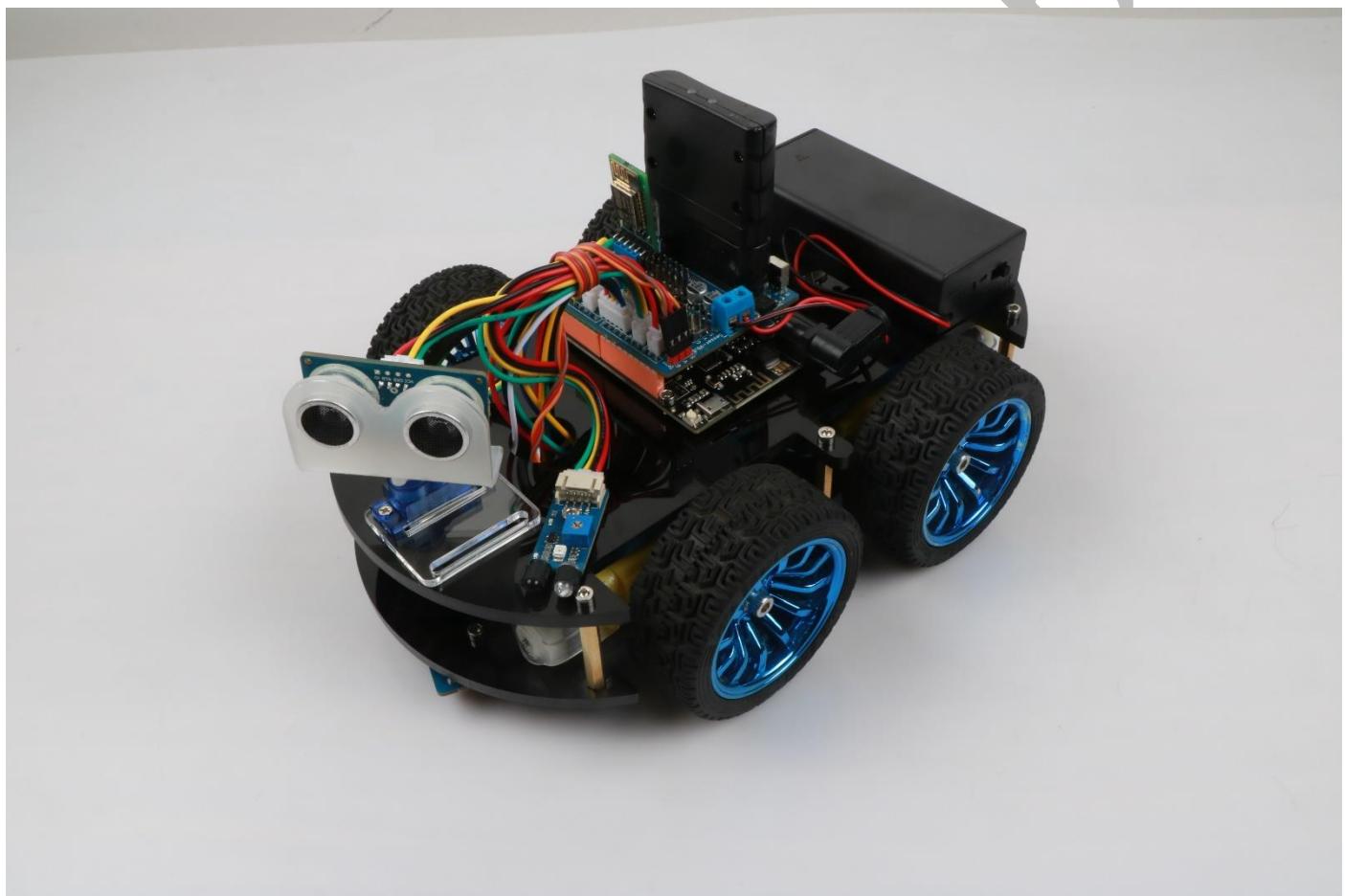




Hummer-bot_4.0

Mixly Graphical Programming Course V.1.0





Version revision

date	version number	describe	author
2019-7-9	V.1.0	create documents	Twisted

KeyWish-robot



Catalog

Chapter 1 Understanding Hummer-bot and Its Programming Environment.....	5
1.1 Introduction of Life.....	5
1.2 Summary.....	5
1.2.1 Introduction to Hummer-bot	5
1.2.2 Hummerbot Function List.....	5
1.3 Hummer-bot Introduction to Hardware	6
1.3.1 Introduction to Analogy	6
1.3.2 Introduction of Main Control Board	7
1.3.3 Introduction to Extension Board	8
1.4.1 Software and Driver Installation	8
1.4.2 Introduction to Compiler Environment.....	12
1.5 Introduction Of Blocks Area.....	13
1.6 Introduction of building blocks.....	14
1.6.1 Introduction of Hummer-bot Building Block	15
Chapter 2 Initial Knowledge Programming.....	17
2.1 Hello word	17
2.1.1 Add Hummer-Bot Library.....	17
2.1.2 First Experience of Programming.....	21
2.2 Connect Mixly With Hummer-Bot.....	22
2.3 Upload program to Hummer-Bot robot	22
Chapter Three Robots Move.....	24
3.1 DC motor	24
3.1.1 Principle of DC Motor	24
3.1.2 DC motor test.....	24
3.1.3 Programming control and debugging the robot to take the square route.....	27
3.1.4 Discover the Rule of Program and Optimize Program	29
3.2 Ultrasound module.....	30
3.2.1 Principle of Ultrasound	30
3.2.2 Usage of Ultrasound Module	30
3.2.3 Testing of the Ultrasound Obstacle Avoidance Module	31
3.2.4 Writing Robot Program for Tango Dance	32
3.2.5 Optimizing Dancing Robot Program	34
3.3 Ultrasonic obstacle avoidance robot	35



3.3.1 Operating Principle of Steering Engine	35
3.3.2 Steering gear test.....	36
3.3.3 Programming of Obstacle Avoidance Robot	37
3.4 Infrared Barrier Avoidance Robot	39
3.4.1 Principle of Infrared Obstacle Avoidance Module	39
3.4.2 Usage of Infrared Obstacle Avoidance Module.....	40
3.4.2 Test of Infrared Obstacle Avoidance Module.....	41
3.4.3 Robot Writing Runway	42
3.5 Ultrasound+Infrared Barrier Avoidance Robot	43
Chapter IV Tracking Robot	47
4.1 Principle of Tracking Module	47
4.2 Use of Trace Module	47
4.3 Testing of Infrared Tracking Module	48
4.4 Writing Tracking Robot Program	49
Chapter 5 Infrared Telecontrol Robot.....	52
5.1 Principle of Infrared Remote Control	52
5.2 The Use of Remote Controller.....	52
5.3 Infrared remote control test.....	53
5.4 Programming of Remote Control Robot.....	54
Chapter 6 Bluetooth Remote Control Robot.....	57
6.1 Bluetooth Control Principle	57
6.2 Bluetooth Module and Hummer-Bot Robot APP KeywishBot Connection Method	58
6.3 Bluetooth module testing	59
6.4 Programming Thought of Bluetooth Remote Control Robot	60
Chapter 7 PS2 Telecontrol Robot	62
7.1 Principle of PS2 remote control handle	62
7.2 PS2 remote control test	63
7.3 Programming Thought of PS2 Remote Control Robot.....	64
Chapter 8 Rgb Robots	67
8.1 RGB lamp	68
Chapter 9: Light-finding Robots	70
9.1 Light-finding Module and Principle	70
9.2 Light-fingding Program	70



Chapter 1 Understanding Hummer-bot and Its Programming Environment

1.1 Introduction of Life

Hello, everyone. Welcome to the world of Hummer-bot robots!

What is a robot? Robots don't necessarily have to be long as human beings call them robots. Generally, robots can perform their work automatically. They can be commanded by human beings and run pre-programmed programs. Such things can be called robots.

What are the responsibilities of robots? Robot, originally Robo, means slave, that is, human servant, so the duty of robot is to help human work.

1.2 Summary

1.2.1 Introduction to Hummer-bot

Through the study of this tutorial, we will experience the interactivity of the physical world and software. Hummer-bot consists of three components: hummer-bot smart car, Arduino UNO master board and Mixly graphical programming software. Mixly is a graphical programming learning software for STEAM education, which is developed based on Scratch 2.0. It not only enables users to create interesting stories, games, animations and other works, but also supports Arduino hardware programming. We can use this software to control the Arduino UNO motherboard on Hummer-bot, so that Hummer-bot can issue various commands and control it. In this software, programming is no longer tedious code, the core logic of programming is accomplished by graphical representation and drag-and-drop interaction. The whole programming is like building an interesting building block toy. Through Mixly's stage area or Hummer-box's behavior, we can intuitively see the actual effect of various script designs.

1.2.2 Hummerbot Function List

- 1) Obstacle Avoidance Function of Ultrasound
- 2) Ultrasound + Infrared Barrier Avoidance Function
- 3) Automatic Tracking Function
- 4) Infrared remote control function
- 5) Bluetooth remote control function
- 6) PS2 remote control function
- 7) Bluetooth Remote Mode Switching Function
- 8) Tracking function
- 9) RGB lamp



1.3 Hummer-bot Introduction to Hardware

1.3.1 Introduction to Analogy

The hardware part of the Hummer-bot robot we use consists of the body model built by acrylic board, various sensors, motor wheel combination, control board and power supply battery, as shown in Figure 1-3-1-1.

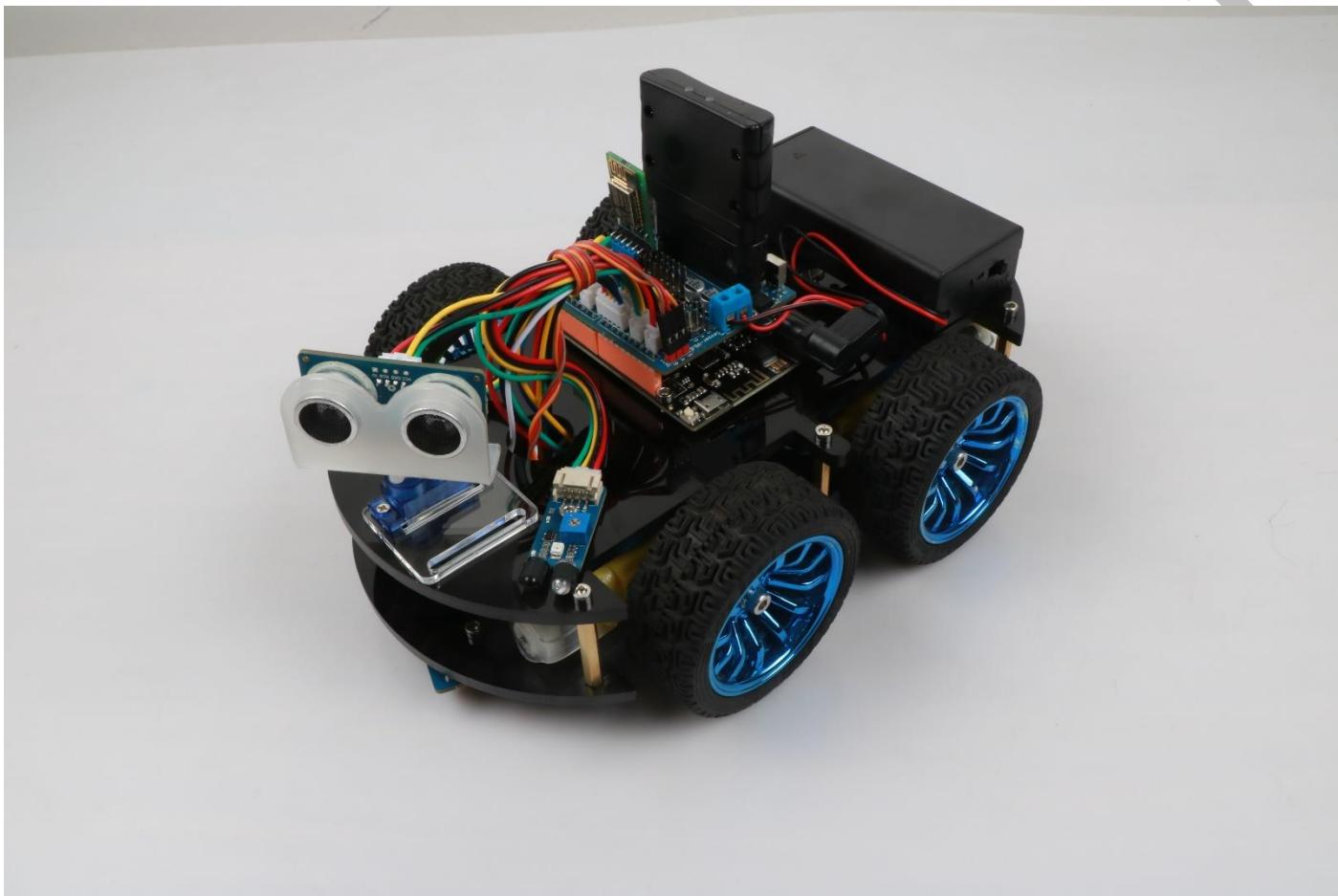


Figure 1-3-1-1

In this robot, every part of it is similar to the structure of our human body and plays a vital role. It is similar to human beings, as shown in Table 1-3-1-2.

Robot	Human beings	Effect
Control board	Brain	Accept instructions, process information, and give instructions to control limbs and organs
Motors and wheels	limb	Responsible for limb movements
Battery	heart	Providing power

HUMMER-BOT ROBOT

sensor	Facial features	Acceptance of perceived information
electronic circuit	nerve	Connect each part of the limb to form a whole.

Table 1-3-1-2

1.3.2 Introduction of Main Control Board

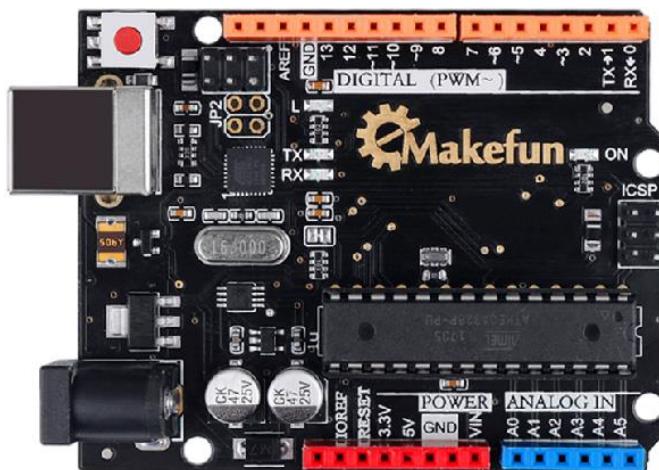


Figure 1-3-2-1



1.3.3 Introduction to Extension Board

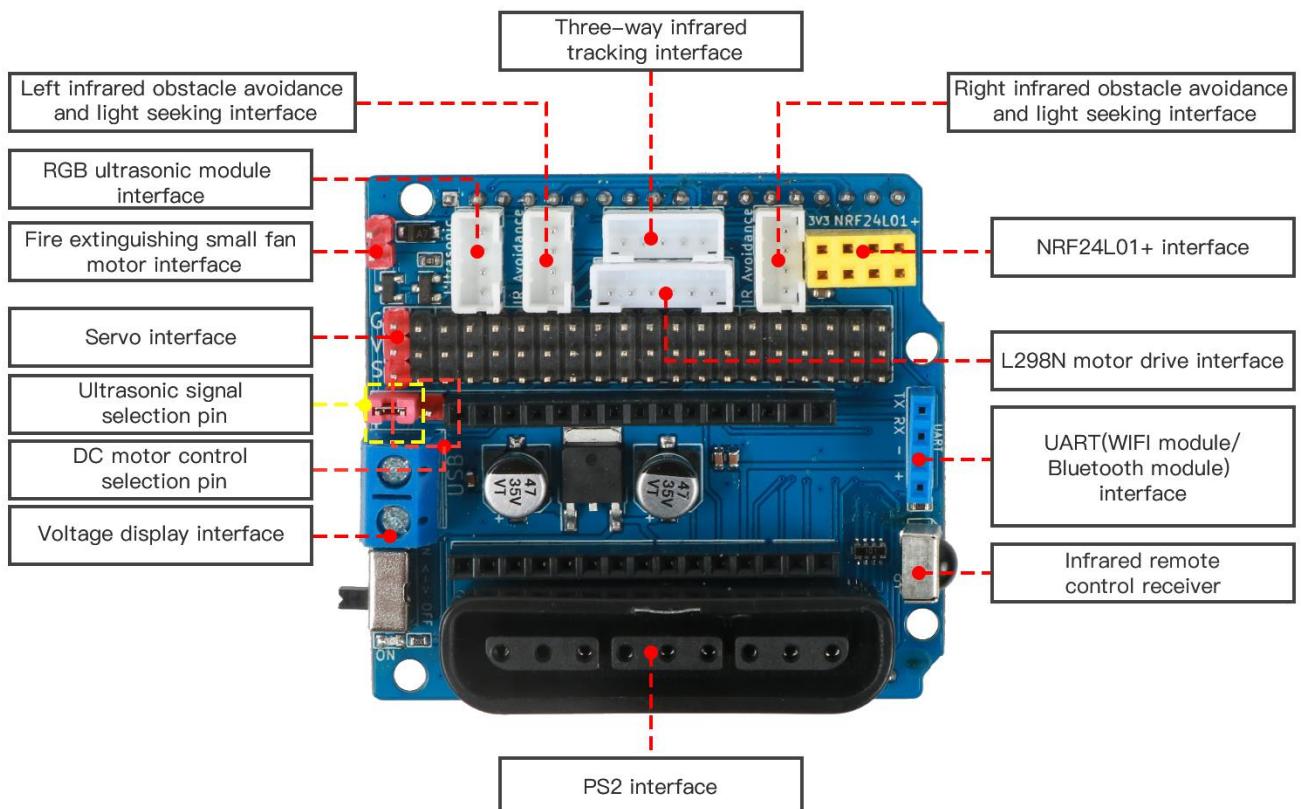


Figure 1-3-3-1

1.4.1 Software and Driver Installation

Installation Procedures of Windows Version Mixly Software



Figure 1-4-1-1

After downloading and decompressing the Windows version, double-click Mixly.exe to open the Mixly software

版本	发布日期
0.998	2018-07-08
0.997	2018-01-23
0.996	2017-11-06
0.995	2017-04-21
0.984	2016-10-29
0.96	2016-01-04
0.95	2015-11-16
0.94	

Figure 1-4-1-2



Figure 1-4-1-3

Installation Procedures of Mac Version Mischi Software

The Mac version of Mischi software installation steps include three steps:

1. Install JDK

Mac version mainly runs in Java environment, so when installing, you need to install JDK. In this disk, you can directly install jdk-8u71-macosx-x64.dmg file, or according to your own system, you can choose the corresponding JDK in Oracle official website.

2. Download Mischi Software Compression Package

At present, the official website only supports the download of Mischi Mac software on Baidu Disk. Without Baidu Disk, Baidu can search for Mischi Mac version to download. After downloading, decompression, and running Mixly.jar file, Mischi software can be opened on Mac. If the software can not run, please select "General" page in "Preference Settings" - "Security and Privacy", and allow the software to use.

3. Installation driver

If the Mixly software is installed, we write the code. If we want to run on the device, we need to install the serial driver. The serial driver is CH341SER_MAC for the download file. The software can be installed directly. After successful installation, we can see the new device in the "About the Native Machine" - "Overview" page - "System Report" - "USB". At this point, the basic installation is completed.



首页 / 软件平台 / Mixly 官方版 / Mixly For Arduino (Mac)



Mixly For Arduino (Mac)

0.998

★ 赞 2

下载地址: [百度云盘](#)

版本说明

Mixly0.998主要修改

- 1.修正了DS1302 RTC问题
- 2.所有板子都支持Factory，并且加入了多行代码直接复制或输入，方便使用混合编程或开发自己的库
- 3.修改了字体，中英文混排更美观
- 4.代码界面使用ACE,更加方便和简介
- 5.增加了Python图形化编程
- 6.允许在多个Mixly之间复制 `ctrl+shift+c, ctrl+shift+v`
- 7.自动识别普通视图和高级视图
- 8.修复编译时图形块突然消失，切换板子管脚不切换的bug
- 9.修复打开Mixly时还会加载上次缓存中图形块的bug
- 10.修复编译时图形块突然消失，切换板子管脚不切换的bug
- 11.增加RFID,ADXL345模块
- 12.解决了配置文件丢失的问题
- 13.解决了代码区不能滚动的问题
- 14.`microbit(py)`增加了`oled, lcd1602, ws2812, ds1307`的支持。
- 15.`micropython, python`实现了语法的统一。



米思齐团队

版本历史

0.998

2018-07-08

0.997

2018-01-23

0.995

2017-04-21

© 2014 - 2018, Mixly Team. 京ICP备13037033号-1

Figure 1-4-1-4

- Mixly0.997_MAC.zip
- jdk-8u71-macosx-x64.dmg
- CH341SER_MAC.ZIP

Figure 1-4-1-5

The interface of the Mac version of Missy software is shown in Figure 1-4-1-5.

