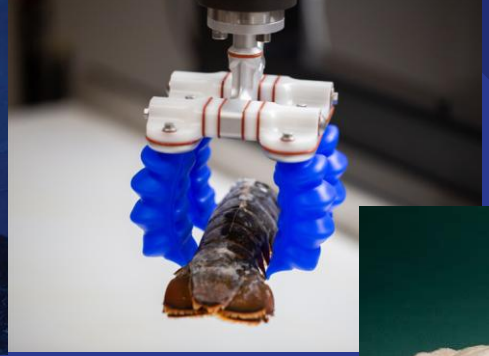


Robots!

Mark Harris (adapted from a class by Rachel Harris)

What are robots?

A robot is a machine that gathers information about its environment (senses) and uses that information (thinks) to follow instructions to do work (acts).

A robot: SENSES -> THINKS -> ACTS

What are the parts of a robot?

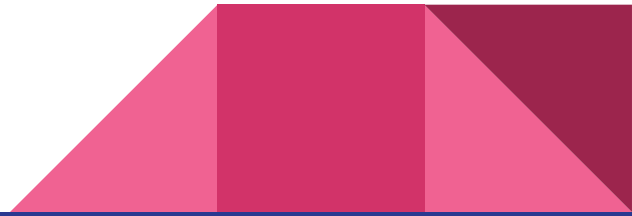
Computer (to make decisions), Input ports (connected to sensors), and Outputs (connected to motors, for example).

Which part does which thing?

Definition: “A robot is a machine that gathers information about its environment (senses) and uses that information (thinks) to follow instructions to do work (acts).”

Senses – using SENSORS; thinks – using COMPUTER; acts – using MOTORS,

What IS a robot? What ISN’T a robot?



Controls

Sensing

- Analog to human senses – sight, hearing, smell, touch, taste

Moving

- Motors
- Pistons

Elektro: The First Modern Humanoid Robot

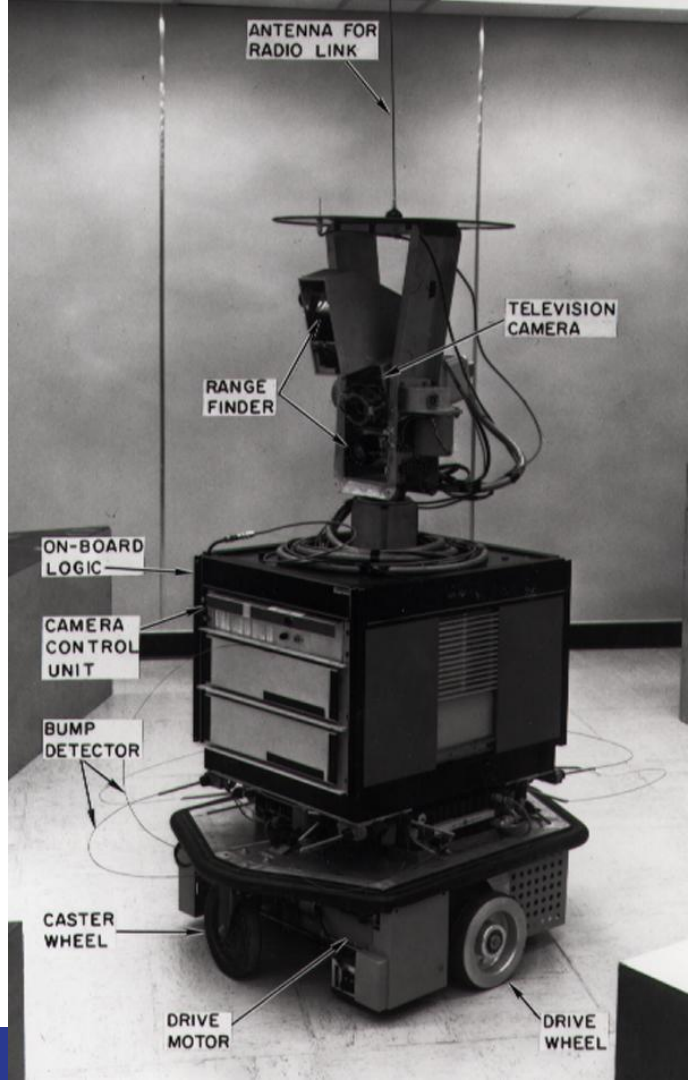
- 1939 New York World's Fair
- Westinghouse Electric Corporation
- Size – two meters tall, 120 kg weight
- 26 movements, including walking, moving head and arms, speaking 700 words, inflating balloons, and smoking cigarettes
- Elektro's dog was named Sparko, and he could also do basic commands

Unimate – the first industrial robot

- George Devol, 1950s
- General Motors
  - Later Chrysler, Ford, and Fiat
- Tasks - die casting handling and spot welding
  - Pouring coffee

Shakey, the first autonomous and intelligent robot

- 1966-1972
- First robot blending logical reasoning and physical action
- Included computer vision and natural language processing
- Stanford Research Institute and Defense Advanced Research Project Agency (DARPA)
- Traveling from one location to another, turning the light switches on and off, opening and closing the doors, climbing up and down from rigid objects, and pushing movable objects around

