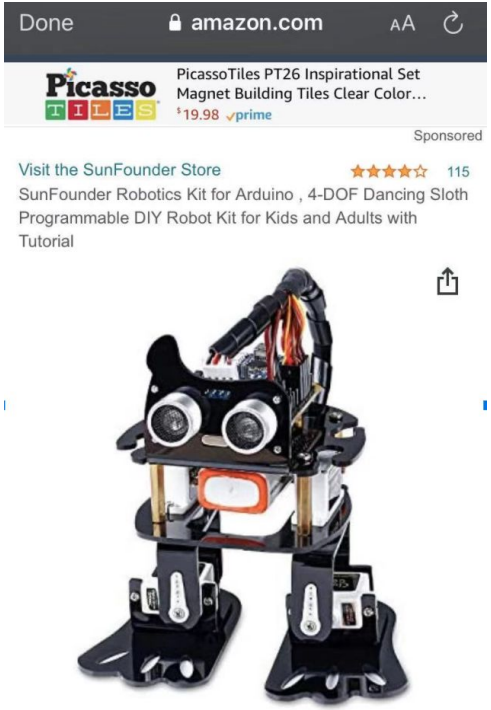
Dancing Robot

Benny Cai / Stella Zhang
Instructor: Truong Nguyen
11.28.2020
ECE 196, Fall 2020



A screenshot taken from amazon.com

Introduction

- **Goal**: build and program a dancing robot using Arduino.

  The robot will be able to:

1. dance (code designed by SunFounder, not us)
2. move forward, and turn right when it senses an obstacle within 5 cm (code designed by us)

