ACEBOTT

Biped Robot Tutorial

Perface

Our Company

ACEBOTT STEM Education Tech Co.,Ltd

Founded in China's Silicon Valley in 2013, ACEBOTT is a STEM education solution leader We have a team of 150 individuals, including members from research and development, sales, and logistics. Our goal is to provide high-quality STEM education products and services to our customers. We are working together with STEM education experts and our business partners to produce successful STE products together Our self-own factory also provides CEM services for our clients including Logo customization on product packaging and PCB.

Our Tutorial

This course and Biped robot learning kit are designed for children and teenagers aged 8 and above. It aims to provide a deeper understanding of the ESP32 controller board, Biped robot knowledge, and electronic hardware. If you want to learn about Biped robot, this kit offers knowledge and operational steps to help you build your own robot arm.

Through this kit, you can:

1. Learn how to effectively use the ESP32 controller board, including downloading code, understanding its features, and coding in the Arduino IDE.
2. Gain a solid programming foundation in the basics of the C languages, as ESP32 utilizes the simplified C/C++ programming languages to control circuits and sensors.
3. Explore the working principles of the servo module and understand the collaborative work of multiple servos in the Biped robot project.
4. Enhance your maker skills by building your own Biped robot using the ACEBOTT kit through step-by-step tutorials.

5. Realized basic mobile movements, comprehensive dance movements, web control, App control and other basic functions in the Biped robot project.
6. Gain a comprehensive understanding of Biped robot concepts, preparing for more advanced learning in the future.

Overall, the ACEBOTT Biped robot is a learning kit designed specifically for beginners and is based on the ESP32. Using this kit, users can gain a comprehensive understanding of the controller board and servos in a Biped robot. By following the tutorials provided in the kit, students of different age groups can acquire valuable knowledge about Biped robot and successfully build their own robot arm projects.

Customer service

ACEBOTT is a dynamic and fast-growing STEM education technology company that strives to offer excellent products and quality services that meet your expectations. We value your feedback and encourage you to drop us a line at support@acebott.com with any comments or suggestions you may have.

Our experienced engineers are dedicated to promptly addressing any problems or questions you may have about our products. We guarantee a response within 24 hours during business days.

Follow Us

Scan the QR codes to Follow Us for troubleshooting & the latest news.
We have a very large community that is very helpful for troubleshooting and we also have a support team at the ready to answer any questions.

ACEBOTT FB Group QR Code YouTube QR Code

Contents

Lesson 1 Introduction To Biped Robot 1

Ⅰ.Knowledge Of Hardware 2

Ⅱ.The Bipedal Robot Movement Principle 4

Lesson 2 Basic Motion Of Biped Robot 7

Ⅰ.Forward And Backward Movement Of The Biped Robot 7

Ⅱ.Left And Right Rotation Of The Biped Robot 11

Ⅲ.The Serial Port Control Robot Movement 15

Lesson 3 The Following Function Of Biped Robot 19

Ⅰ.Ultrasonic Sensor 19

Ⅱ.Robot Follow Function Program 20

Lesson 4 Obstacle Avoidance Of Biped Robot 23

Ⅰ.Biped Robot Obstacle Avoidance Program 23

Lesson 5 Biped Robot Dance Moves 1 27

Ⅰ.Robot Dance Program 1 27

Lesson 6 Biped Robot Dance Moves 2 30

Ⅰ.Robot Dance Program 2 30

Lesson 7 Web Control Of Biped Robot 33

Ⅰ.Web Control Program 33

Ⅱ.Login Page 34

Lesson 8 Biped Robot APP Control 36

Ⅰ.APP Download 36

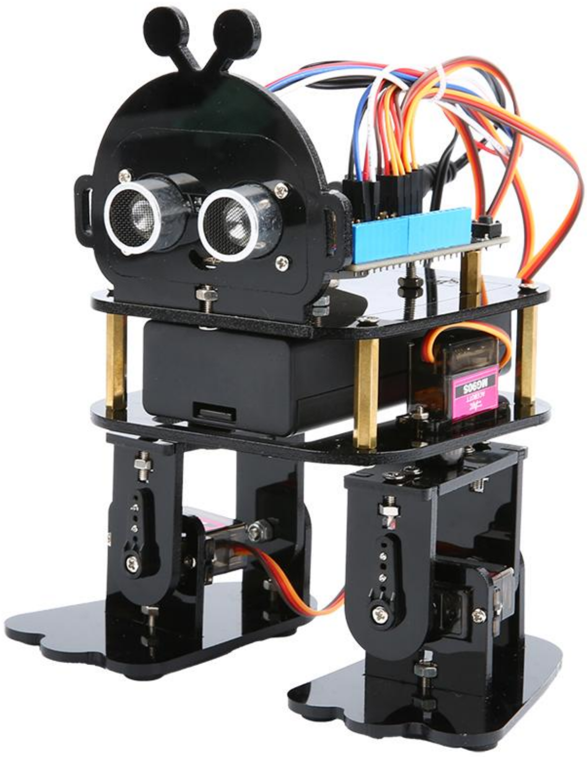
Ⅱ.APP Controls Biped Robot 38

Lesson 1 Introduction To Biped Robot

Bipedal robot is a robot that simulates human gait and has two moving legs, which can realize bipedal walking and related actions of the robot. The research of biped robots can greatly promote the progress of human-computer interaction, artificial intelligence and biomechanics. They help scientists gain insight into issues such as gait control and adaptation to complex environments, laying the foundation for future robotics.

At present, biped robots have made remarkable achievements in terms of stability and movement ability. It has a wide range of applications in related fields such as rescue, care, education and entertainment. In the future production and life, biped robots can also help humans solve many problems, such as a series of dangerous or heavy work such as rescue.

This tutorial uses a biped robot composed of four servos, with two servos on the left and right sides, which constitute the thigh joint and the calf joint of the biped robot.

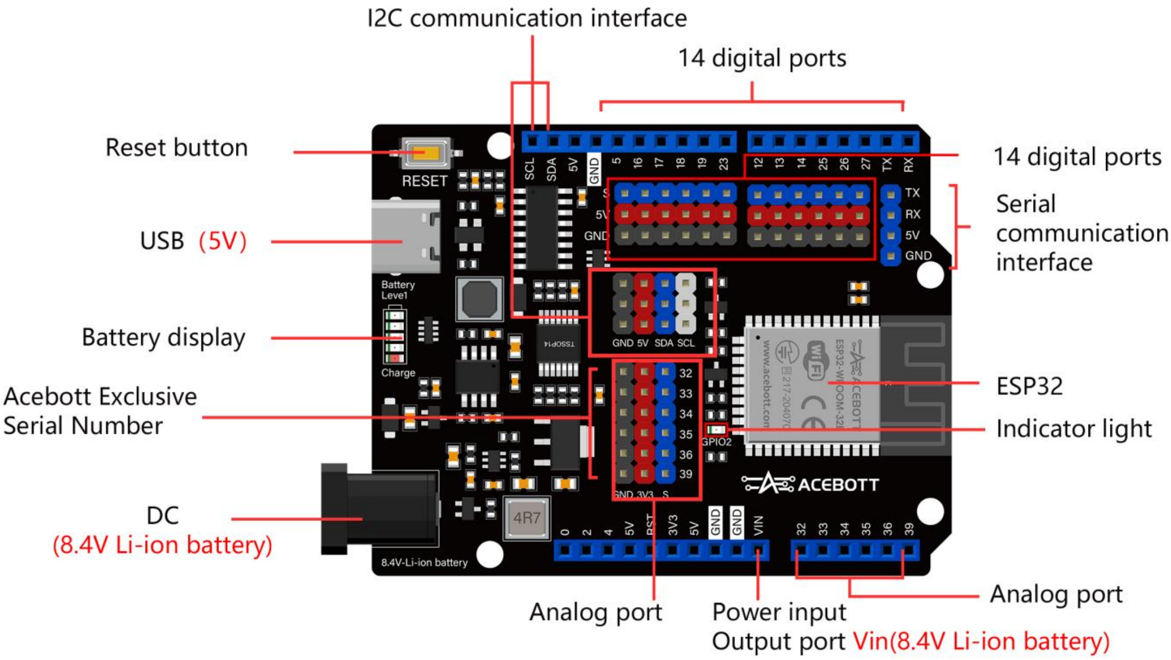


Ⅰ.Knowledge Of Hardware

1.ESP32 controller board

The ESP32 controller board is a low-power, high-performance microcontroller that is ideal for iot development. It has a 240MHz dual-core processor, 520KB of RAM and 4MB of flash memory. Built-in WiFi and Bluetooth 4.2 module for wireless communication. With 34 GPIO ports, various peripherals can be connected and controlled.

The ESP32 controller board used in this kit is equipped with a rechargeable function, and the controller board carries 5 LED modules for battery power display, the first 4 LED lights correspond to the current battery power. When the battery is full, the four LED lights are all blue, and when the power is reduced, the number of lights will be correspondingly reduced. The fifth light indicates whether the battery is being charged, when it is red, it means that the battery is being charged, when it is not, it means that the battery is full or not being charged.

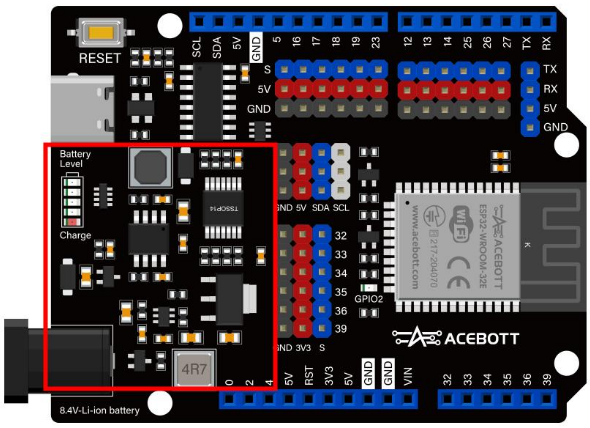


Attention:

(1) The controller board is not allowed to input more than 12V voltage power supply, otherwise it will cause damage to the controller board!

(2) Do not allow the use of dry batteries, because it will cause battery explosion! Only two 8.4V lithium batteries can be used in series.

(3) It is not recommended to touch some components of the chip power supply by hand to prevent burns.