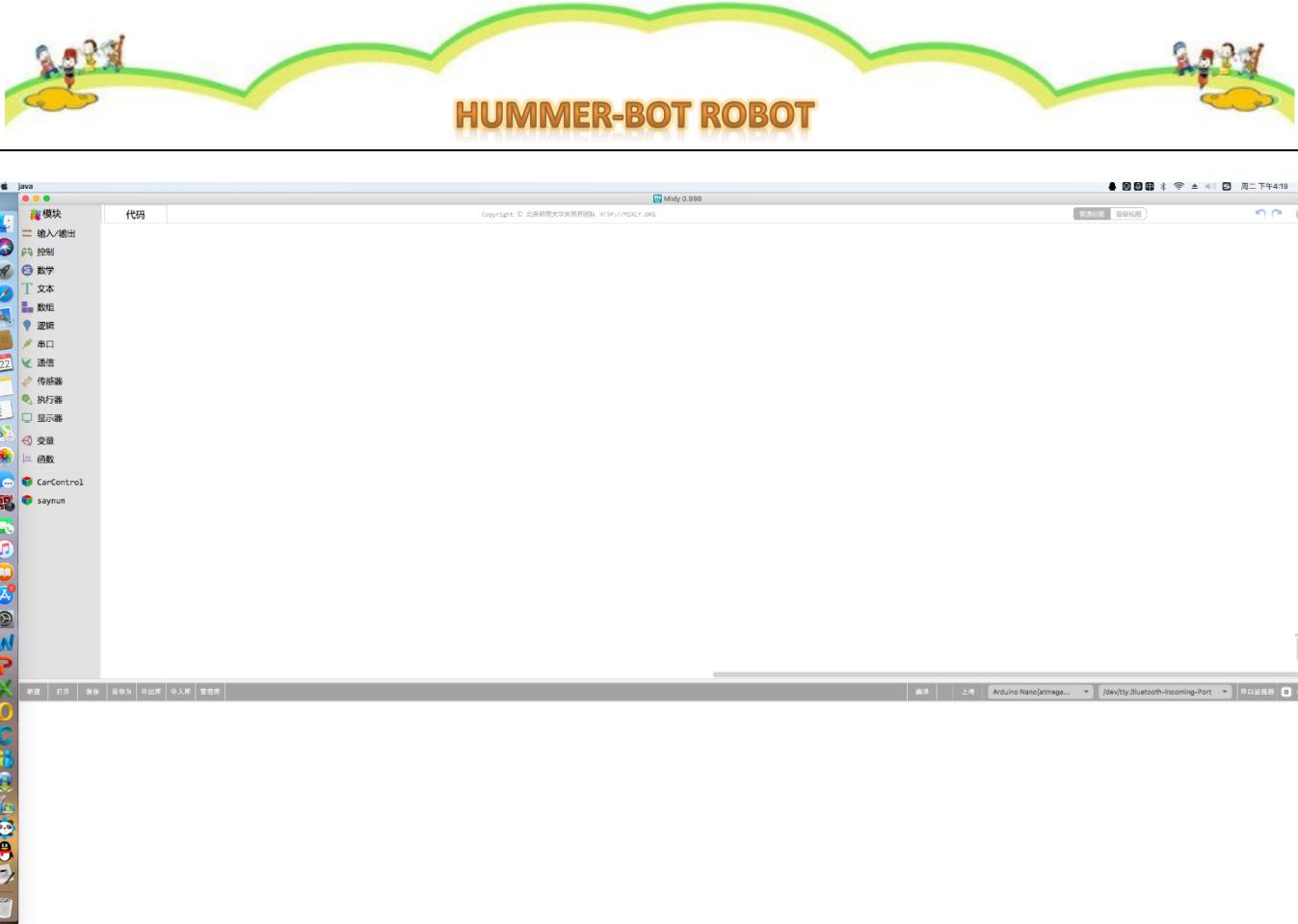


Figure 1-4-1-5

1.4.2 Introduction to Compiler Environment

We know that in order for a robot to move, besides the hardware cooperation, the more important thing is to write a program for it. We already know the hardware part of the robot, and then we will know its software part. Hummer-Bot's software part is programmed on the graphical programming software Mixly. With this software, we can control the robot by writing various commands that we want it to execute. Mixly's software interface is shown in Figure 1-4-2-1.

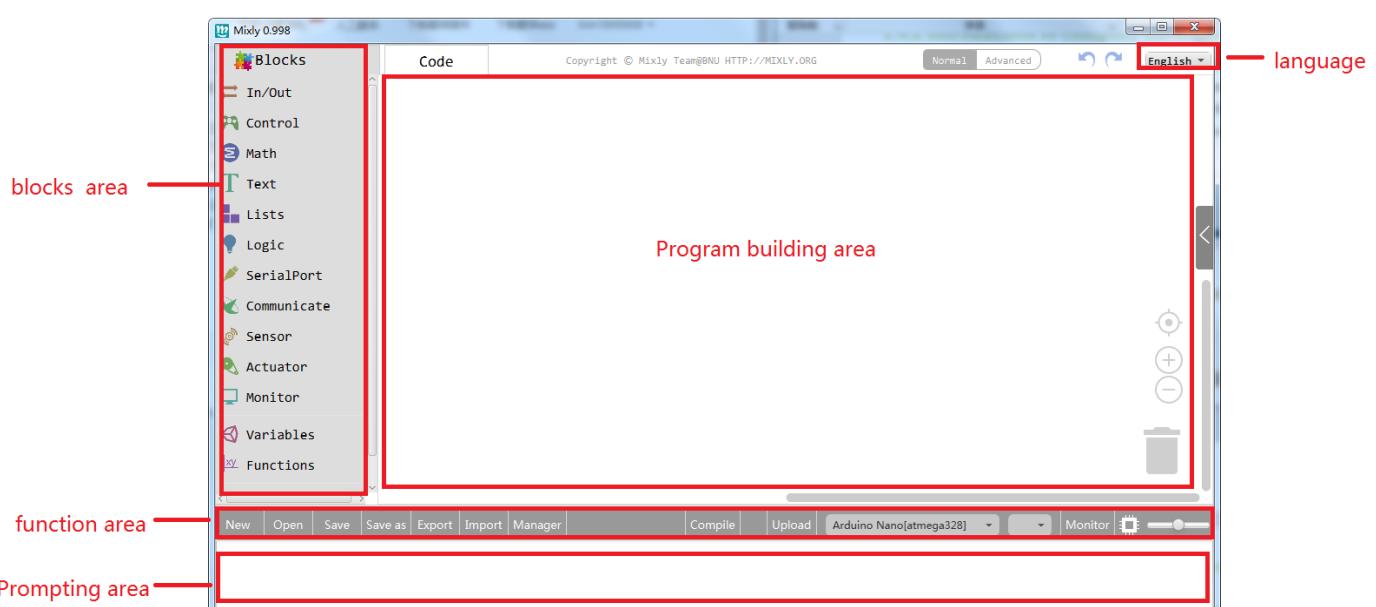




Figure 1-4-2-1

- blocks area:** The same type of building blocks are divided into the same module and given the same color. Each building block represents a control instruction.
- function area:** Create and save project files, import, export and manage libraries, connect and upload programs through serial ports, and select the area for operation by the control board.
- program building area:** Place the building blocks pulled out of the base module area and the reservoir module area.
- language switching button:** You can switch between simplified Chinese, traditional Chinese, English and Spanish.

1.5 Introduction Of Blocks Area

Mixly's blocks area are divided into some building blocks brought by Mixly software and the library building blocks imported by Mixly itself. These blocks can correspond to C language code one by one. We can write our favorite program by splicing different blocks. We don't even need to know what code each block corresponds to. Because when we drag and pull out the blocks, the program is written by us. We can see these C language codes by clicking on the "code" field. Figures 1-5-1 and 1-5-2 are modules and codes, respectively.

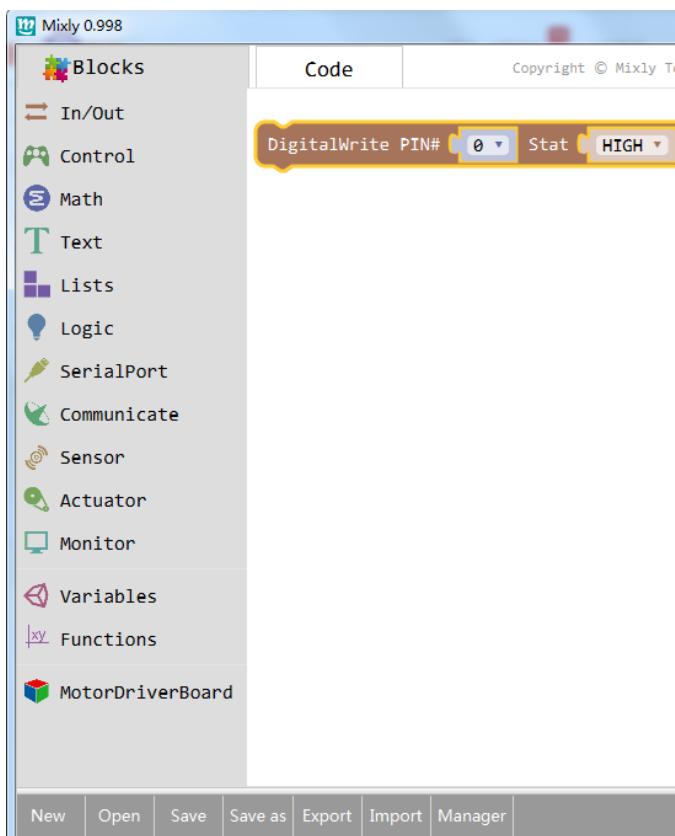


Figure 1-5-1

```
1 void setup(){
2   pinMode(0, OUTPUT);
3 }
4
5 void loop(){
6   digitalWrite(0,HIGH);
7 }
8 }
```

Figure 1-5-2



1.6 Introduction of building blocks

Mixly's script label has many types of building blocks, such as input/output, control, mathematics, text, array, serial port, communication, etc. If you are interested in it, you can try it out by yourself. Let's not go into details here. We mainly know four types of building blocks: control, mathematics, serial port and logic.

- 1) Control building blocks are the building blocks that control the execution process of the program. The main program is shown in Fig. 1-6-1.

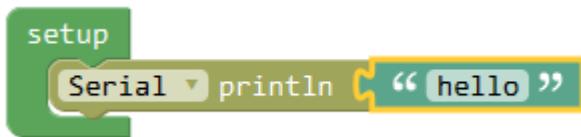


Figure 1-6-1

```

1 void setup(){
2   Serial.begin(9600);
3   Serial.println("hello");
4 }
5
6 void loop(){
7 }
8

```

Figure 1-6-2

Initialize the setup function in the corresponding code of building blocks, dragging some building blocks into the initial building blocks means that the dragged building blocks program will run once. If the building blocks are dragged outside the initial building blocks, then these programs will enter the loop function, and then they will execute these programs in a loop. As shown in figure 1-6-2

- 2) The main function of building blocks of digital and logical operation types is to do mathematical operation, which is used as the condition of judgment, to compare size and logical judgment with, or with, or without, as shown in Figure 1-6-3.

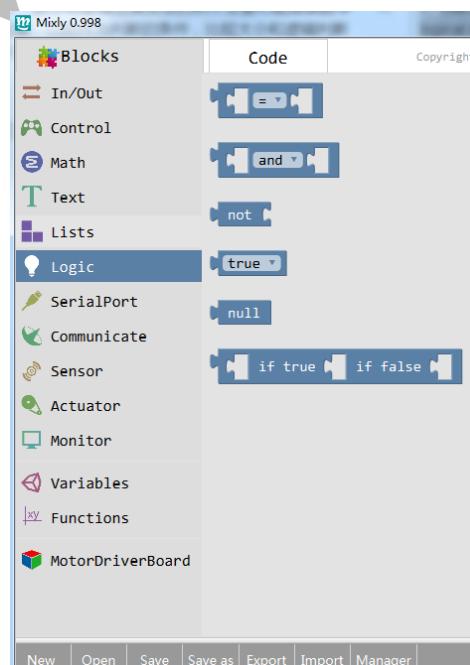




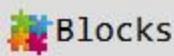
Figure 1-6-3

1.6.1 Introduction of Hummer-bot Building Block

Hummer-bot is a library we wrote for Hummer-bot. After importing the library, it will appear in the library module area below the basic module area. Click on it and it will appear various small building blocks, as shown in Figure 1-6-1-1. Next, we will learn how to write programs for Hummer-bot by assembling these modules.

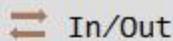
HUMMER-BOT ROBOT

Mixly 0.999



Code

Copyright



Hummerbot Initialization



Hummerbot



forward ▾ Speed

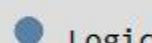
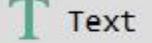
0



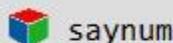
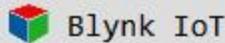
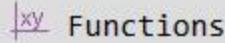
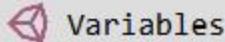
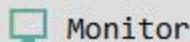
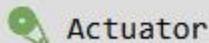
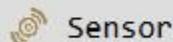
Control mode



bluetooth control ▾



Logic



Set Servo



Servo pin 13

Correction angle 90

Set Servo



90

Set motor pin



Left negative pole 6

Left positive pole 10

Right negative pole 5

Right positive pole 9

Turn a corner



Angle 0

Speed 0

Speedup

Speeddown



Figure 1-6-1-1

Chapter 2 Initial Knowledge Programming

2.1 Hello word

If we want the Hummer-Bot robot to move, we need to store instructions (programs) in its brain (control board) beforehand for the Hummer-Bot robot. How to compile instructions for the Hummer-Bot robot? Now let's take you through the experience of writing a Hummer-Bot robot print Hello word program.

2.1.1 Add Hummer-Bot Library

Before programming, we need to add the Hummer-Bot library. The steps are as follows:

- 1) Download Hummer-Bot library and save it on your computer
- 2) Open the Mixly software and click "Import Library", as shown in Figure 2-1-1-1;
- 3) Click on the Hummer-Bot.xml file and then click Open, as shown in Figure 2-1-1-2.

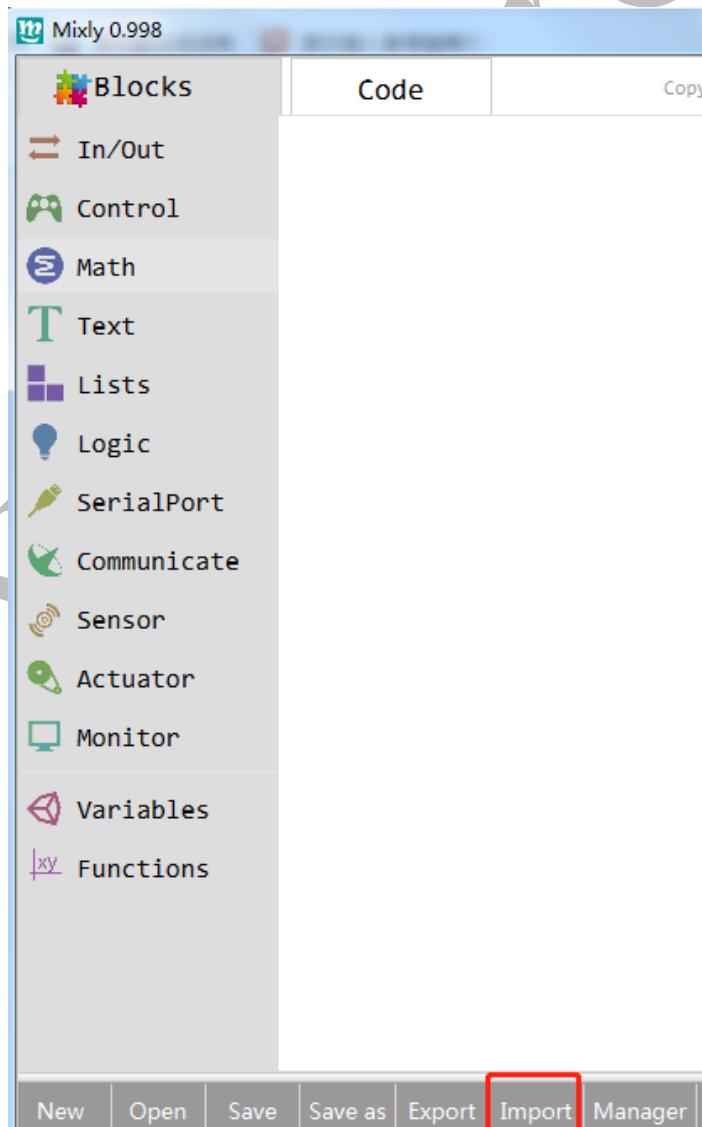


Figure 2-1-1-1

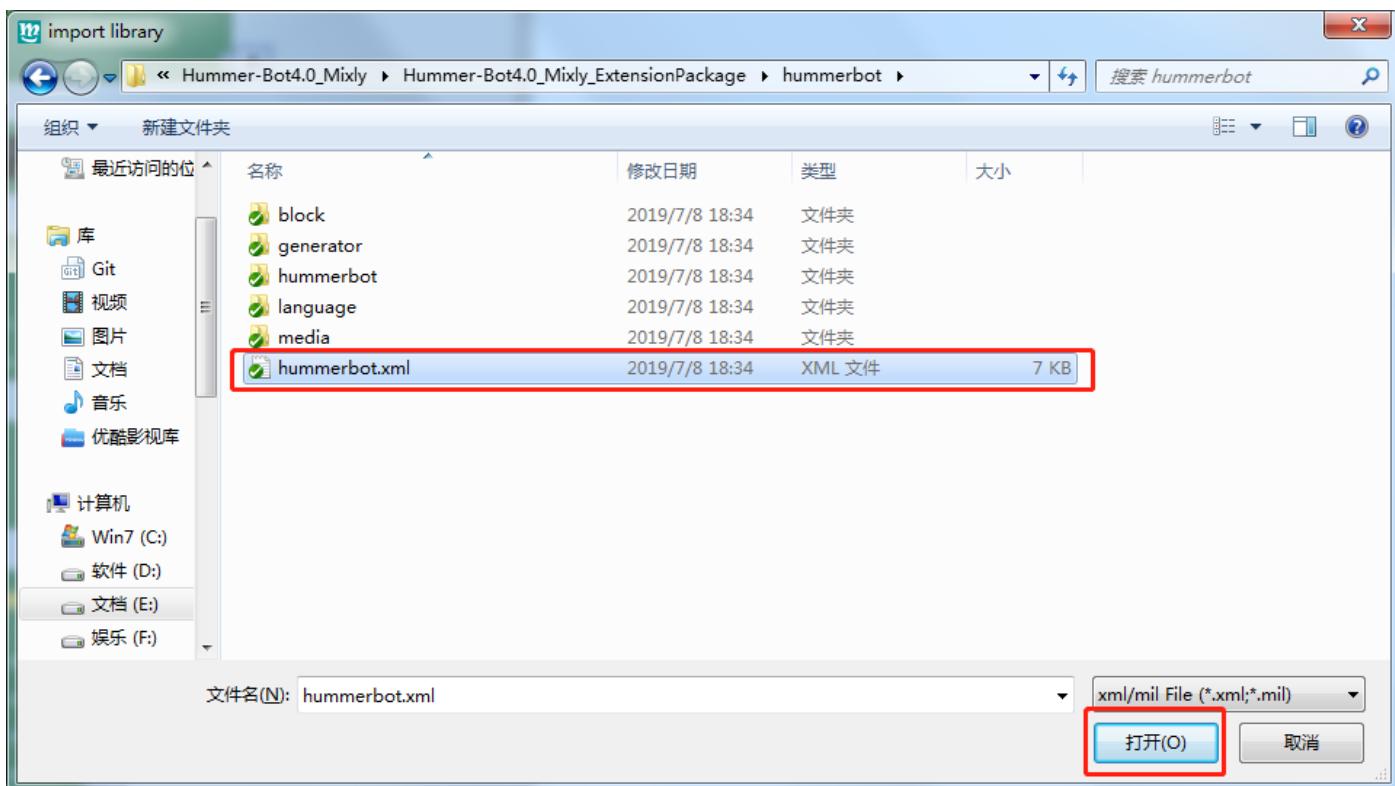


Figure 2-1-1-2

- 4) After successful import, the prompt area will show that the import of custom library is successful! In the library module area, you can see the Hummer-Bot library module as shown in Figure 2-1-1-3.

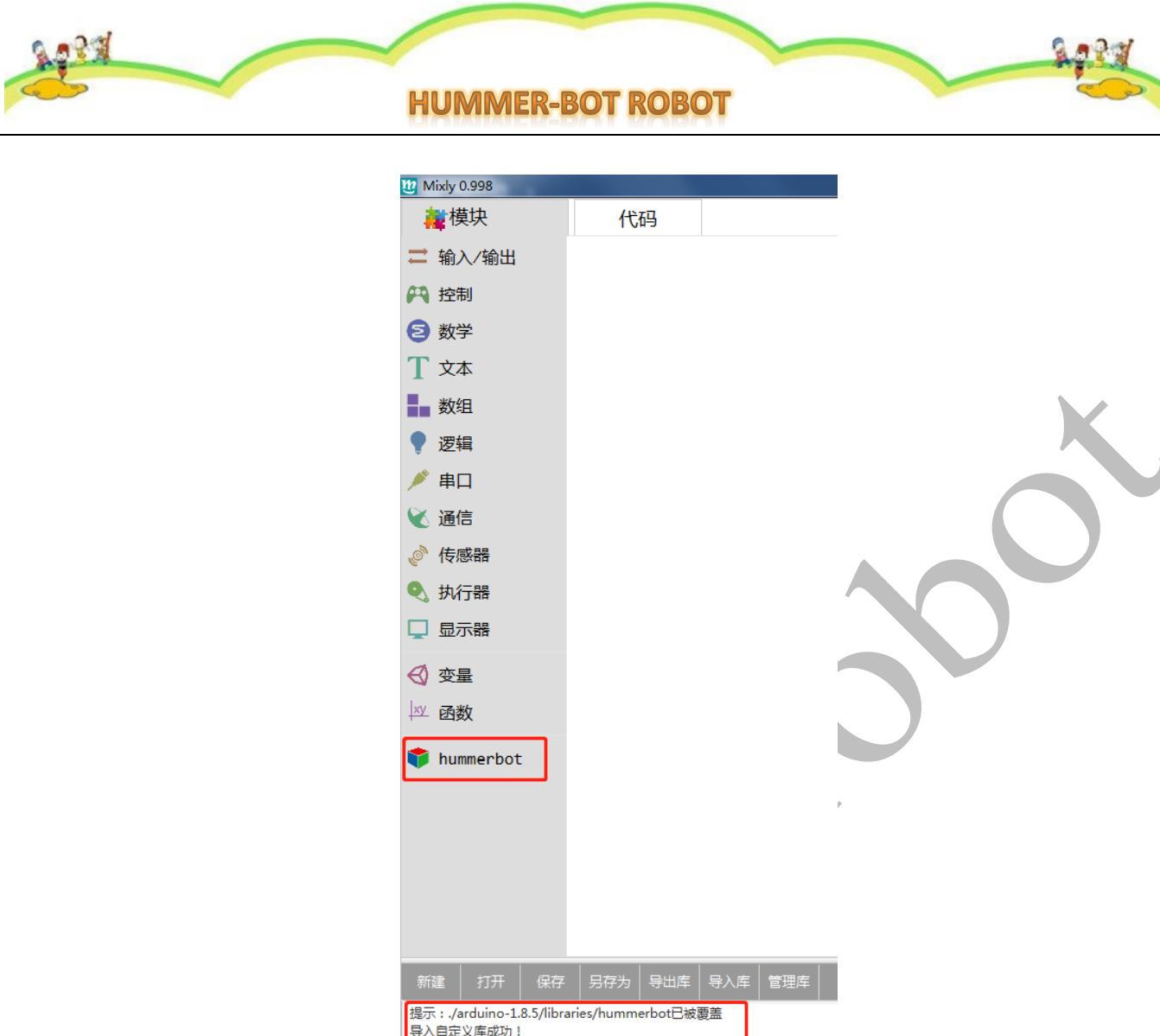


Figure 2-1-1-3

- 5) 点击“hummerbot”，就会显示 hummerbot 积木 Figure 形编程块，如 Figure 2-1-1-4 所示。

HUMMER-BOT ROBOT

Mixly 0.999

Blocks

In/Out

Control

Math

Text

Lists

Logic

SerialPort

Communicate

Sensor

Actuator

Monitor

Variables

Functions

Blynk IoT

hummerbot

CarControl

saynum

Code

Copyright

Hummerbot Initialization



Hummerbot forward Speed

0

Control mode bluetooth control

Set Servo



Servo pin

13

Correction angle

90

Set Servo



90

Set motor pin



Left negative pole

6

Left positive pole

10

Right negative pole

5

Right positive pole

9

Turn a corner



Angle

0

Speed

0

Speedup

Speeddown



Figure 2-1-1-4

2.1.2 First Experience of Programming

After adding the Hummer-Bot library, let's experience Hummer-Bot programming. Let's first write a program for Hummer-Bot robot to print Hello word.