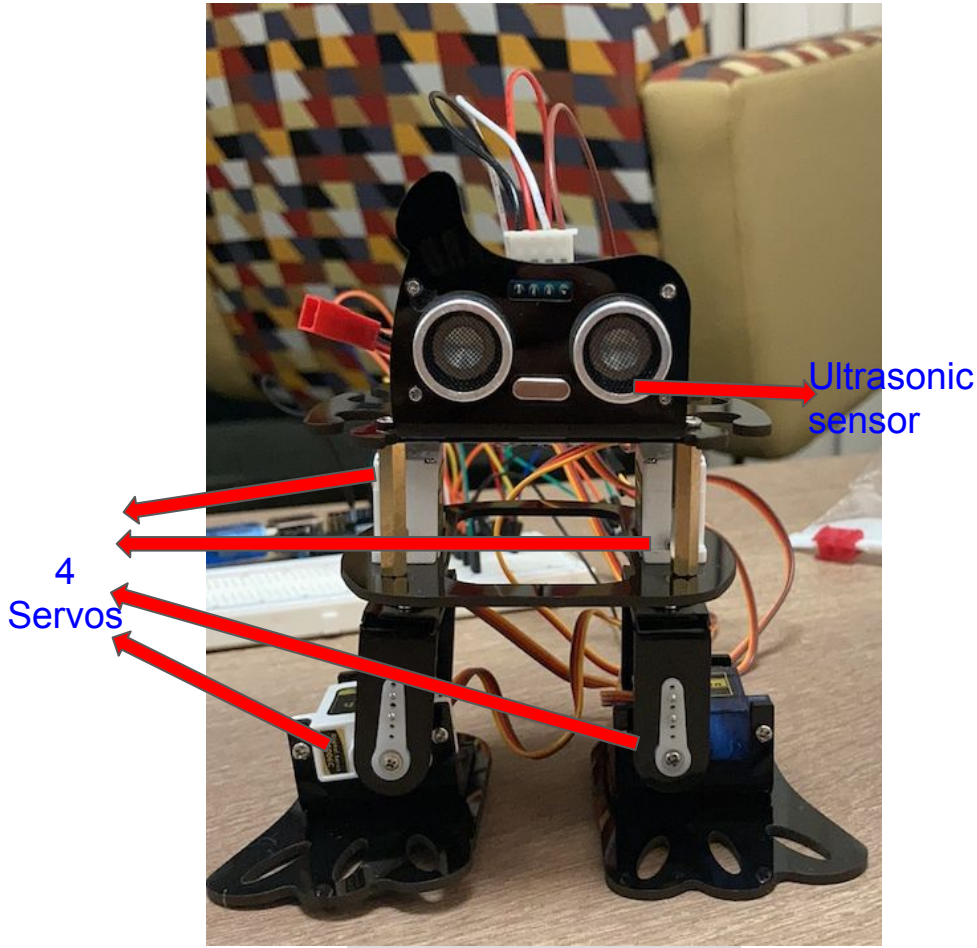
Problem Solving Approach

- Understand the robot’s structure
  - Two legs and feet and a pair of eyes
- Test each component
  - Four servos and an ultrasonic sensor
- Build the robot
  - Calibrate the internal angle of each servo
- Design and connect the circuit
  - All components connected in series
- Program the robot
  - Implement the dancing code.
  - Design the code for moving forward and turning right.
- Upload and make it dance/move!

The robot’s structure

- Two legs and feet: controlled by four servos
  - able to perform various movements
  - Each servo is able to rotate from 0° to 180°. (This is the built-in maximum range.)
  - Two servos on the feet rotate vertically. (From the angle of the image, rotates clockwise and counterclockwise)
  - Two servos on the body/leg rotate horizontally. (From the angle of the image, rotates leftwards and rightwards)

- Eyes: an ultrasonic distance sensor
  - able to sense distance from obstacle

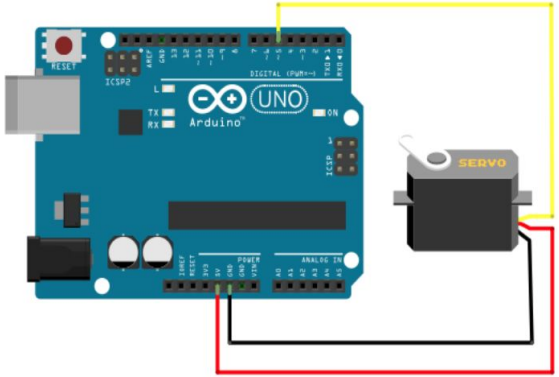


Our Finished Product

Test servo:

- We tested the four servos individually by connecting each one of them with our Arduino Uno board according to the circuit on the right, and programming them using the code on the right. As the short clip indicates, as an example, each one of them is able to rotate from 0° to 180°.

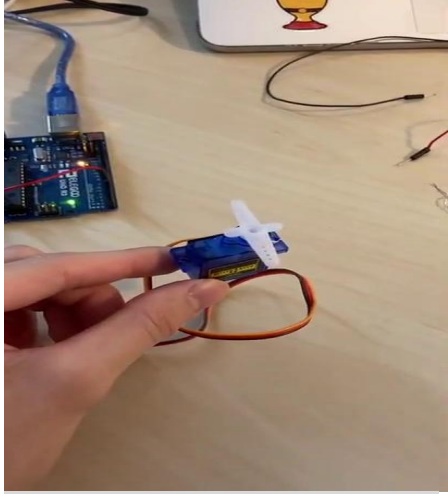
- **Remark**: one mistake we made was that we didn't test the case when all four servos are powered by a 9V battery. This issue would be elaborated on a later slide.



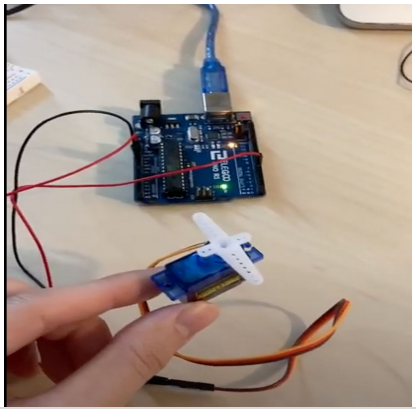
Circuit referenced from "Introduction to Arduino" by PIB

```
#include <Servo.h>

int servo_pin = 5;

Servo servoMain; // Create object

void setup() {
  servoMain.attach( servo_pin );
}

void loop() {
  servoMain.write( 180 );  // Highest angle delay(1000);
  delay(1000);
  servoMain.write( 0 ); // Lowest angle delay(1000);
  delay(1000);
}
```