2.Servo

(1) Servo introduction

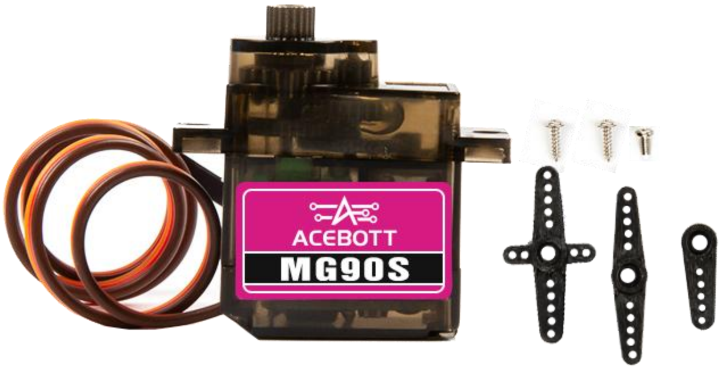
The main structure of the servo mainly has several parts: housing, transmission gear group, motor, adjustable potentiometer, control circuit board, steering disc.

Its working principle is to control the circuit board to receive the control signal of the signal source, and drive the motor to rotate; The gear group will reduce the speed of the motor by many times, and the output torque of the motor will be amplified by the corresponding multiple, and then output; The potentiometer rotates with the final stage of the gear set to measure the actual Angle of rotation of the servo shaft; The control circuit board receives the actual motor Angle from the potentiometer feedback and compares it with the target Angle. If there is an error, the servo is controlled to rotate to the target Angle position.

The working process is as follows: control signal → electronic control board → motor rotation → gear group deceleration → steering wheel rotation → feedback motor actual Angle → Control circuit board adjusts the position of the motor to the target Angle according to feedback.

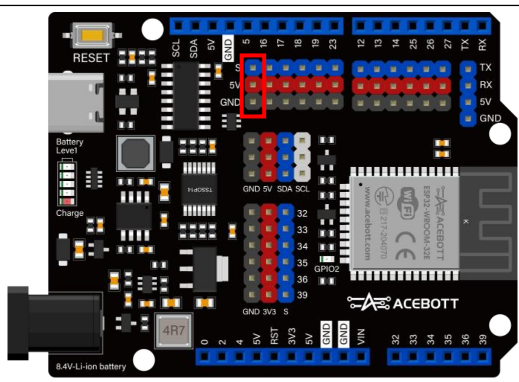
(2) Pin definition of servo

①Usually the servo has 3 control lines: power line, ground line and signal line.



②Servo pin definition: Brown line - GND, red line - 5V, orange line - signal.

③The servo is connected to the ESP32 controller board in the following way, for example, connected to the GPIO5 pin.

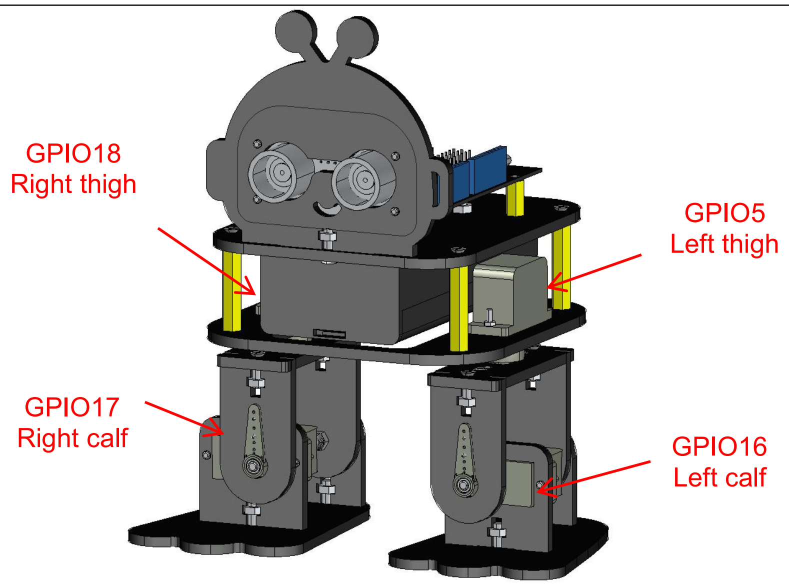
Servo	ESP32 controller board	Schematic drawing
Brown wire	GND	
Red wire	5V	
Orange wire	GPIO5	

Ⅱ.The Bipedal Robot Movement Principle

The movement of biped robot is mainly based on imitating the walking way of human, and different gaits of robot are realized through the alternating movement of two legs. The movement of each leg of the robot is controlled by the joint servo on the leg, and each servo is usually responsible for the movement of one joint. By receiving the control signal, the servo adjusts its rotation position, so that the robot legs move according to the predetermined Angle, and through the coordinated rotation of each servo, the complex actions such as forward, backward, turning and stopping can be realized.

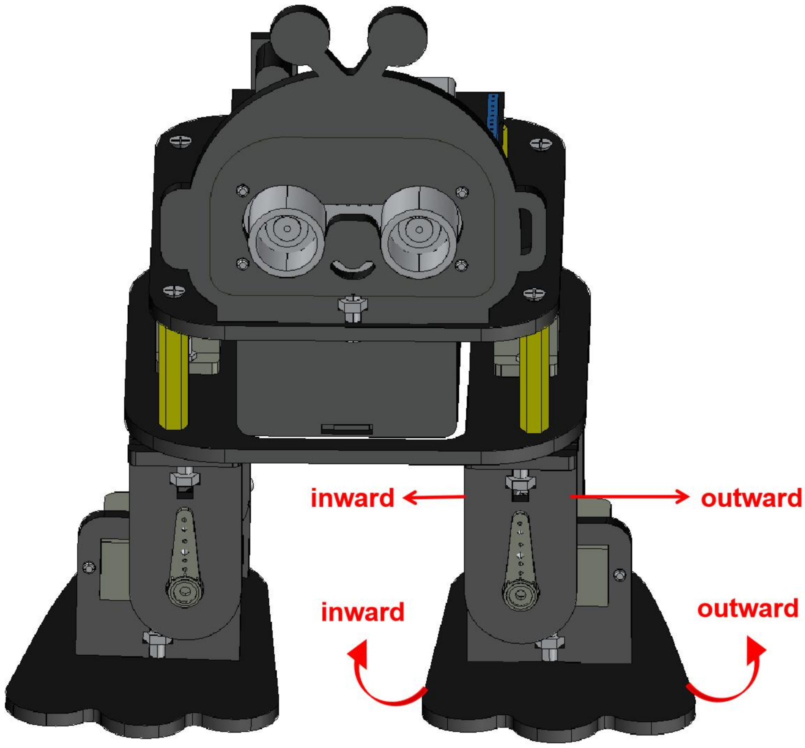
1.Servo pin of biped robot

The bipedal robot uses a total of 4 servos, and the corresponding pin numbers are as follows:

NO.	Pin Number	Servo position	Schematic drawing
1	GPIO5	Left thigh	
2	GPIO16	Left calf	
3	GPIO17	Right calf	
4	GPIO18	Right thigh	

2.Motion rules of biped robot servo

NO.	Servo pin	Servo position	Motion rule
1	GPIO5	Left thigh	The larger the servo Angle, the more inward the thigh joint turns.
2	GPIO16	Left calf	The larger the servo Angle, the more inward the calf joint turns.
3	GPIO17	Right calf	The larger the servo Angle, the more outward the calf joint turns.
4	GPIO18	Right thigh	The larger the servo Angle, the more outward the thigh joint turns.


inward
outward
inward
outward

Lesson 2 Basic Motion Of Biped Robot

Ⅰ.Forward And Backward Movement Of The Biped Robot

Moving forward and backward is one of the basic movements of biped robot. Through the basic movement learning, we can prepare for the combined movements of the robot.

Attention:

After the biped robot is powered on, it is forbidden to turn the servo directly by hand to prevent damage to the servo.

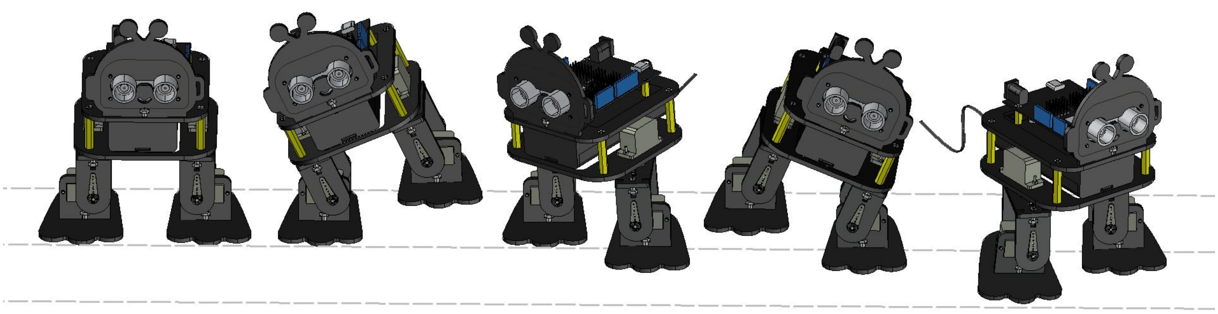
1.Forward moving program

Open "Move_Forward.ino" in English\Arduino\ 6.program \Lesson 2\Move_Forward, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

Sample Code:

```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>
void setup() {// initialize servo
   servo_5.attach(5, SERVOMIN, SERVOMAX);
   servo_16.attach(16, SERVOMIN, SERVOMAX);
   servo_17.attach(17, SERVOMIN, SERVOMAX);
   servo_18.attach(18, SERVOMIN, SERVOMAX);
   Servo_PROGRAM_Zero();// 90 90 90 90
}
int Servo_Forward_Step = 4;
//Define an array of forward movements of the robot
int Servo_Forward [][ALLMATRIX] PROGMEM = {
   //GPIO5, GPIO16, GPIO18, GPIO17, time
   {70, 110, 70, 120, 300}, //Left leg up, left leg forward
   {60, 90, 60, 90, 300}, //Left leg landing
   {110, 50, 110, 60, 300}, //Right leg up, right leg forward
   {105, 90, 105, 90, 300}, //Right leg landing
};
void loop() {
   Servo_PROGRAM_Run(Servo_Forward, Servo_Forward_Step);
}
```

After uploading the program, the theoretical gait of the biped robot corresponding to the program is shown in the figure below.



Attention:

①In the actual operation of the biped robot, due to the inability of the center of gravity to lift the legs and the friction with the ground, the walking action will be inconsistent with the figure, but when writing and understanding the program, it is necessary to write according to the action in the figure.

②However, you can alternately hold down the sole of its foot to observe its movement. At the beginning, hold down its right foot with your hand, and then wait for it to complete the third action in the figure above, and then hold down its left foot with your hand, so that it can complete the forward movement according to the movement gait in the figure above.

