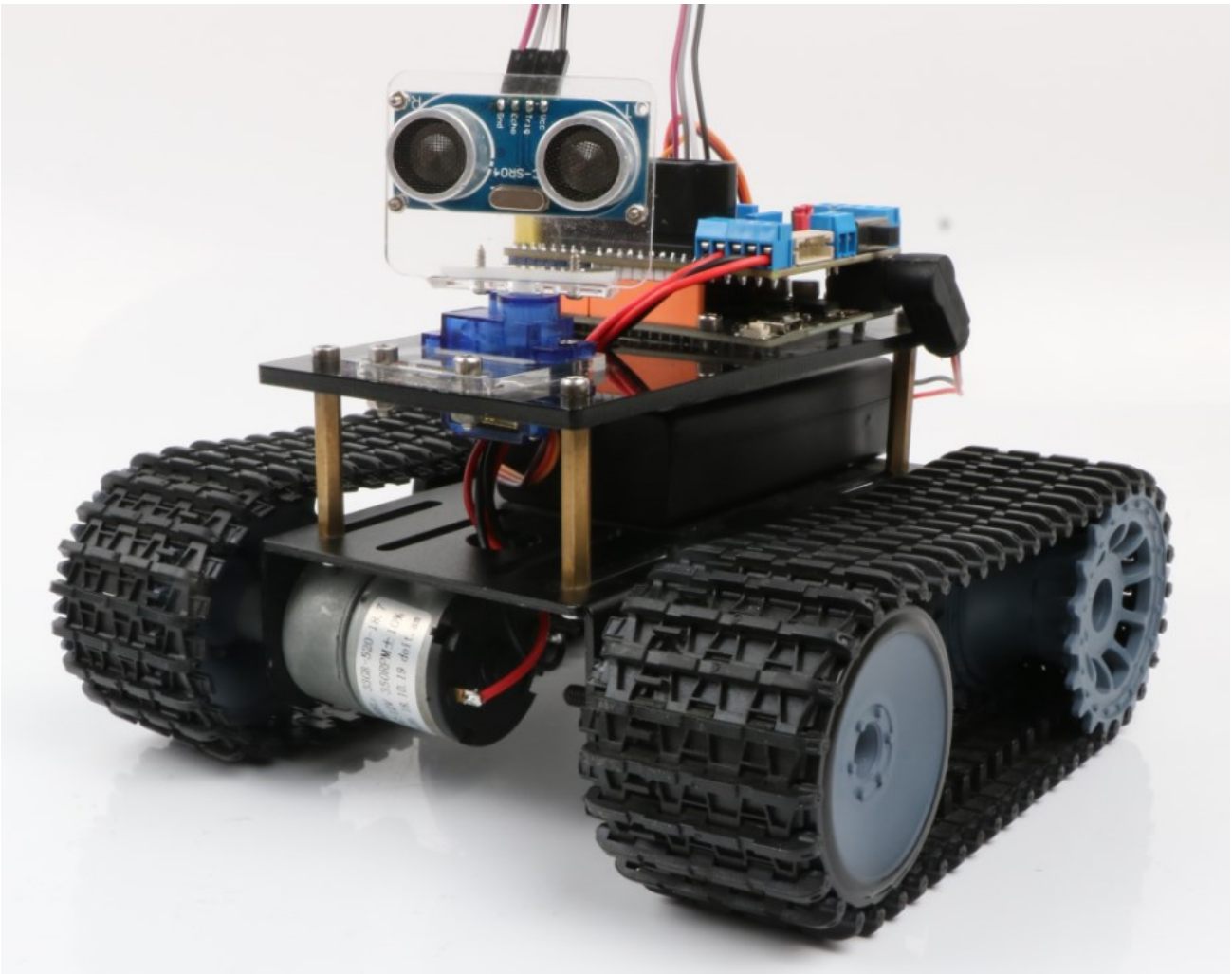
Panther-Tank-5.0 Instruction Manual V.1.4



Get last update from https://github.com/keywish/keywish-panther-tank

Revised Version Of History

Date	Version	Description	Author
2019/8/14	V.1.0	Create	Carl.Du
2019/8/27	V1.1	Update the picture	Abbott.Chen
2019/7/30	V.1.2	Modify picture error, increase charging instructions	Carl.Du
2019/9/27	V.1.3	Modify the expansion board and add the PS2 control arm description	Carl.Du
2019/12/12	V.1.4	Modify expansion board, add expansion board version switch	Carl.Du

CONTENTS

Chapter 1 PREFACE

1.1 Purpose

Our purpose is to offer a learning platform for DIY lovers, makers and beginners, help to get a better understanding of Arduino, and its expansion system design methods and principles, as well as the corresponding hardware debugging methods. Further deepen the understanding of the design and application of Arduino and its extended system.

The instruction manual mainly introduces the installation, hardware and software for Panther- Tank from the easy to the difficult and complicated. There are two parts for the instruction manual.

The first part mainly introduces how to use the common developing software and the method of downloading,debugging. And the second part mainly introduces about hardware and software. For the hardware part, it mainly introduces the functions, principles of every module, and for the software part, there are many examples for customers, it mainly introduces the applications of the Tank.

There are lots of detailed schematic diagrams and example codes for each module, which is strictly tested to ensure the accuracy and precision. Moreover, you can easily find the library files in the corresponding file folder and download through the UART/simulator to the BLE uno board for the corresponding functions . You can debug each module according to the course or directly assemble the car to enjoy the fun of a maker.

1.2 Product Introduction

"Tank" is ATMEGA328P-PU as the main control chip, and H450 is used as a multi-functional crawler car for motor drive chip. Compared with the traditional car, "Tank" is also equipped with wireless control (Bluetooth, infrared remote control). It can automatically avoid obstacles. Of course, Maker can also add or subtract other functions through its own Idea, such as adding automatic tracking, PS2 gamepad, adding wifi control, robotic arm, etc.