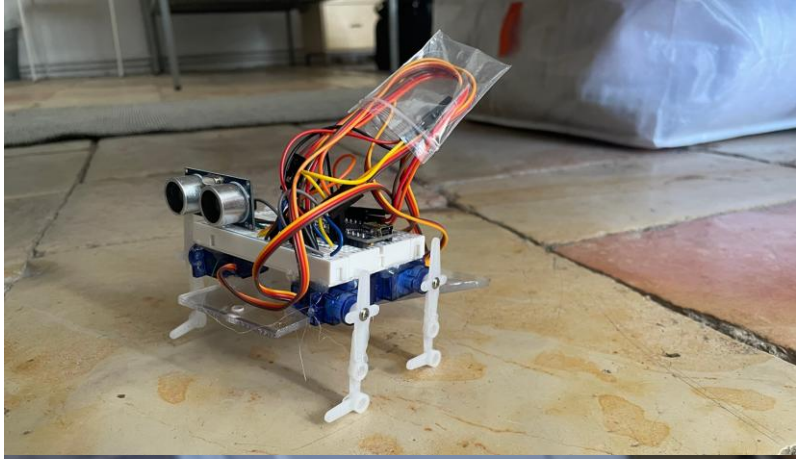


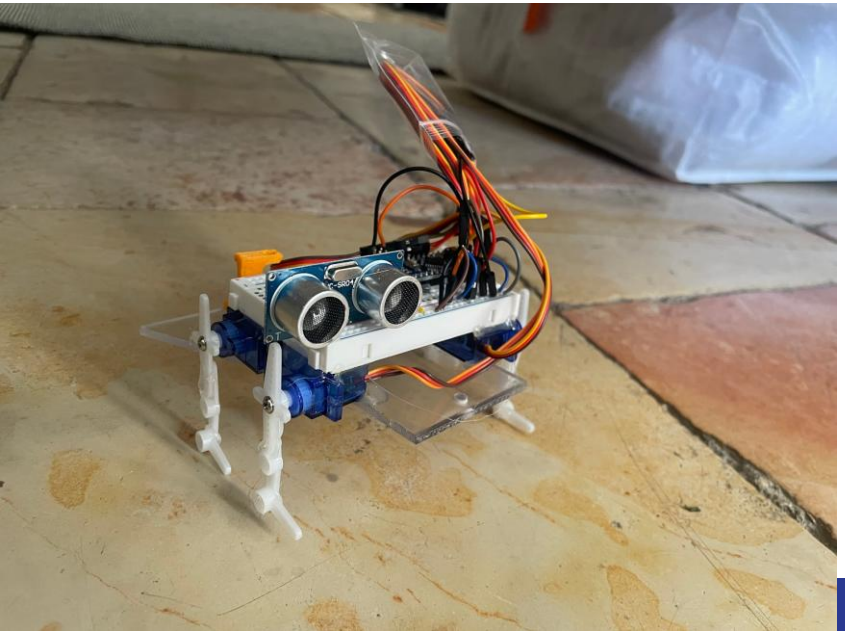
Modern Robots

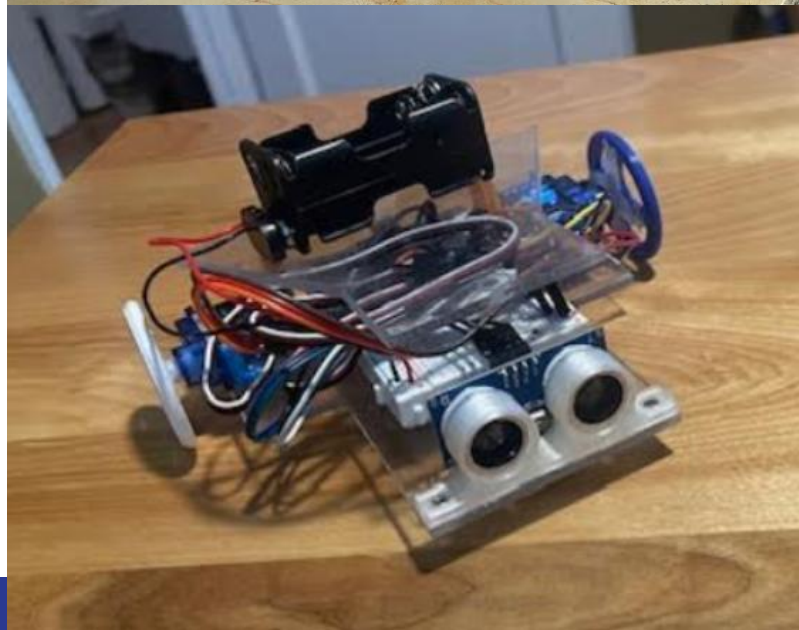
- Robots mow your lawn, sweep your house, and wash your dishes
- Robots work in factories and on farms
- Robot drones clean windows on high rise buildings and take photos and video
- Robot drones fight

Our Robots

- Ultrasonic bot to navigate maze
- What parts of the robot can you recognize?







Mechanical

What is mechanical engineering?

- Includes mechanics, dynamics, thermodynamics, materials science, and design.
- What parts of the robot are “mechanical” parts?
- What mechanical things have you built before?

Hardware

Hardware, or “Mechanical Engineering” is just building things.

- Manufacturing = making parts
- Assembly = putting parts together

Subtractive Manufacturing

Making a part by taking pieces out of it

- Saws, drills, scissors, etc
- Other examples?

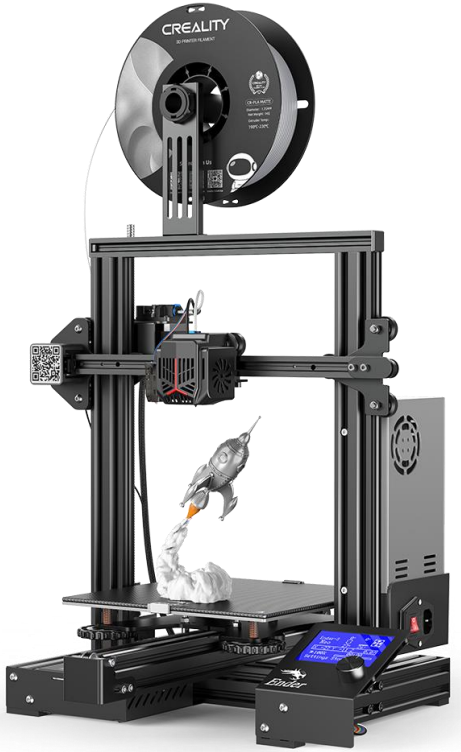
Additive Manufacturing

Making a part by adding material

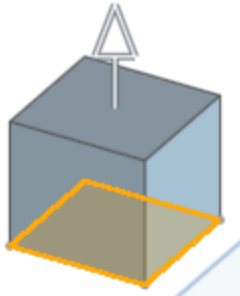
Normally done by melting material into layers

3d printing, layering blocks, what else?