- 1) First, the "Hummer Initialization" in the building block area is dragged to the program architecture area with the mouse, and then the "Serial Baud Rate 9600" is dragged to the bottom of the "Hummer Initialization" building block, as shown in Figure 2-1-2-1.



Figure 2-1-2-1

- 2) Drag the "Serial Printing (Automatic Line Breaking)" building block under the "Serial Baud Rate 9600" in the serial module, as shown in Figure 2-1-2-2.

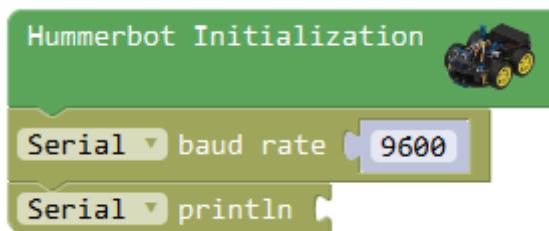


Figure 2-1-2-2

- 3) In the text module, drag the building blocks of "hello" to the right of Serial Printing (automatic line breaking) and enter "Hello word" as shown in Figure 2-1-2-3.

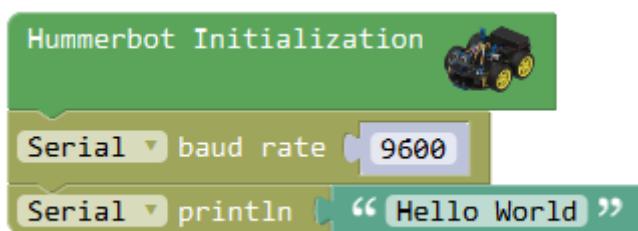


Figure 2-1-2-3

This is the Hummer-Bot Robot Printing Hello Word program. After the program is written, we need to transfer the program to the brain of the Hummer-Bot robot (control motherboard) to do the desired action according to the program we wrote. How to transfer the program to the brain of the Hummer-Bot robot (control motherboard)? Only when Mixly is connected to the robot motherboard, can we transfer the written program from the computer to the brain of



HUMMER-BOT ROBOT

the Hummer-Bot robot (control motherboard). Here's how Mixly connects to the robot motherboard.

2.2 Connect Mixly With Hummer-Bot

- 1) Using a USB data line, one end is inserted into the computer, and the other end is inserted into the robot master board, which connects the robot master board with the computer.
- 2) Click the drop-down box on the right of "Upload" to select the model of the main console, and then click the second drop-down box on the right to select the USB serial port, as shown in Figure 2-2-1. When the selection is completed, Mixly and Starter-box are successfully connected.

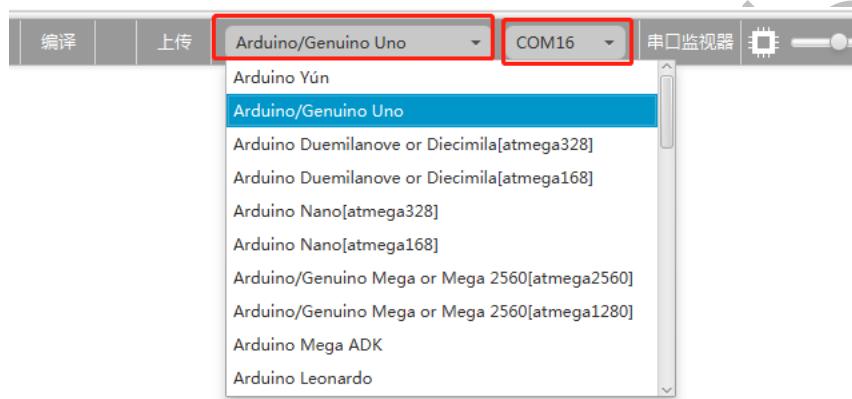


Figure 2-2-1

2.3 Upload program to Hummer-Bot robot

When the program is finished and Mixly and Hummer-Bot robots are connected correctly, we can transmit the program to the brain of Hummer-Bot robots (main control board). The specific steps are as follows:

- 1) Click on upload, and the building blocks in the program architecture area begin to upload to the control board. We need to wait for a while. After the upload is completed, we will prompt "upload completed", as shown in Figure 2-3-1.

After completing the above steps, the Hummer-Bot robot's brain (main control board) already has the program we wrote. If the upload succeeds, it will be prompted to upload successfully. Of course, we can click on "code" at any time to see the program that is actually uploaded to the control board. These are the C language codes corresponding to building block graphics, as shown in Figure 2-3-2.



Figure 2-3-1

The screenshot shows the Mixly 0.998 software interface. On the left, there is a 'Blocks' tab and a 'Code' tab. The 'Code' tab contains the following C++ code:

```
1 #include<Arduino.h>
2 #include<SoftwareSerial.h>
3 #include<ProtocolParser.h>
4 #include<BluetoothHandle.h>
5 #include<KeyMap.h>
6 #include<debug.h>
7 #include<StarterBot.h>
8
9 ProtocolParser *mProtocol = new ProtocolParser();
10 StarterBot Ebot(mProtocol, EM_IN1_PIN, EM_IN2_PIN, EM_IN3_PIN, EM_IN4_PIN);
11 byte Ps2xStatus, Ps2xType;
12
13 void setup(){
14     Ebot.init();
15     Ebot.SetSpeed(0);
16     Ebot.SetControlMode((E_SMARTCAR_CONTROL_MODE)(0));
17     Ps2xType = Ebot.mPs2x->readType();
18
19     Serial.begin(9600);
20 }
21
22 void loop(){
23     Serial.println("Hello World");
24 }
```

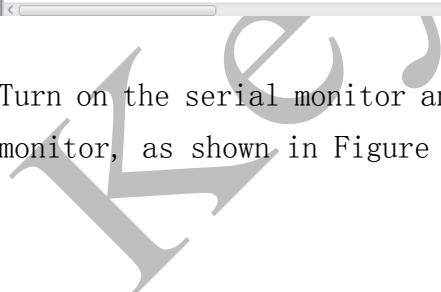
On the right, there is a terminal window showing the output of the avrdude command:

```
avrduke: verifying ...
avrduke: 4568 bytes of flash verified
avrduke done. Thank you.
```

A red box highlights the message "Upload success!" in the bottom left corner.

Figure 2-3-2

Turn on the serial monitor and we will see Hello Word printing continuously on the serial monitor, as shown in Figure 2-3-3.



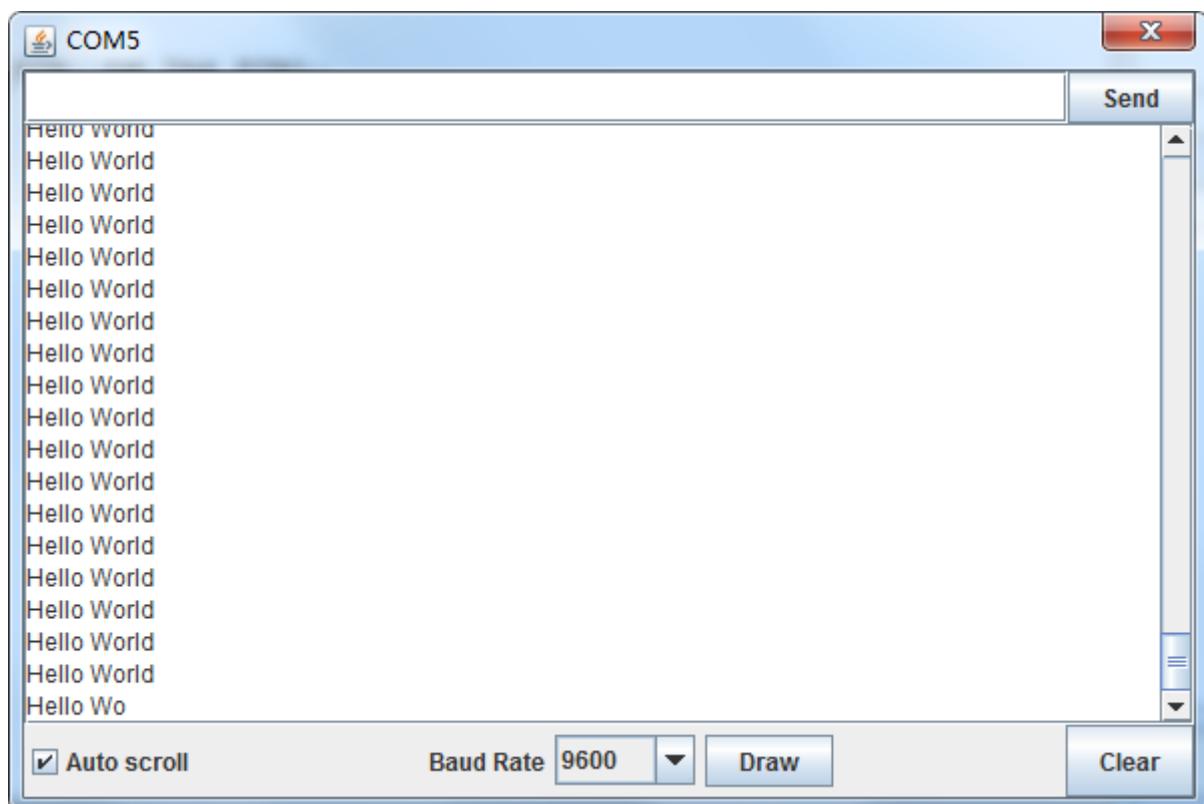


Figure 2-3-3

Chapter Three Robots Move

3.1 DC motor

3.1.1 Principle of DC Motor

The reason why cars are active is that they have engines to power them. Robots also have DC motor modules that allow them to move, so what is a motor? In our science textbook, there is an introduction of electromagnetic induction. The motor is rotated by electromagnetic induction. It has an iron core winding copper wire inside and a rotor outside. When the iron core is electrified, there is electromagnetic induction to make the rotor move. This is the motor.

Hummer-Bot robot has two DC motors. We can use DC motors to control building blocks to make DC motors turn, thus driving Hummer-Bot robot to move.

3.1.2 DC motor test

In the robot module, we can find three control blocks of DC motor module, which are motor pin building blocks, direction setting, speed building blocks, turning angle setting and speed building blocks, as shown in Figure 3-1-2-1.



HUMMER-BOT ROBOT

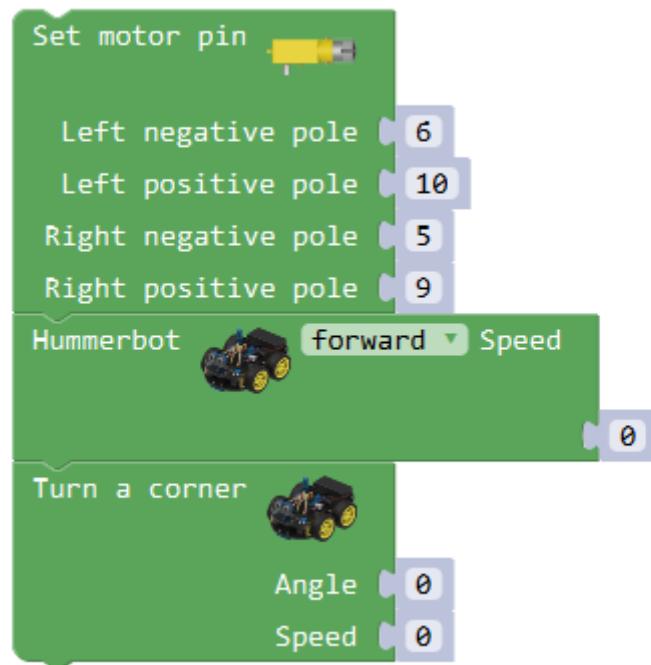


Figure 3-2-1

- Setting up the motor pin building blocks: Used to set the pins connecting the two motors of the control robot to the main control board. The default pins are selected in this tutorial.
- Set the direction and speed, speed building blocks: Set the direction of the robot movement and the speed in that direction;
- Set the turning angle and speed: Set the turning angle and speed when the robot moves.

Write the following program on Mixly. They can control the motor forward and reverse respectively, and then let the robot move forward, backward, left and right directions. They run the program separately and observe carefully the direction of two DC motors when the robot moves in each direction. The programming of the four directions is shown in Figure 3-2-2.

HUMMER-BOT ROBOT

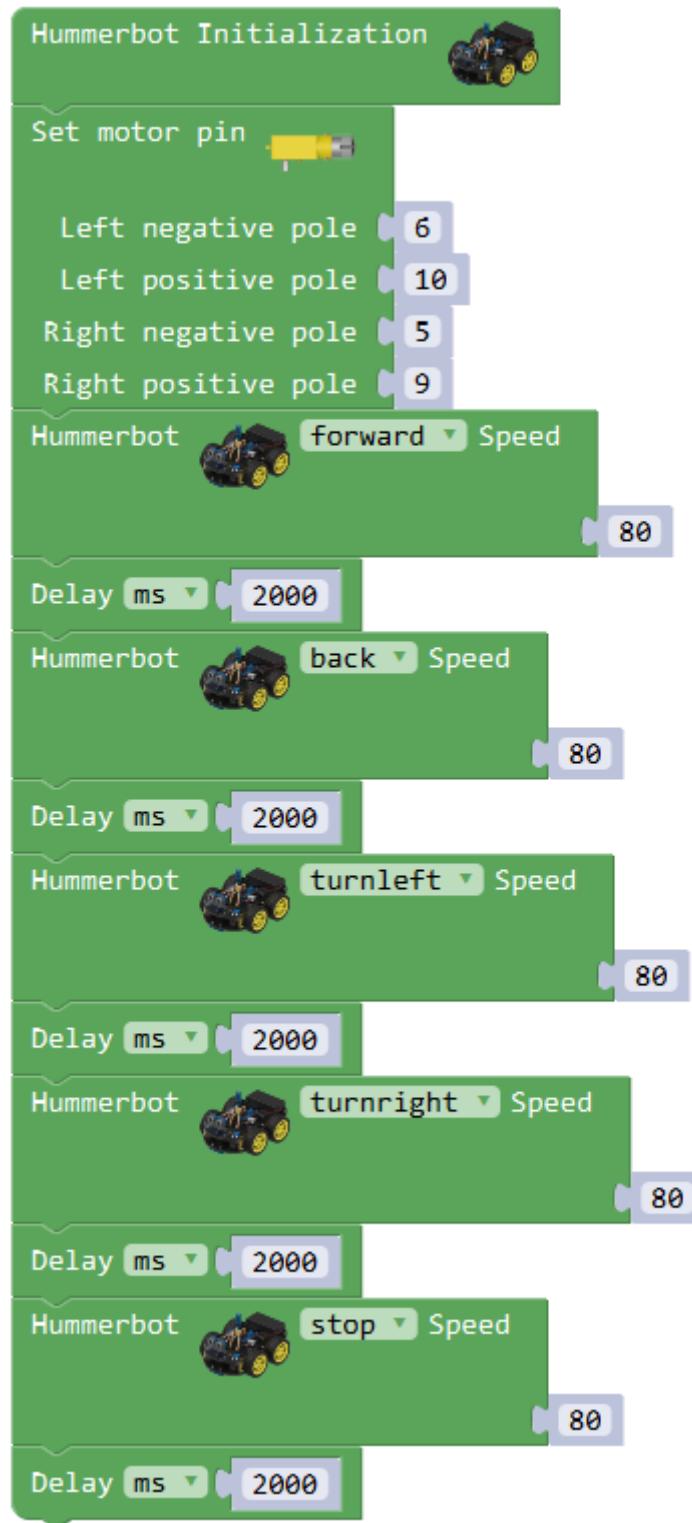


Figure 3-2-2-2

Through the above practical operation, we should have understood the programming method of letting the robot move in a single direction by controlling the motor. Now let's explore the programming method of letting the robot walk in a square.



3.1.3 Programming control and debugging the robot to take the square route.

The way the robot moves in a square is to go forward, turn left, move forward, turn left, move forward, turn left, turn left (reverse can also be done). The path of the robot is shown in Figure 3-1-3-1, and the reference program is shown in Figure 3-1-3-2.

KeyWish-robot



HUMMER-BOT ROBOT

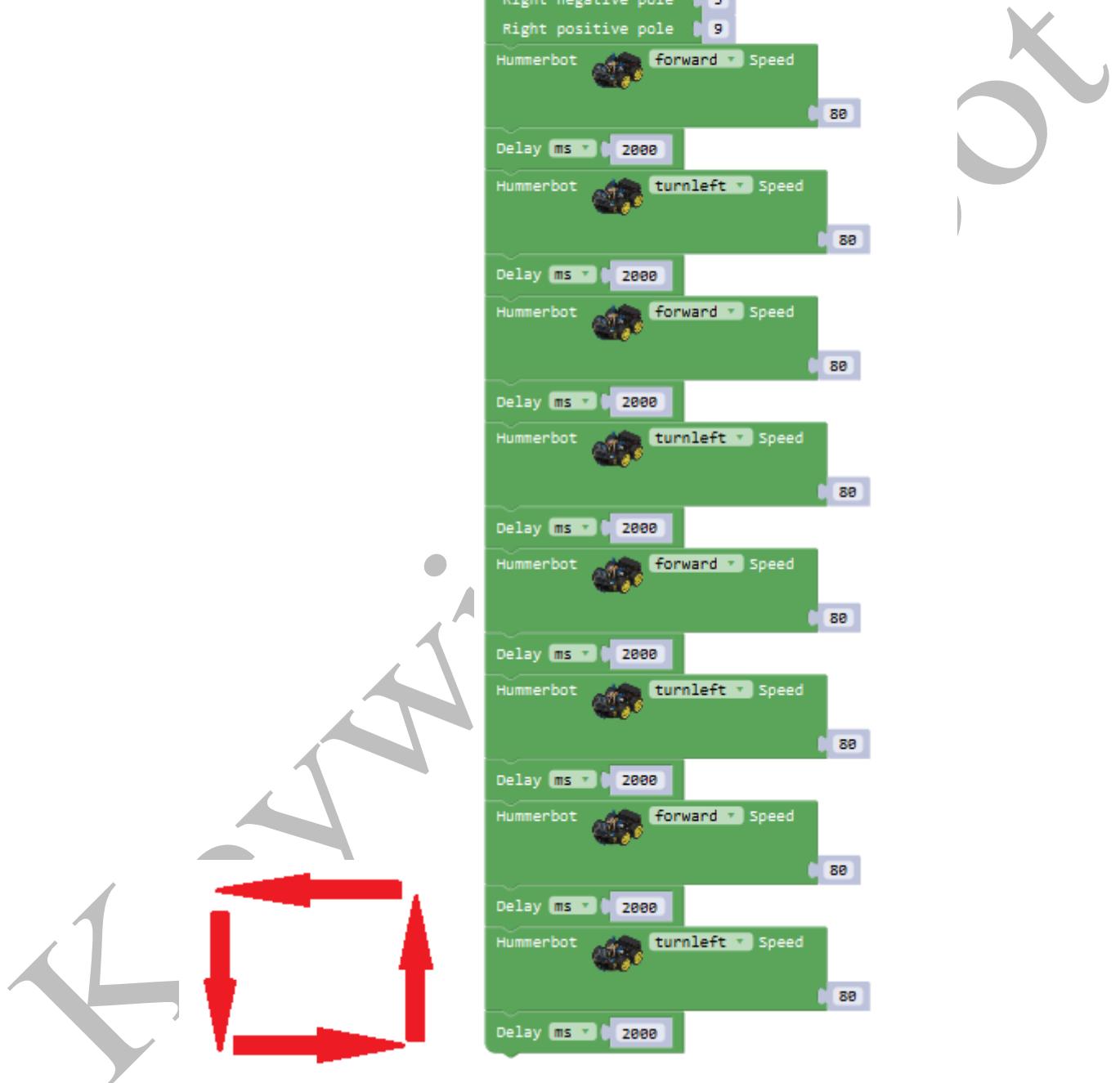


Figure 3-3-3-1

Figure 3-1-3-2

note:

- 1) The delay blocks under the "forward" building blocks are to control the forward distance of the robot, while the delay blocks under the left turn are to control the turning angle of the robot. When writing the program, the delay time should be modified according to the actual walking condition of



the robot.

- 2) At the end of the program, the motor must stop turning, otherwise the robot will remain in its original state of motion and turn left all the time.

3.1.4 Discover the Rule of Program and Optimize Program

By observing the program of Figure 3-1-3-2, we can find that the path of the robot walking square is actually to control the wheel to repeat four times forward and turn left 90 degrees. The optimization program is shown in Figure 3-1-4-1.