"Tank" is equipped with all kinds of materials, technical manuals, routines, etc., and teaches you from entry to proficiency. Every electronic enthusiast can easily get started and realize the functions they want.

Product features

- High power all metal geared motor
- Integral stamping molding kit,easier Installation,tighter
- 2400mAH,7.4v ,rechargeable li-battery,longer battery life,and more dynamic
- 2 RGB turn lights
- Buzzer Turn around reminder

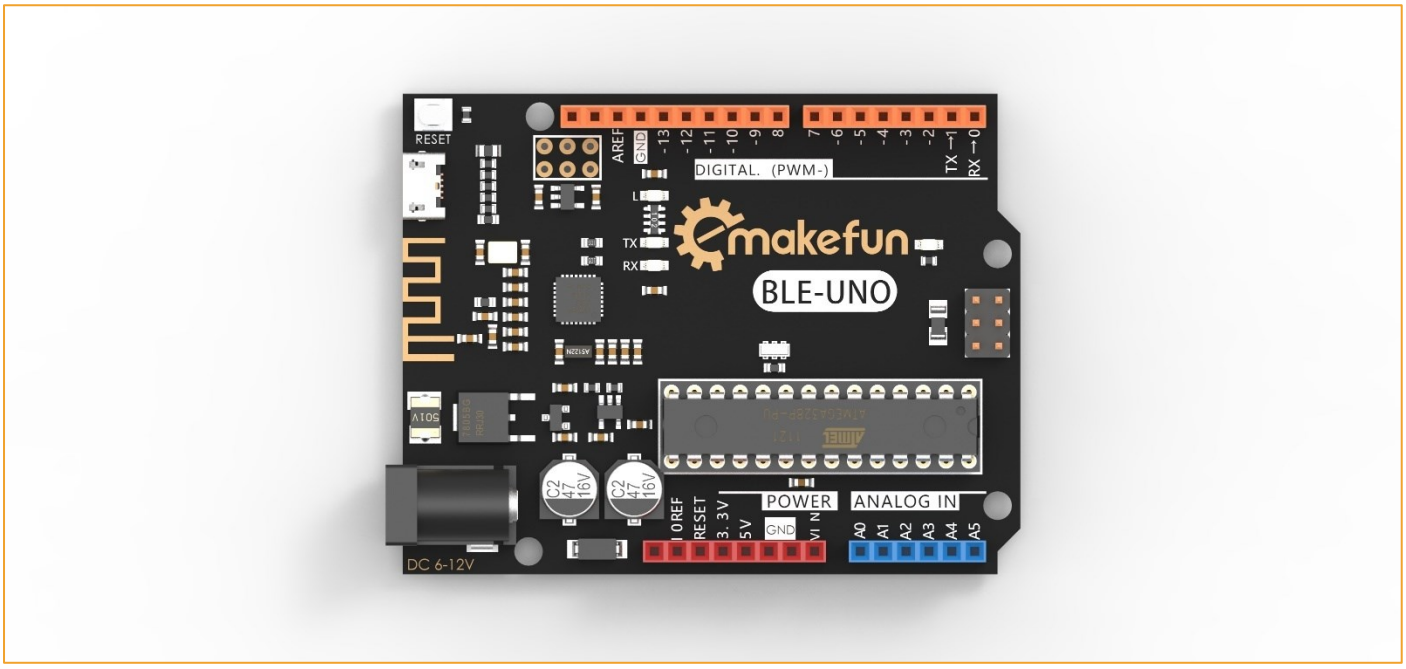
- Infrared remote control
- Android App control

Chapter 2 Preparation

2.1 On BLE UNO Master Control Board and Extension Board

2.1.1BLE UNO R3 Master Control Board

In " Hummer-Bot ", we used the BLE-UNO R3 as the main control board, which has 14 digital input/output pins (6 of which can be used as PWM output), 6 analog inputs, and a 16 MHz ceramic resonator, 1 USB connection, 1 power socket, 1 ICSP head and 1 reset button. It contains everything that supports the microcontroller; You just need to connect it to a computer via a USB cable or start with an AC-DC adapter or battery.



`Figure 2.1.1 Arduino BLE - UNO R3 Board

Technical specifications:

- Working voltage: 5V
- BLE chip : TI CC2540
- BLE work channel :2.4G
- Bluetooth transmission distance: 50m open distance
- Interface : Micro-USB
- Input voltage: USB powered or external 7V~12V DC input

- Output voltage: 5V DC output and 3.3V DC output and external power input
- Microprocessor: ATmega328 (Chip data sheet is in the documentation)
- Bootloader： Arduino Uno
- Clock frequency: 16 MHz
- Support USB interface protocol and power supply (without external power supply)
- Support ISP download function
- Digital I/O port: 14 (4 PWM output ports)
- Analog input port: 6
- DC Current I/O Port: 40mA
- DC Current 3.3V Port: 50mA
- Flash memory: 32 KB (ATmega328) (0.5 KB for bootloader)
- SRAM ： 2 KB (ATmega328)
- EEPROM： 1 KB (ATmega328)
- Size：75x55x15mm