2.Backward moving program

Open "Move_Backward.ino" in the English\Arduino\6. Program \Lesson 2\Move_Backward, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

Sample Code:

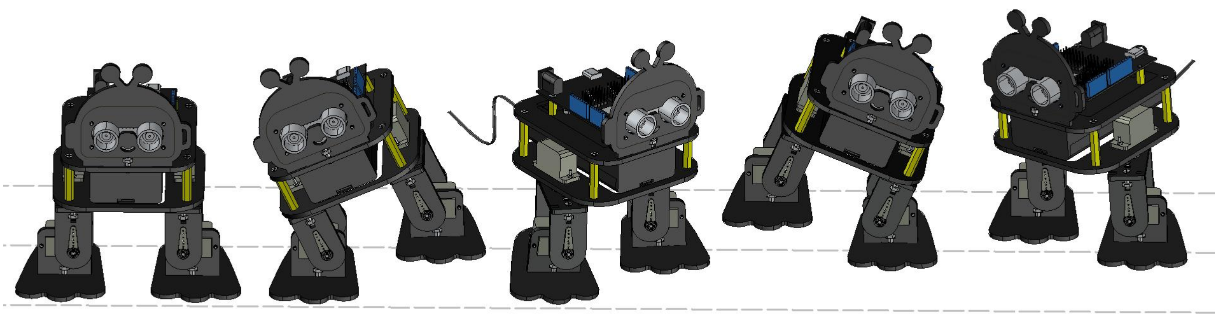
```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>

void setup() {// initialize servo
  servo_5.attach(5, SERVOMIN, SERVOMAX);
  servo_16.attach(16, SERVOMIN, SERVOMAX);
  servo_17.attach(17, SERVOMIN, SERVOMAX);
  servo_18.attach(18, SERVOMIN, SERVOMAX);
  Servo_PROGRAM_Zero();// 90 90 90 90
}

int Servo_Backward_Step = 4;
//Define an array of Backward movements of the robot
int Servo_Backward [][ALLMATRIX] PROGMEM = {
  //GPIO5, GPIO16, GPIO18, GPIO17, time
  {120, 75, 110, 110, 300}, //Right leg up, right leg backward
  {90, 100, 110, 75, 300}, //Right leg landing
  {60, 75, 65, 90, 300}, //Left leg up, left leg backward
  {60, 90, 60, 90, 300}, //Left leg landing
};

void loop() {
  Servo_PROGRAM_Run(Servo_Backward, Servo_Backward_Step);
}
```

After uploading the program, the theoretical gait of the biped robot corresponding to the program is shown in the figure below.



Attention:

①In the actual operation of the biped robot, due to the inability of the center of gravity to lift the legs and the friction with the ground, the walking action will be inconsistent with the figure, but when writing and understanding the program, it is necessary to write according to the action in the figure.

②However, you can alternately hold down the sole of its foot to observe its movement. At the beginning, hold down its right foot with your hand, and then wait for it to complete the third action in the figure above, and then hold down its left foot with your hand, so that it can complete the backward movement according to the movement gait in the figure above.

Ⅱ.Left And Right Rotation Of The Biped Robot

The left and right rotation of the biped robot takes a few more steps than the forward and backward steps, and it is necessary to set the Angle at which the robot needs to rotate and determine the position of each step.

1.Left rotation program

Open "Turn_Left.ino" in the English\Arduino\ 6.program \Lesson 2\Turn_Left, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

Sample Code:

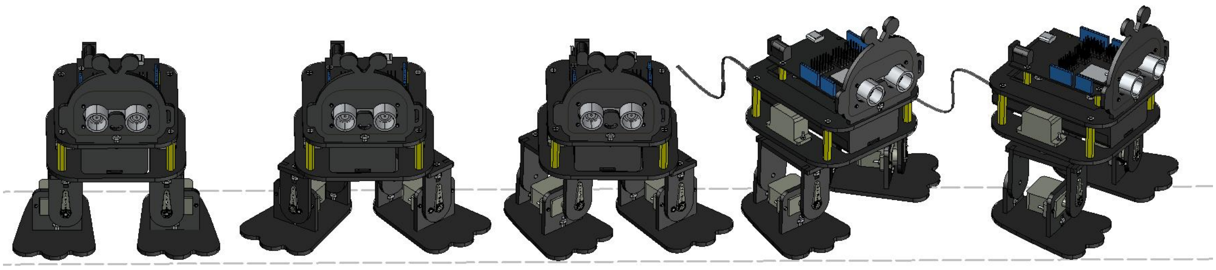
```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>

void setup() {// initialize servo
  servo_5.attach(5, SERVOMIN, SERVOMAX);
  servo_16.attach(16, SERVOMIN, SERVOMAX);
  servo_17.attach(17, SERVOMIN, SERVOMAX);
  servo_18.attach(18, SERVOMIN, SERVOMAX);
  Servo_PROGRAM_Zero();// 90 90 90 90
}

int Servo_Left_Step = 5;
//Define an array of Left movements of the robot
int Servo_Left [][ALLMATRIX] PROGMEM = {
  //GPIO5, GPIO16, GPIO18, GPIO17, time
  {60, 60, 125, 90, 300}, //The left thigh rotates outward, the right thigh rotates outward,
and the left calf rotates outward
  {60, 90, 125, 90, 300}, //Turn the left calf inward
  {60, 120, 55, 120, 300}, //The left calf rotates inward, the right thigh rotates inward, and
the right calf rotates outward
  {60, 90, 55, 90, 300}, //The left calf rotates outward and the right calf rotates inward
  {80, 90, 90, 90, 150}, //The left thigh rotates inward and the right thigh outward
};

void loop() {
  Servo_PROGRAM_Run(Servo_Left, Servo_Left_Step);
}
```

After uploading the program, the theoretical gait of the biped robot corresponding to the program is shown in the figure below.



Attention:

①During the actual operation of the biped robot, due to the inability of the center of gravity to lift the legs and the friction with the ground, the walking action will not be consistent with the figure, but when writing and understanding the program, it is necessary to write according to the action in the figure.

②However, you can alternately hold down the sole of its foot to observe its movement, that is, when the biped robot appears the second action in the figure above, first hold down its left foot with your hand, and then wait for it to complete the third action in the figure above, and then hold down its right foot with your hand, then you can complete the left turn according to the movement gait in the figure above.

2.Right rotation program

Open "Turn_Right.ino" in the English\Arduino\ 6.program \Lesson 2\Turn_Right, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

Sample Code:

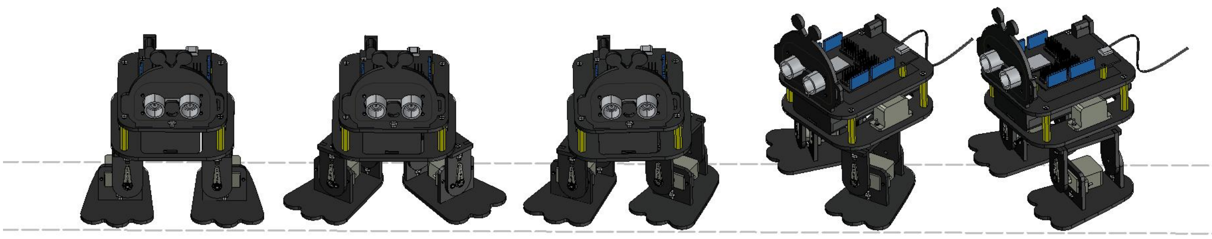
```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>

void setup() {// initialize servo
   servo_5.attach(5, SERVOMIN, SERVOMAX);
   servo_16.attach(16, SERVOMIN, SERVOMAX);
   servo_17.attach(17, SERVOMIN, SERVOMAX);
   servo_18.attach(18, SERVOMIN, SERVOMAX);
   Servo_PROGRAM_Zero();// 90 90 90 90
}

int Servo_Right_Step = 5;
//Define an array of Right movements of the robot
int Servo_Right [][ALLMATRIX] PROGMEM = {
   //GPIO5, GPIO16, GPIO18, GPIO17, time
   {60, 90, 120, 120, 300}, //The left thigh rotates outward, the right thigh rotates outward,
and the right calf rotates outward
   {60, 90, 120, 90, 300}, //Turn the right calf inward
   {120, 60, 120, 60, 300}, //The left thigh rotates inward, the left calf outward, and the
right calf inward
   {120, 90, 120, 90, 300}, //The left calf was rotated inward and the right calf outward
   {90, 90, 80, 90, 150}, //The left thigh rotates outward and the right thigh rotates inward
};

void loop() {
   Servo_PROGRAM_Run(Servo_Right, Servo_Right_Step);
}
```

After uploading the program, the theoretical gait of the biped robot corresponding to the program is shown in the figure below.



Attention:

①In the actual operation of the biped robot, due to the inability of the center of gravity to lift the legs and the friction with the ground, the walking action will be inconsistent with the figure, but when writing and understanding the program, it is necessary to write according to the action in the figure.

②However, you can alternately hold down the sole of its foot to observe its movement mode, that is, when the biped robot appears the second action in the above figure, first hold down its right foot with your hand, and then wait for it to complete the third action in the above figure, and then hold down its left foot with your hand, so that it can complete the right turn according to the movement gait in the above figure.

Ⅲ.The Serial Port Control Robot Movement

By inputting corresponding commands through the serial port, the biped robot can make corresponding actions according to the instructions of the serial port.