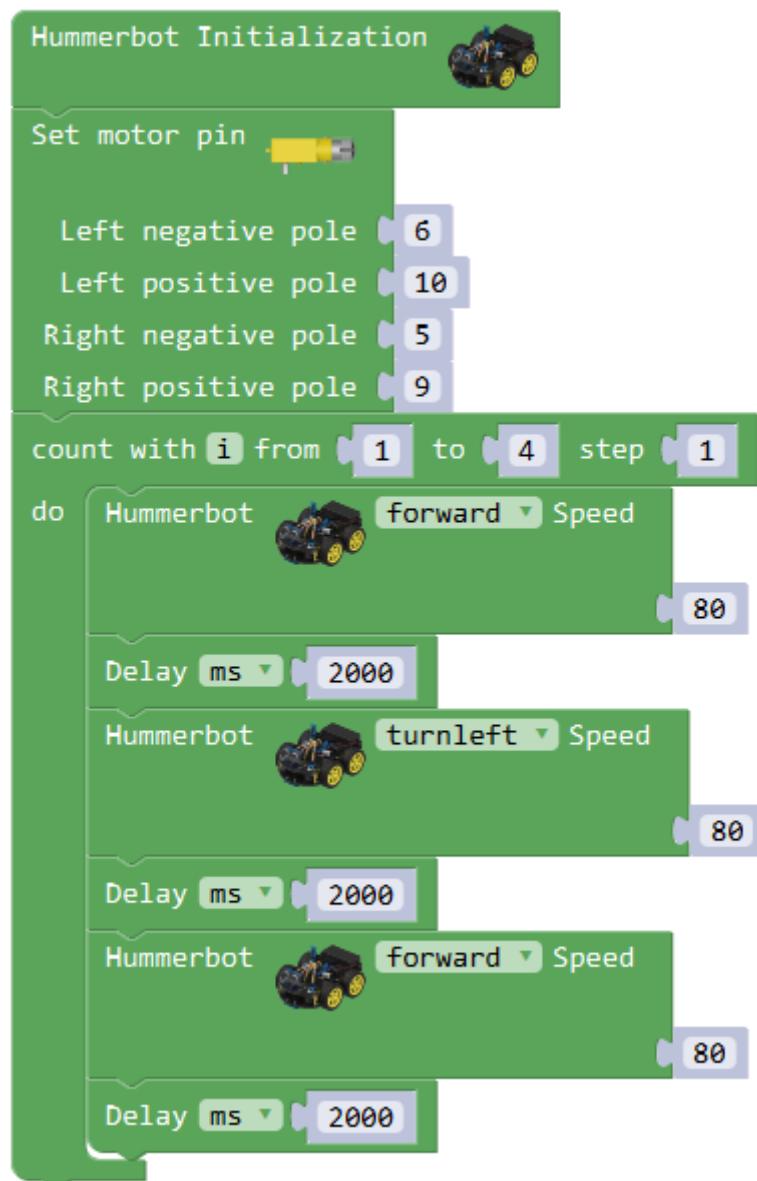


Figure 3-1-4-1



3.2 Ultrasound module

3.2.1 Principle of Ultrasound

Ultrasound sensor (Figure 3-2-1-1) is a device that detects distance by transmitting ultrasound. Ultrasound is an inaudible sound wave, which has the characteristics of returning when it touches an object. Ultrasound sensor has two "eyes", one of which emits ultrasound, and the other "eyes" receives the ultrasound emitted from obstacles. When one eye emits ultrasound, it begins to time, and when the other eye receives the returned ultrasound, it stops the time. Mathematically, we have learned that $\text{distance} = \text{speed} * \text{time}$, then the distance is measured by ultrasound.= The speed of the ultrasonic wave (2) so that the distance can be calculated.

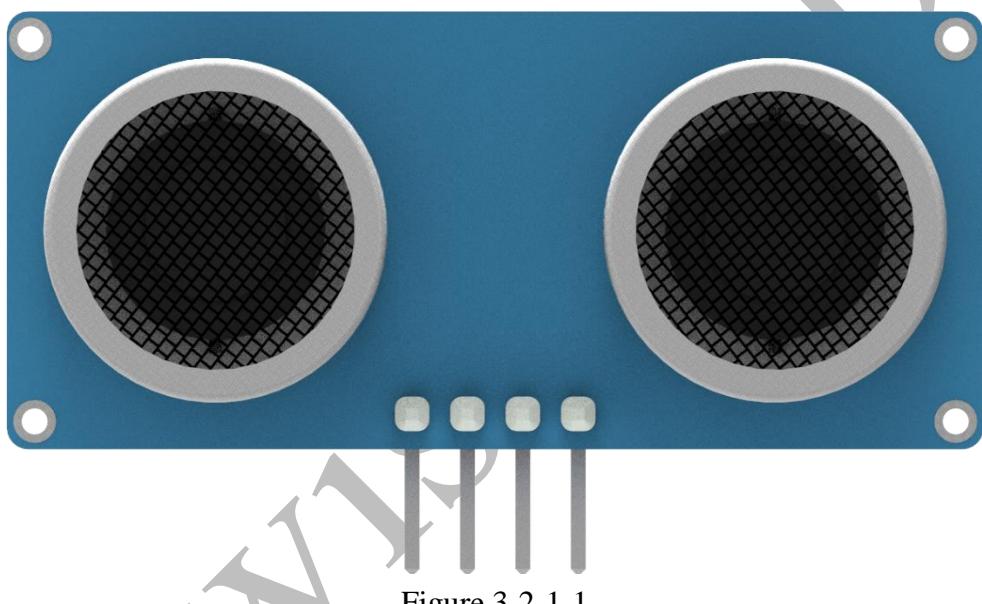


Figure 3-2-1-1

3.2.2 Usage of Ultrasound Module

We find the control building blocks of the ultrasonic module in the control module of the robot. As shown in Figure 3-2-2-1, the building blocks can detect the distance between the robot and the obstacle in front. Note: In order to use the ultrasonic module, we should first choose the mode to control the building blocks and set the mode to the ultrasonic obstacle avoidance mode. When the obstacles completely block the ultrasonic module or face the ultrasonic module to the distant place, the ultrasonic wave emitted by the ultrasonic module can not be received, so the distance between the robot and the obstacle can not be measured. The ultrasonic module installed on the robot can not be detected. The detection range is 5CM ~ 400CM.

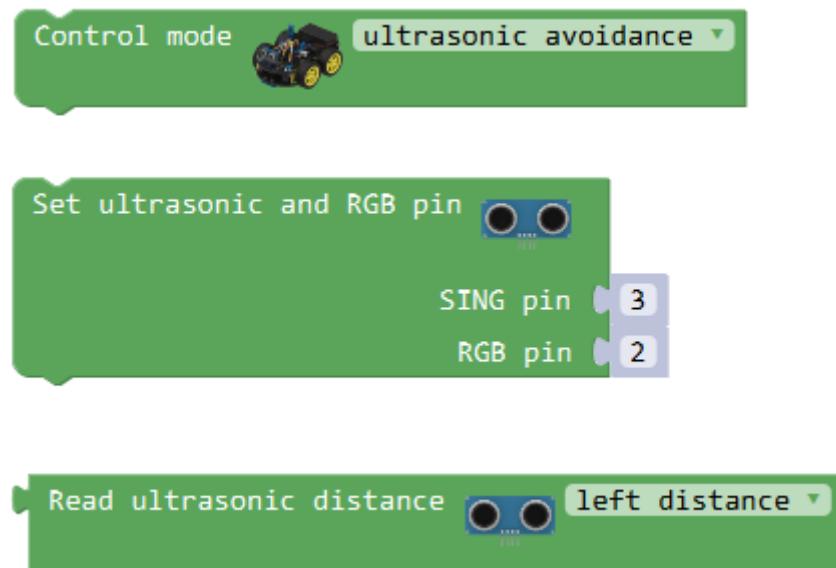


Figure 3-2-2-1

3.2.3 Testing of the Ultrasound Obstacle Avoidance Module

We can first write a program to test the ultrasonic obstacle avoidance module, using serial port printing. When we hand close to the ultrasonic obstacle avoidance module and far away from the ultrasonic obstacle avoidance module, we can observe the ultrasonic distance measured by serial port printing, and we can more intuitively see the process of measuring the distance of the ultrasonic obstacle avoidance module. Now we first write a test program.

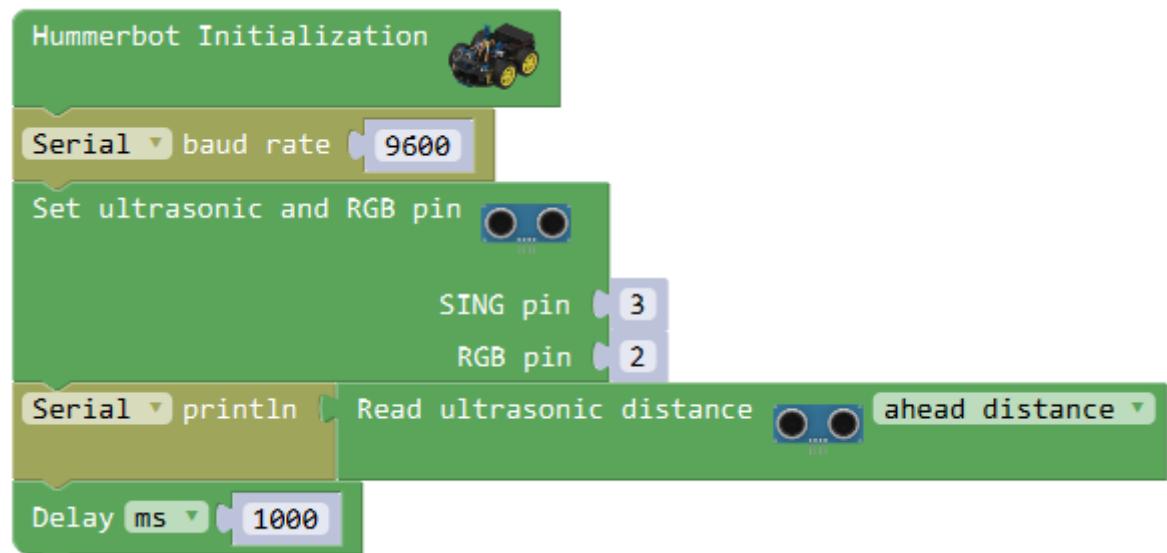


Figure 3-2-2

We write the program as shown in Figure 3-2-2, then upload the program successfully, open the serial port, and then close or far from the ultrasonic module, the serial port will print the corresponding distance.



3.2.4 Writing Robot Program for Tango Dance

Tango dance is a kind of double dance. The dance step is to let the robot dance with our palms when one person is close. If the hand is close to it and the hand moves away from it, the distance between the hand and the robot should be measured by the ultrasonic module to determine whether the robot moves forward or backward. The larger the measuring distance of ultrasound means that the hand is far away from the robot, and after a certain distance, the robot moves forward; the smaller the measuring distance of ultrasound means that the hand is close to the robot, and after a certain distance, the robot retreats; let's set the median value to 15CM first, then let's start programming.

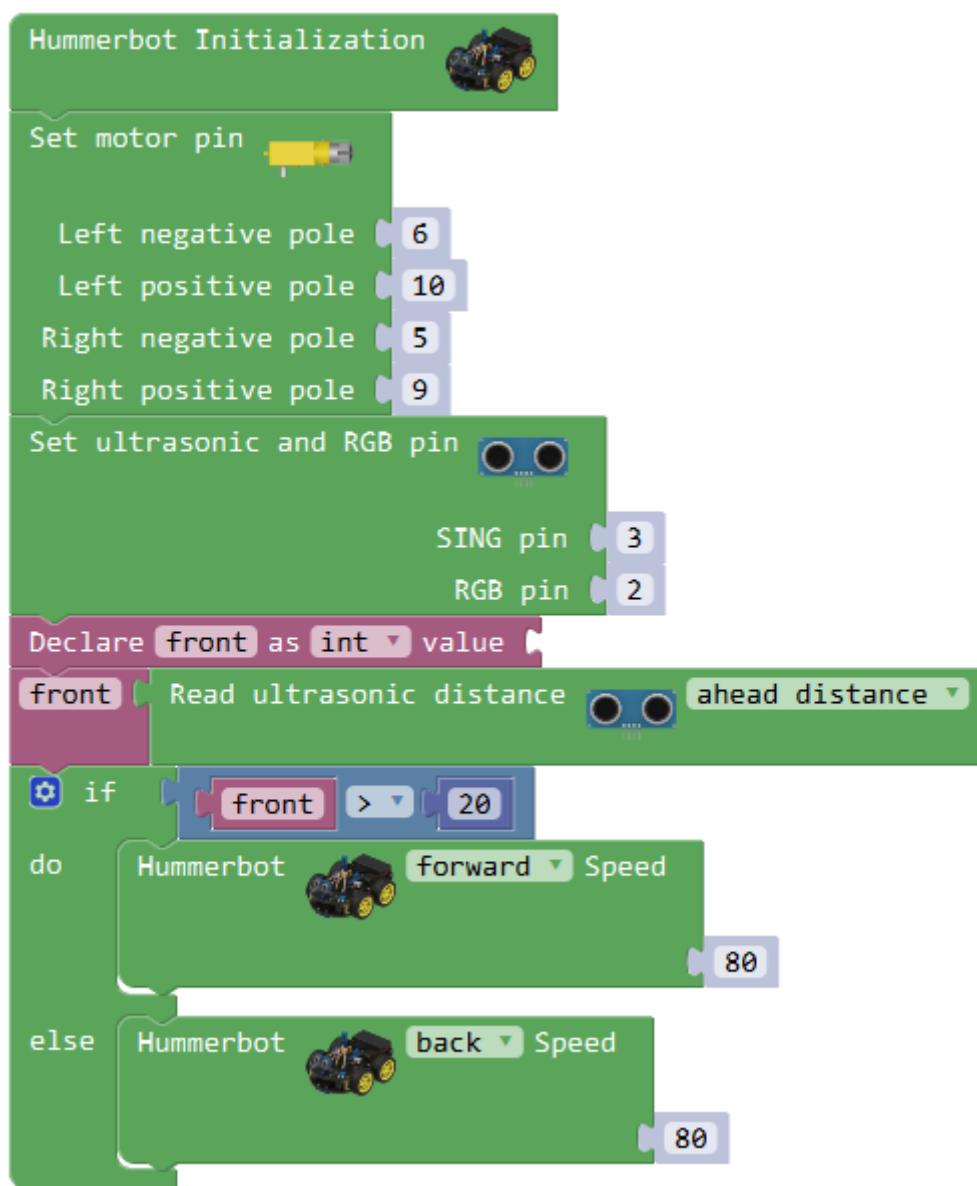


Figure 3-2-4-1



HUMMER-BOT ROBOT

Run the program shown in Figure 3-2-4-1 above. When the hand is far away, the robot moves forward; when the hand is near, the robot retreats. But when the hand is still, the robot will move back and forth repeatedly, because we did not set a stop motion interval for the robot. How to stop the robot in an interval?

The Value Measured by the Ultrasound Module	Motion State of Robot
Greater than 20	Forward
Between 12~20	Stop
Less than 12	Back off

Table 3-2-4-2

As shown in Table 3-2-4-2 above, a new judgment statement is needed for this situation between 12 and 20. As shown in Figure 3-2-4-3, when the distance measured by ultrasound is greater than 12CM and less than 20CM, the robot stops.

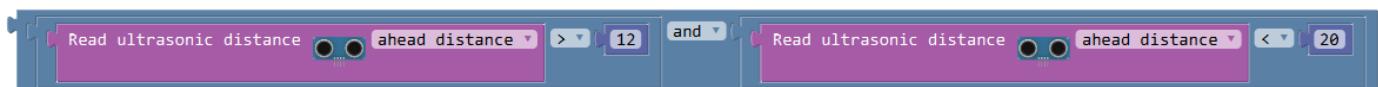
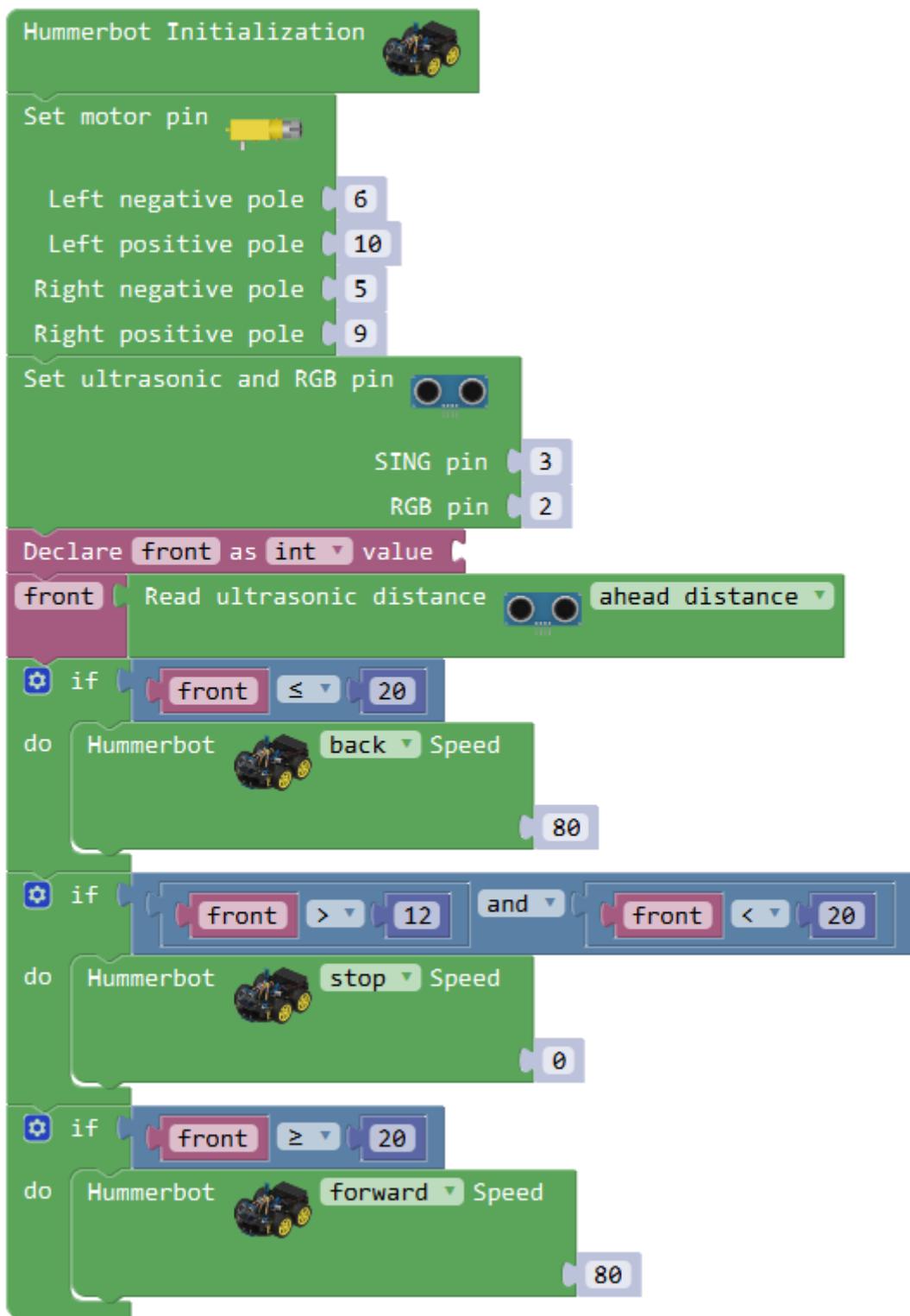


Figure 3-2-4-3



3.2.5 Optimizing Dancing Robot Program



The image shows a Scratch script titled "Hummerbot Initialization". The script begins with a "Set motor pin" block followed by four "Set ultrasonic and RGB pin" blocks for Left negative pole (pin 6), Left positive pole (pin 10), Right negative pole (pin 5), and Right positive pole (pin 9). It then declares a variable "front" as an integer with a value of 0. The script uses a "Read ultrasonic distance" block to get the "ahead distance". It contains three nested loops. The first loop checks if "front" is less than or equal to 20; if true, it makes the Hummerbot go "back" at speed 80. The second loop checks if "front" is greater than 12 and less than 20; if true, it makes the Hummerbot go "stop" at speed 0. The third loop checks if "front" is greater than or equal to 20; if true, it makes the Hummerbot go "forward" at speed 80.

```
Hummerbot Initialization
Set motor pin [pin 1]
Set ultrasonic and RGB pin [pin 6]
Set ultrasonic and RGB pin [pin 10]
Set ultrasonic and RGB pin [pin 5]
Set ultrasonic and RGB pin [pin 9]
Set ultrasonic and RGB pin [pin 3]
Set ultrasonic and RGB pin [pin 2]
Declare [front] as [int] [0]
front Read ultrasonic distance [ahead distance]
if [front ≤ 20] then
  do [Hummerbot [back v] [80]]
end
if [front > 12 and front < 20] then
  do [Hummerbot [stop v] [0]]
end
if [front ≥ 20] then
  do [Hummerbot [forward v] [80]]
end
```

Figure 3-2-5-1



3.3 Ultrasonic obstacle avoidance robot

We've learned the roles of robots in moving forward, backward, left, right and ultrasonic modules. Now let's explore how robots use ultrasound to avoid obstacles in motion. The so-called obstacle avoidance of robots is to let robots move around obstacles (left or right or backward) if there are obstacles in front of them. If the object continues to move, the robot needs to turn left and right and judge the distance between left and right directions, then the ultrasonic module needs to rotate left and right to "explore the way". At this time, the steering gear is used to realize the left and right turn of the ultrasonic module. How does the steering gear work? Next we will introduce the use of the steering gear.

3.3.1 Operating Principle of Steering Engine

The steering gear is mainly composed of the following parts: steering wheel, deceleration gear set, position feedback potentiometer, DC motor, control circuit and so on, as shown in Figure 3-3-1-1. The Bumblebee and Optimus Prime joints we see in the movies need to be controlled by the steering gear. Especially when the robot is walking, the mechanical sound of clicking and clicking is generated by the steering gear rotating on the robot body. Figure 3-3-1-2 is the most commonly used SG90 steering gear physical diagram at this stage.

