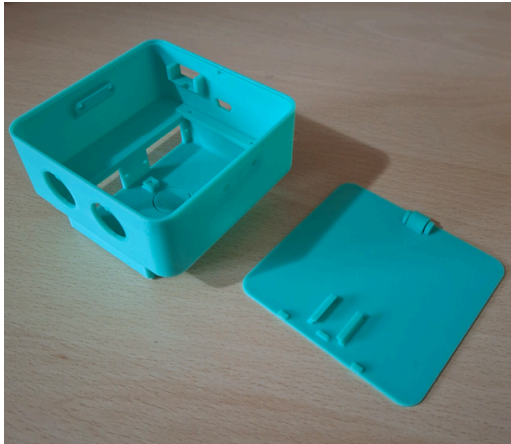


COBOT: learn robotics through 3d printing

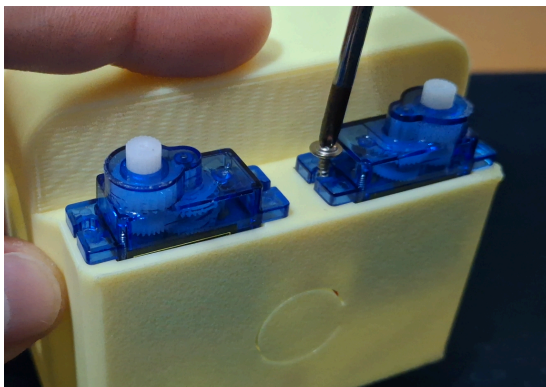
Assembly guide | Cobot v1.0

1. Print the chassis, cover and legs



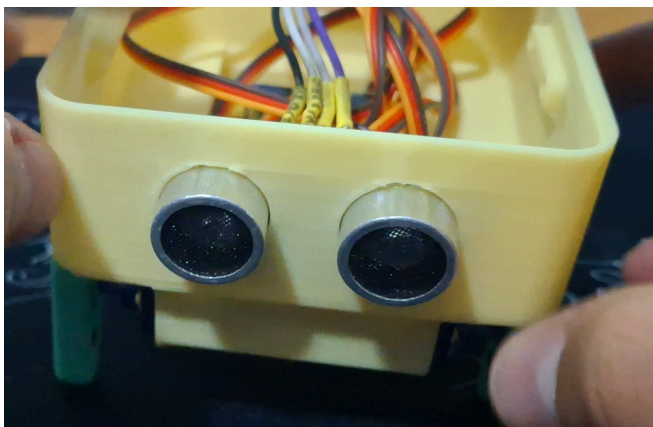
2. Attach the 4 servos

You can use the servo screws or another type of 2mm (diameter) screws



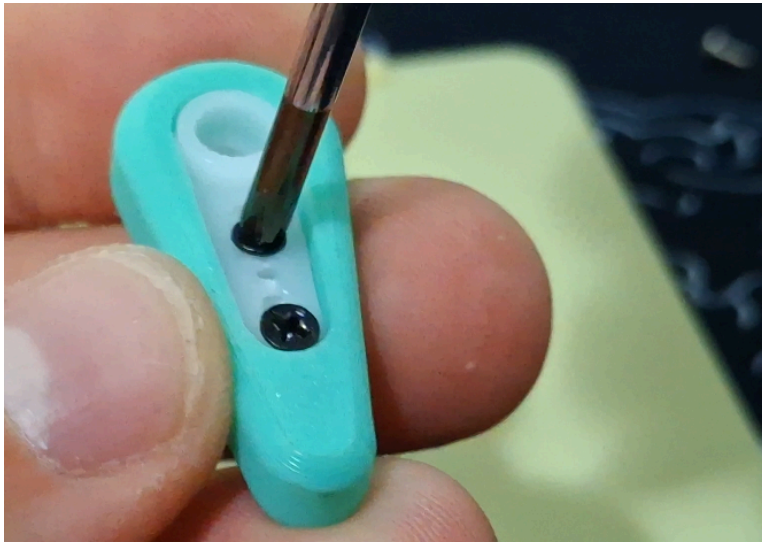
3. Mount HC-SR04

Apply some glue or tape for mounting the distance sensor in front of robot



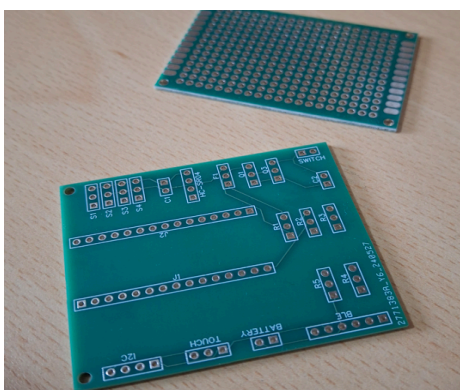
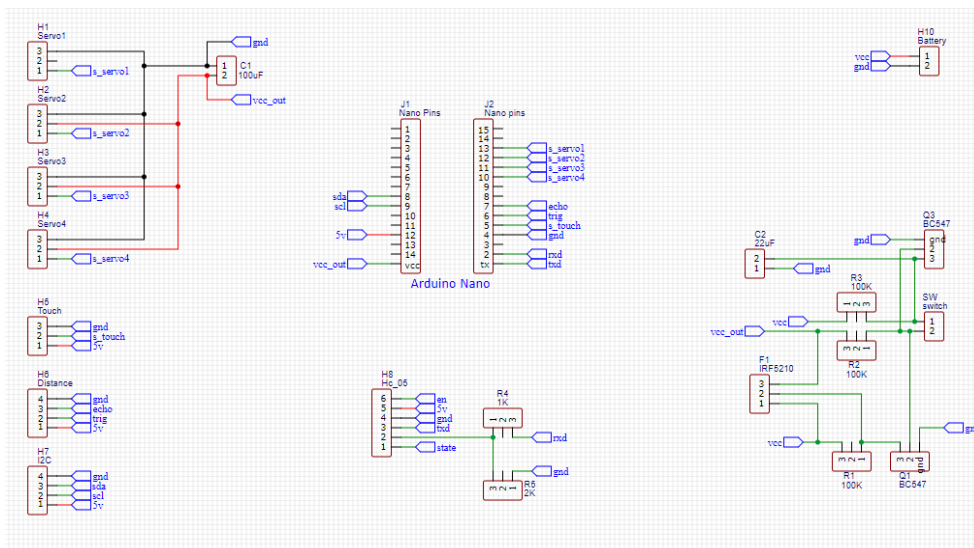
4. Mount Legs

use 2mm screw for attach servo support to legs (there are small hole for easy install)



5. Electrical connections

You can make your own simpler version by following the sheet. Use a breadboard, make your own pcb or use only wire (as you want). In the build video I use only wire connections.



6. Arduino Code

```
/*

#####
#####

#####
#####

                        Made by Luca Dilo 03/07/2024 v.1

#####
#####

#####
#####

***** algorithm = 0 *****
[ setup: set all servo to std pos
  use this in build face to align servo 90 deg pos and legs ]

***** algorithm = 1 *****
[ all servos 90 -> (0, 180), but high_left = low_right and high_right = low_left, (hl, lr) =
-(hr-lr)