Open "Serial_Control.ino" in the English\Arduino\6. Program \Lesson 2\Serial_Control, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

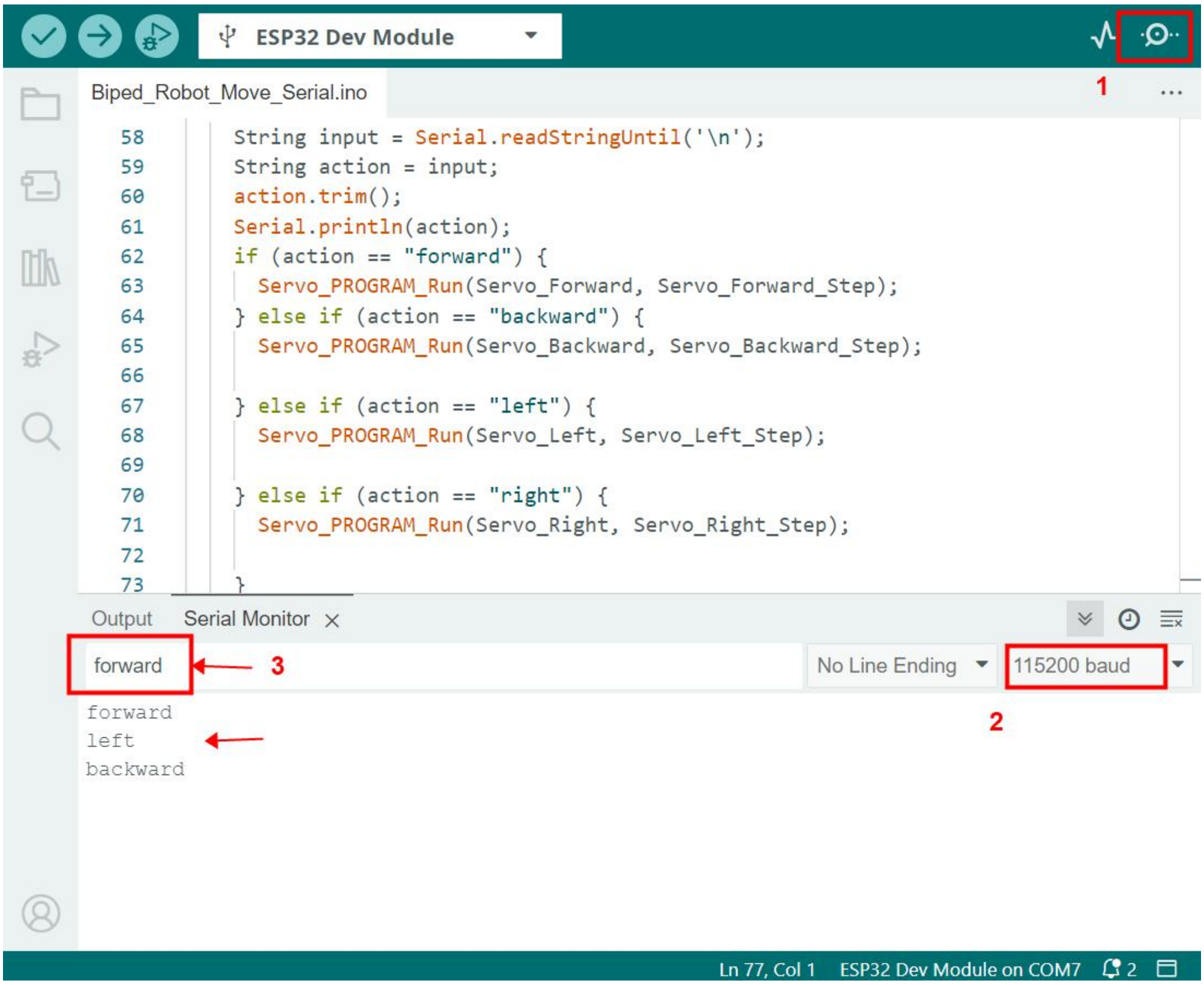
Sample Code:

```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>
void setup() {// initialize servo
  Serial.begin(115200);// set the baud rate to 115200
  servo_5.attach(5, SERVOMIN, SERVOMAX);
  servo_16.attach(16, SERVOMIN, SERVOMAX);
  servo_17.attach(17, SERVOMIN, SERVOMAX);
  servo_18.attach(18, SERVOMIN, SERVOMAX);
  Servo_PROGRAM_Zero();// 90 90 90 90
}
int Servo_Forward_Step = 4;//Forward
int Servo_Forward [][ALLMATRIX] PROGMEM = {
  {70, 110, 70, 120, 300},//GPIO5, GPIO16, GPIO18, GPIO17, time
  {60, 90, 60, 90, 300},
  {110, 50, 110, 60, 300},
  {105, 90, 105, 90, 300},
};
int Servo_Backward_Step = 4;//Backward
int Servo_Backward [][ALLMATRIX] PROGMEM = {
  {120, 75, 110, 110, 300},
  {90, 100, 110, 75, 300},
  {60, 75, 65, 90, 300},
  {60, 90, 60, 90, 300},
};
int Servo_Left_Step = 5;//Left
int Servo_Left [][ALLMATRIX] PROGMEM = {
  {60, 60, 125, 90, 300},
  {60, 90, 125, 90, 300},
  {60, 120, 55, 120, 300},
  {60, 90, 55, 90, 300},
  {80, 90, 90, 90, 150},
};
int Servo_Right_Step = 5;//Right
int Servo_Right [][ALLMATRIX] PROGMEM = {
  {60, 90, 120, 120, 300},
  {60, 90, 120, 90, 300},
  {120, 60, 120, 60, 300},
  {120, 90, 120, 90, 300},
  {90, 90, 80, 90, 150},
};
```

```
int Servo_Stop_Step = 1;//Stop
int Servo_Stop [][ALLMATRIX] PROGMEM = {
  {90, 90, 90, 90, 300},
};

void loop() {
  if (Serial.available()) {//Determine whether there is serial port data
    String input = Serial.readStringUntil('\n');
    String action = input;
    action.trim();
    Serial.println(action);//Print the input serial port data
    if (action == "forward") {//When "forward" is input, the robot moves forward
      Servo_PROGRAM_Run(Servo_Forward, Servo_Forward_Step);
    }
    else if (action == "backward") {//When "backward" is input, the robot moves
backward
      Servo_PROGRAM_Run(Servo_Backward, Servo_Backward_Step);
    }
    else if (action == "left") {//When "left" is input, the robot rotates to the left
      Servo_PROGRAM_Run(Servo_Left, Servo_Left_Step);
    }
    else if (action == "right") {//When "right" is input, the robot rotates to the right
      Servo_PROGRAM_Run(Servo_Right, Servo_Right_Step);
    }
    Servo_PROGRAM_Run(Servo_Stop, Servo_Stop_Step);
  }
}
```

After uploading the program, open the serial port monitor, type "forward" in the serial port input box, and press the "Enter" key on the keyboard, and the biped robot will perform a forward action. Similarly, you can enter the instructions "backward", "left", and "right" in order to control the robot to perform the corresponding action.


ESP32 Dev Module
Biped_Robot_Move_Serial.ino
1
String input = Serial.readStringUntil('\n');
String action = input;
action.trim();
Serial.println(action);
if (action == "forward") {
Servo_PROGRAM_Run(Servo_Forward, Servo_Forward_Step);
} else if (action == "backward") {
Servo_PROGRAM_Run(Servo_Backward, Servo_Backward_Step);
} else if (action == "left") {
Servo_PROGRAM_Run(Servo_Left, Servo_Left_Step);
} else if (action == "right") {
Servo_PROGRAM_Run(Servo_Right, Servo_Right_Step);
}
Output
Serial Monitor
forward
3
No Line Ending
115200 baud
2
forward
left
backward
Ln 77, Col 1
ESP32 Dev Module on COM7

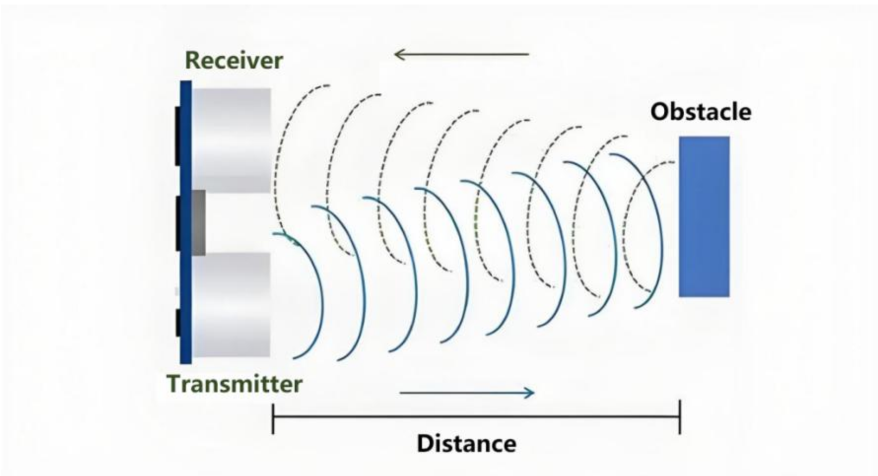
Lesson 3 The Following Function Of Biped Robot

Robot following function is the ability that enables a robot to automatically follow a target object. This function is useful in many application scenarios, such as intelligent assistant, automatic navigation, logistics and distribution.

At present, there are a variety of sensors that can realize the robot's following function, such as: cameras, LIDAR, ultrasonic sensors and so on. The biped robot in this tutorial realizes the following function through the ultrasonic sensor. In use, the target to be followed needs to be placed at a certain distance in front of the ultrasonic sensor of the biped robot. When the distance between the target and the biped robot is less than the set threshold, the robot will retreat; When the distance between the target and the biped robot is greater than the set threshold, the robot will advance; When the distance between the target and the biped robot is equal to the set threshold, the robot stops moving.

Ⅰ.Ultrasonic Sensor

Ultrasonic sensor is a sensor used to measure distance. The principle of ultrasonic sensor is based on the transmission and reception of ultrasonic waves. It is widely used in distance measurement, object detection, level measurement and other scenes. Ultrasonic sensors emit ultrasonic signals through internal transmitters, and when the ultrasonic waves meet the surface of the object, they are reflected back to form an echo. These echoes are picked up by the receiver of the ultrasonic sensor. By calculating the time it takes for the ultrasonic wave to travel out and back, the sensor can determine the distance between the wave and the object.