2.1.2 MotorDriverBoard Expansion board interface diagram

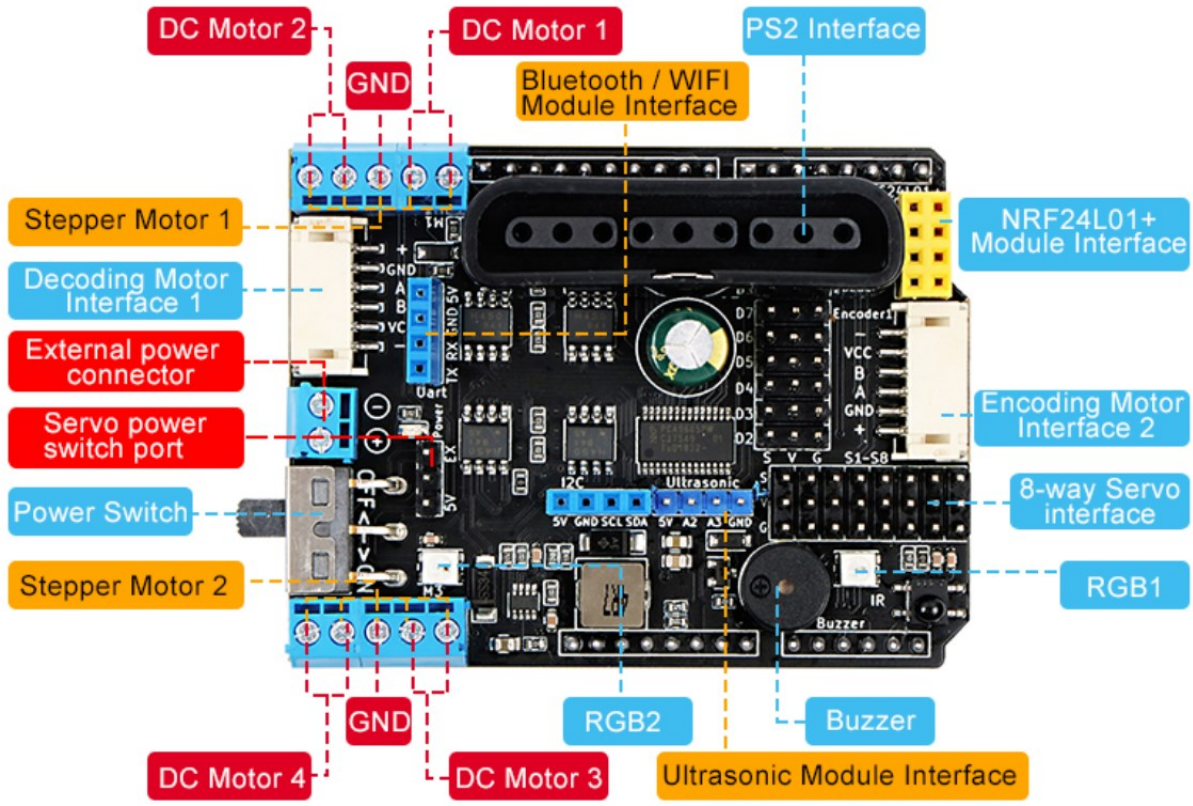


Figure 2.1.2 Expansion Board Interface Diagram

2.2 Development environment Arduino IDE

2.2.1 Install the IDE

AduinoIDE is an open source software and hardware tool written by open source software such as Java, Processing, and avr-gcc. It is an integrated development environment that runs on a computer. It can write and transfer programs to the board. The major feature of the IDE is cross-platform compatibility for Windows, MaxOSX, and Linux. Only a simple code base is needed, and the creators can create personalized home internet solutions through the platform, such as remote home monitoring and constant temperature control and so on.

In this tutorial, we use the version is 1.6.0, download address is : https://www.arduino.cc/en/Main/OldSoftwareReleases#previous， After opening the link, we can see the interface as shown in Figure 2-2-1. In this interface, we can see the different versions of the IDE and different operating environments. Everyone can download according to their own computer system, of course, There will be a downloaded installation package on our companion CD, but only the Windows version, because this tutorial is all running under Windows system.

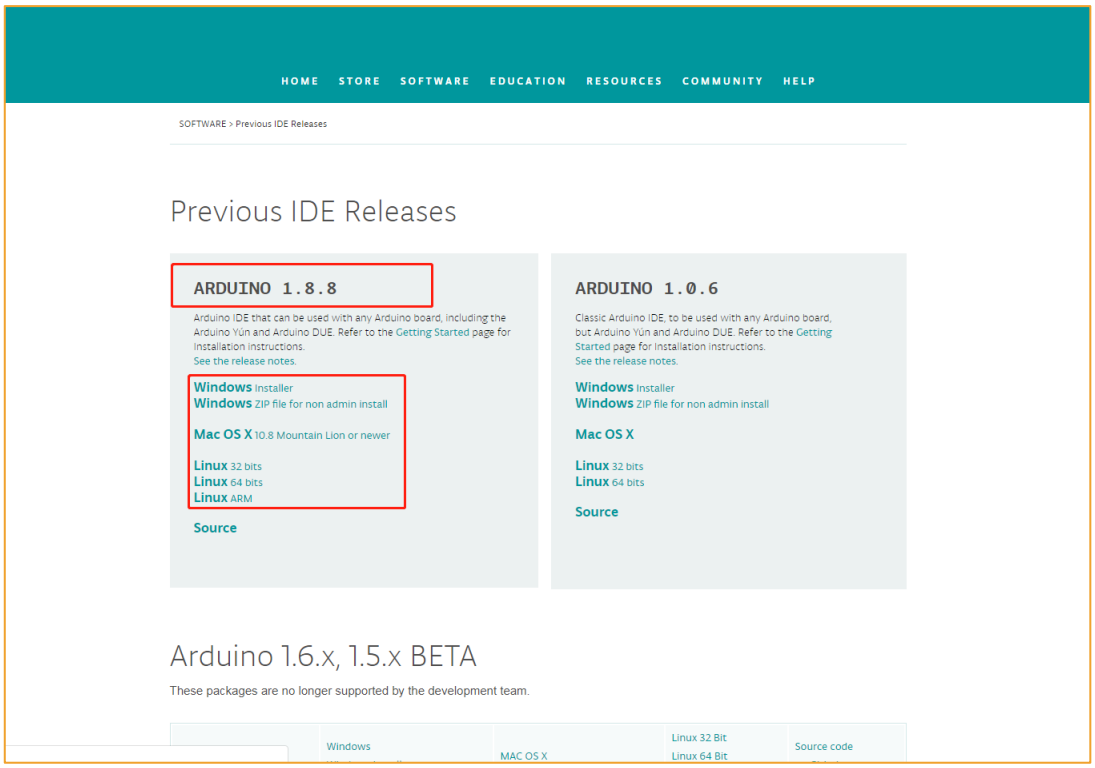


Figure 2-2-1 ArduinoIDE download interface

After the downloading, we will get a compressed package as shown in Figure 2-2-2. The compressed package will be decompressed. After decompression, the files in Figure 2-2-3 are extracted. The “drivers” is the driver software. When the “Arduino.exe” is installed, it will be Install the driver automatically. Because the installation of "arduino.exe" is very simple, it will not be explained here. It is recommended to exit the

anti-virus software during the installation process, otherwise it may affect the installation of the IDE. After the installation is complete, click "arduino.exe" again to enter the IDE programming interface.

