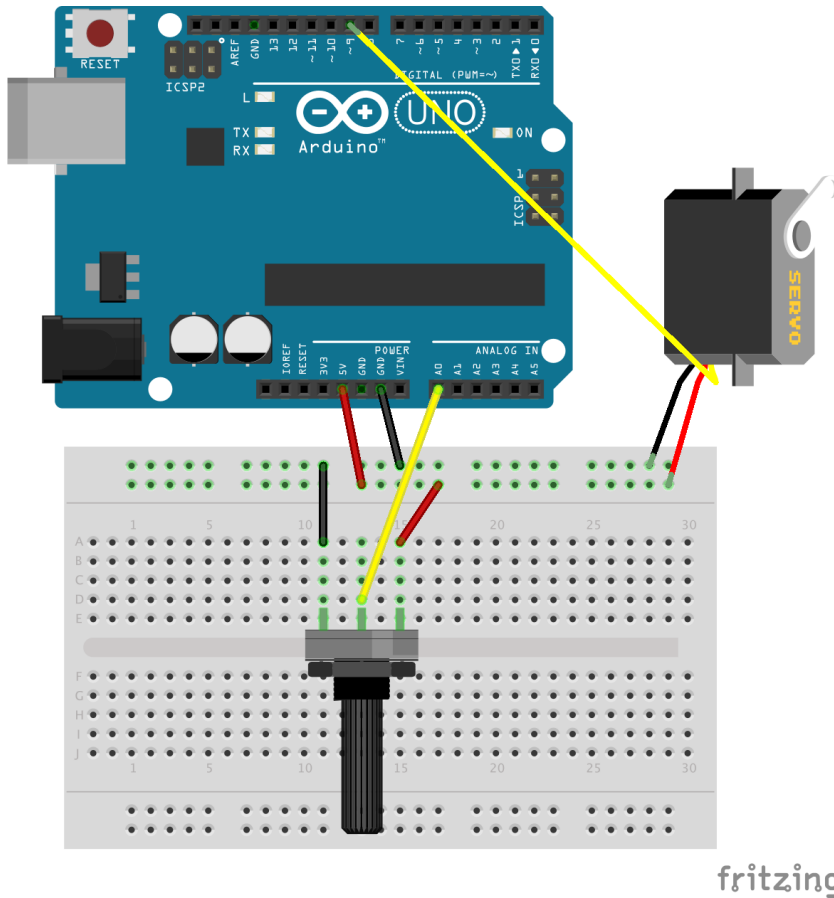
void loop() {
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value
  delay(15); // waits for the servo to get there
}
```

Activity 2: Wire a robotic arm.

1. Have students wire the robotic arm *themselves first* before guiding them.
2. Once the servo is moving, have them use the tape and paper in the bag to create an arm.

Expected Answer:

Layout:



Extra Credit Activity 3: Train Crossing Gate

If students would like a challenge, have them code and wire a train-crossing gate. Create a gate using the servo and potentiometer. Using a red and green led, light up the red LED when the gate closes and light up the green LED when the gate opens like a train-crossing gate!

Expected Answer:

Code:

```
Train_Crossing §
/*
  Electronics Lesson 5: Train Crossing
  -----
  Description
  -----
  Controls servo using the potentiometer. Lights up green LED when more than 90 degrees and lights red LED hwn e less than 90 degrees.
*/

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin A0 used to connect the potentiometer
int val; // variable to read the value from the analog pin

const int green = 7; // the number of the LED pin for green LED
const int red = 6; // the number of the LED pin for red LED

void setup() {
  // initialize the LED pin as an output:
  pinMode(green, OUTPUT);
  pinMode(red, OUTPUT);

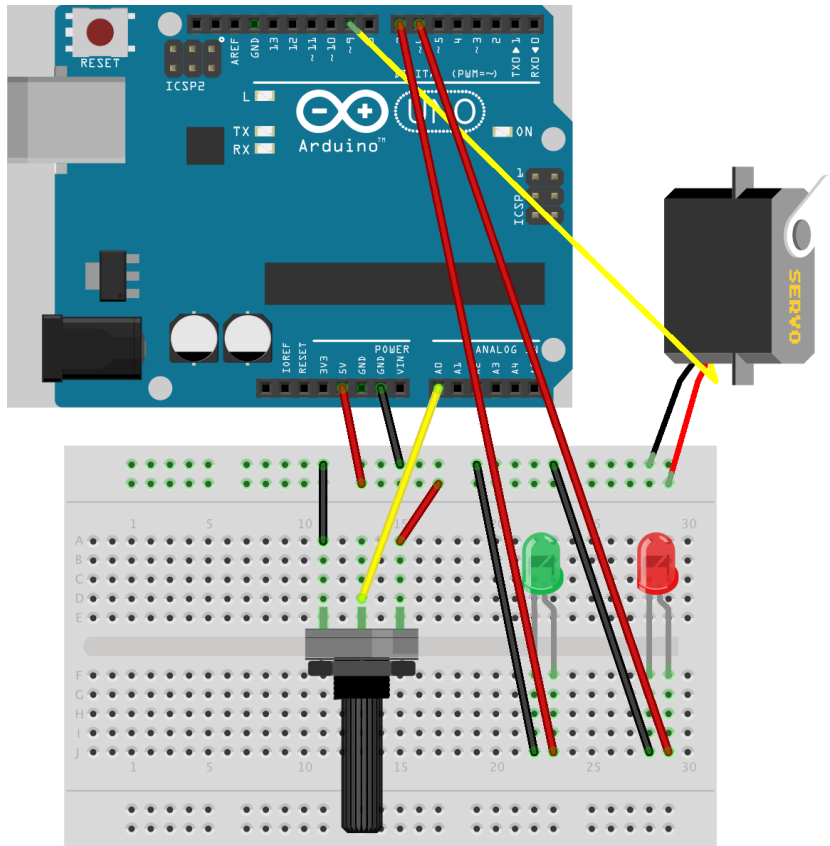
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value

  if (val > 90) { //
    digitalWrite(green, HIGH);
    digitalWrite(red, LOW);
  } else {
    digitalWrite(green, LOW);
    digitalWrite(red, HIGH);
  }

  delay(15); // waits for the servo to get there
}
```

Layout:



fritzing