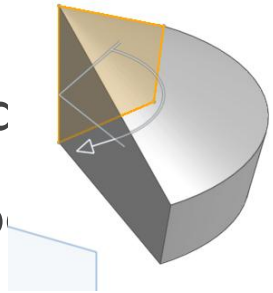
1. Design part on the computer (CAD)
2. Slice part in software
3. Print part

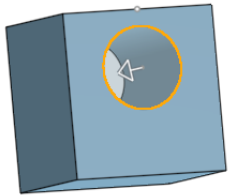


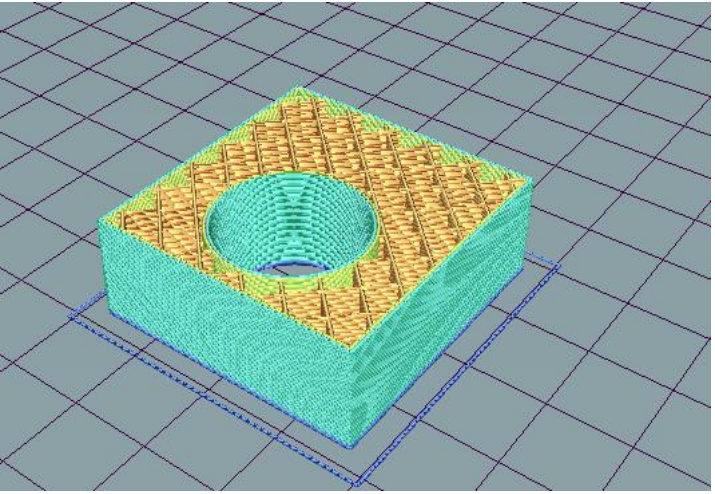
CAD - Computer Aided Design



- Use a computer to draw a 3d shape
- Extrude - take 2D shape and thicken it along a perpendicular axis
- Revolve - take 2D shape and rotate it along a plc
- Cut (extrude or revolve) - take solid and 2D shap remove the shape from the material







Slicer

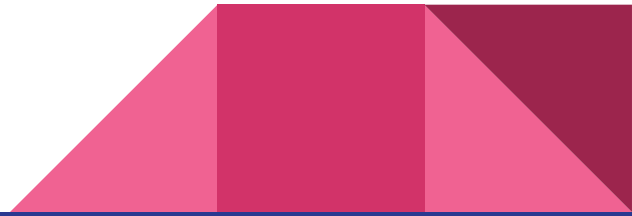
```
; Default start code
G28 ; Home extruder
G1 Z15 F100
M107 ; Turn off fan
G90 ; Absolute positioning
M82 ; Extruder in absolute mode
M190 S50
; Activate all used extruder
M104 T0 S210
G92 E0 ; Reset extruder position
; Wait for all used extruders to reach
M109 T0 S210
;Layer count: 165
;LAYER:0
M107
G0 F9000 X78.896 Y79.566 Z0.300
;TYPE:SKIRT
G1 F1800 X121.103 Y79.566 E2.10572
G1 X121.103 Y120.433 E4.14458
```

1. Save CAD part as .stl (lots of triangles making a 3d surface)
2. Load stl into slicer, adjust settings
3. Software makes a bunch of “slices” of part - cross sections at each height
4. Software writes g-code, which tells printer where to go and what to extrude
5. Printer prints the part!

Printer

- Motors: help nozzle move over big 3d space (normally x, y, z)
- Extruder: pushes filament through nozzle and heats it up
- Filament melts on the bed into whatever shape you tell it,

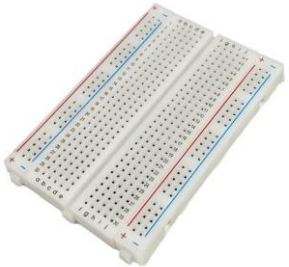
And then you get a part.



Robotics Groups

- Divide into groups of two people
  - One robot per group
- Sit next to each other at a table.
- Take all clutter off your table. This is your assembly area.
- Have a napkin or paper towel to keep your area clean.
- Everyone must participate

Hardware for Each Group:

- 1 battery pack 
- 2-4 servos 
- 4 servo horns or wheels/screws 
- 1 breadboard 

Hardware Questions for Our Robots

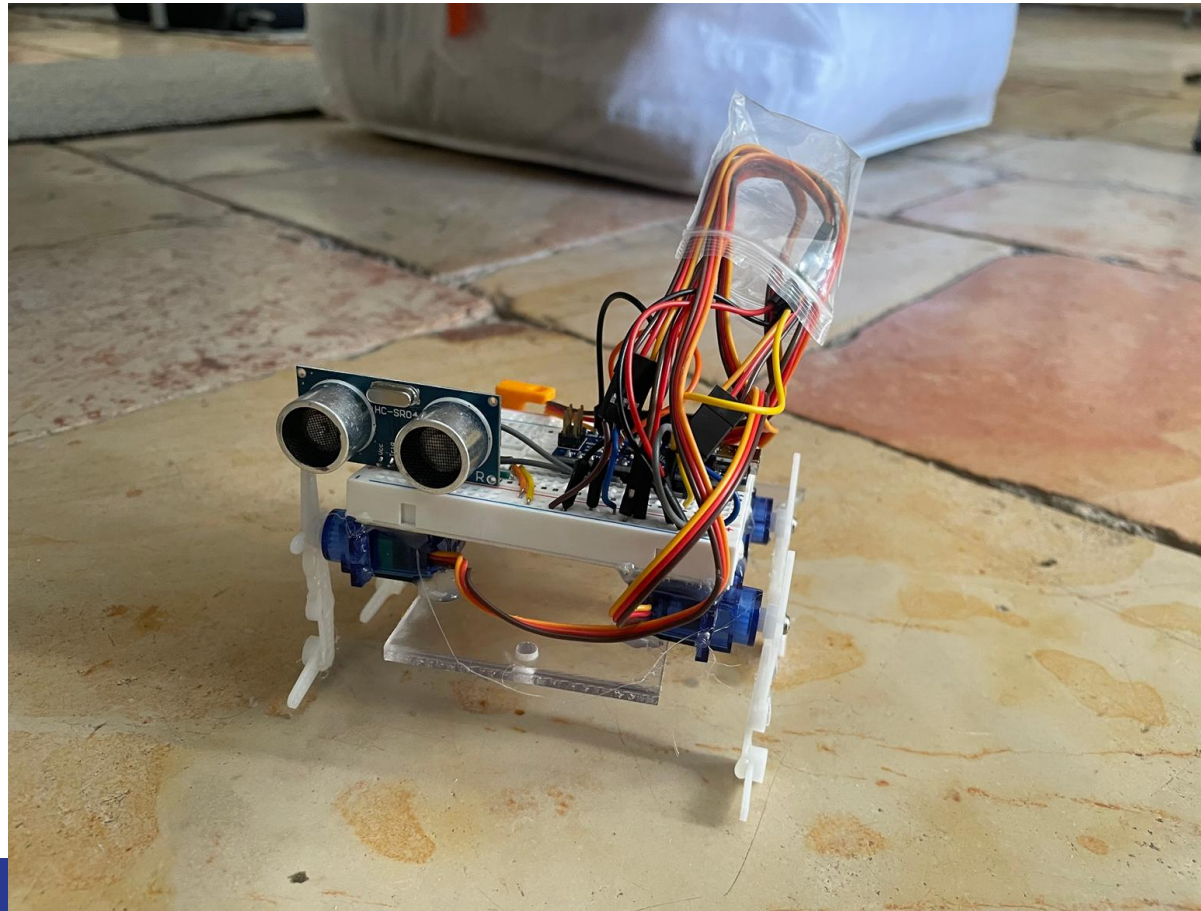
- Feet - What shape of feet are best for the robot to walk? How long should the legs be? What will you make them out of?
- Wheels – How many wheels? Where will they be?
- Center of Mass - Where is the weight on the robot? Where do you want it? Does the robot tip backwards/forwards/to the side?