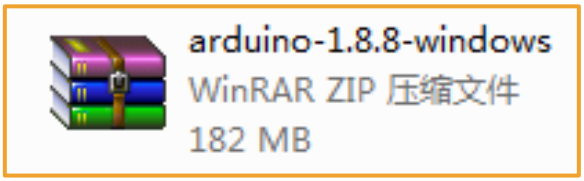


Figure 2-2-2 Arduino IDE Installation Package

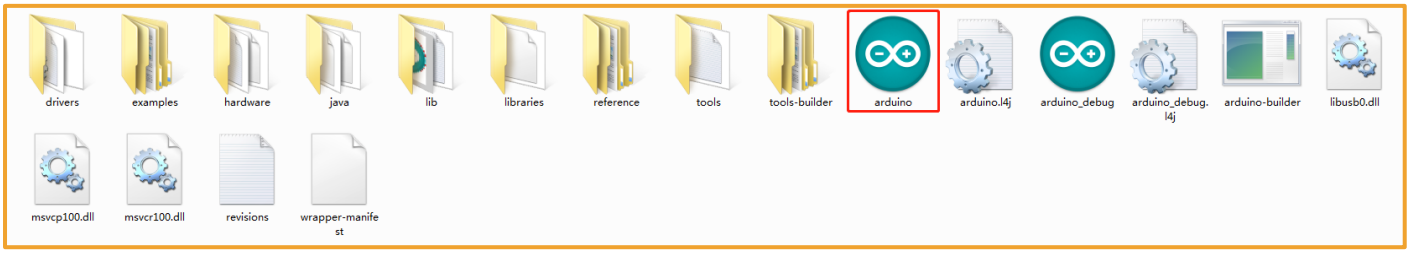


Figure 2-2-3 Extracted files

When finish the installation of the IDE，connect to the Arduino motherboard，click“My Computer”→“Properties”→“Device Manager”→ “Viewing Ports (COM and LTP)”，If you can see as the Figure 2-2-4

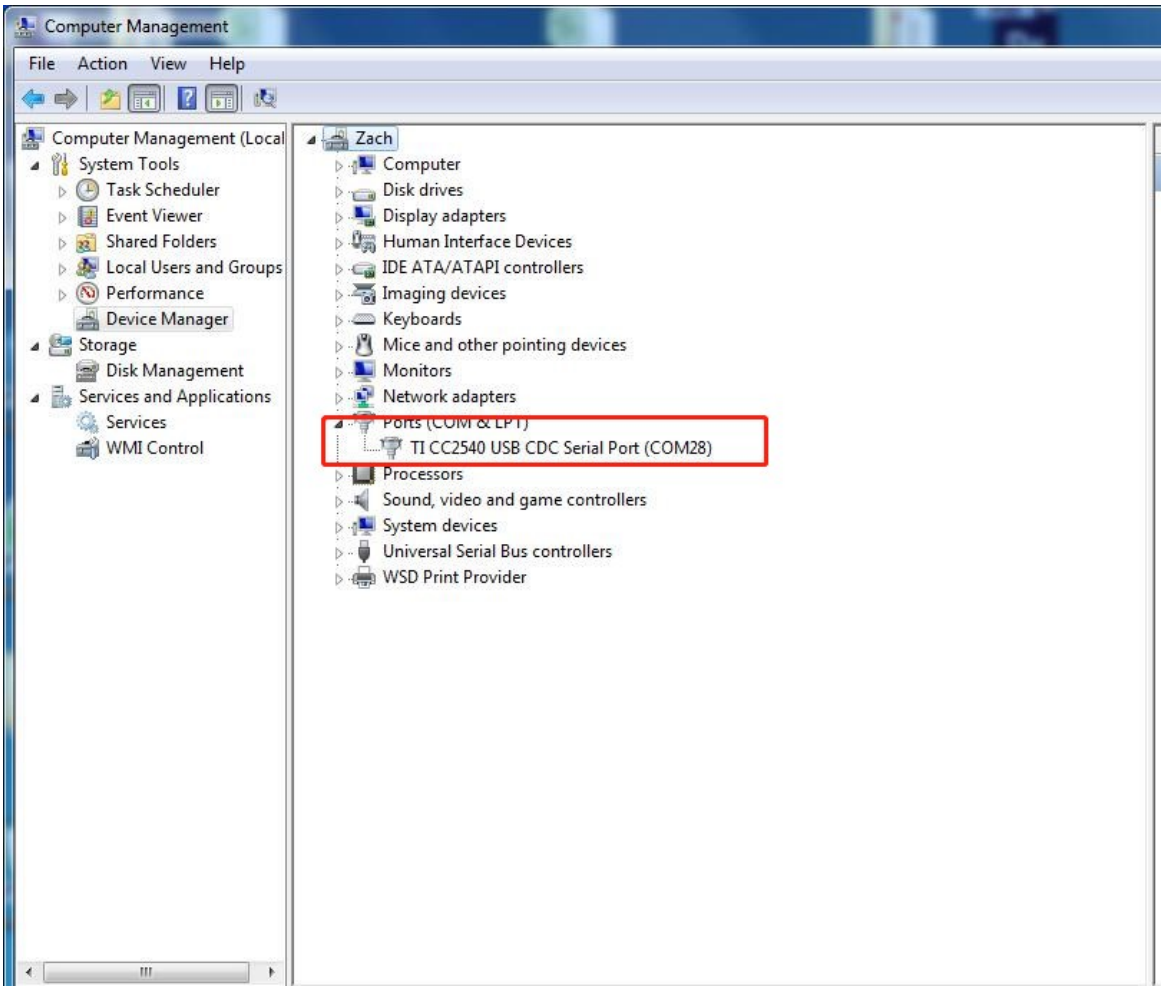


Figure 2-2-4 Driver installation success interface

that indicates the driver has been installed successfully，At this time we open the IDE, select the corresponding development board model and port in the toolbar to use normally. If you see Figure 2-2-5, it means that the computer does not recognize the development board and you need to install the driver yourself.