Ⅱ.Robot Follow Function Program

Open "Move_Follow.ino" in the English\Arduino\ 6.program \Lesson 3\Move_Follow, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

Sample Code:

```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>
void setup() {// initialize servo
  servo_5.attach(5, SERVOMIN, SERVOMAX);
  servo_16.attach(16, SERVOMIN, SERVOMAX);
  servo_17.attach(17, SERVOMIN, SERVOMAX);
  servo_18.attach(18, SERVOMIN, SERVOMAX);
  Servo_PROGRAM_Zero();// 90 90 90 90
  Ultrasonic_Init();
}
// Forward
int Servo_Forward_Step = 4;
int Servo_Forward [][ALLMATRIX] PROGMEM = {
  {70, 110, 70, 120, 300}, //GPIO5, GPIO16, GPIO18, GPIO17, time
  {60, 90, 60, 90, 300},
  {110, 50, 110, 60, 300},
  {105, 90, 105, 90, 300},
};
```

```
// Backward
int Servo_Backward_Step = 4;
int Servo_Backward [][ALLMATRIX] PROGMEM = {
  {120, 75, 110, 110, 300},
  {90, 100, 110, 75, 300},
  {60, 75, 65, 90, 300},
  {60, 90, 60, 90, 300},
};
// Stop
int Servo_Stop_Step = 1;
int Servo_Stop [][ALLMATRIX] PROGMEM = {
  {90, 90, 90, 90, 300},
};

void loop() {
  UT_distance = Ranging(Trig_PIN, Echo_PIN);
  //When the ultrasonic detection distance is less than 15CM, the robot moves backward
  if (UT_distance < 15) {
    Servo_PROGRAM_Run(Servo_Backward, Servo_Backward_Step);// Backward
  }
  //When the ultrasonic detection distance is 15~20CM, the robot stops moving
  else if (15 <= UT_distance && UT_distance <= 20) {
    Servo_PROGRAM_Run(Servo_Stop, Servo_Stop_Step);// Stop
  }
  //When the ultrasonic detection distance is 20~35CM, the robot moves forward
  else if (20 <= UT_distance && UT_distance < 35) {
    Servo_PROGRAM_Run(Servo_Forward, Servo_Forward_Step);// Forward
  }
  else{
    Servo_PROGRAM_Run(Servo_Stop, Servo_Stop_Step); // Stop
  }
}
```

After uploading the program, you can use your hand as the following target of the biped robot, place your hand in front of the ultrasonic wave of the robot, and then approach and stay away from the robot to observe the following effect of the robot.

Lesson 4 Obstacle Avoidance Of Biped Robot

The robot's obstacle avoidance function enables it to move safely in complex environments and avoid collisions with obstacles. The obstacle avoidance function of the biped robot in this tutorial is to detect the distance of the object in front by using the transmission and reception of ultrasonic waves, so as to achieve the obstacle avoidance function. When the ultrasonic sensor detects an obstacle in front, the biped robot turns to avoid the obstacle, and when there is no obstacle in front, the biped robot continues to move forward.

Ⅰ.Biped Robot Obstacle Avoidance Program

Open "Move_Avoid.ino" in the English\Arduino\ 6.program \Lesson 4\Move_Avoid, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

Sample Code:

```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>
void setup() {// initialize servo
  servo_5.attach(5, SERVOMIN, SERVOMAX);
  servo_16.attach(16, SERVOMIN, SERVOMAX);
  servo_17.attach(17, SERVOMIN, SERVOMAX);
  servo_18.attach(18, SERVOMIN, SERVOMAX);
  Servo_PROGRAM_Zero();// 90 90 90 90
  Ultrasonic_Init();
}

int Servo_Forward_Step = 4;//Forward
int Servo_Forward [][ALLMATRIX] PROGMEM = {
  {70, 110, 70, 120, 300}, //GPIO5, GPIO16, GPIO18, GPIO17, time
  {60, 90, 60, 90, 300},
  {110, 50, 110, 60, 300},
  {105, 90, 105, 90, 300},
};

int Servo_Backward_Step = 4;//Backward
int Servo_Backward [][ALLMATRIX] PROGMEM = {
  {120, 75, 110, 110, 300},
  {90, 100, 110, 75, 300},
  {60, 75, 65, 90, 300},
  {60, 90, 60, 90, 300},
};

int Servo_Left_Step = 5;//Left
int Servo_Left [][ALLMATRIX] PROGMEM = {
  {60, 60, 125, 90, 300},
  {60, 90, 125, 90, 300},
  {60, 120, 55, 120, 300},
  {60, 90, 55, 90, 300},
  {80, 90, 90, 90, 150},
};
```

```
int Servo_Right_Step = 5;//Right
const int Servo_Right [][ALLMATRIX] PROGMEM = {
  {60, 90, 120, 120, 300},
  {60, 90, 120, 90, 300},
  {120, 60, 120, 60, 300},
  {120, 90, 120, 90, 300},
  {90, 90, 80, 90, 150},
};

int Servo_Stop_Step = 1;//Stop
int Servo_Stop [][ALLMATRIX] PROGMEM = {
  {90, 90, 90, 90, 300},
};

void loop() {
  UT_distance = Ranging(Trig_PIN, Echo_PIN);
  //When the ultrasonic detection distance is less than 15CM, the robot stops moving and
then moves backward 6 times
  if (UT_distance <= 15) {
    Servo_PROGRAM_Run(Servo_Stop, Servo_Stop_Step);
    for(int i=1;i<=6;i++){
      Servo_PROGRAM_Run(Servo_Backward, Servo_Backward_Step);
    }
  }
```

```
  // When the ultrasonic detection distance is 15-20cm, the robot will randomly choose to turn left or right
  if (UT_distance > 15 && UT_distance <= 20){
    int randNumber = random(1, 3);//Produces a random integer of 1 or 2
      switch (randNumber) {
        case 1://When the random number is 1
        for(int i=1;i<=10;i++){//Rotate to the right 10 times
          Servo_PROGRAM_Run(Servo_Right, Servo_Right_Step);
        }
        delay(500);
        break;
        case 2://When the random number is 2
        for(int i=1;i<=10;i++){//Rotate to the left 10 times
          Servo_PROGRAM_Run(Servo_Left, Servo_Left_Step);
        }
        delay(500);
        break;
      }
    }
  //When the ultrasonic detection distance is greater than 20CM, the robot moves forward
  else {
    Servo_PROGRAM_Run(Servo_Forward, Servo_Forward_Step);
  }
}
```

After uploading the program, when the biped robot advances and encounters an obstacle in front of it, the robot will choose to retreat according to the distance with the obstacle, move left or right to avoid the obstacle, until there is no obstacle in front of it, and then continue to move forward.

Lesson 5 Biped Robot Dance Moves 1

We have learned the basic movements of the robot before and after moving around, I believe you can't wait to combine these basic movements, and then achieve personalized dancing movements on the robot. This tutorial will provide a biped robot dancing sample program to open your mind, and later you can define more interesting dancing effects yourself.

Ⅰ.Robot Dance Program 1

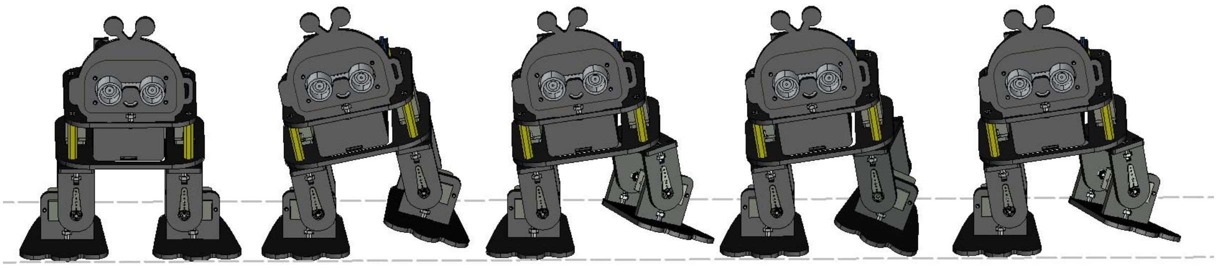
Open "Move_Dance1.ino" in the English\Arduino\ 6.program \Lesson 5\Move_Dance1, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

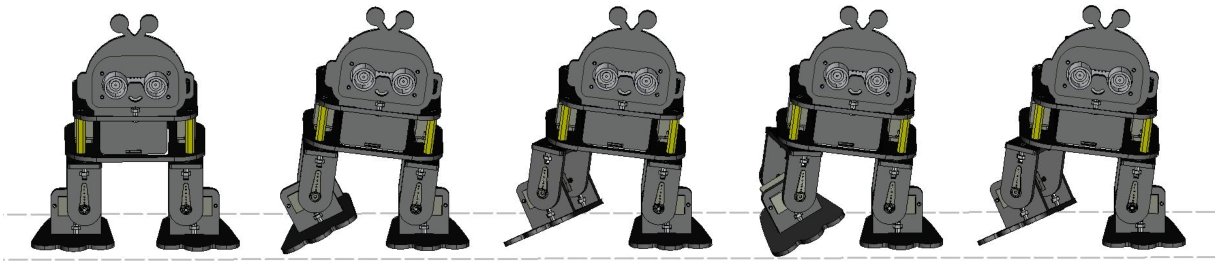
Sample Code:

```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>
void setup() {//initialize servo
   servo_5.attach(5, SERVOMIN, SERVOMAX);
   servo_16.attach(16, SERVOMIN, SERVOMAX);
   servo_17.attach(17, SERVOMIN, SERVOMAX);
   servo_18.attach(18, SERVOMIN, SERVOMAX);
   Servo_PROGRAM_Zero();//90 90 90 90
}
int Servo_Dance1_Step = 22;
int Servo_Dance1 [][ALLMATRIX] PROGMEM = {
   //GPIO5, GPIO16, GPIO18, GPIO17 , time
   {50,90,80,90, 300}, //Left thigh to the outward rotation
   {50,130,80,100, 300}, //Left calf to the inward rotation
   {80,130,80,100, 300}, //Left thigh to the inward rotation
   {60,130,80,100, 300}, //Left thigh to the outward rotation
   {80,130,80,100, 300}, //Left thigh to the inward rotation
   {60,130,80,100, 300}, //Left thigh to the outward rotation
   {80,130,80,100, 300}, //Left thigh to the inward rotation
   {60,130,80,100, 300}, //Left thigh to the outward rotation
   {80,130,80,100, 300}, //Left thigh to the inward rotation
   {60,110,80,105, 300}, //Left thigh to the outward rotation， Left calf to the inward rotation
   {90,90,90,90, 300}, //Action initialization of biped robot
   {100,90,125,90, 300}, //Right thigh to the outward rotation
   {100,75,125,50, 300}, //Right calf to the inward rotation
   {100,75,100,50, 300}, //Right thigh to the inward rotation
   {100,75,120,50, 300}, //Right thigh to the outward rotation
   {100,75,100,50, 300}, //Right thigh to the inward rotation
   {100,75,120,50, 300}, //Right thigh to the outward rotation
   {100,75,100,50, 300}, //Right thigh to the inward rotation
   {100,75,120,50, 300}, //Right thigh to the outward rotation
   {100,75,100,50, 300}, //Right thigh to the inward rotation
   {100,90,120,70, 300}, //Right thigh to the outward rotation， Right calf to the outward
rotation
   {90,90,90,90, 300}, //Action initialization of biped robot
};
void loop() {
   Servo_PROGRAM_Run(Servo_Dance1, Servo_Dance1_Step);
`
```

After uploading the program, the biped robot will shake its left and right ankles and repeat it four times, although its action looks simple, it also needs to set the Angle of each servo, so that each servo can cooperate with each other to achieve a flexible effect.