The code we used for testing the servo



A short clip of testing a servo

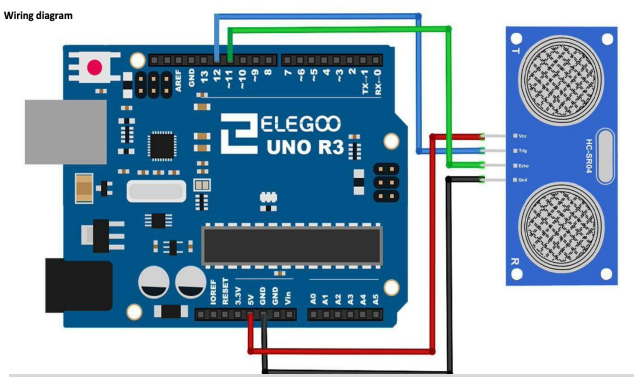


A screenshot in case the video doesn't work

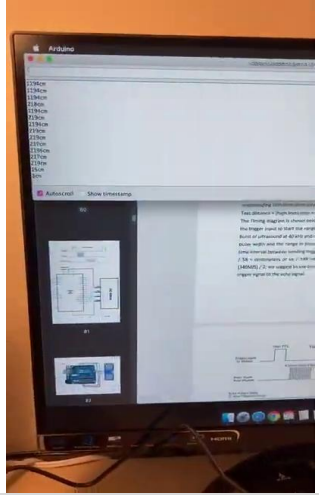
Test ultrasonic sensor:

- On the right are the circuit and the code we used for testing our ultrasonic sensor.

- To test its functionality, we put a hand in front of the sensor to simulate an obstacle. From the video/picture, we can see that the serial monitor is able to print out its distance from the hand continuously, indicating it's functioning properly.



Circuit referenced from "THE MOST COMPLETE STARTER KIT TUTORIAL FOR UNO" by Elegoo

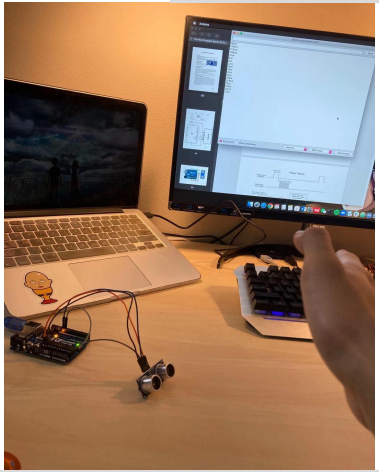


A short clip of testing the sensor

SR04_Example | SR04.cpp | SR04.h

```
//www.elegoo.com
//2016.12.08
#include "SR04.h"
#define TRIG_PIN 12
#define ECHO_PIN 11
SR04 sr04 = SR04(ECHO_PIN,TRIG_PIN);
long a;

void setup() {
   Serial.begin(9600);
   delay(1000);
}

void loop() {
   a=sr04.Distance();
   Serial.print(a);
   Serial.println("cm");
   delay(1000);
}
```