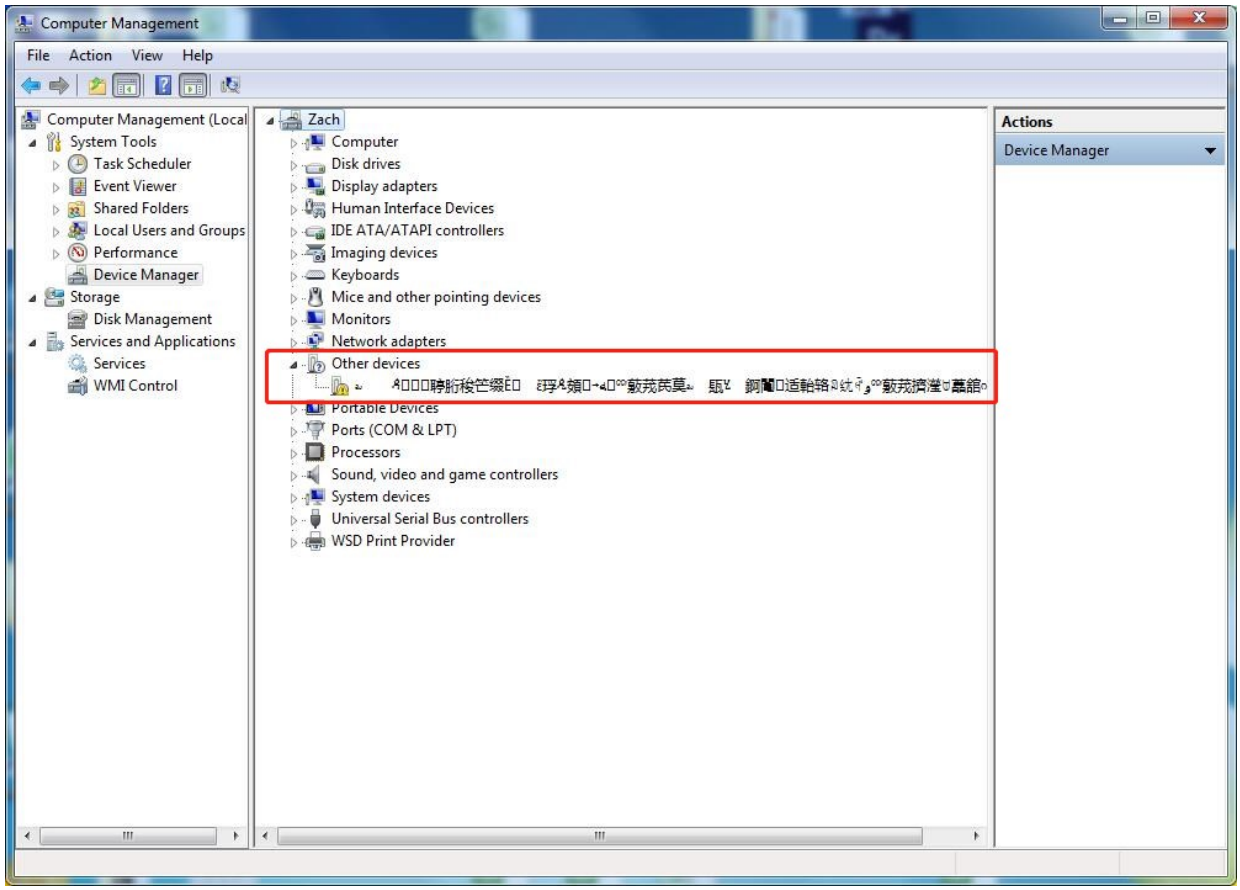


Figure 2-2-5 Driver is not successfully installed interface

Notice:

1）If you connect the controller board to the computer, the computer does not respond. Right-click "My Computer" and select Open Device Manager then find viewing port (com & lpt). If there is no com or lpt, or only an unknown device, there is a problem with the controller board or the USB cable.

2）Right-click "My Computer" and select Device Manager, find the viewing ports (COM and LPT). If there is a yellow Arduino UNO exclamation point, this means you need to install the driver yourself.

3）If you install the driver again and again, it eventually fails. Please uninstall the driver and re-install> install the driver automatically> restart the computer.

2.2.2 Install Driver

If your computer is a Windows 7 system

1）Right-click on "My Computer" and open the Device Manager, find viewing the ports (COM and LPT). At this point you will see a "USB Serial Port", right-click "USB Serial Port" and select the "Update Driver Software" option.