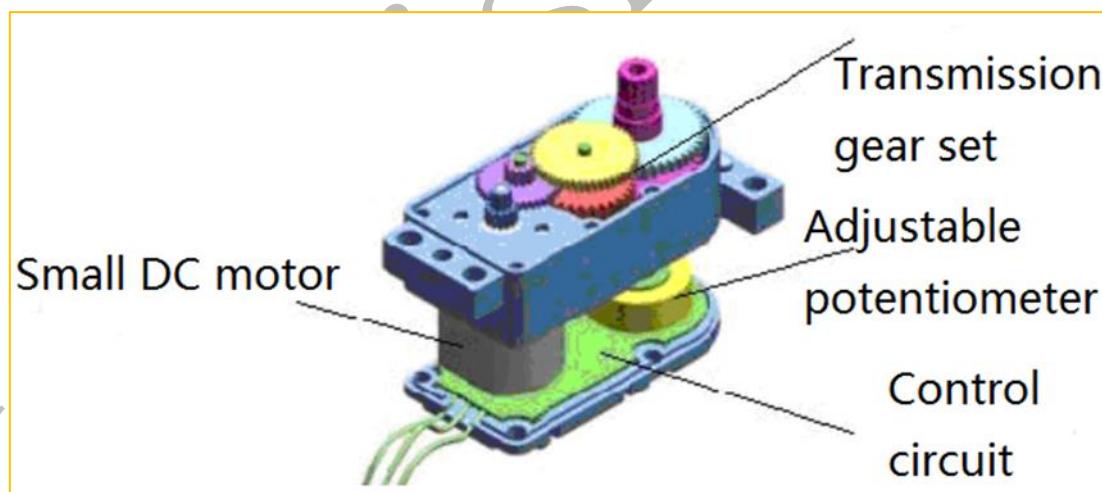


Figure 3-3-1-1

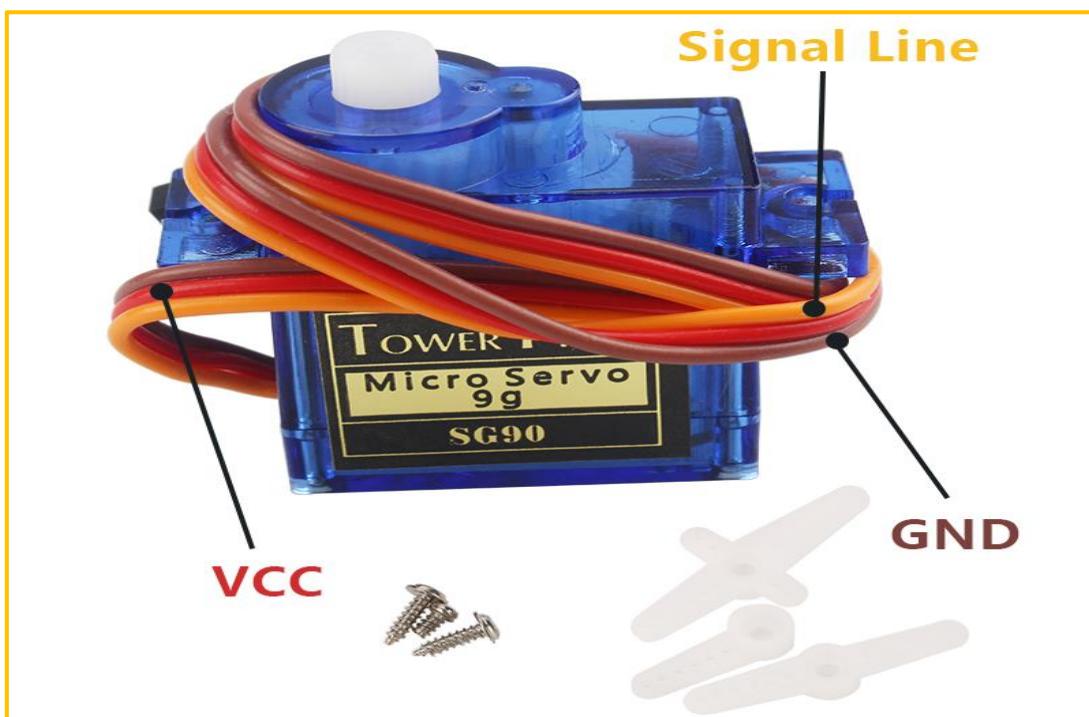


Figure 3-3-1-2 SG90

When the control circuit board receives the control signal of the self-confidence signal line, it controls the motor to rotate, and the motor drives a series of gears, then reduces the speed and drives them to the output steering wheel. Its work flow is: control signal control circuit board motor rotation gear set deceleration steering wheel rotation position feedback potentiometer control circuit board feedback.

3.3.2 Steering gear test

The steering gear on the Hummer-Bot robot is installed in the front, and the ultrasonic module is fixed on the steering gear. When the ultrasonic module is to measure the distance between the robot and the front obstacle, the steering gear is 90 degrees. When the ultrasonic module measures the distance between the robot and the left obstacle, the steering gear rotates 90 degrees to the left. When the ultrasonic module measures the distance between the robot and the right obstacle, the steering gear rotates 90 degrees to the right. The steering gear rotates 90 degrees to the right. Next, we write a program to test the steering gear by using the steering gear to make the ultrasonic wave move forward, then left and then right.

HUMMER-BOT ROBOT

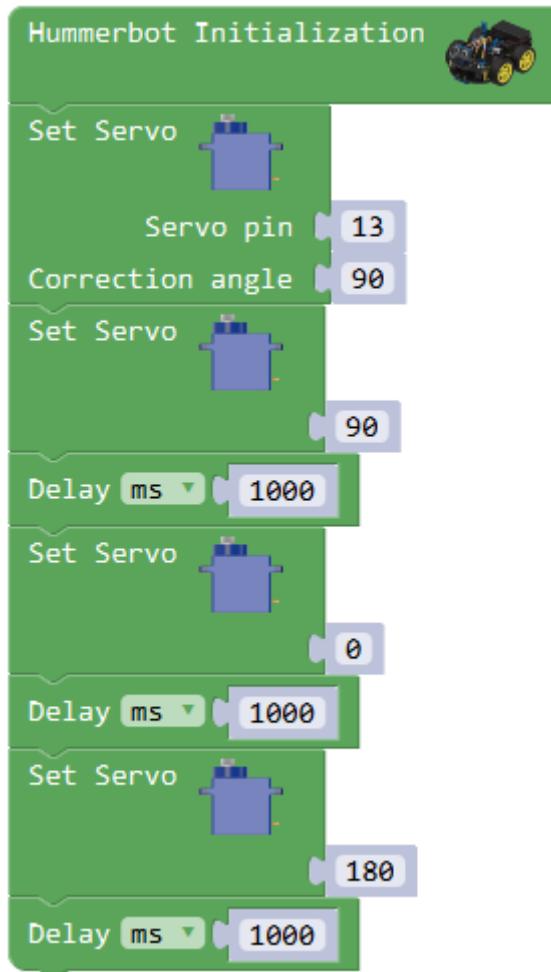


Figure 3-3-2-1

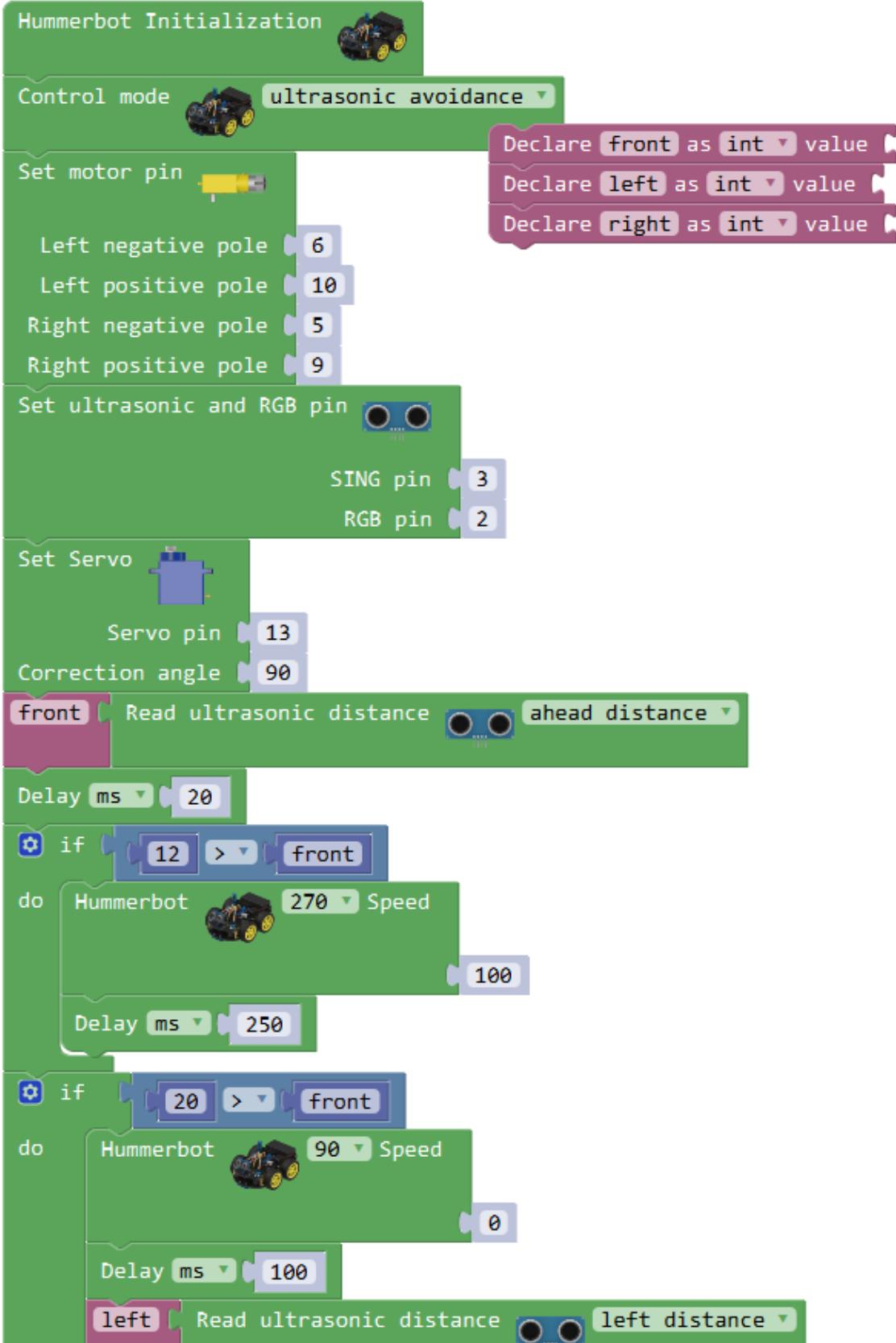
Above is the testing procedure of the steering gear. After uploading the program to the robot, turning on the power supply, we can see that the steering gear makes the ultrasonic module turn forward, then left, and then right.

3.3.3 Programming of Obstacle Avoidance Robot

The main work flow of the ultrasonic obstacle avoidance robot is as follows: after power-on, the steering gear makes the ultrasonic module automatically turn to 90 degrees (alignment in front of the robot). The ultrasonic module measures the distance of the obstacle in front. If the value is larger than the set safe distance (12CM), the robot continues to move forward, otherwise stops. At this time, the steering gear makes the ultrasound rotate 90 degrees to the right first, and the ultrasonic mode. The distance between the block and the right obstacle is measured. Then the steering gear rotates the ultrasonic wave 180 degrees to the left. The ultrasonic module measures the distance between the ultrasonic module and the left obstacle. Then the steering gear lets the ultrasonic module return to 90 degrees. The robot compares the two measured distances. If the left is larger than the right, the robot turns left, then drives, then turns right, and vice versa.

HUMMER-BOT ROBOT

When the distance is less than the safe distance, the robot will turn around and walk. Following this train of thought, let's start to write the program of obstacle avoidance robot.



The Scratch script for the Hummer-bot robot consists of the following blocks:

- Hummerbot Initialization**:
 - Control mode: ultrasonic avoidance
 - Set motor pin:
 - Left negative pole: 6
 - Left positive pole: 10
 - Right negative pole: 5
 - Right positive pole: 9
 - Set ultrasonic and RGB pin:
 - SING pin: 3
 - RGB pin: 2
 - Set Servo:
 - Servo pin: 13
 - Correction angle: 90
- front** [Read ultrasonic distance [ahead distance v]]
- Delay [ms v] [20]**
- if [12 > front]**
 - do** [Hummerbot [270 v Speed] [100]]
 - Delay [ms v] [250]**
- if [20 > front]**
 - do** [Hummerbot [90 v Speed] [0]]
 - Delay [ms v] [100]**
- left** [Read ultrasonic distance [left distance v]]

HUMMER-BOT ROBOT



K

3.4 Infrared Barrier Avoidance Robot

3.4.1 Principle of Infrared Obstacle Avoidance Module

The principle of the infrared obstacle avoidance module is similar to that of the ultrasonic module. The difference is that the ultrasonic module measures the distance by transmitting ultrasonic wave. The infrared obstacle avoidance module measures the distance by transmitting infrared ray. The infrared obstacle



avoidance module has a pair of infrared emission and receiving tubes, which emits infrared ray. When the emitted infrared ray meets the obstacle (reflecting surface), the infrared ray is reflected back and received. Receiving and receiving, the green indicator lights up. After the infrared signal is processed by the control motherboard of the robot, the detection distance is calculated. The infrared obstacle avoidance module is shown in Figure 3-4-1-1.

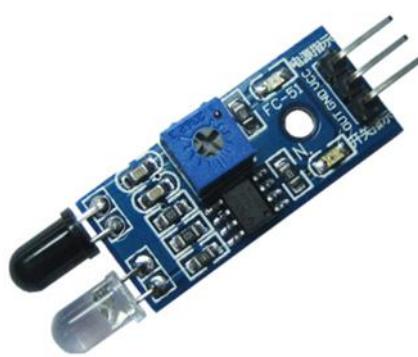


Figure 3-4-1-1

3.4.2 Usage of Infrared Obstacle Avoidance Module

In the control module of the robot, we find the control building blocks of the infrared obstacle avoidance module. As shown in Figure 3-4-2-1, the building blocks can detect the distance between the robot and the left and right obstacles. Note: In order to use the infrared obstacle avoidance module, we first need to choose the mode to control the building blocks and set the mode to the infrared obstacle avoidance mode. When the obstacles completely block the ultrasonic module or face the ultrasonic module to the distance, the infrared radiation emitted by the infrared obstacle avoidance module can not be received because it is too far or too close to detect the distance between the robot and the obstacle, so it is used. When infrared obstacle avoidance module ranging, it is appropriate to keep the distance between the obstacle and the infrared obstacle avoidance module between 5 and 15.

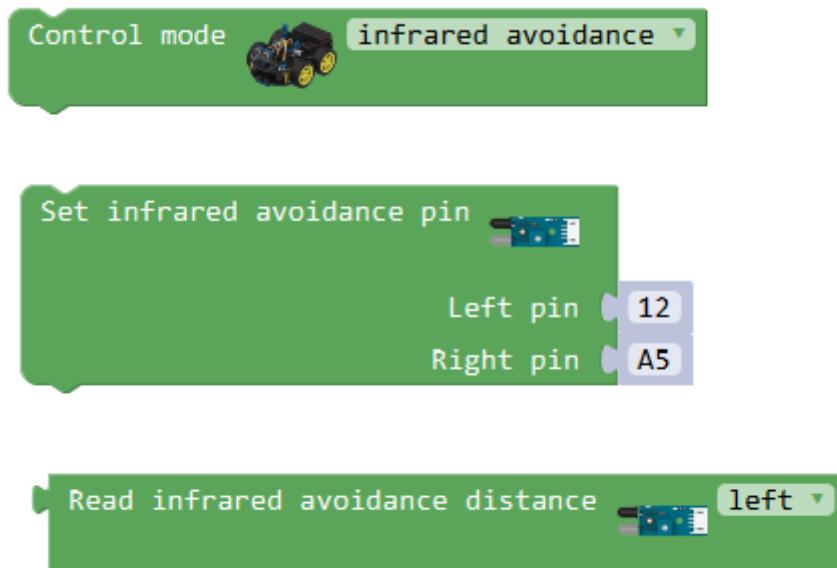


Figure 3-4-2-1

3.4.2 Test of Infrared Obstacle Avoidance Module

We can first write a program to test the infrared obstacle avoidance module, using serial port printing, when we hand close to the infrared obstacle avoidance module and far away from the infrared obstacle avoidance module, we can observe the contents of serial port printing, we can more intuitively see the infrared obstacle avoidance module measuring distance process, below we first write a test program.

HUMMER-BOT ROBOT

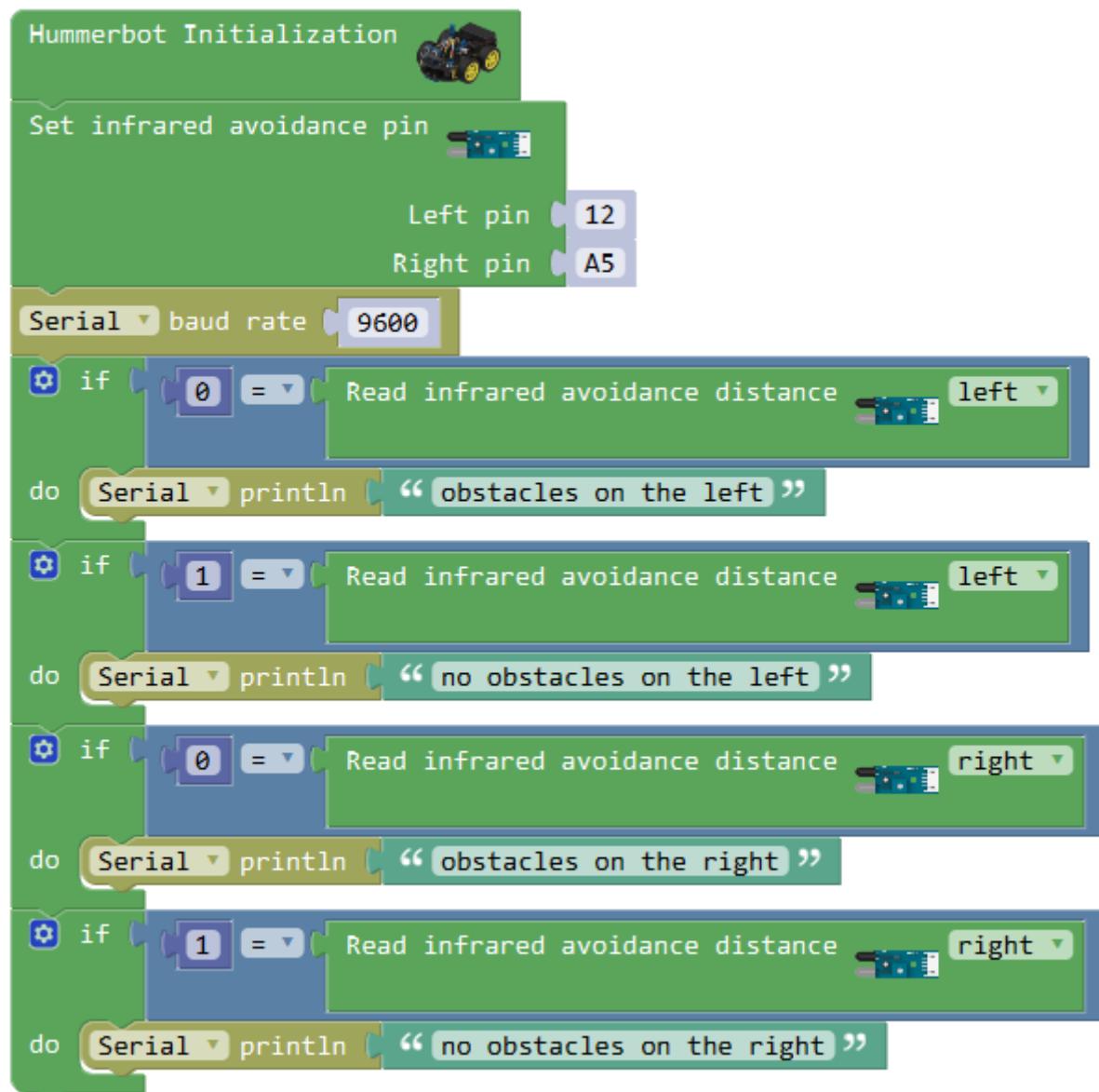


Figure 3-4-2-1

We write the program as shown in Figure 3-4-2-1, then upload the program successfully, open the serial port, and then close or far from the infrared obstacle avoidance module, the serial port will print the corresponding text.

3.4.3 Robot Writing Runway

If we want to let the robot run smoothly on the track to avoid encountering the track, we need to keep a certain distance between the robot and both sides of the track at all times. At this time, we can use the infrared obstacle avoidance module to achieve this effect. The basic idea is to set the safe distance of the infrared obstacle avoidance module first, the robot moves forward, and the infrared obstacle avoidance module measures it with the right track. The distance between the edge and the left side of the track is measured by the infrared obstacle avoidance module. Then the robot compares the distance measured twice.

HUMMER-BOT ROBOT

If the left is larger than the right, the robot turns left, then drives, and vice versa. If the distance measured on both sides is less than the safe distance, the robot will stop walking. Follow this train of thought to learn how to write a racetrack robot.



Figure 3-4-3-1

Run the program shown in Figure 3-4-3-1 above, and the robot moves forward; when the robot's left side is near the edge of the runway, the robot turns right. When the right side of the robot is close to the edge of the runway, the robot turns left. When the left and right sides of the robot are close to obstacles, the robot stops.

3.5 Ultrasound+Infrared Barrier Avoidance Robot

We have already known the use and programming of the ultrasonic module and the infrared obstacle avoidance module separately. Next, we will combine the ultrasonic module and the infrared obstacle avoidance module to use together. The programming idea is to set the mode as the ultrasonic infrared obstacle avoidance mode first, then set the ultrasonic module, the safe distance of the infrared obstacle avoidance module, and then detect the obstacle in front of the robot by ultrasonic. Obstacles, infrared



HUMMER-BOT ROBOT

obstacle avoidance module detects whether there are obstacles on the left and right sides of the robot. Now let's write the program of the ultrasonic infrared obstacle avoidance robot.