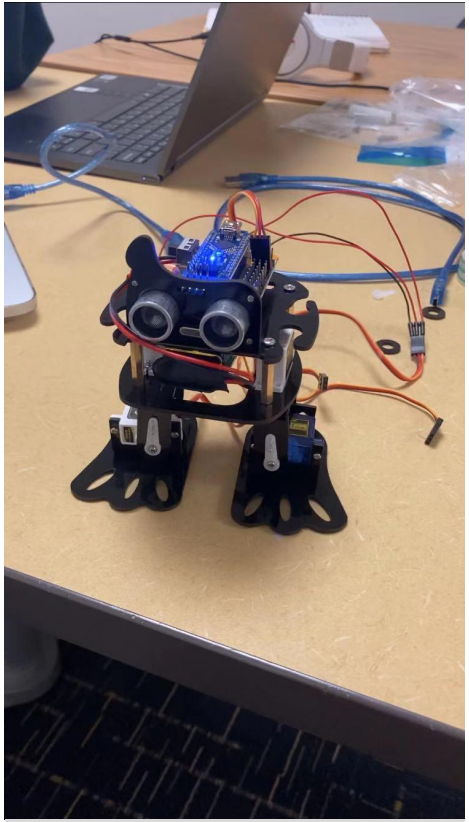
Code referenced from www.elegoo.com



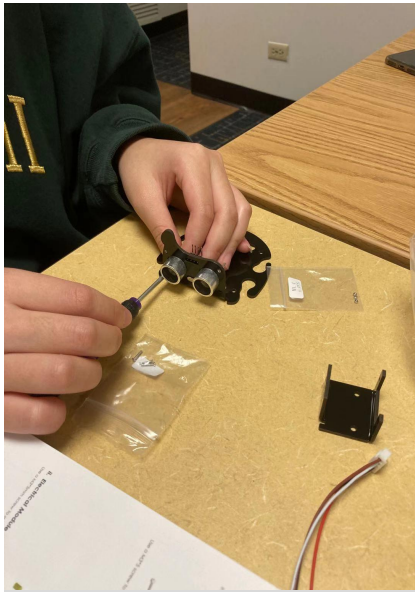
A picture in case the video doesn't work

Build the robot

- We spent quite a long time assembling the robot because the screws are so small and it's so hard to assemble them :(
- **Remark**: One mistake we made during the installation was that we didn't calibrate the internal angle of the servos before installing its legs and feet.
- We want its feet to land perfectly on the ground when the internal angle of the servos on its **feet** is 90° (in the middle), so its feet are able to rotate clockwise and counterclockwise each by 90° at max, (as the rotation range of a servo is 0°~180°).
- We want its feet to point towards the front when the internal angle of the servos on its **leg** is 90°, so its leg are able to rotate leftwards and rightwards each by 90° at max.
- However, we didn't make sure the internal angles of the four servos are 90° when the feet and legs are in the right position. So, we had to disassemble the robot :(



The first version of our robot

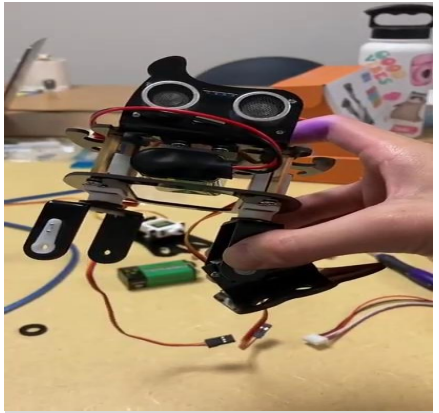


Us building the robot

Calibrate the internal angle of each servo

- On the right is the code we used for determining the internal 90° of the servo. From the code we know that there are four rotations in each iteration (90°,0°,150°,0°), and the servo would reach 0° position twice. For the other two rotations, the one that is physically closer to the 0° position is going to be 90°.
- After we found the internal 90° of the servo, we took off the corresponding leg/feet, and reinstalled it with the right direction, and we're done with the calibration!
- On the right is two example clips of us calibrating the robot's feet.
- Before calibrating the **left** foot, we can see that when the servo reaches 90°, the left foot didn't point directly to the below.
- After calibrating the **right** foot, using the same code, we can see that the right foot landed perfectly on the ground when the servo reaches 90°.

```
void loop(){
    myservo1.write(90);

    delay(1000);
    myservo1.write(0);

    delay(1000);
    myservo1.write(150);

    delay(1000);
    myservo1.write(0);

    delay(1000);
```