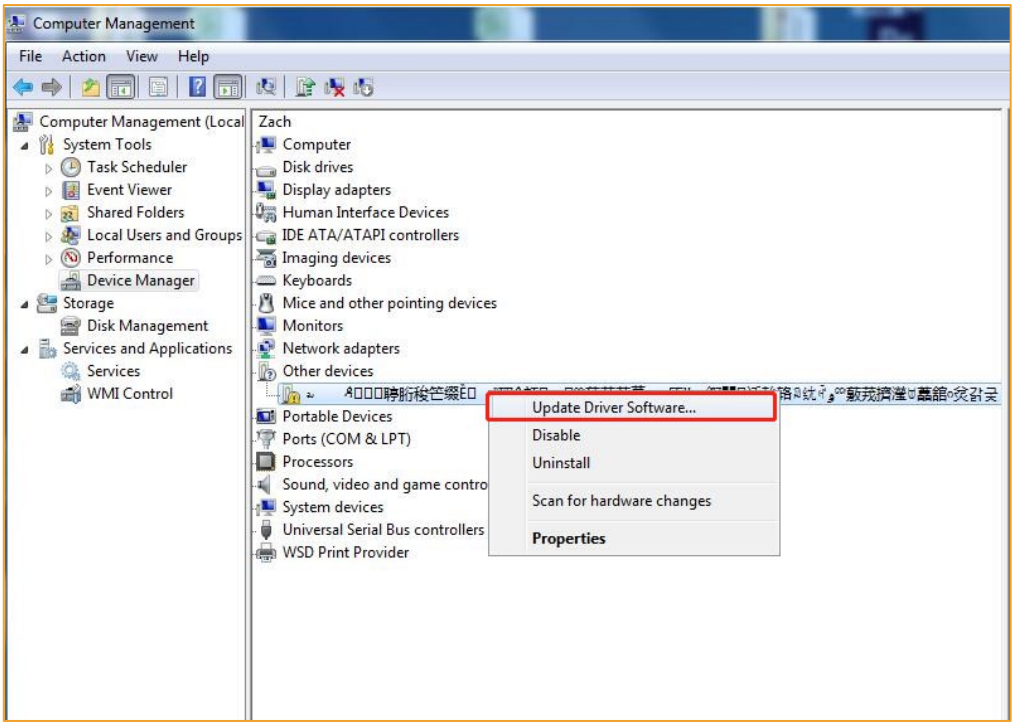


Figure 2-2-6 Updated Driver Interface

2）Next, select the "Browse my computer for driver software" option.

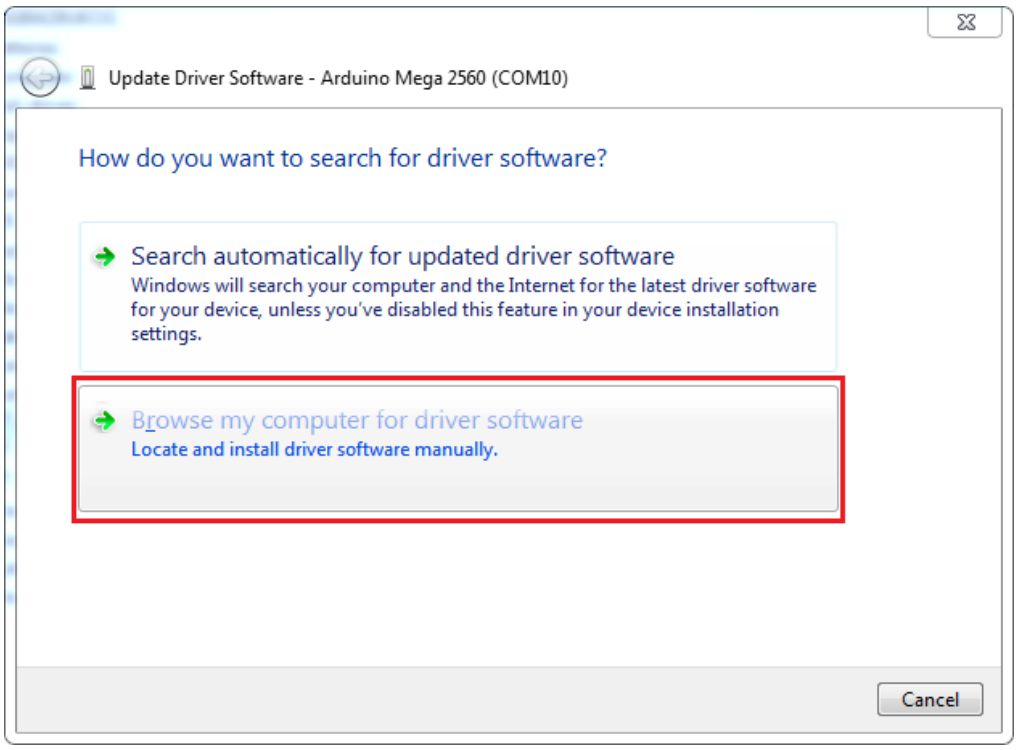


Figure 2-2-7 Driver Update Selection Screen

3）Driver path: " BLE-UNO _Driver \ ccxxxx_usb_cdc.inf" click "next", as shown in figure 2-2-8