KeyWish-robot

HUMMER-BOT ROBOT

```

Hummerbot Initialization
Control mode ultrasonic infrared avoidance
Set motor pin
Left negative pole 6
Left positive pole 10
Right negative pole 5
Right positive pole 9
Set infrared avoidance pin
Left pin 12
Right pin A5
Set ultrasonic and RGB pin
SING pin 3
RGB pin 2
Set Servo
Servo pin 13
Correction angle 90
ufront Read ultrasonic distance ahead distance
ileft Read infrared avoidance distance left
iright Read infrared avoidance distance right
Delay ms 20
if iright ≠ 0 and ileft = 0
do Turn a corner
Angle 20
Speed 70
else if iright = 0 and ileft ≠ 0
do Turn a corner
Angle 160
Speed 70
else Hummerbot 90 Speed
45
if ufront < 20
do Hummerbot 90 Speed
0
Declare upfront as int value
Declare ileft as int value
Declare iright as int value
Declare uleft as int value
Declare tright as int value

```



The Scratch script for the "Hummerbot" robot is designed to navigate through an environment using ultrasonic sensors and a corner detection algorithm. The script starts with a "when green flag clicked" hat block. It begins by setting the robot's speed to 90. A "do" loop then executes the following sequence:

- If the ultrasonic sensor ("ufront") value is less than 20, the robot moves forward at 90 speed for 45 steps.
- If the ultrasonic sensor ("ufront") value is between 20 and 12, or if both the left and right ultrasonic sensors ("ileft" and "iright") are equal to 0, the robot moves forward at 270 speed for 60 steps.
- After each forward movement, there is a 300ms delay.
- After the second forward movement, the robot returns to its original speed of 90.
- The script then reads the ultrasonic distances from the left and right sensors.
- If the right distance is greater than or equal to the left distance, the robot moves forward at 0 speed for 100 steps, followed by a 400ms delay.
- If the right distance is less than the left distance, the robot moves forward at 180 speed for 100 steps, followed by a 400ms delay.
- If the right distance is less than or equal to 12 and the left distance is also less than or equal to 12, the robot performs a "Turn a corner" maneuver. This involves turning 0 degrees at a speed of 100 for 800 ms, followed by a 90 speed move for 60 steps.

Figure 3-5-1



Figure 3-5-2

Fig. 3-5-1 and Fig. 3-5-2 are the procedures of the ultrasonic infrared obstacle avoidance robot. In the above procedures, the delay time after turning should be adjusted according to the speed of the robot, which should be paid attention to when programming.

Chapter IV Tracking Robot

4.1 Principle of Tracking Module

Tracking module works by using the characteristics of different reflective properties of infrared rays on the surface of objects with different colors. During the movement of the robot, infrared light is continuously emitted to the ground. When the emitted infrared rays are not reflected back or reflected back, but the intensity is not high enough, the robot thinks that the area is the route it is going to take; when the infrared light is reflected back to the machine, the robot thinks that the area is the route it is going to take. Man recognizes that the robot thinks the area is not the route it takes. The robot is based on the reflected infrared light to determine the position of the black line and its walking path. The Hummer-Bot robot is equipped with three tracking modules, one on the left, one on the right, and the principle of tracking is shown in Figure 4-1-1.

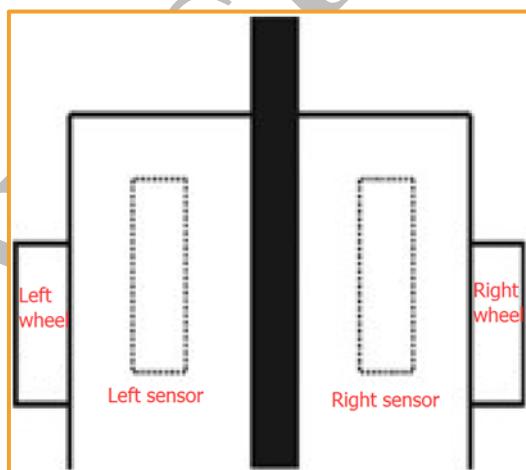


Figure 4-1-1

4.2 Use of Trace Module

In the control module of the robot, the control building blocks of the infrared tracking module are found. As shown in Figure 4-2-1, using the building blocks, the robot can automatically run along its own track. Note: In order to use the infrared tracking module, we first need to select the mode to control the building blocks, and set the mode to the infrared tracking mode. Besides the mode setting, we need to define a variable to save the values read by each infrared tracking, so that when the robot runs off, it can correct the direction according to the values read last time and find the route again. Read the trace value building blocks,

HUMMER-BOT ROBOT

the drop-down menu has all black, all white, left 1, left 2, center, right 1, right 2 options, their interpretation is as follows:

- All black: black lines were detected in all three tracking modules.
- Whiteness: None of the three tracking modules detected black lines.
- Left 1: The black line was detected by the tracking module on the left.
- Left 2: Black lines were detected in the left and middle tracing module.
- Center: The black line was detected by the tracing module in the middle.
- Right 1: The black line was detected by the tracking module on the right.
- Right 2: The black line was detected by the tracing module on the right and in the middle.

Figure 4-2-1

4.3 Testing of Infrared Tracking Module

We can first write a program to test the infrared tracing module, using serial port printing, using black tape near the infrared tracing module and far away from the infrared tracing module, we can observe the contents of serial port printing, we can more intuitively see the tracing effect of each module when the infrared tracing module tracks. Now we first write a test program.

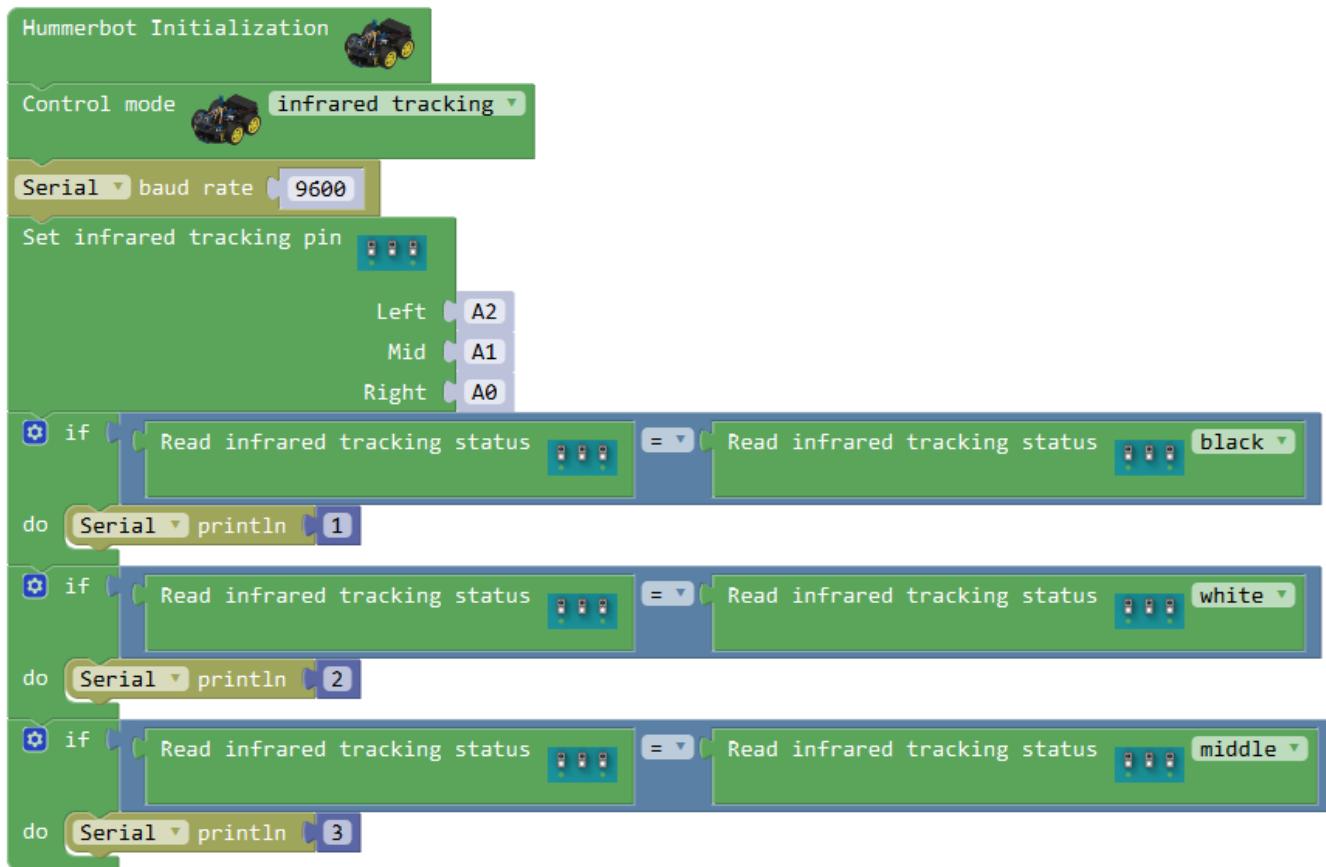


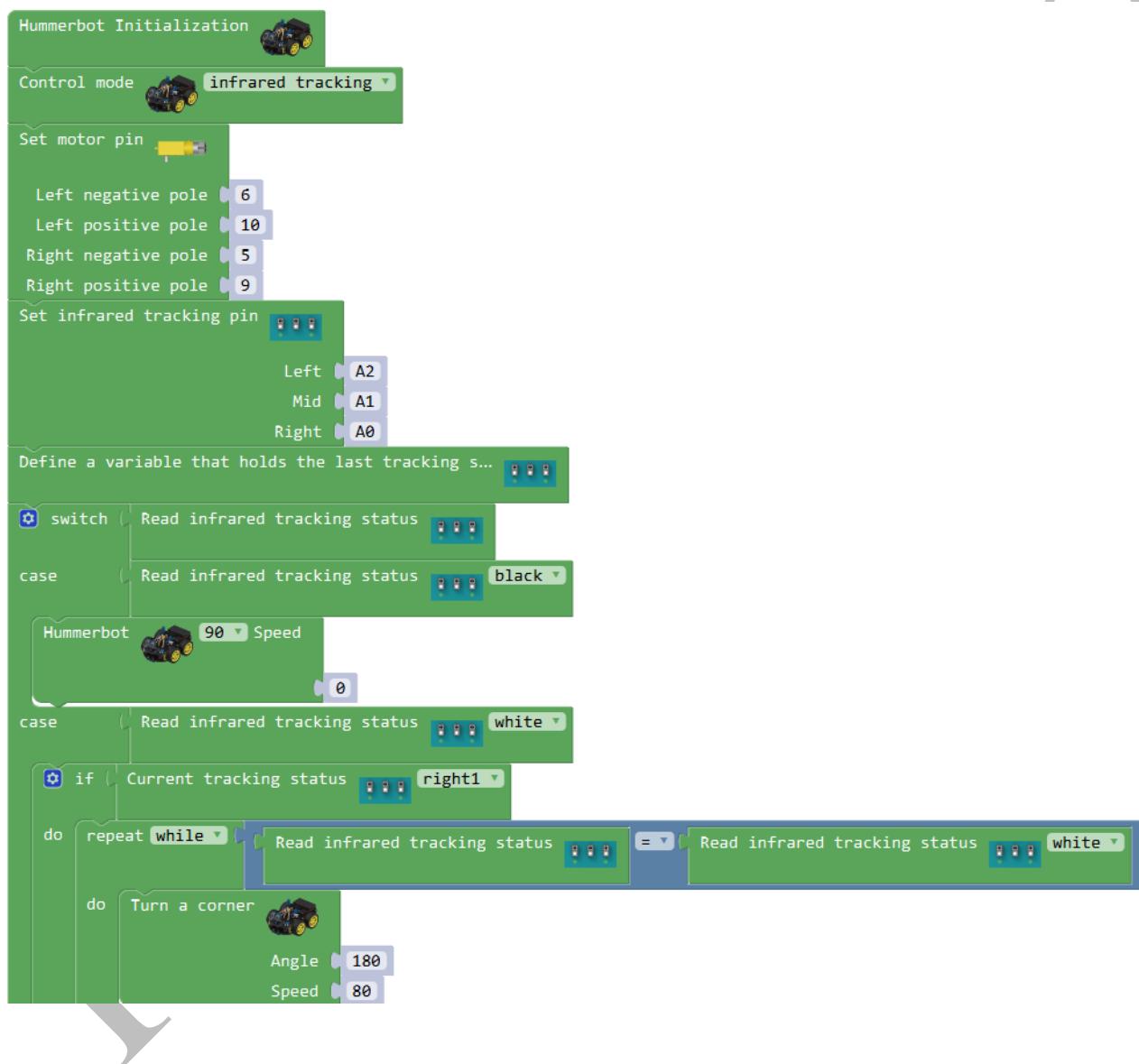
Figure 4-3-1

HUMMER-BOT ROBOT

As shown in Figure 4-3-1 above, when we open the serial port, we close the black strip to any one of the tracking modules, and the serial port will print the corresponding number.

4.4 Writing Tracking Robot Program

The programming idea of the tracking robot is to set the infrared tracking mode first, and then set the connecting pins of the infrared tracking module according to its actual wiring mode. Then the robot executes the corresponding left-to-left, forward and right-to-right movement according to the detection results of the left-to-center and right-to-right tracking modules. Let's write the program of the tracking robot together.

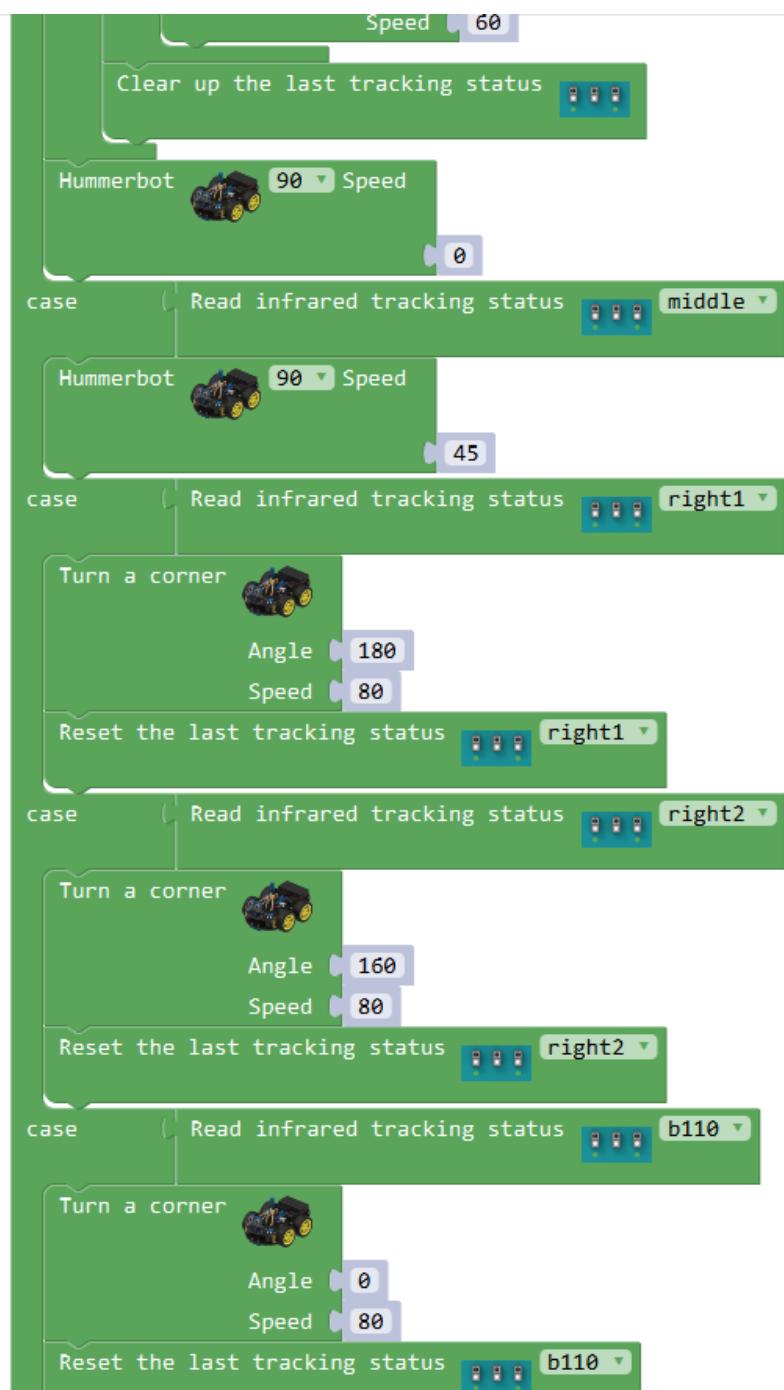


HUMMER-BOT ROBOT

```
Clear up the last tracking status [infrared sensor]
if [Current tracking status = b110] do [repeat [while [Read infrared tracking status [infrared sensor] = white]] do [Turn a corner [Angle 0, Speed 80]]]
    Clear up the last tracking status [infrared sensor]
if [Current tracking status = right2] do [repeat [while [Read infrared tracking status [infrared sensor] = white]] do [Turn a corner [Angle 160, Speed 60]]]
    Clear up the last tracking status [infrared sensor]
if [Current tracking status = left2] do [repeat [while [Read infrared tracking status [infrared sensor] = white]] do [Turn a corner [Angle 20, Speed 60]]]
    Clear up the last tracking status [infrared sensor]
```

KeyW

HUMMER-BOT ROBOT



K

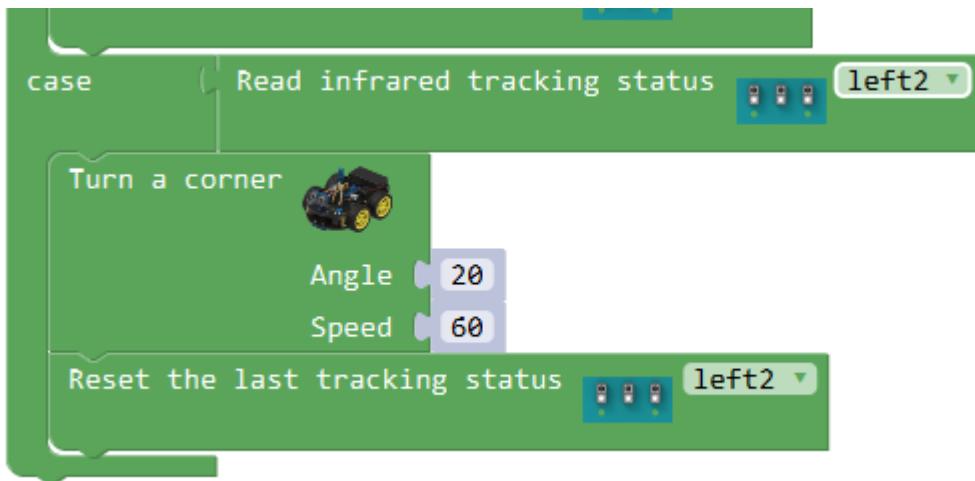


Figure 4-4-1

Chapter 5 Infrared Telecontrol Robot

5.1 Principle of Infrared Remote Control

The remote control system is generally composed of a remote controller (transmitter) and a receiver. When you press any key on the remote controller, the remote controller will issue an instruction. After receiving the instruction from the remote controller, the receiver will transmit the instruction to the robot's brain. The robot will think about what kind of action to do according to the remote control instruction, and then control its limbs (four wheels) to do the corresponding. Actions, remote controls and receivers are shown in Fig. 5-1-1.



Figure 5-1-1

5.2 The Use of Remote Controller

To control the robot through the remote controller, we first set the operation mode to infrared remote control mode, and set the infrared remote control receiving pin according to the actual connection port, then



define the effect of each key press of the remote controller, so that when we press the key of the remote controller, the robot will do the action defined in our program.

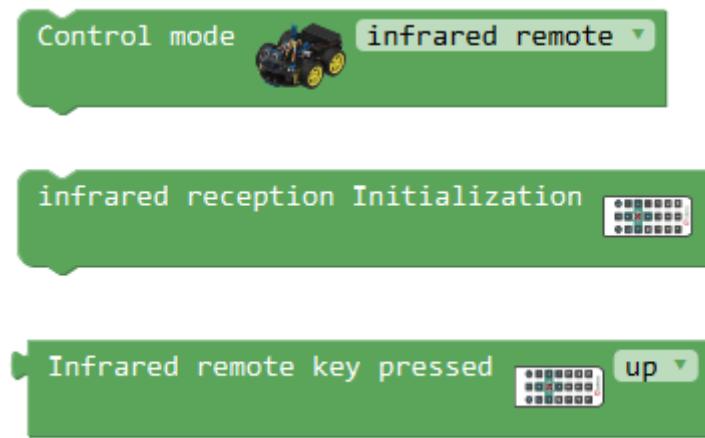


Figure 5-2-1

5.3 Infrared remote control test

We can first write a program to test infrared remote control, using serial port printing, when the remote control button is pressed, the serial port prints the corresponding key values, we observe the contents of serial port printing, we can see which key of the infrared remote control is pressed. Now we first write a test program.

HUMMER-BOT ROBOT

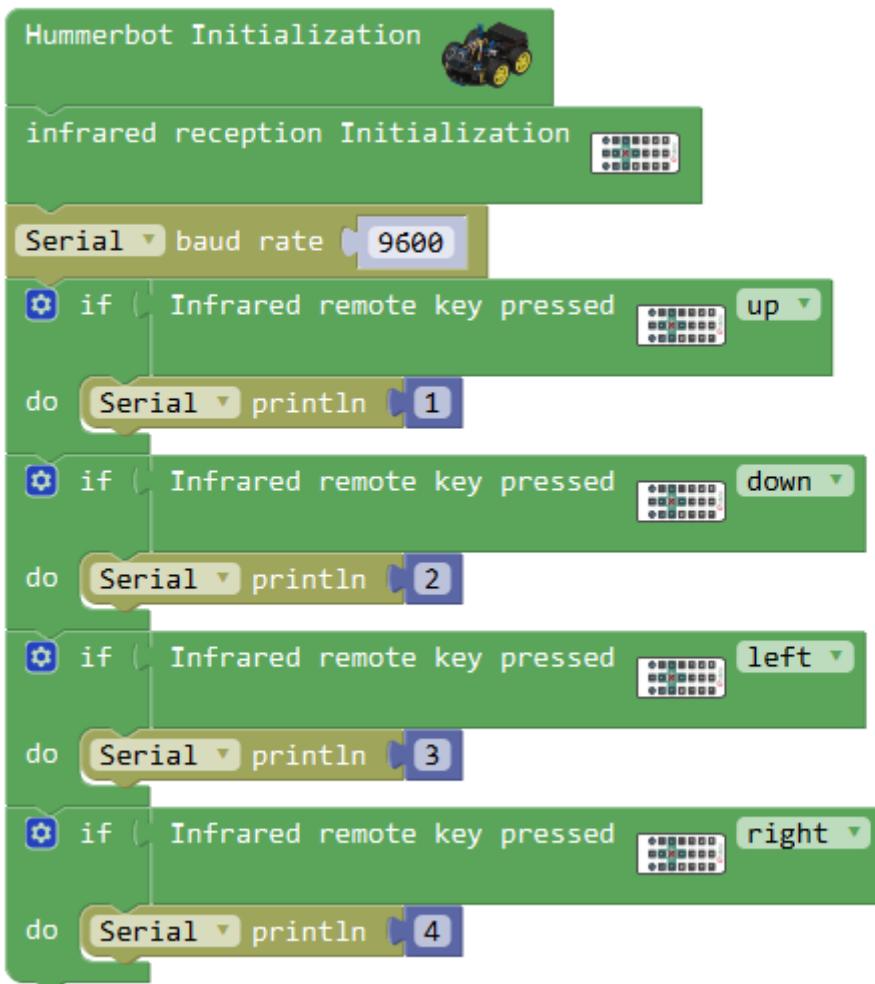


Figure 5-3-1

We write the program as shown in Fig. 5-3-1, then upload the program successfully, open the serial port, then hold the remote control to the robot, press up, down left, right, the serial port will print the corresponding key numbers.