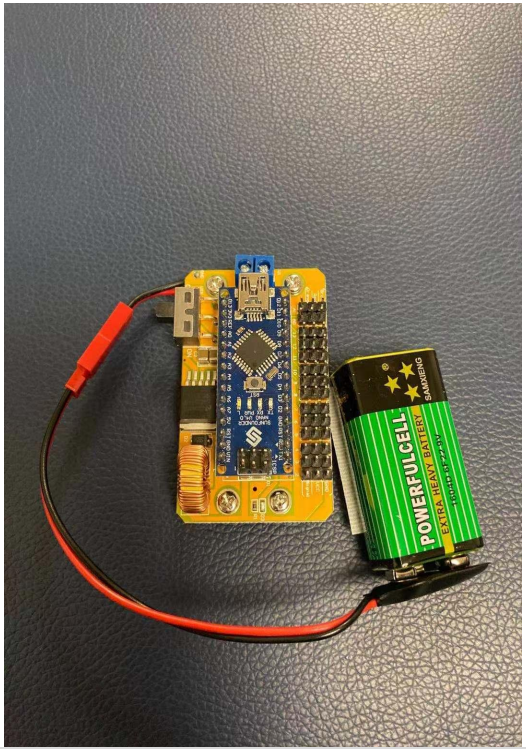


before calibrating the **left** foot

after calibrating the **right** foot

A problem met when connecting the circuit: NOT ENOUGH POWER :<

- Initially, we used the Arduino Nano board provided in the kit and a 9V battery to power the circuit.
- When we connected all four servos to the Nano board and tested them, it turned out that the 9V battery are not able to power all four of them.
- The 9V battery was able to power one servo and make it move during our testing round, but it failed to make four servos move at the same time!
- We had to abandon the Nano board and used our own Uno board.



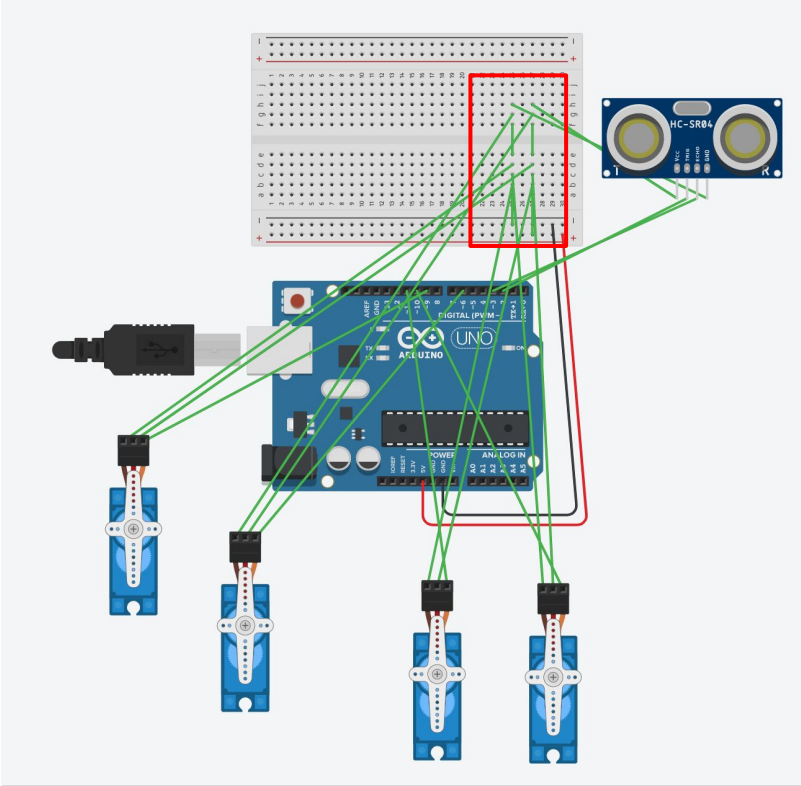
The Nano board provided and a 9V battery

Design the circuit

- The Uno board doesn't have as many pins as the Nano does, so we have to use a breadboard.