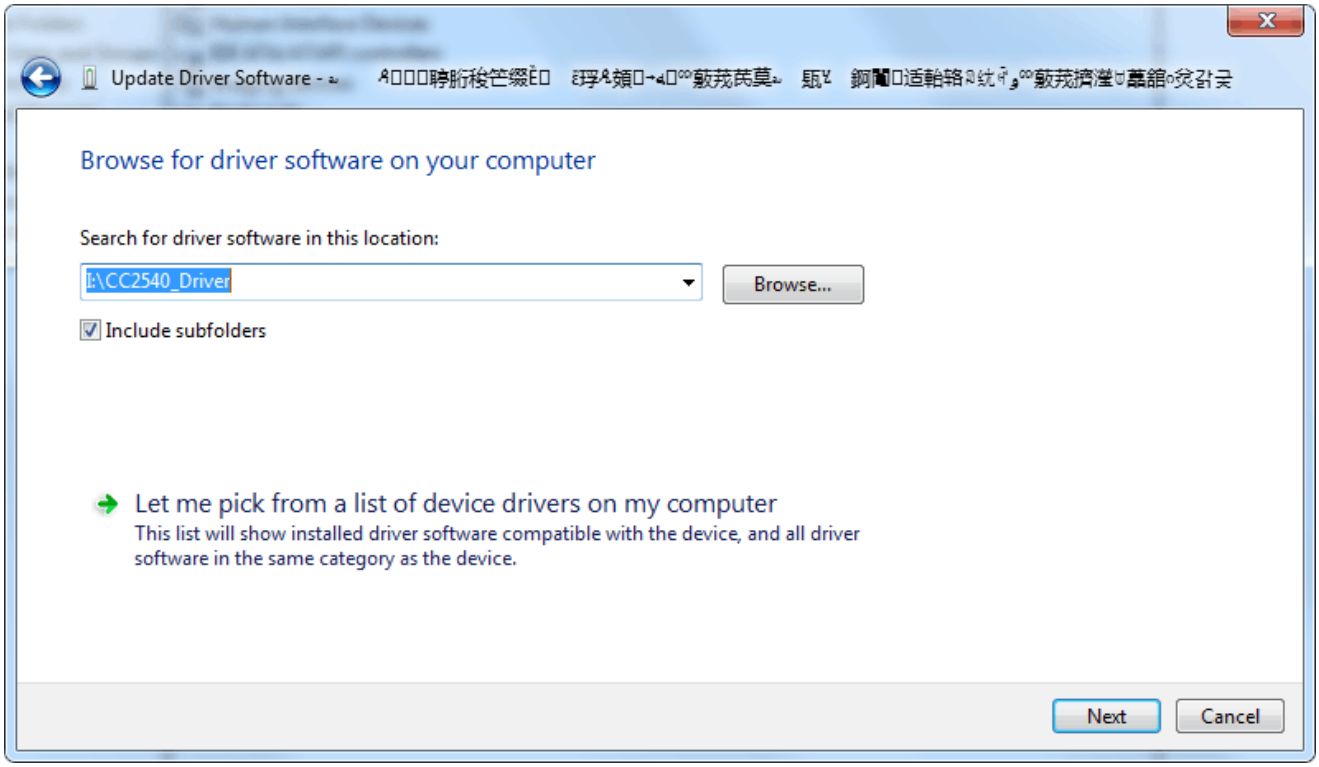
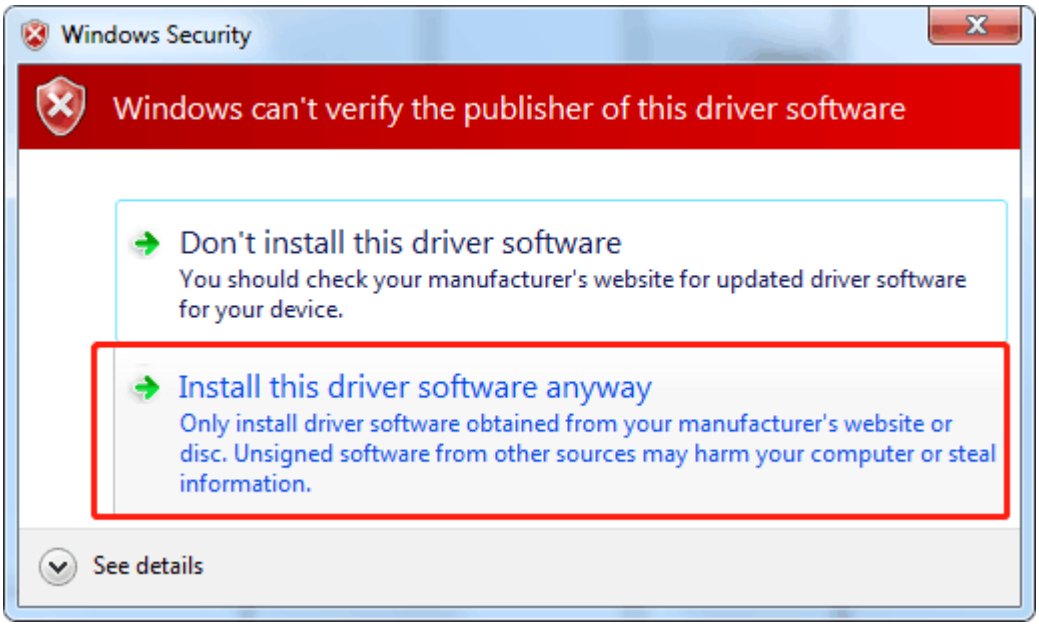


Figure 2-2-8 Driver file selection interface

4）If you have already installed, the following figure will automatically inform you that the driver was successful.