Robot Assembly

Battery pack and servos under breadboard

Legs screw onto servos or wheels attach to servos

Design legs or place wheels however you want



Electrical

- What is electrical engineering? Study, design, and application of equipment, devices, and systems that use electricity, electronics, and electromagnetism
- What parts of the robot are electrical?
- What electrical things have you used before?

Electrical

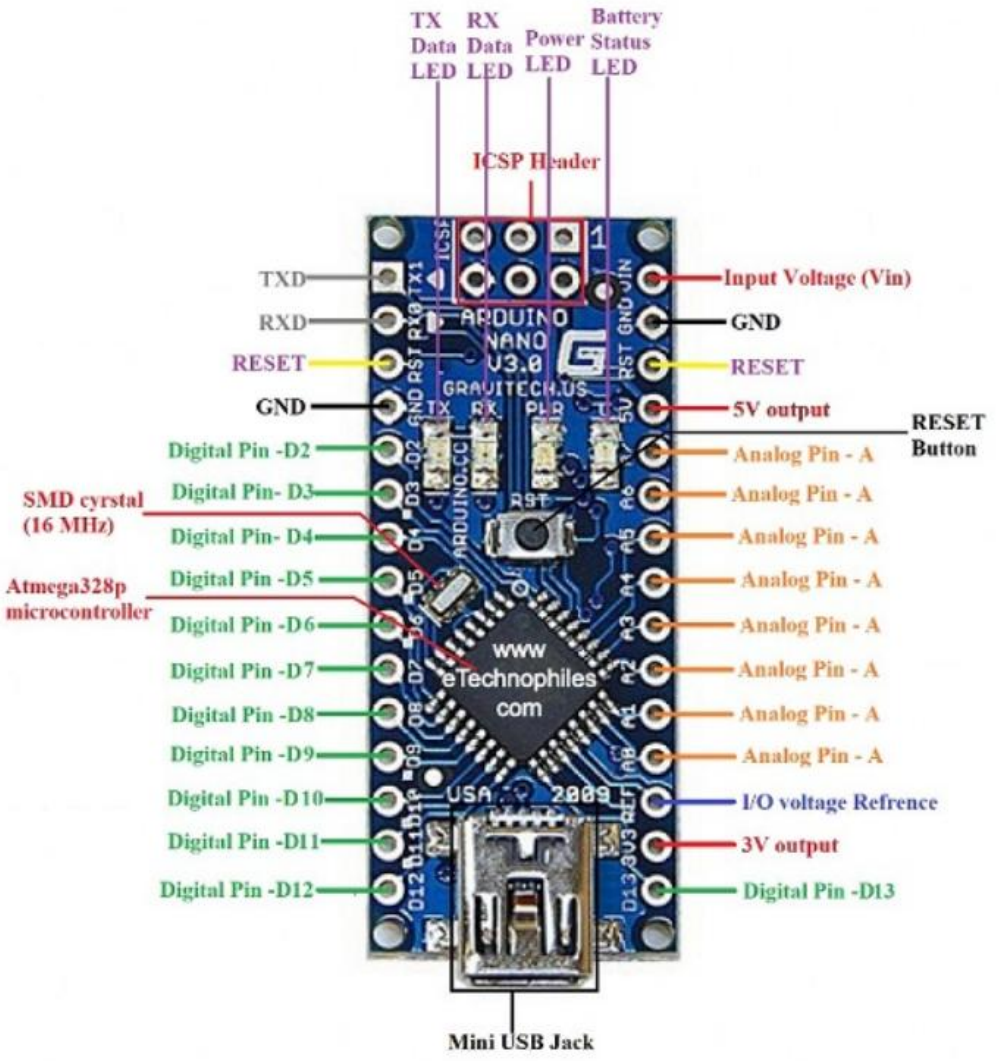
Arduino - computer

Input ports - sensors

Output ports - motor/actuator

Batteries

Wires/breadboard



Ultrasonic sensor

Uses sound waves to determine distance

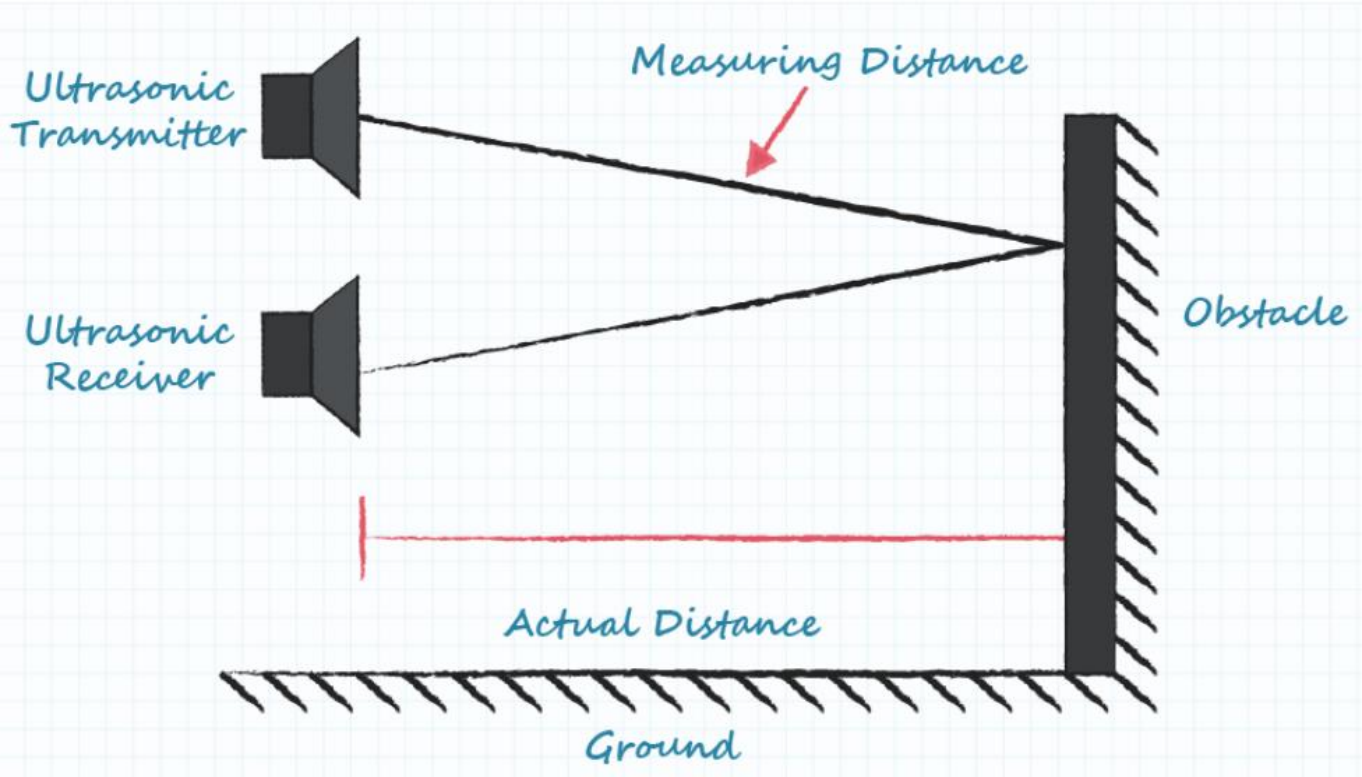
“Ping” = send 1 pulse from sensor

4 pins:

GND/PWR

TRIG (transmitter)

ECHO (receiver)



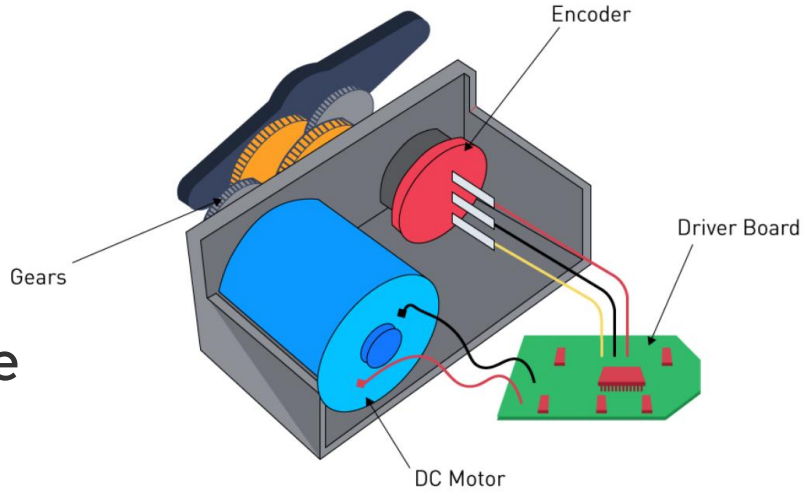
Servos

Motor: Uses power and ground to run - current determines direction

Encoder: generates signals that indicate the angle or displacement of the motor shaft in order to provide feedback on the motor's position.

Driver: takes position data from encoder and controls motor based on it

Gears: attached to the motor to change speed or torque



Arduino

ALL robots are controlled by a computer. The computer we're using is the Arduino Nano.

Since robots cannot think on their own like us, they have to follow specific instructions to make decisions. These instructions are given to their computer in the form of a program. The computer can then read the program and act like the robot's brain, controlling the robot based on just running each instruction in the program in sequence.