5.4 Programming of Remote Control Robot

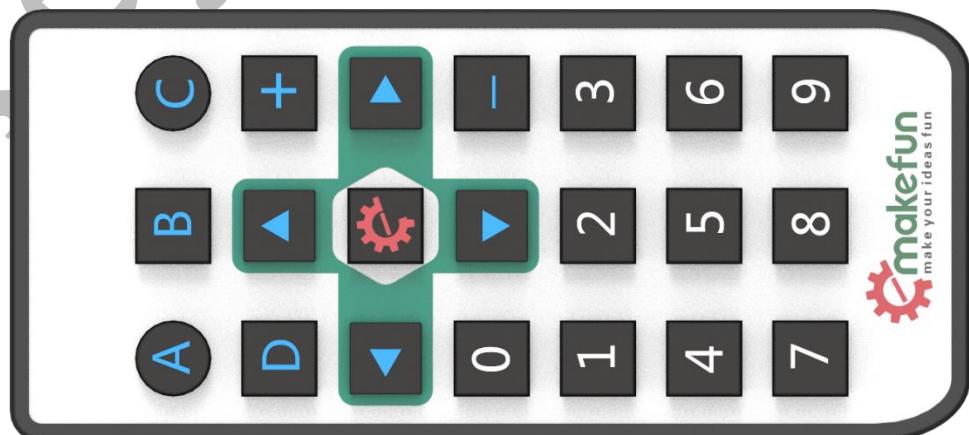


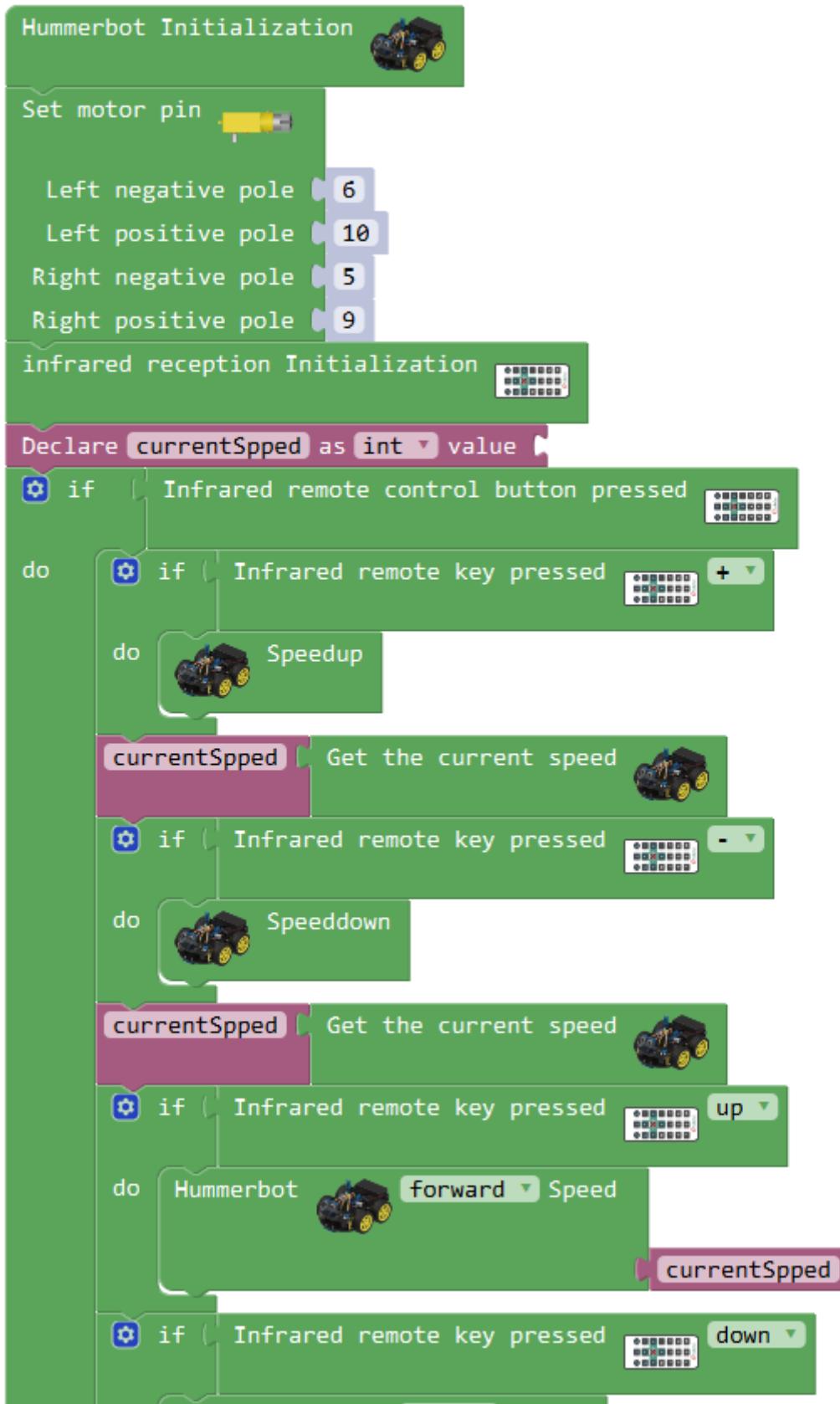
Figure 5-4-1



We write a remote control key function as defined in Fig. 5-4-1, and set the speed regulation 1 (+) for acceleration and the speed regulation 2 (-) for deceleration. We have to do a 0.12S delay after the acceleration and deceleration procedures, which is to make the robot accelerate and decelerate a little bit more smoothly. Without the delay, the speed will directly change to the maximum or minimum.

KeyWish-robot

HUMMER-BOT ROBOT



HUMMER-BOT ROBOT

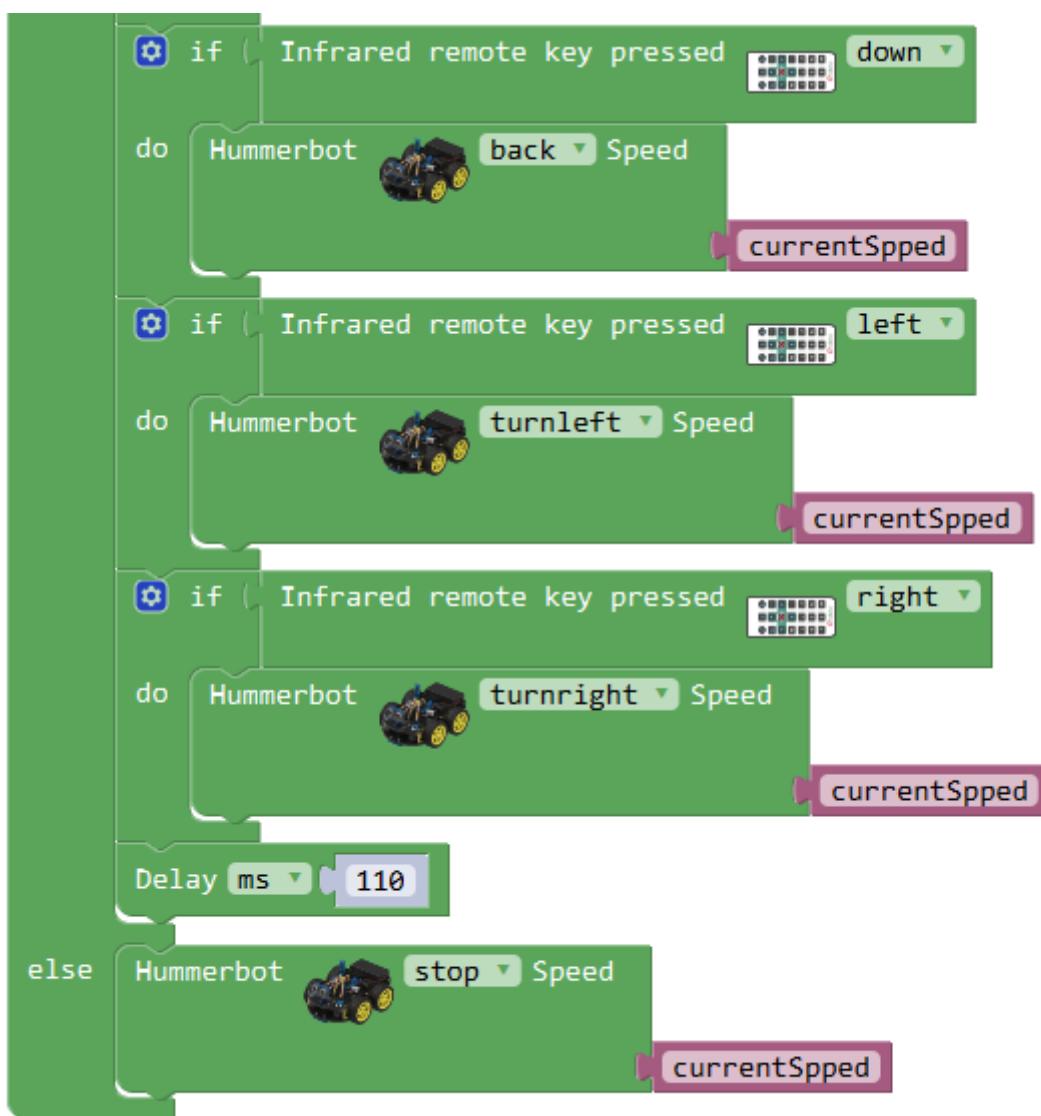


Figure 5-4-2

Note: When setting the steering speed, the speed value can not be set too small, otherwise the robot will not be strong enough when turning, and will be stuck in place. Therefore, the best setting speed value is above 40.

Chapter 6 Bluetooth Remote Control Robot

6.1 Bluetooth Control Principle

Bluetooth is a kind of long-distance communication mode. We send the instructions to the Bluetooth module on the robot from the APP end of the mobile phone through the Bluetooth of the mobile phone. Then the Bluetooth module on the robot sends the instructions to the robot brain (main control board). The robot controls its limbs (four wheels) to do the corresponding actions. At the same time, the robot will also install Bluetooth through itself. The module sends the action being done to the mobile app, so that the mobile app can be remotely controlled by Bluetooth.



6.2 Bluetooth Module and Hummer-Bot Robot APP KeywishBot Connection Method

The Hummer-Bot robot is equipped with a Bluetooth module (as shown in Figure 6-2-1). The blue light on the Bluetooth module flickers when the power is turned on. After turning on the APP, select Hummer-Bot , Bluetooth module and the Hummer-Bot robot APP KeywishBot on the mobile phone (as shown in Figure 6-2-3) and connect via Bluetooth. When the connection is successful, the blue light will be on for a long time, and then select "remote". The control interface enters the gravity induction control interface (Fig. 6-2-5). Handle control mode can also be switched as shown in Figure 6-2-6.



Figure 6-2-1

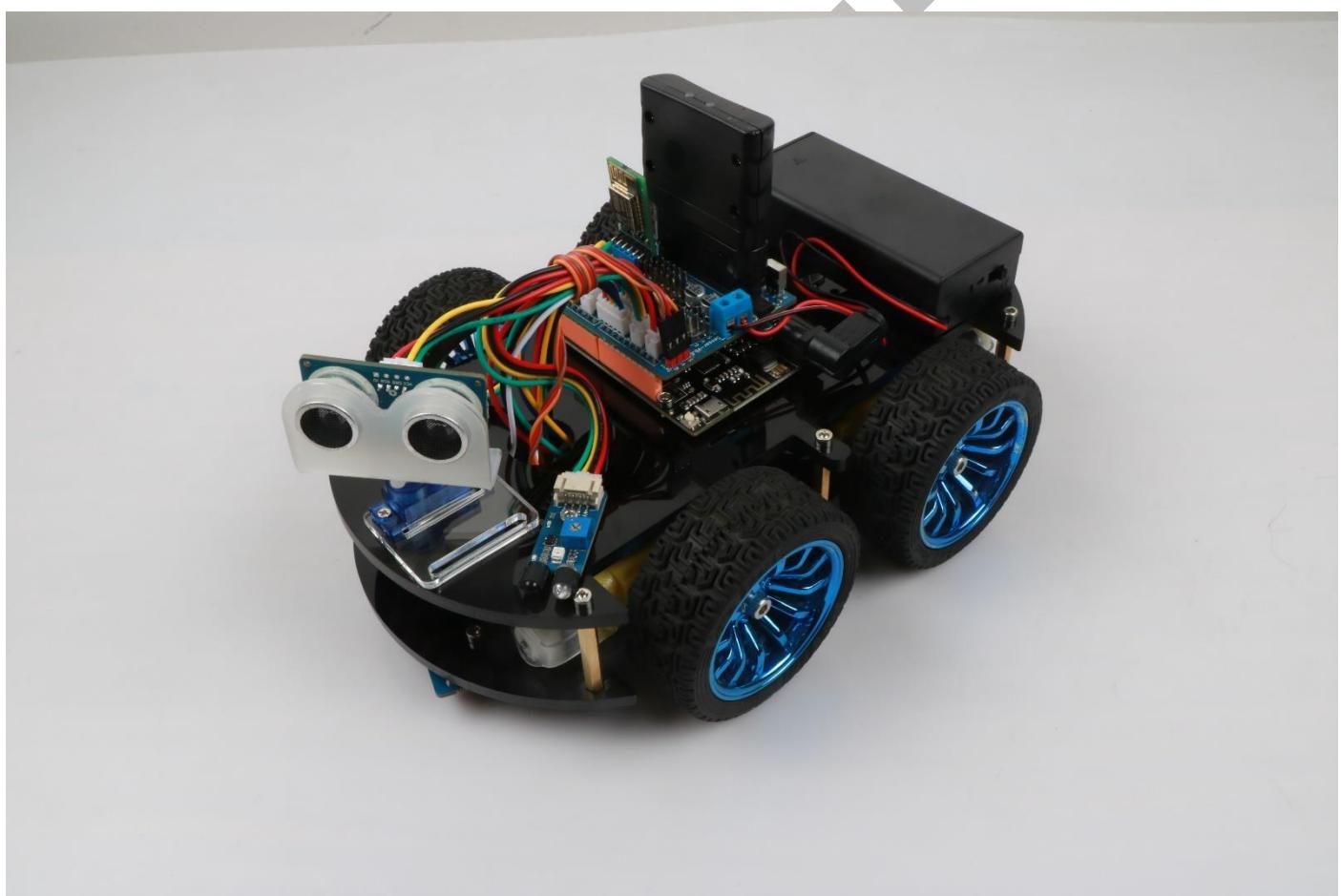


Figure 6-2-3

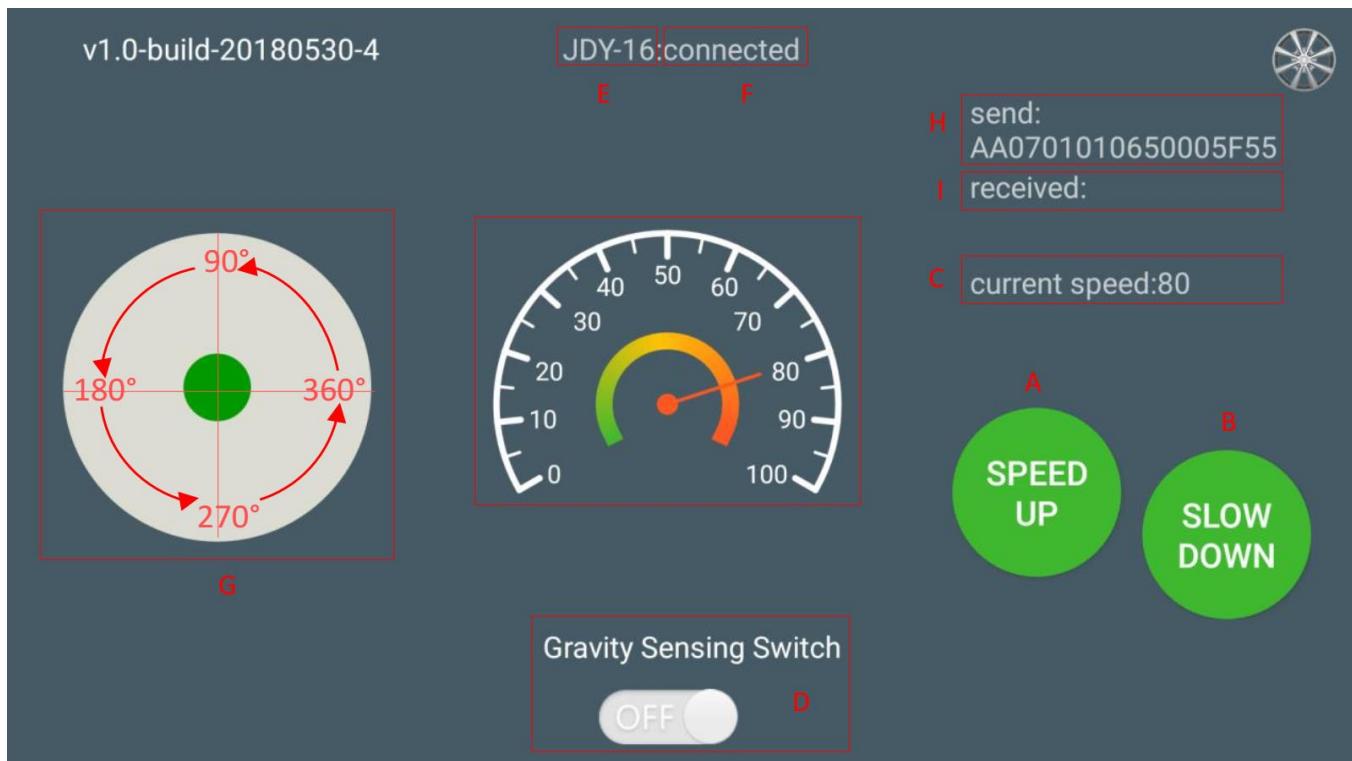


Figure 6-2-5



Figure 6-2-6

6.3 Bluetooth module testing

We can first write a program to test Bluetooth module, using serial port printing. When the Bluetooth module of the robot receives data and the serial port prints the data received by Bluetooth, we can see the

HUMMER-BOT ROBOT

effect of Bluetooth data transmission by observing the content sent by Bluetooth and the content printed by serial port. Here we write a test program.

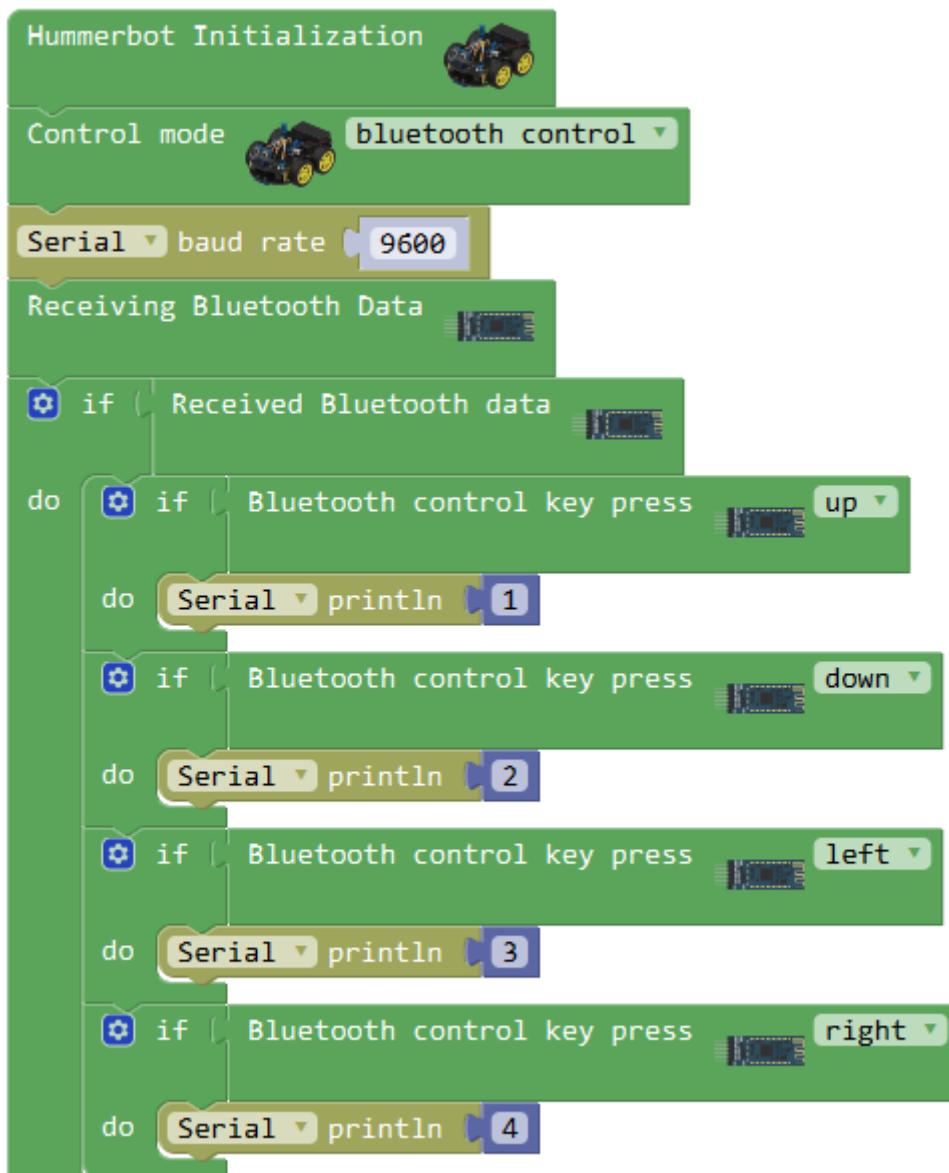


Figure 6-3-1

We have written the program as shown in Figure 6-3-1. After uploading the program successfully, we open the serial port, then connect it with the robot Bluetooth using mobile APP, press the left and right buttons, and the serial port will print the corresponding key numbers.