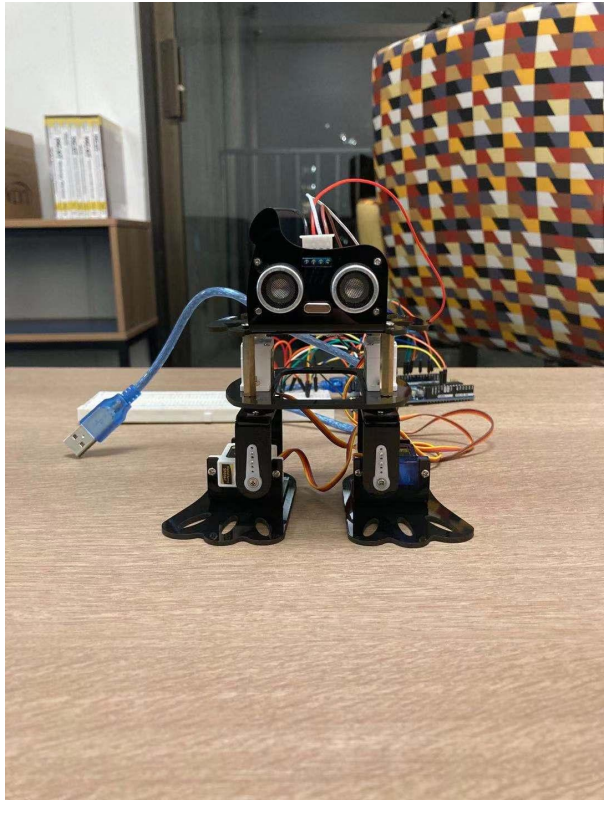
- The use of the breadboard (Physics concept):
  - We want to make sure that the four servos and the ultrasonic sensor are connected in **parallel**.

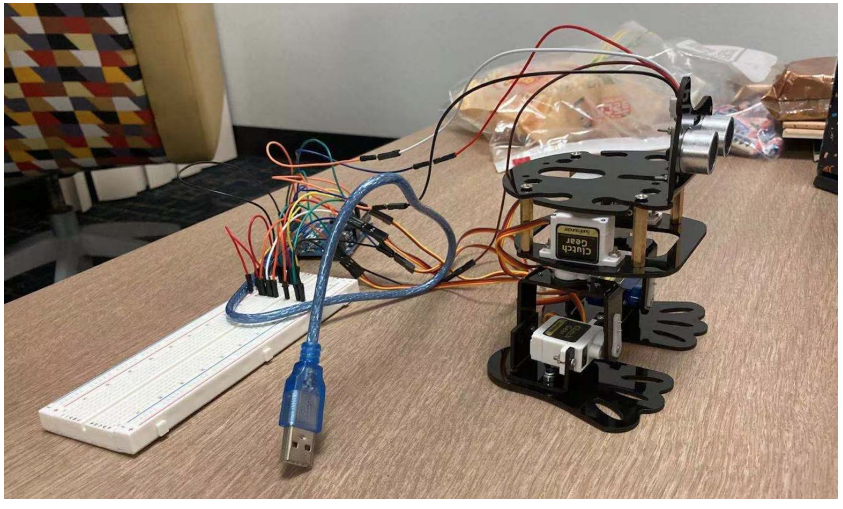
- We apologize that the circuit we designed looks extremely messy. But, ignoring all the messy wiring, the important part is what's inside the red square: all the components are connected within the same two vertical lines: one for the ground, the other for the power. This indicates that all the components are connected in parallel.

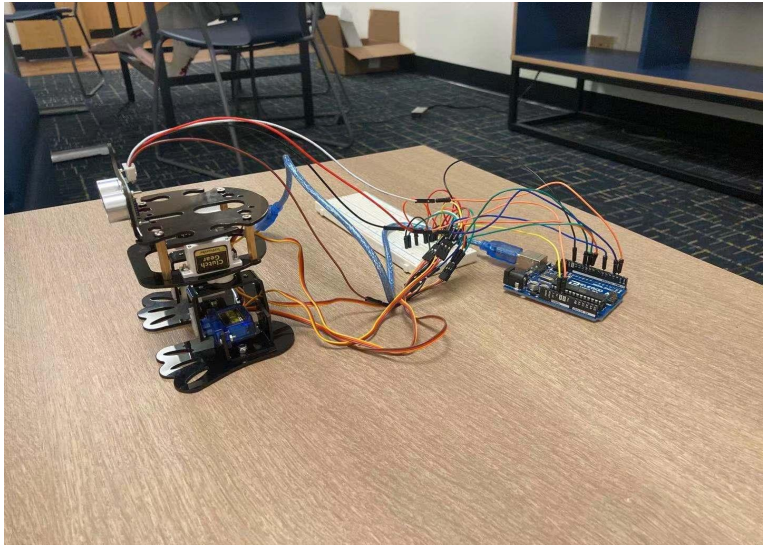


The circuit we designed by ourselves using TinkerCad

Final product:







Program the robot: Dancing

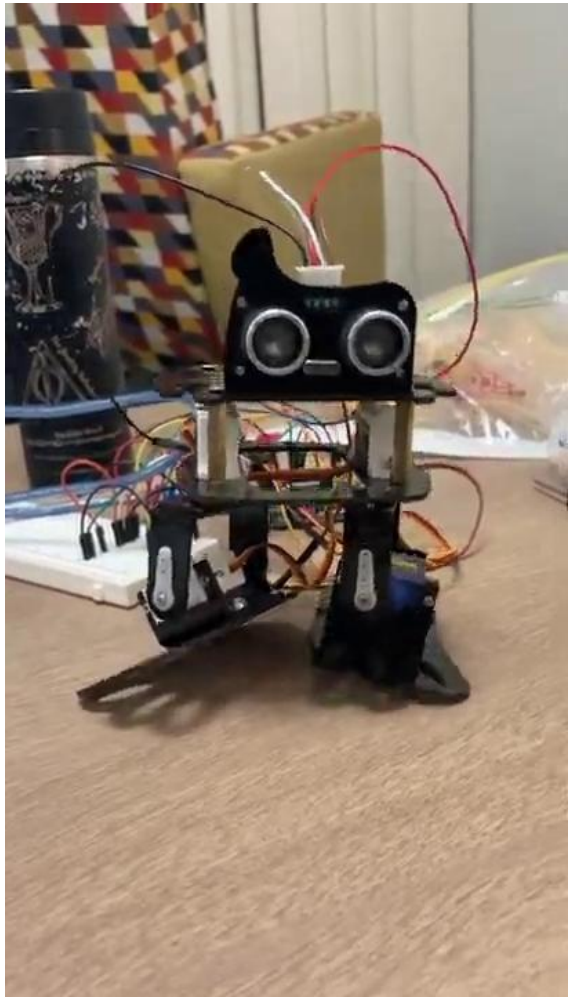
- Code referenced from https://www.sunfounder.com/.

- We believe Stella will be able to design a more beautiful dancing because she is a professional hip-hop dancer in UCSD!

```
void Dancing3(int Times = 1, int Vel = 40, int Delay = 250, int low = 0, int high = 0)
{
    for(int time3 = 0; time3 < Times; time3++) {
        for(int z=0; z<6; z++) {
            if ( time3 > 1 && time3 < 4) {
            vel_Dance3 = Vel;
            delay_Dance3 = Delay;
            }
            else {
            vel_Dance3 = 40;
            delay_Dance3 = 200;
            }

            RU.slowmove (array_cal[0] + array_dance3[z][0] , vel_Dance3);
            RL.slowmove (array_cal[1] + array_dance3[z][1] , vel_Dance3);
            LU.slowmove (array_cal[2] + array_dance3[z][2] , vel_Dance3);
            LL.slowmove (array_cal[3] + array_dance3[z][3] , vel_Dance3);
            delay(delay_Dance3);
        }
    }
    for(int z=6; z<8; z++) {
            RU.slowmove (array_cal[0] + array_dance3[z][0] , vel_Dance3);
            RL.slowmove (array_cal[1] + array_dance3[z][1] , vel_Dance3);
            LU.slowmove (array_cal[2] + array_dance3[z][2] , vel_Dance3);
            LL.slowmove (array_cal[3] + array_dance3[z][3] , vel_Dance3);
            delay(delay_Dance3);
        }
```

A screenshot of the dancing code

Dancing video



Program the robot: move forward and backward

- Set up

```
#include <Servo.h>
#include "SR04.h"
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;

#define TRIG_PIN 2
#define ECHO_PIN 3
SR04 sr04 = SR04(ECHO_PIN,TRIG_PIN);
long a;
int i;

void setup(){
  Serial.begin(9600);
  myservo1.attach(6);    // LL
  myservo2.attach(9);    // RL
  myservo3.attach(10);     // LU
  myservo4.attach(11);     // RU
```