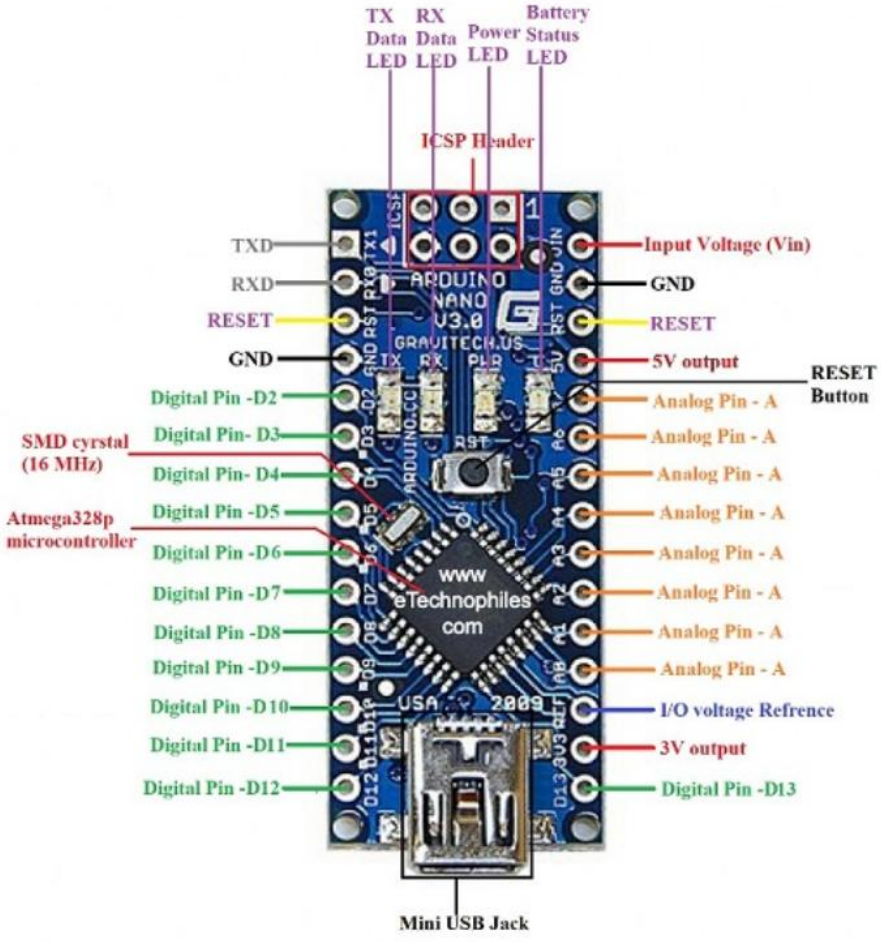
Arduino

Vin/GND: how we power the arduino

Digital pins: can be inputs or outputs, we tell them what to be in the code

Analog pins: another form of inputs we don't use these

RESET: pressing this button or using this pin erases the code from the Arduino

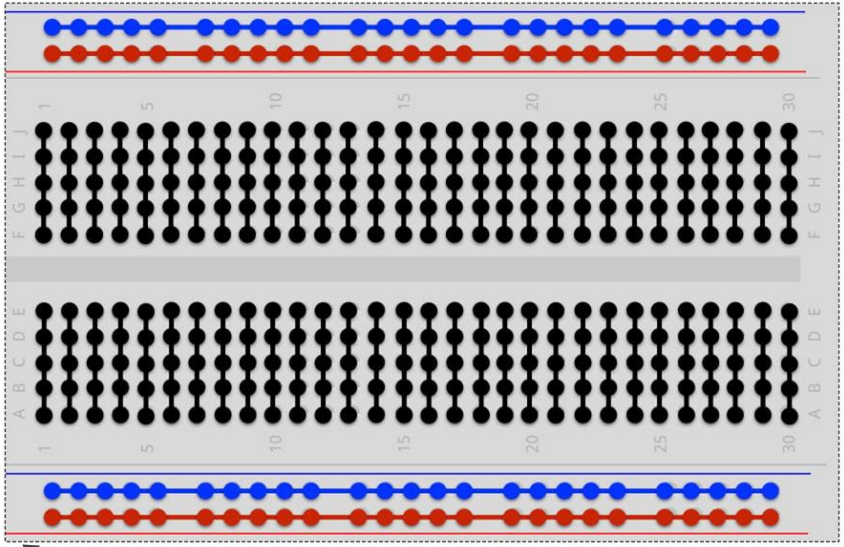


Wires/Breadboard

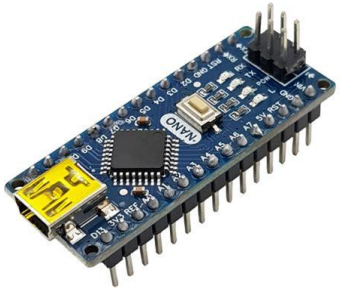
Wires act as roads for electrons to follow

Breadboards are lots of stationary wires

We can plug pins into these wires to create intersections and connect the “roads” to each other so that the signal can get where it needs to be.