6.4 Programming Thought of Bluetooth Remote Control Robot

The programming idea of Bluetooth remote control robot is to set the mode as Bluetooth operation mode, initialize the serial port, and then the Bluetooth module of the program waits to receive the

HUMMER-BOT ROBOT

instructions from the mobile phone APP. The robot will act accordingly according to the instructions. Now we will write the program of Bluetooth remote control robot together.

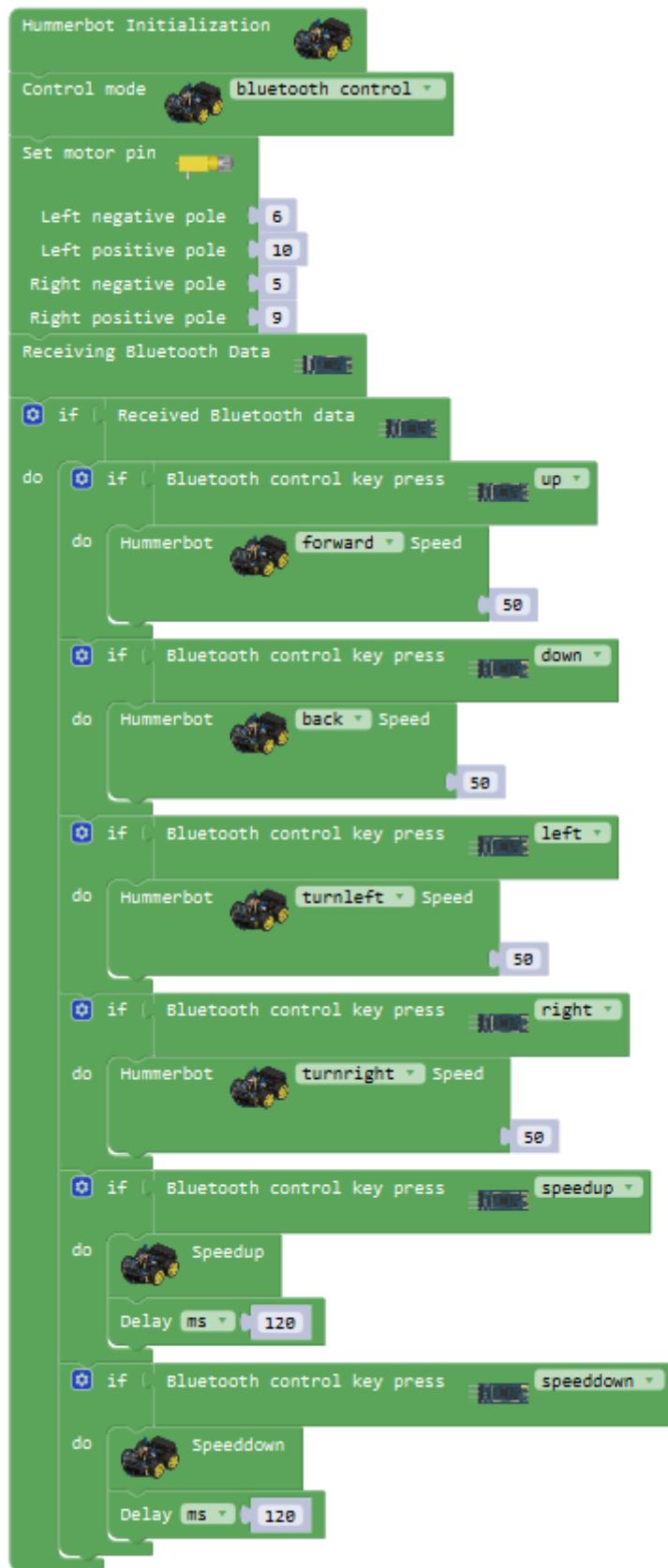




Figure 6-4-1

In the above programming, notice that we have made a 0.12S delay after the acceleration and deceleration program. This is to make the acceleration and deceleration of the robot a little more smooth. If there is no delay, the speed will directly change to the maximum or minimum.

Chapter 7 PS2 Telecontrol Robot

7.1 Principle of PS2 remote control handle

The PS2 handle is composed of a handle and a receiver (as shown in Figure 7-1-1). The handle needs two sections of 7 1.5V power supply. Put the handle switch on ON. Without searching for the receiver, the light on the handle will flash continuously. For a certain period of time, before searching for the receiver, the handle will enter standby mode, and the light on the handle will go out. At this time, press START." Key, wake up handle.

The working power supply of the receiver is 3 ~ 5V, so it can not be connected back or overvoltage, otherwise the receiver will burn out.

After normal power-on, the handle and the receiver are automatically paired and connected. In the unsuccessful state, the green light of the receiver flashes, and the light on the handle also flickers. When the pairing is successful, the green light on the receiver is always on, the light on the handle is always on, and the button "MODE" (with different batches of the handle, the above logo may be "AnalyOG", but it will not affect the use), so the "red light mode" can be selected. "Green light mode".



Figure 7-1-1



When the handle is connected to the receiver, we can use the handle to send button commands. When the receiver receives these button commands, the robot's brain (main control board) will let his limbs (four wheels, steering gear) act accordingly according to the received commands.

7.2 PS2 remote control test

We can first write a program to test the PS2 remote control, using serial port printing, when the PS2 remote control keys press, serial port printing the corresponding key values, we observe the contents of serial port printing, you can see which keys of the PS2 remote control are pressed, next we first write a test program.

HUMMER-BOT ROBOT

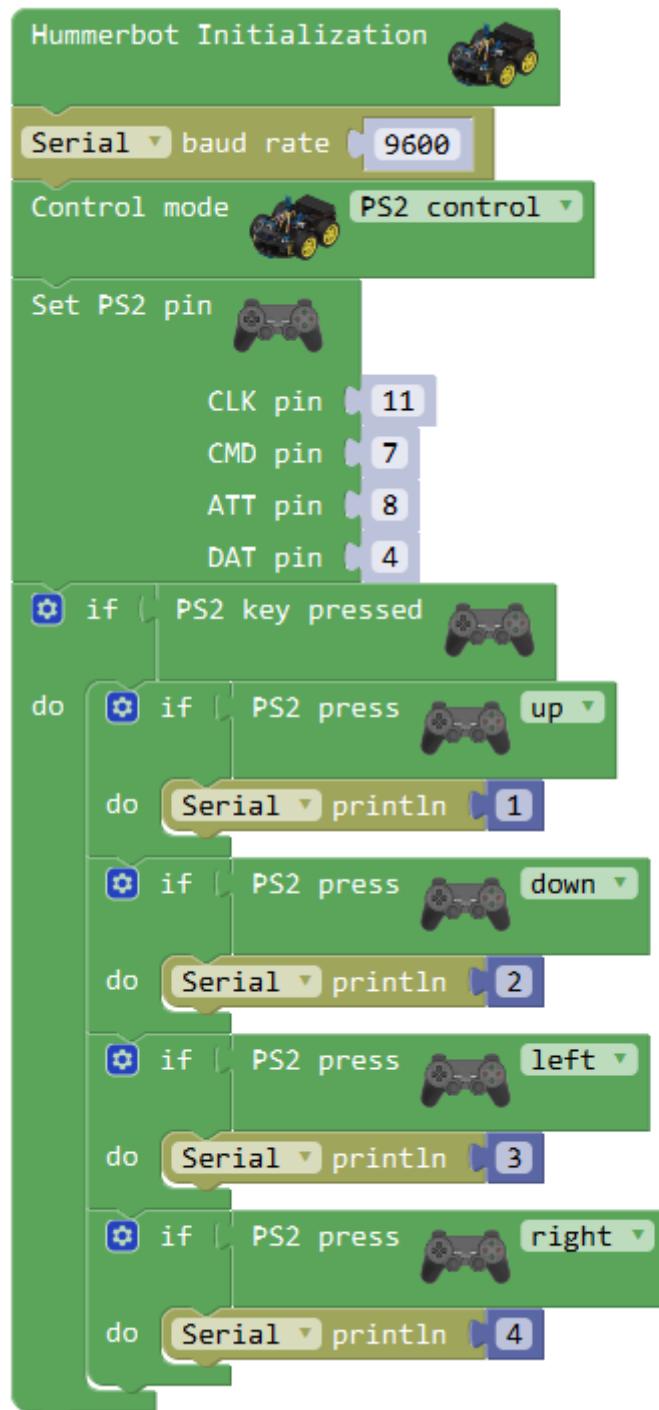


Figure 7-2-1

Fig. 7-2-1 bit PS2 test program, after uploading the program to the robot, open the serial port monitor, press the buttons around PS2, and the serial port will print the corresponding number.

7.3 Programming Thought of PS2 Remote Control Robot

The programming idea of PS2 remote control robot is to set the mode as PS2 control mode first, then define the key function on the PS2 handle according to its own needs. Our next programming is

HUMMER-BOT ROBOT

programmed according to the key definition shown in Figure 7-3-1. Then the robot executes the corresponding left, forward, right movement according to the received key value of the handle, or accelerates, decelerates, operates the steering gear, etc., PS2 handle. The remote control car defines all key functions as shown in Figure 7-3.



Figure 7-3-1

- Sign UP: Forward
- Sign DOWN: Back
- Sign LEFT: Turn left
- Sign RIGHT: Turn right
- Sign A: Acceleration
- Sign B: Left turn of steering gear
- Sign C: Deceleration
- Sign D: steering gear right turn

Now let's write the program of PS2 remote control robot.

HUMMER-BOT ROBOT

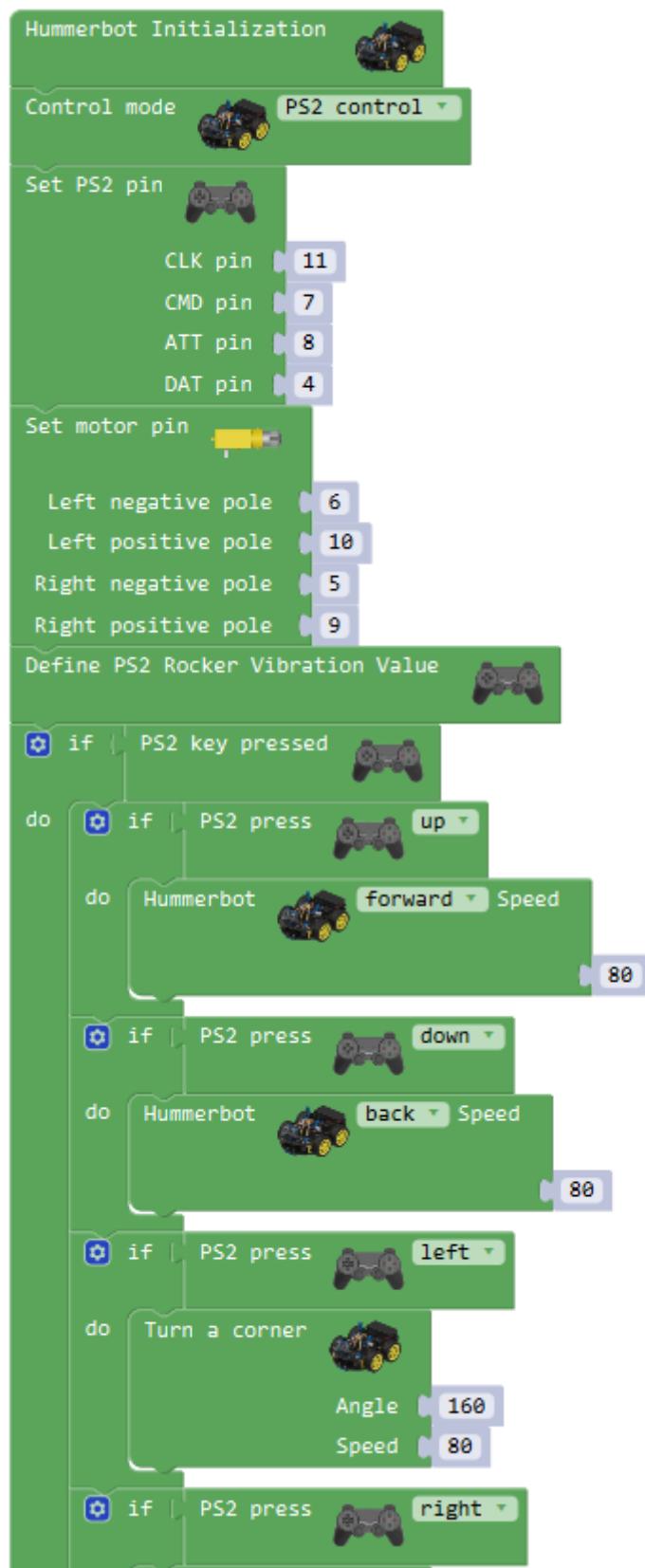


Figure 7-3-2

HUMMER-BOT ROBOT

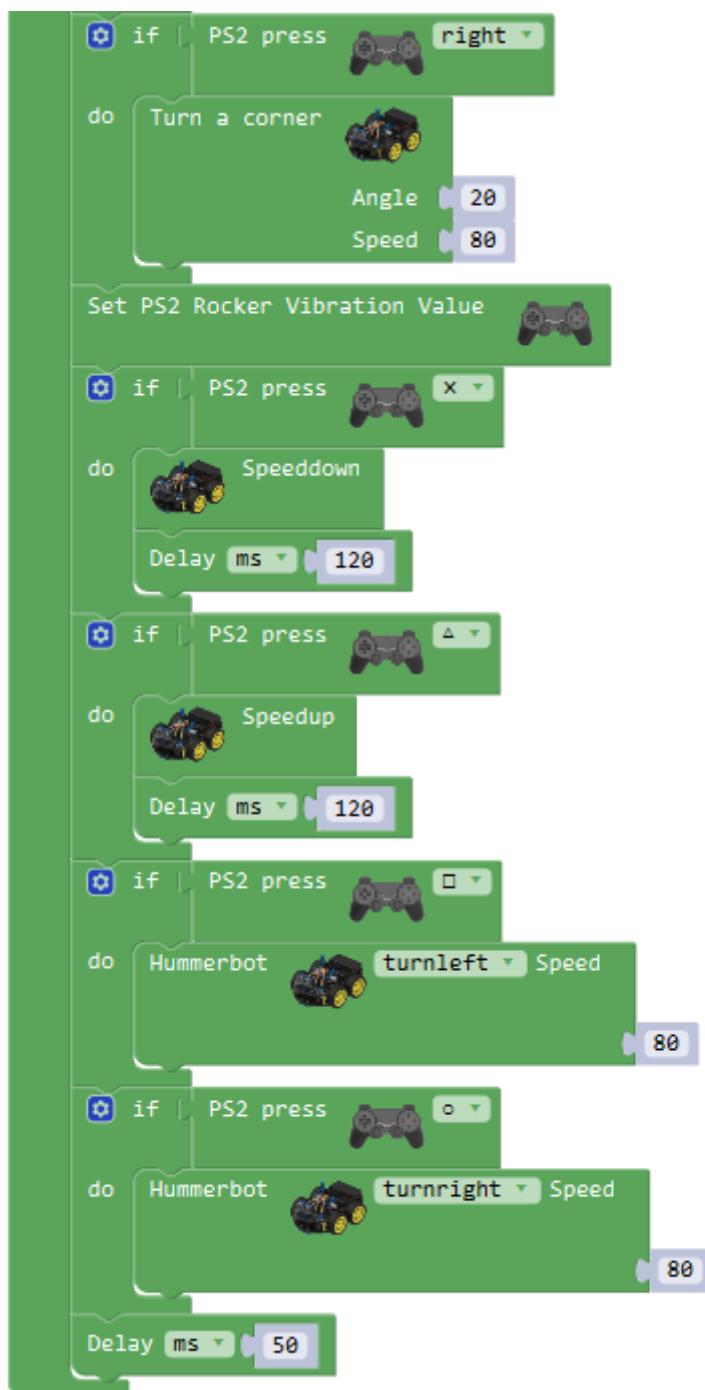


Figure 7-3-3

Transfer the program of Figure 7-3-2 and Figure 7-3-3 to the robot. We turn on the power of the robot, connect the PS2 handle to the receiver, press the button on the PS2 handle, and the robot will perform the corresponding action.

Chapter 8 Rgb Robots



8.1 RGB lamp

We usually see some colorful lights in the evening. The colorful lights are very gorgeous. There are also two colorful lights on Hummer-Bot. They are installed on the ultrasonic module. These two LED lights can emit colorful light. Why can one light emit colorful light? This is related to the three primary colors of light we have learned in physics. The three primary colors of light are red, green and blue, which is why? Three colors can be mixed together to show different colors, which is the principle of Hummer-Bot RGB LED lamp. The program building blocks in Hummer-Bot are shown in Figure 8-1 below.



Figure 8-1

After recognizing the building blocks of LED lights, let's write a simple program to make the two lights light up.



HUMMER-BOT ROBOT

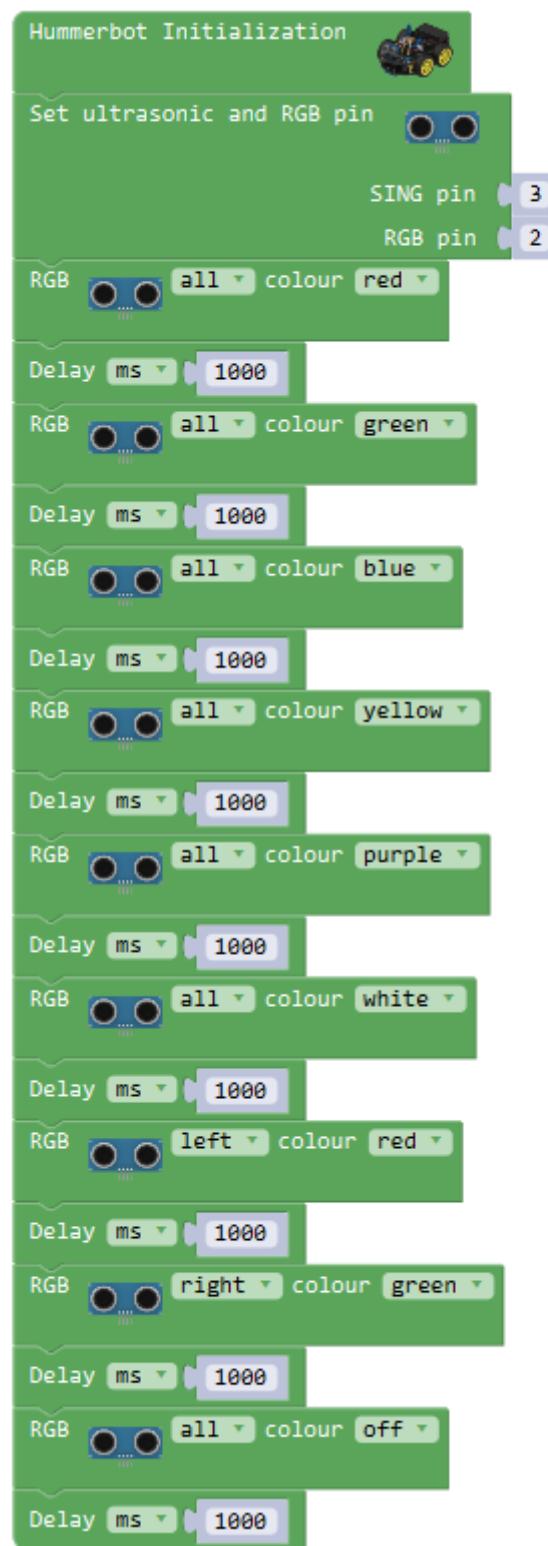


Figure 8-2

By running the program shown in Figure 8-2 above, we can see that when we turn on the Hummer-Bot power supply, the two lights will turn on, and the color will be changed every second. The color can be set by ourselves.



Chapter 9: Light-finding Robots

9.1 Light-finding Module and Principle

The light-finding function of the light-finding module is to output analog signals by judging the light intensity of the surrounding environment through the photodiode. When the photoresistor is illuminated by strong light, its resistance value drops rapidly, the passing current increases, the resistance value of the photoresistor increases rapidly in the dark environment, and the passing current decreases. The main control board judges whether there is a light source. The tracking module and program building blocks are shown in Figures 9-1-1 and 9-1-2.

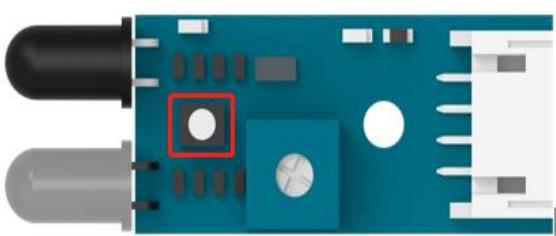


Figure 9-1

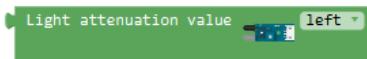


Figure 9-2

9.2 Light-finding Program

Understanding the principle of the light-finding module and building blocks, we can build a Hummer-Bot light-finding program, as shown in Figure 9-2-1.

HUMMER-BOT ROBOT

```

Hummerbot Initialization
Set Servo
  Servo pin 13
  Correction angle 90
Set motor pin
  Left negative pole 6
  Left positive pole 10
  Right negative pole 5
  Right positive pole 9
Set photosensitive pin
  Left pin A3
  Right pin A4
Control mode light seeking
Declare LeftValue as float value
  Light attenuation value left
Declare RightValue as float value
  Light attenuation value right
Declare Angle as int value 0
Hummerbot forward Speed 80
if LeftValue > 30 and RightValue > 30
do Hummerbot stop Speed 0
else if LeftValue ≥ RightValue
do Angle float RightValue ÷ LeftValue × 90
else if LeftValue < RightValue
do Angle 180 - float RightValue ÷ LeftValue × 90
Turn a corner
  Angle Angle 50

```



Figure 9-3

KeyWish-robot