Attention: Repetitions of dance moves are no longer shown.





Lesson 6 Biped Robot Dance Moves 2

The dancing program of the last lesson was mainly the ankle shaking movement of the robot. In order to add some new movements to enrich the dance movements, this lesson will show more complicated dancing movement2. Of course, if you have a better idea, you can always add more moves.

Ⅰ.Robot Dance Program 2

Open "Move_Dance2.ino" in the English\Arduino\ 6.program \Lesson 6\Move_Dance2, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

Sample Code:

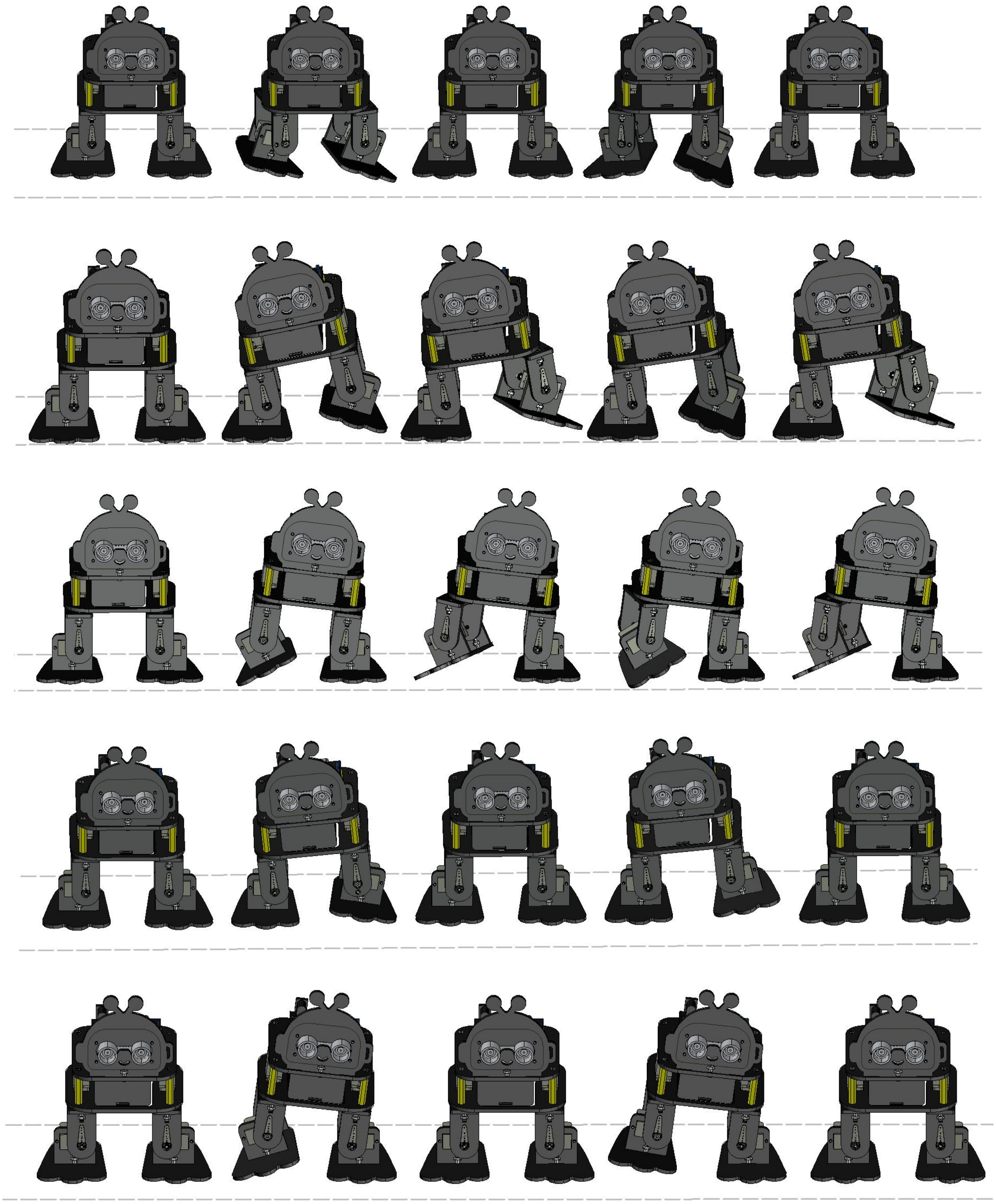
```
#include <Arduino.h>
#include <ACB_Biped_Robot.h>

void setup() {// initialize servo
   servo_5.attach(5, SERVOMIN, SERVOMAX);
   servo_16.attach(16, SERVOMIN, SERVOMAX);
   servo_17.attach(17, SERVOMIN, SERVOMAX);
   servo_18.attach(18, SERVOMIN, SERVOMAX);
   Servo_PROGRAM_Zero();// 90 90 90 90
}
int Servo_Dance2_Step = 50;
int Servo_Dance2 [][ALLMATRIX] PROGMEM = {
   {75, 105, 68, 110, 600},//First
   {75, 90, 60, 90, 600},
   {75, 68, 110, 68, 600},
   {100, 90, 110, 90, 600},
   {60, 120, 60, 120, 200},
   {60, 90, 60, 90, 200},
   {110, 60, 110, 60, 200},
   {110, 90, 110, 90, 200},
   {60, 110, 60, 110, 200},
   {60, 90, 60, 90, 200},
   {110, 60, 110, 60, 200},
```

```
    {110, 90, 110, 90, 200},
    {50,90,80,90, 300},//Second
    {50,130,80,100, 300},
    {80,130,80,100, 300},
    {60,130,80,100, 300},
    {80,130,80,100, 300},
    {60,130,80,100, 300},
    {80,130,80,100, 300},
    {60,130,80,100, 300},
    {80,130,80,100, 300},
    {60,110,80,105, 300},
    {90,90,90,90, 300},
    {100,90,125,90, 300},//Third
    {100,75,125,50, 300},
    {100,75,100,50, 300},
    {100,75,120,50, 300},
    {100,75,100,50, 300},
    {100,75,120,50, 300},
    {100,75,100,50, 300},
    {100,75,120,50, 300},
    {100,75,100,50, 300},
    {100,90,120,70, 300},
    {90,90,90,90, 300},
    {90, 100, 92, 95, 300},//Fourth
    {90, 60, 92, 85, 300},
    {90, 120, 92, 95, 300},
    {90, 60, 92, 85, 300},
    {90, 120, 92, 95, 300},
    {90, 60, 92, 85, 300},
    {90, 120, 92, 95, 300},
    {90, 90, 90, 90, 300},
    {90, 80, 90, 100, 300},//Fifth
    {90, 95, 90, 60, 300},
    {90, 75, 90, 120, 300},
    {90, 95, 90, 60, 300},
    {90, 75, 90, 120, 300},
    {90, 95, 90, 60, 300},
    {90, 75, 90, 120, 300},
    {90, 90, 90, 90, 300},
};
void loop() {
    Servo_PROGRAM_Run(Servo_Dance2, Servo_Dance2_Step);
}
```

After uploading the program, the biped robot will first move forward from left to right and repeat four times, then shake its left and right ankles four times each, and finally repeat the left and right moonwalk four times each to form a complete dance movement.

Attention: Repetitions of dance moves are no longer shown.



Lesson 7 Web Control Of Biped Robot

With the continuous development of wireless communication technology and Internet of Things technology, remote control equipment technology has been widely used in many fields, which allows users to achieve remote and accurate control of terminal equipment. There are many types of wireless communication technology, this tutorial mainly introduces how to use WiFi communication technology to achieve remote control of biped robot.

WiFi communication technology is a wireless local area network (WLAN) technology that allows electronic devices such as smartphones, tablets, laptops, etc., to connect wirelessly to the Internet or local area network. WiFi communication technology connects devices to the same network through a wireless router or access point (AP), and devices can receive and send data to each other.

Web control device is one of the main applications of WiFi communication technology, which is widely used in smart home and smart industry. The web control device is connected to the device and the control terminal through the Internet. The interaction between the device and the controller can be achieved through a simple HTTP protocol, when the device is connected to the controller, the controller provides a simple Web interface, and the user can access the controller through a web page, thereby controlling the device.

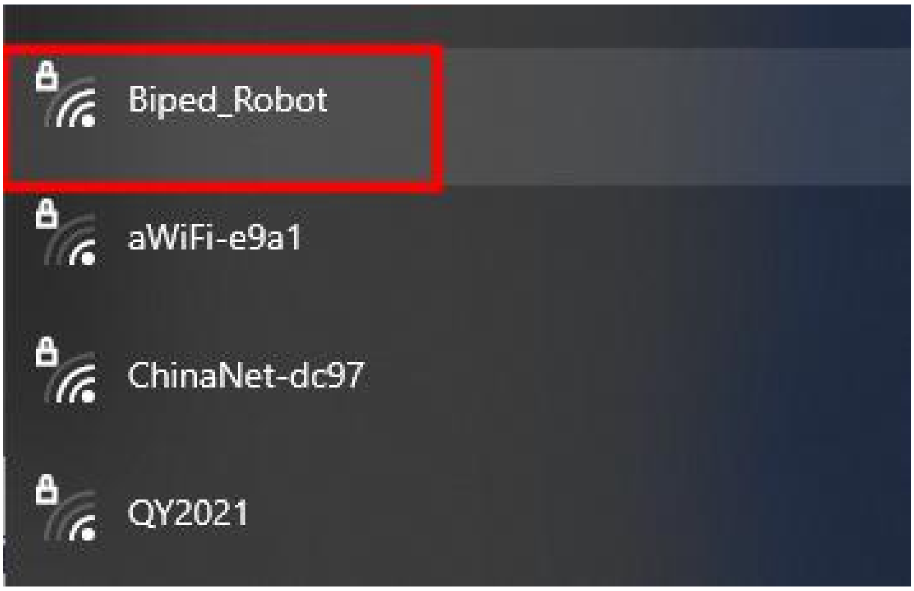
Next, we'll use the web to remotely control biped robots.