Electronics for Each Group

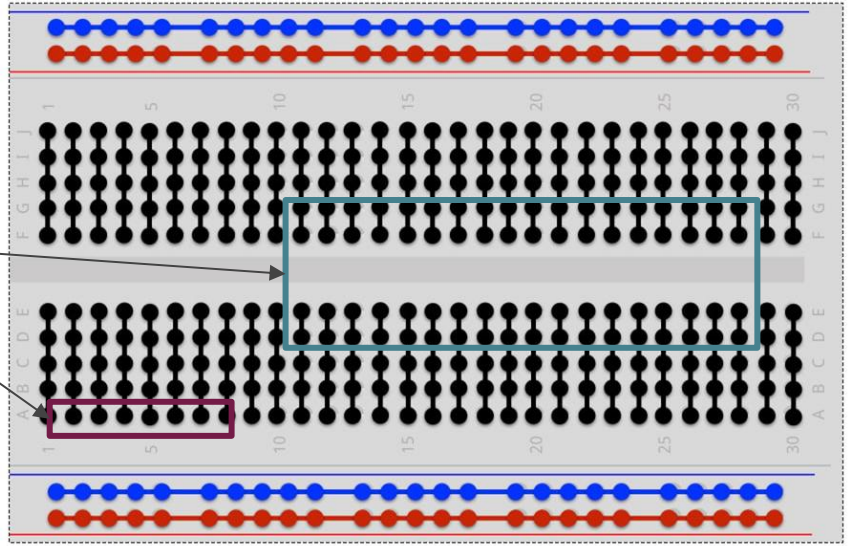


- 1 microcontroller
- 1 ultrasonic sensor
- Wires - variety of sizes

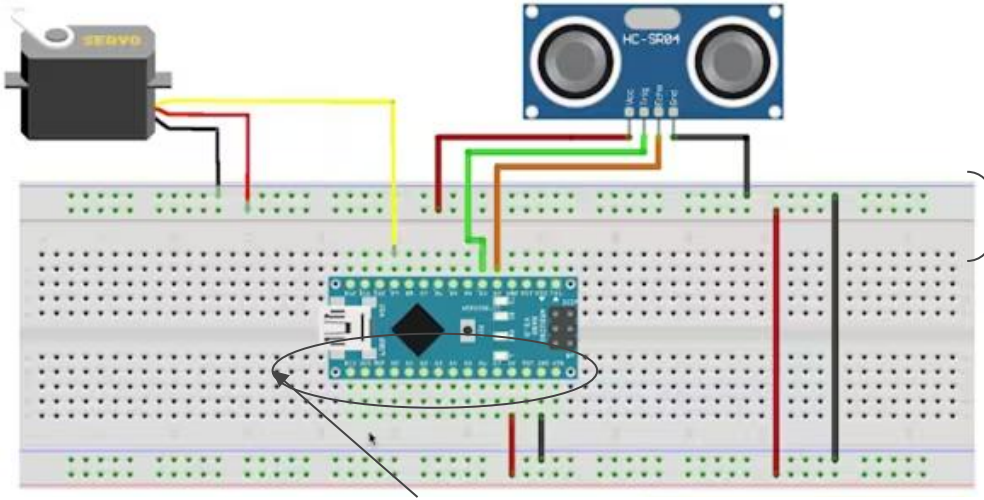
Electronics Setup

Ultrasonic sensor

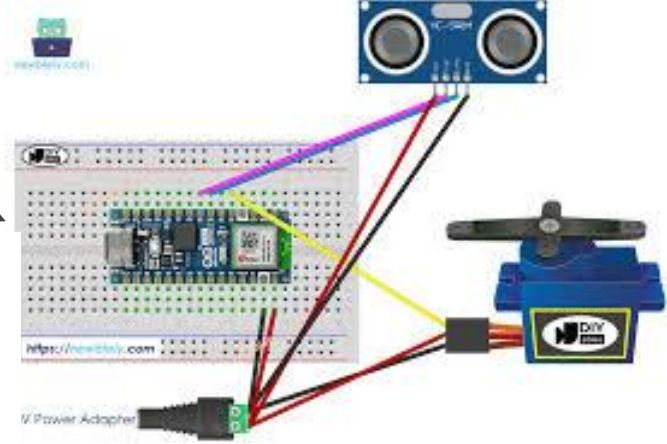
microcontroller



Wiring Diagram



Another way of wiring the same thing

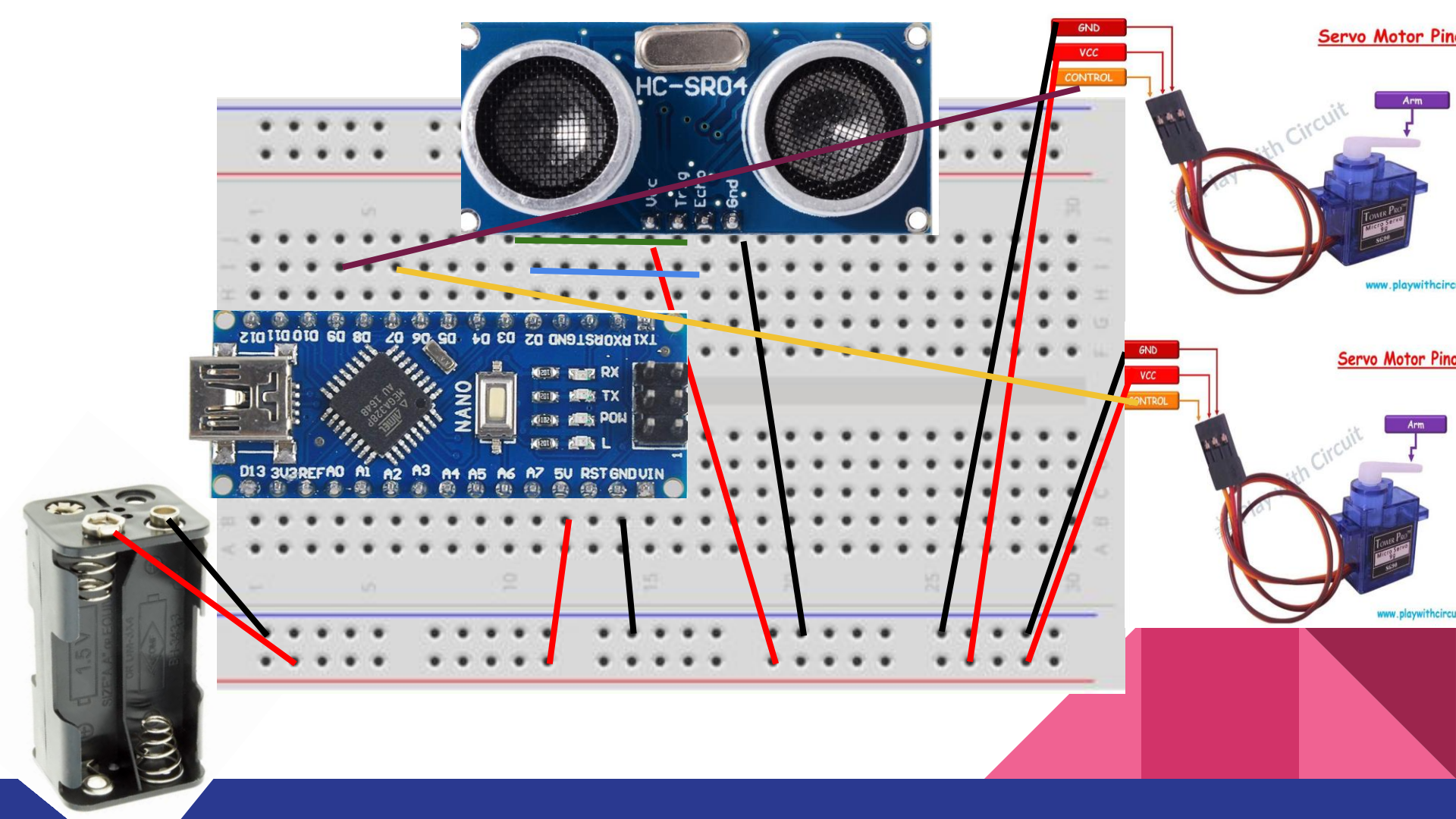


PWR / GND for:

- Both servos (output)
- Ultrasonic sensor (output)
- Arduino (output)
- Battery (input)

Servo: Signal wires (yellow) to D7, D9
Both left to D9, both right to D7

Ultrasonic sensor: TRIG to D3, ECHO to D2


HC-SR04
Vcc
Trig
Echo
Gnd
GND
VCC
CONTROL
Servo Motor Pin
Arm
www.playwithcirc
NANO
RX
TX
POW
L
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN
TX1 RX0 RST GND D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12
1.5V

Software

What is software engineering?
Designing, developing, testing, and maintaining software applications.

What parts of the robot are “software” parts?
What software things have you used before?

Software

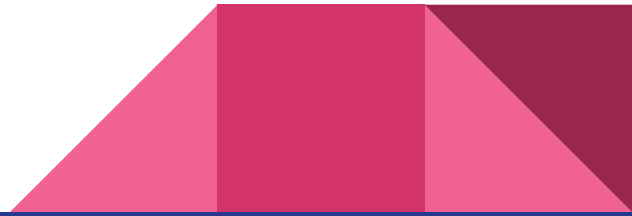
Coding language = how we tell the computer what to do (C++)

Normal parts of code:

Functions: a command that you define once, and call anywhere

Loops: something that runs more than once in a row, depending on how you call it

Variables: something that can be set to whatever value you need



Header

These are things we'll use throughout the whole code - they have to be the first things the computer is told.

```
1   #include <Servo.h>
2
3   // Create servo objects
4   Servo leftServo;
5   Servo rightServo;
6
7   // Ultrasonic sensor pins
8   const int trigPin = 3;
9   const int echoPin = 2;
10
11  // Distance threshold (in mm) to trigger turning
12  const int stopDistanceMM = 80;
```

#include : call pre-existing libraries with data we can use

Servo : object in library that has the internal functions for our motors

Const int : variable.