  by watching BD spot movements: alternate high and low, left and right ]

***** algorithm = 2 *****
[ hight legs = low legs ]

***** algorithm = 3 *****
[ little jump ]

***** algorithm = 4 *****
[ bow ]

***** algorithm = 5 *****
[ hello with high_left leg ]

***** algorithm = 6 *****
[ example of combining movements ]
*/

#include <Servo.h> // ***** INSTALL SERVO LIB *****
#include "Timer.h" // ***** INSTALL TIMER LIB *****

Timer timer;
```

```

Servo high_left;
Servo high_right;
Servo low_left;
Servo low_right;

int pos = 0;
int pos2 = 0;

int vel = 5; // delay between movements (less is more vel.)

const int trigPin = 4;
const int echoPin = 5;

long duration;
int distance;

int algorithm_selected = 6; // select algorithms

void setup() {
  Serial.begin(9600);

  low_left.attach(12);
  high_right.attach(10);
  high_left.attach(11);
  low_right.attach(9);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.println("Moving all servo in standard position ...");

  // move all servo in std pos
  low_left.write(90);
  high_right.write(90);
  high_left.write(90);
  low_right.write(90);

  // wait 1 sec then start algorithm
  delay(1000);

  Serial.println("done");

  Serial.println("Start Timer ...");
  timer.start();
  Serial.println("done");

  Serial.println("Starting loop algorithm");
}