Ⅰ.Web Control Program

Open "Biped_Robot_Web.ino" in the English\Arduino\ 6.program \Lesson 7\Biped_Robot_Web, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

Ⅱ.Login Page

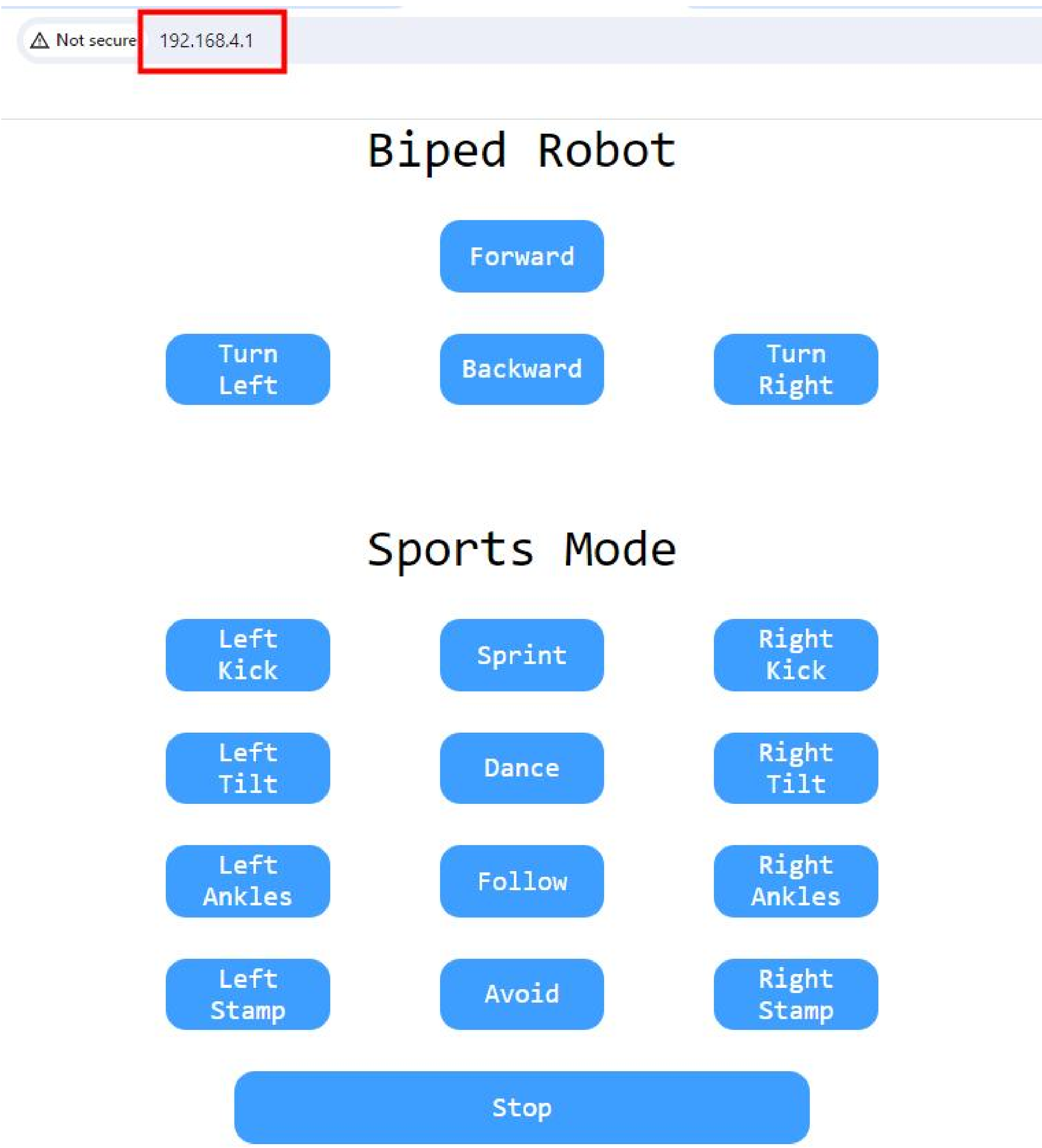
After the upload is successful, then use the computer or mobile phone wireless network to scan the WIFI, connect to the WIFI hotspot named "Biped_Robot", the password is 12345678, as shown in the following figure.



Attention: The name and password of the hotspot are already defined in the program, but the user can customize the modification, when we have multiple biped robots, we can distinguish each biped robot by different WiFi names.

```
const char* ssid = "Biped_Robot";//WiFi name
const char* password = "12345678";//WiFi password
```

After the connection is successful, enter "192.168.4.1" in the address bar of the browser. The page interface is as follows:


Not secure
192.168.4.1
Biped Robot
Forward
Turn Left
Backward
Turn Right
Sports Mode
Left Kick
Sprint
Right Kick
Left Tilt
Dance
Right Tilt
Left Ankles
Follow
Right Ankles
Left Stamp
Avoid
Right Stamp
Stop

Lesson 8 Biped Robot APP Control

In the previous tutorial, we have learned to use the rocker and web page to control the biped robot. In order to control the biped robot more conveniently, we choose to use the mobile APP as the client to realize the control of the biped robot through the mobile APP. Next, we will learn how to control the work of bipedal robots through mobile phone apps.

Ⅰ.APP Download

(1)If you are using an iOS device, search for the keyword "ACEBOTT" in the App Store and download it. If you are using an Android device, search for the keyword "ACEBOTT" in the Google Play Store and download it. The icon is shown as below.

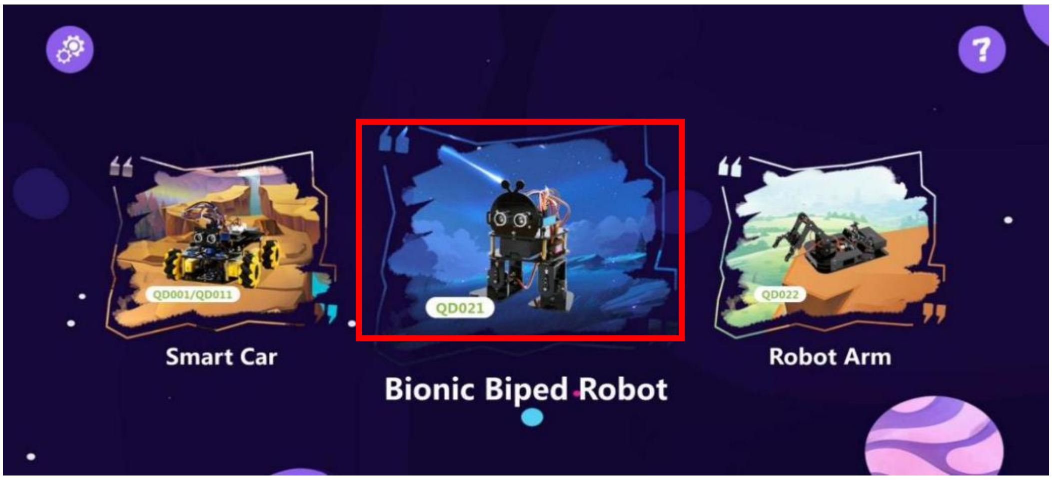
Attention:

①This tutorial applies to ACEBOTT APP version 2.0 and above. You can check the Acebott APP version by clicking the Settings button in the upper left corner of the app. Please make sure that you have the right version of the app.

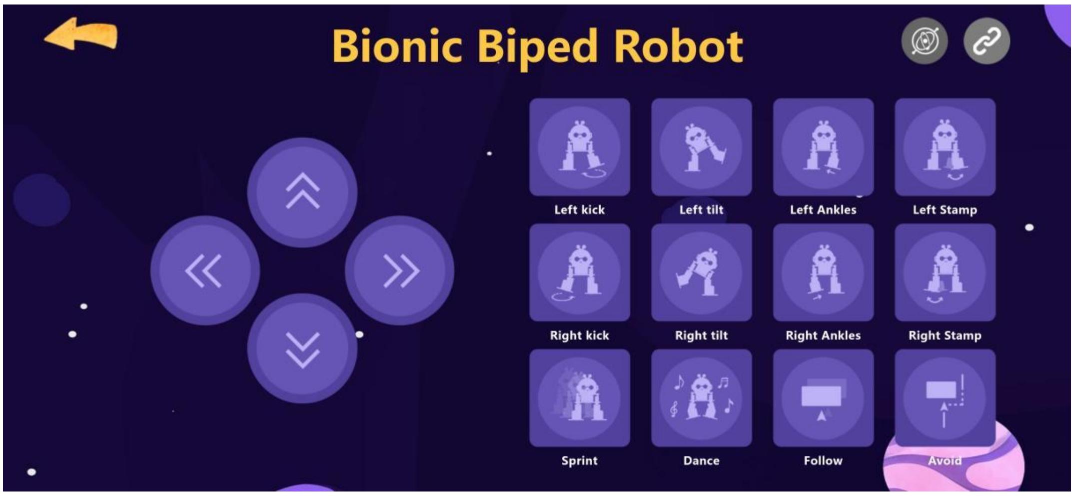
②To update the ACEBOTT software version, you can download the latest APP version by following the instructions in this tutorial.

(2)Open the app to enter the splash screen.

(3) Enter the selection interface and select the Bionic Biped Robot.



(4)Enter the Biped Robot control interface (now can not be directly controlled, need to burn the program).



Ⅱ.APP Controls Biped Robot

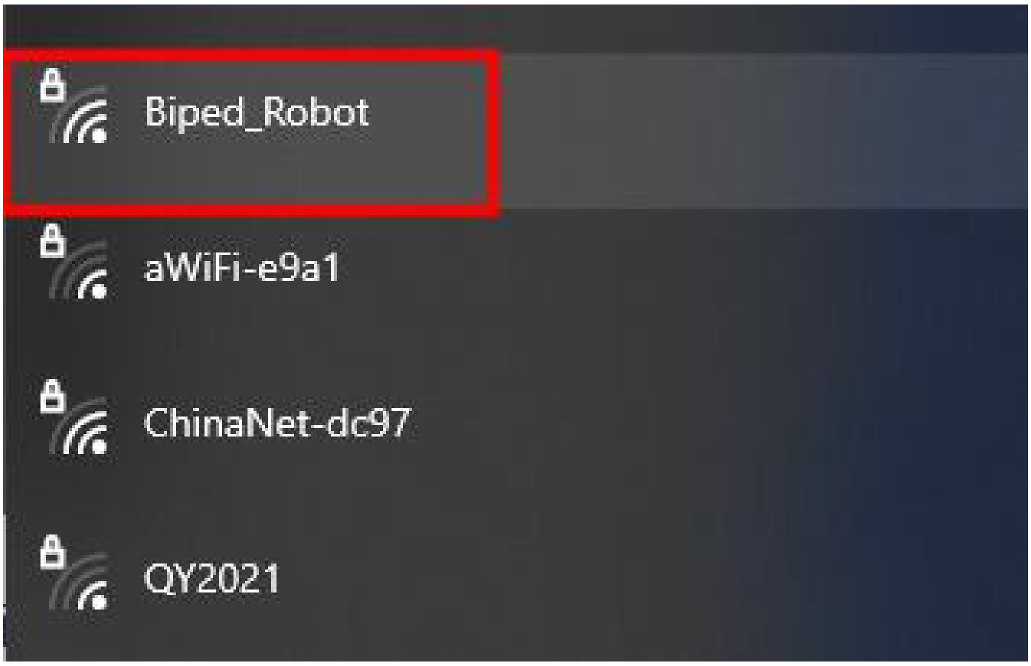
1.Upload the Arduino program to control the biped robot

Before using the APP to control the biped robot, it is necessary to upload the Arduino program for the biped robot to communicate with the APP to the biped robot.