Initialize and attach the four servos

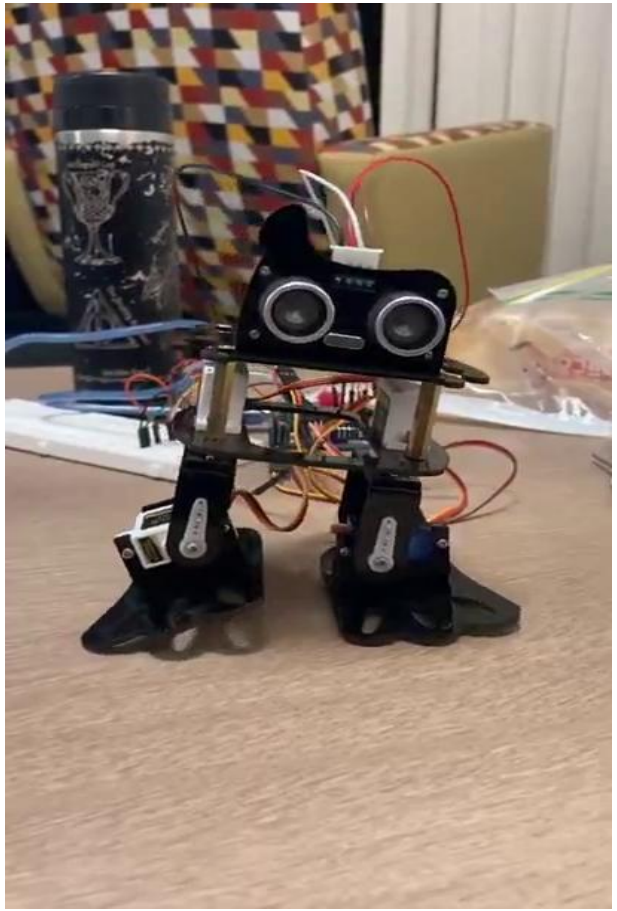
```
void loop(){
    a=sr04.Distance();
    Serial.print(a);
    Serial.println("cm");
    delay(1000);
```

Sensing distance

Move forward

```
else {
   // move forward
  // move left leg
  for (i = 90; i < 130; i++) {
    myservo2.write(i);
    myservo1.write(i);
    delay(20);
  }

  for (i = 90; i > 60; i--) {
    myservo3.write(i);
    delay(20);
  }

  for (i = 130; i > 90; i--) {
    myservo2.write(i);
    myservo1.write(i);
    delay(20);
  }

  for (i = 60; i < 90; i++) {
    myservo3.write(i);
    delay(20);
  }
```

```
// move right leg
for (i = 90; i > 50; i--) {
  myservo2.write(i);
  myservo1.write(i);
  delay(20);
}

for (i = 90; i < 120; i++) {
  myservo4.write(i);
  delay(20);
}

for (i = 50; i < 90; i++) {
  myservo2.write(i);
  myservo1.write(i);
  delay(20);
}

 for (i = 120; i > 90; i--) {
  myservo4.write(i);
  delay(20);
}
```



Sense obstacle and move backward

```
if (a < 5) {
    // move backward
   // move left leg
  for (i = 90; i < 120; i++) {
    myservo2.write(i);
    myservo1.write(i);
    delay(20);
  }

  for (i = 90; i < 120; i++) {
    myservo3.write(i);
    delay(20);
  }

  for (i = 120; i > 90; i--) {
    myservo2.write(i);
    myservo1.write(i);
    delay(20);
  }

  for (i = 120; i > 90; i--) {
    myservo3.write(i);
    delay(20);
  }
```

```
// move right leg
for (i = 90; i > 50; i--) {
  myservo2.write(i);
  myservo1.write(i);
  delay(20);
}

for (i = 90; i > 60; i--) {
  myservo4.write(i);
  delay(20);
}

for (i = 50; i < 90; i++) {
  myservo2.write(i);
  myservo1.write(i);
  delay(20);
}

 for (i = 60; i < 90; i++) {
  myservo4.write(i);
  delay(20);
}
```

Thank you!

Special thanks to our TA, Phuong Truong, for providing us the kit!