```

```
void loop() {  
  if(algorithm_selected == 0) {  
  
    resetAllServoPos();  
  
  } else if(algorithm_selected == 1) {  
  
    alternateWalk();  
  
  } else if(algorithm_selected == 2) {  
  
    highAndLowSynchroWalk();  
  
  } else if(algorithm_selected == 3) {  
  
    littleJump();  
  
  } else if(algorithm_selected == 4) {  
  
    bow();  
  
  } else if(algorithm_selected == 5) {  
  
    hello();  
  
  } else if(algorithm_selected == 6) {  
  
    if(timer.read() < 3000) {  
      hello();  
    } else if(timer.read() < 6000) {  
      littleJump();  
    } else {  
      resetAllServoPos();  
    }  
  
  }  
  
}
```

```
void readDistance() {  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
  
    duration = pulseIn(echoPin, HIGH);  
    distance = duration * 0.034 / 2;  
  
    Serial.print("Distance: ");  
    Serial.println(distance);  
}
```

```
void littleJump() {  
    low_right.write(180);  
    low_left.write(0);  
    high_right.write(0);  
    high_left.write(180);  
  
    delay(1000);  
  
    low_right.write(90);  
    low_left.write(90);  
    high_right.write(90);  
    high_left.write(90);  
  
    delay(100);  
}
```

```
void bow() {  
    low_right.write(90);  
    low_left.write(90);  
    high_right.write(90);  
    high_left.write(90);  
  
    delay(1000);  
  
    high_right.write(0);  
    high_left.write(180);  
  
    delay(2000);  
  
    high_right.write(90);  
    high_left.write(90);  
  
    delay(4000);  
}
```

```

void hello() {
  low_right.write(90);
  low_left.write(90);

  high_right.write(110);

  high_left.write(0);
  delay(300);
  high_left.write(40);
  delay(300);
}

void alternateWalk() {
  for (pos = 0; pos <= 180; pos += 1) {
    pos2 = map(pos, 0, 180, 180, 0);

    low_right.write(pos2);
    high_right.write(pos);
    low_left.write(pos2);
    high_left.write(pos);

    delay(vel);
  }

  for (pos = 180; pos >= 0; pos -= 1) {
    pos2 = map(pos, 180, 0, 0, 180);

    low_right.write(pos);
    high_right.write(pos2);
    low_left.write(pos);
    high_left.write(pos2);

    delay(vel);
  }
}

```

```

void highAndLowSynchroWalk() {

    for (pos = 180; pos >= 0; pos -= 1) {
        pos2 = map(pos, 180, 0, 0, 180);

        low_right.write(pos2);
        low_left.write(pos);

        delay(vel);
    }

    delay(500);

    for (pos = 0; pos <= 180; pos += 1) {
        pos2 = map(pos, 0, 180, 180, 0);

        low_right.write(pos2);
        low_left.write(pos);

        high_right.write(pos);
        high_left.write(pos2);

        delay(vel);
    }

    for (pos = 180; pos >= 0; pos -= 1) {
        pos2 = map(pos, 180, 0, 0, 180);

        high_right.write(pos);
        high_left.write(pos2);

        delay(vel);
    }
}

void resetAllServoPos() {
    low_left.write(90);
    high_right.write(90);
    high_left.write(90);
    low_right.write(90);
}

```