Open "Biped_robot_app.ino" in English\Arduino\ 6.program \Lesson 8\Biped_Robot_App, connect ESP32 controller board and computer with USB cable, select the correct controller board and port, and upload the code to ESP32 controller board. The controller board should connect the power supply of the battery box and turn the switch of the battery box to the "ON" position.

2.Connect the WiFi of the biped robot

Scan the WIFI of the computer or mobile phone and connect to the WIFI hotspot named Biped_Robot. The password is 12345678, as shown in the following figure.

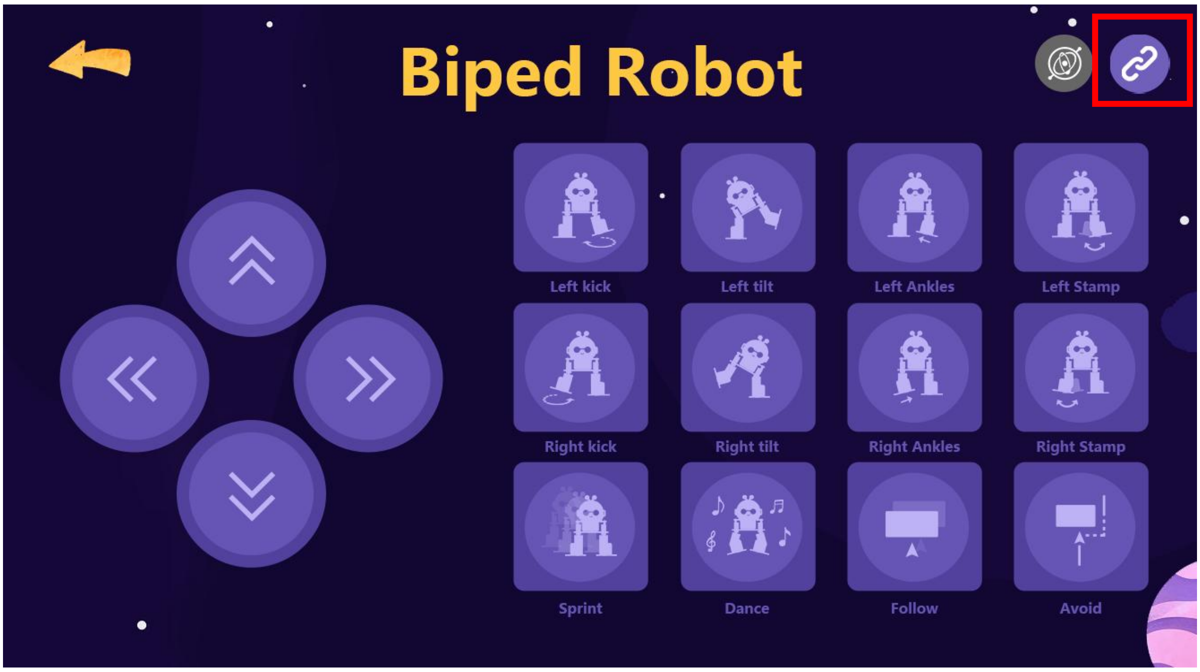


Attention: The name and password of the hotspot are already defined in the program, but the user can customize the modification, when we have multiple biped robots, we can distinguish each biped robot by different WiFi names.

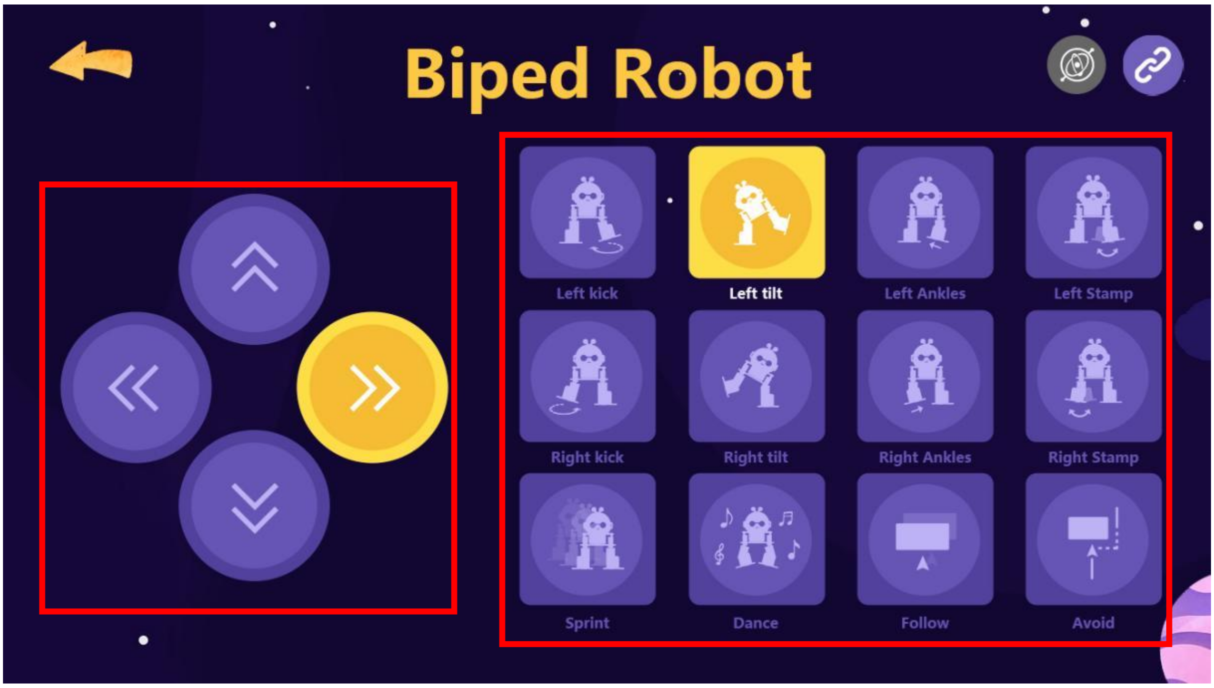
```
const char* ssid = "Biped_Robot";//WiFi name
const char* password = "12345678";//WiFi password
```

3.APP control

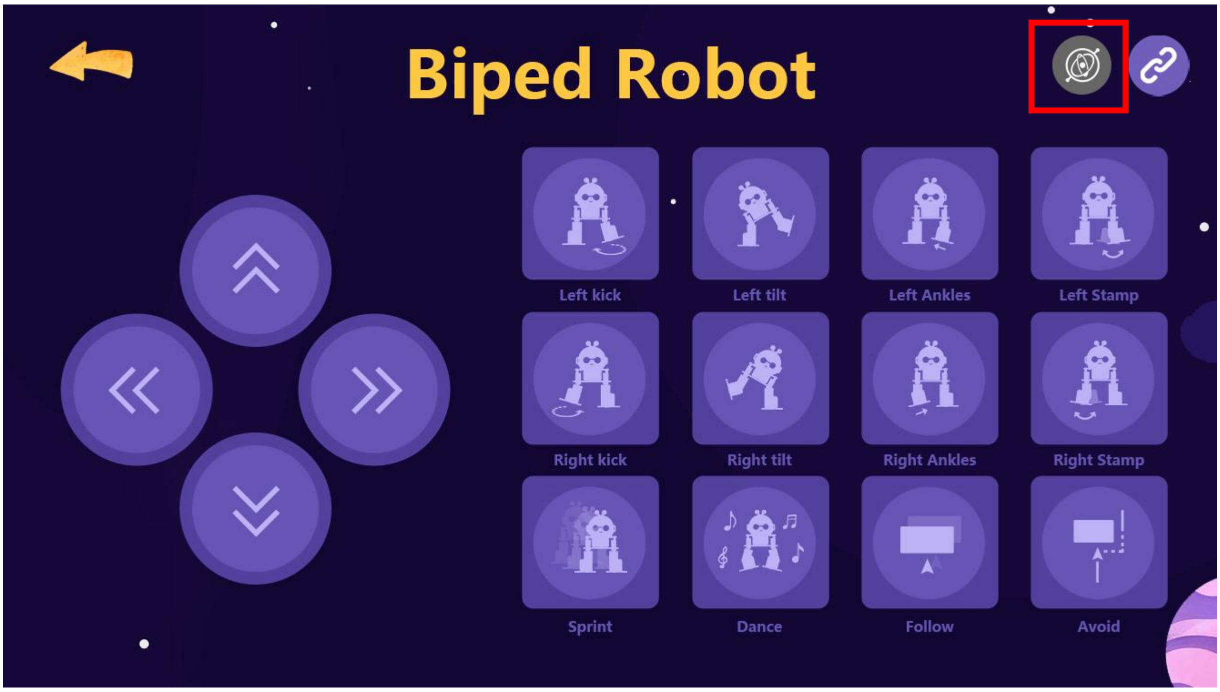
After the WiFi is connected, click the connection icon in the upper right corner of the APP to complete the connection.



After completing the above operations, return to the interface shown below, and then you can control the biped robot. The left side of the operation panel can control the robot's forward, backward, left, and right movements; the right side is the robot action group control, the main actions are: left kick, right kick, left stamp, right stamp, sprint, dance, follow, obstacle avoidance, etc.



In the upper right corner of the biped robot operation interface, gyroscope control is provided. Click this button to control the movement of the bipedal robot through the gyroscope of the mobile phone. If the mobile phone does not have a built-in gyroscope, you can ignore this function.



Follow Us

Scan the QR codes to Follow Us for troubleshooting & the latest news.
We have a very large community that is very helpful for troubleshooting and we also have a support team at the ready to answer any questions.

ACEBOTT FB Group QR Code YouTube QR Cod