- "Const" means it won't change
- "int" means integer (not decimal)

Setup

Code that only has to run once

Serial.begin : shows data from microcontroller on computer when plugged in

Servo.attach : connects servo object to pin on microcontroller

pinMode : tells pins whether they should accept a sensor reading (input) or give an instruction (output)

```
14  void setup() {
15    Serial.begin(9600);
16
17    // Attach servos
18    leftServo.attach(9);
19    rightServo.attach(7);
20
21    // Set ultrasonic pin modes
22    pinMode(trigPin, OUTPUT);
23    pinMode(echoPin, INPUT);
24  }
```

Loop

Code that runs over and over again

Serial.print : send message from microcontroller to computer

If : condition - only runs inner code if the first statement is true

While: runs again and again as long as first statement is true

Delay : code pauses for set time

```
26  void loop() {
27    float distance = readDistanceMM();
28
29    Serial.print("Distance: ");
30    Serial.print(distance);
31    Serial.println(" mm");
32
33    moveForward();
34
35    if (distance > 0 && distance < stopDistanceMM) {
36
37      while (readDistanceMM() < stopDistanceMM) {
38        turnRight();  // keep turning until path clears
39        //delay(100);   // allow time for motion
40      }
41
42      delay(200); // short pause before resuming
43      moveForward(); // resume forward motion
44    }
45
46    delay(100); // reduce sensor read jitter
47  }
```

Functions - Move Forwards

```
50  void moveForward() {
51      for (int pos = 30; pos > 0; pos -= 1) { // goes from 0 degrees to 180 degrees
52      // in steps of 1 degree
53      leftServo.write(pos);              // tell servo to go to position in variable 'pos'
54      delay(5);                       // waits 5 ms for the servo to reach the position
55    }
56        for (int pos = 30; pos > 0; pos -= 1) { // goes from 0 degrees to 180 degrees
57      // in steps of 1 degree
58      rightServo.write(pos);              // tell servo to go to position in variable 'pos'
59      delay(5);                       // waits 5 ms for the servo to reach the position
60    }
61
62  }
```

For : runs a number of times that we specify

Servo.write : tells servo to go to angle…For robots that walk

Functions - Turn Right

```
69  void turnRight() {
70      for (int pos = -10; pos <= 10; pos += 1) { // goes from 0 degrees to 180 degrees
71        // in steps of 1 degree
72        leftServo.write(pos);              // tell servo to go to position in variable 'pos'
73        delay(5);
74      }
75
76      for (int pos = 30; pos > 0; pos -= 1) { // goes from 0 degrees to 180 degrees
77        // in steps of 1 degree
78        rightServo.write(pos);              // tell servo to go to position in variable 'pos'
79        delay(5);
80                                  // waits 15 ms for the servo to reach the position
81    }
82  }
```

Same building blocks as moveForward, different instructions

Functions - Distance

Float : variable, can be decimal

digitalWrite: send value to pin

pulseIn: read value from pin

Return: when this function is called, a value passes to the main code

```
84  // === Distance Reading Function ===
85  float readDistanceMM() {
86    long duration;
87    float distance;
88
89    // Trigger pulse
90    digitalWrite(trigPin, LOW);
91    delayMicroseconds(2);
92    digitalWrite(trigPin, HIGH);
93    delayMicroseconds(10);
94    digitalWrite(trigPin, LOW);
95
96    // Read echo
97    duration = pulseIn(echoPin, HIGH, 20000); // 20ms timeout
98
99    // Convert to mm
100   distance = duration * 0.343 / 2;
101
102   // Handle timeout
103   if (duration == 0) {
104     return -1; // invalid reading
105   }
106   return distance;
107 }
```

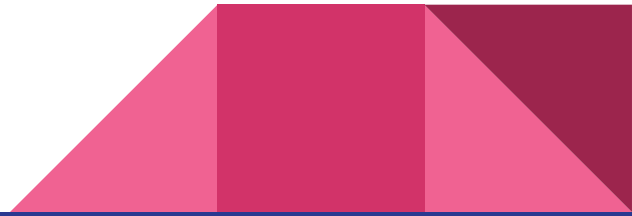
Robots that drive

```
#include <Servo.h> // Include the Servo library

// Define pin numbers for the ultrasonic sensor
#define trigPin 7
#define echoPin 9

// Create a servo object
Servo ServoLeft;
Servo ServoRight;

// Define pin number for the servo motor
const int Left = 4;
const int Right = 3;

// Define variables for distance calculation
long duration;
float distanceInch;

void setup() {
  // Set up the ultrasonic sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Set up the servo motor
  ServoLeft.attach(Left);
  ServoRight.attach(Right);

  // Begin serial communication for debugging
  Serial.begin(9600);
}
```

```
void loop() {
  // Clear the trigPin for a clean reading
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Send a 10 microsecond pulse to trigger the sensor
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the echoPin and get the pulse duration
  duration = pulseIn(echoPin, HIGH);

  // Convert duration to distance in inches
  // Distance = (duration / 2) / 74 (since 1 inch is approx. 74 microseconds)
  distanceInch = duration / 148.0;

  Serial.print("Distance: ");
  Serial.print(distanceInch);
  Serial.println(" inches");

  // Check if the distance is less than 2 inches
  if (distanceInch < 2) {
    Serial.println("Obstacle detected! Turning left...");
    ServoLeft.write(90); // Turn the servo to the specified left angle
    ServoRight.write(90);
    delay(500);
  } else {
    // If no obstacle, keep the servo in the starting position
    ServoLeft.write(0);
    ServoRight.write(0);
  }
}
```

Activities

- Build a maze and navigate it
- Climb a hill – how steep can your robot climb?
- Race other robots
- Robot Demolition Derby



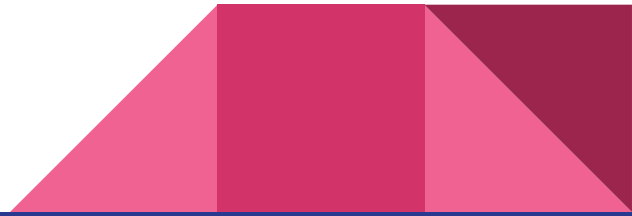
Run the robots!

How did your group’s robot do in the maze and the other activities?

How could you change it to do better?

What else could we do to our robots?

What else could we do with robots in general?



Other resources

- https://www.sciencebuddies.org/blog/robotics-lessons
- https://www.monolithicpower.com/en/learning/mpscholar/electric-motors/servo-motors/basics-of-servo-motor-technology?srsltid=AfmBOoprvcndi3RbWcTKrbxPuhtqWhcLokxnBIAornCqsaic4-mkz-6K
- https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p058/robotics/arduino-walking-robot
- https://www.circuito.io/app?components=514,11022,13959,2345678