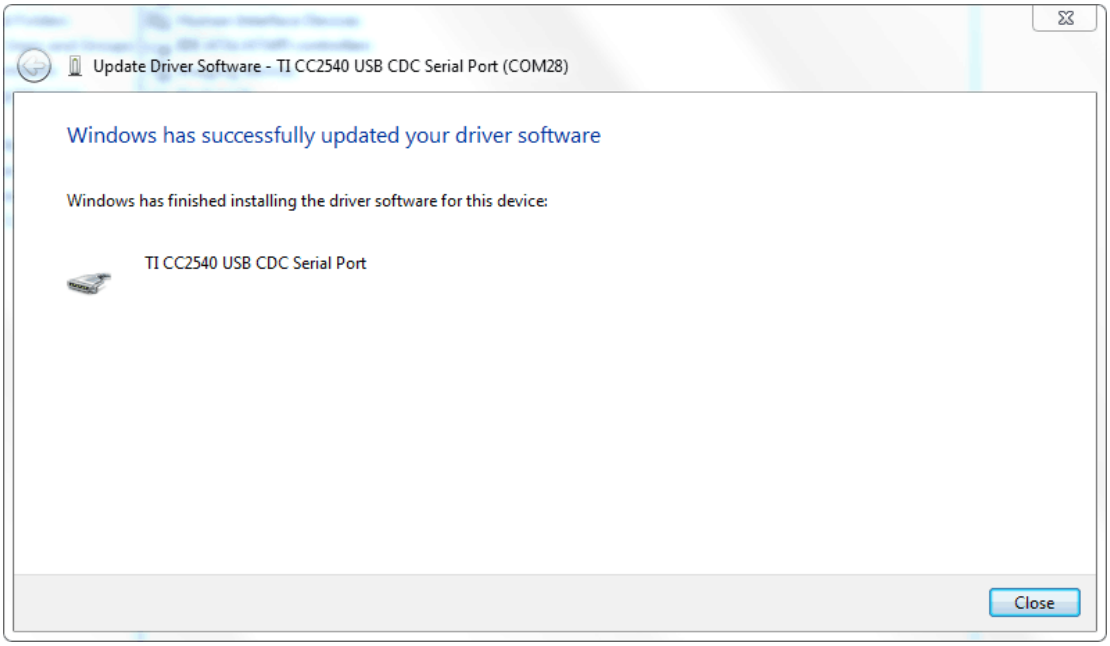


Figure 2-2-9 Driver Installation Successful Interface

At this time, we return to the "Device Manager" interface, the computer has successfully identified Arduino, as shown in the below Figure 2-2-10 .then open the Arduino compilation environment, you can open the Arduino trip.

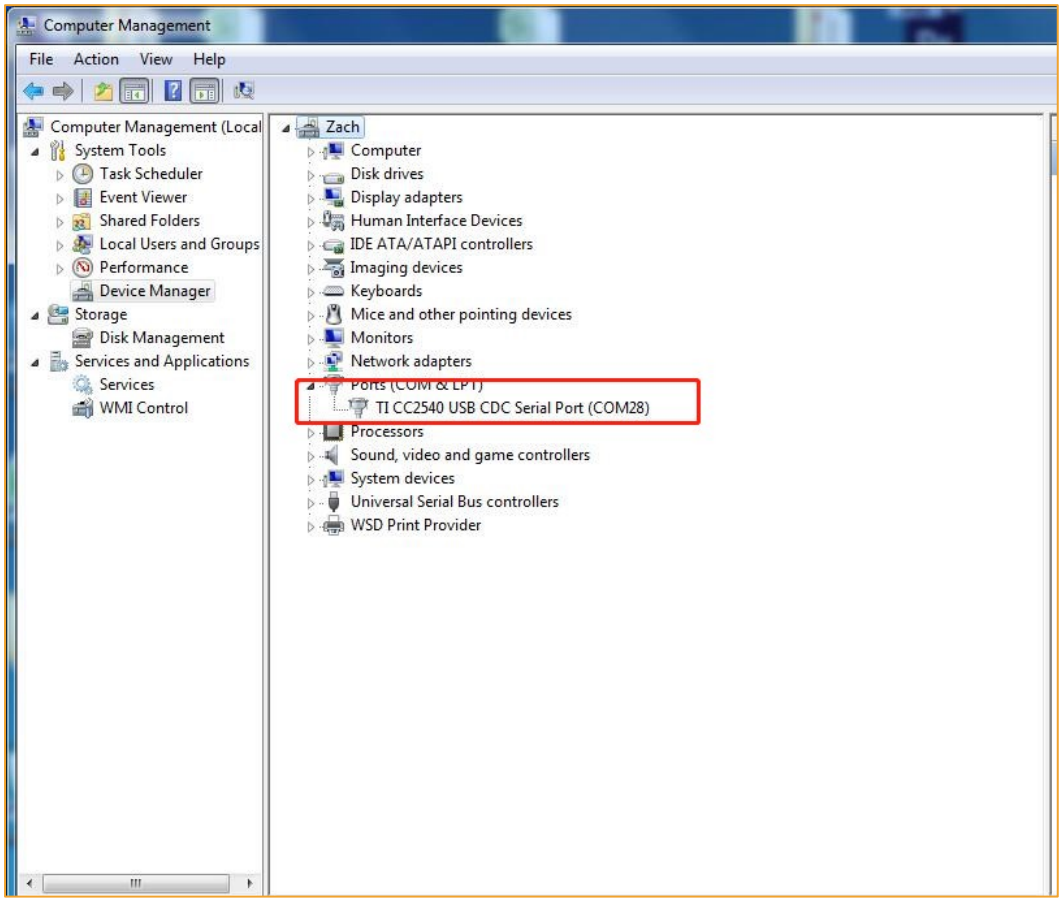


Figure 2-2-10 Driver Success Recognition Interface

Notice In Win10 system, some Arduino are connected to the computer (non-genuine chips are difficult to identify), the system will automatically download the corresponding driver, so you can not install the driver yourself, but in the Win7 system, you have to do it yourself.

In addition, we can see that the USB serial port is identified as COM15 in the above figure, but it may be different with different computer, you may be COM4, COM5, etc., but USB-SERIAL CH340, this must be the same. If you do not find the USB serial port, you may have installed it incorrectly or the system is incompatible.

2、If your computer is a Windows 8 system: Before installing the driver, you should save the files you are editing because there will be several shutdowns during the operation.

1）Press "Windows key" + "R"

2）Iput shutdown.exe / r / o / f / t 00

3）Click the "OK" button.

4）The system will reboot to the "Select an option" screen

5）Select "Troubleshooting" from the "Select an option" screen

6）Select "Advanced Options" from the "Troubleshoot" screen

7）Select “Windows startup settings screen” from “Advanced Options”

8）Click the "Restart" button

9）The system will reboot to the “Advanced Boot Options” screen

10）Select "Disable Driver Signature Enforcement"

11）Once the system is booted, you can install Arduino driver the same as Windows7

3、If your computer is a Windows XP system: The installation steps are basically the same as for Windows 7, please refer to the above Windows 7 installation steps.

2.2.3 IDE Interface Introduction

Nextly,we introduce the Arduino IDE interface, firstly enter the software directory. Then you can see the arduino.exe file and double-click to open the IDE. As shown in Figure 2-2-11.

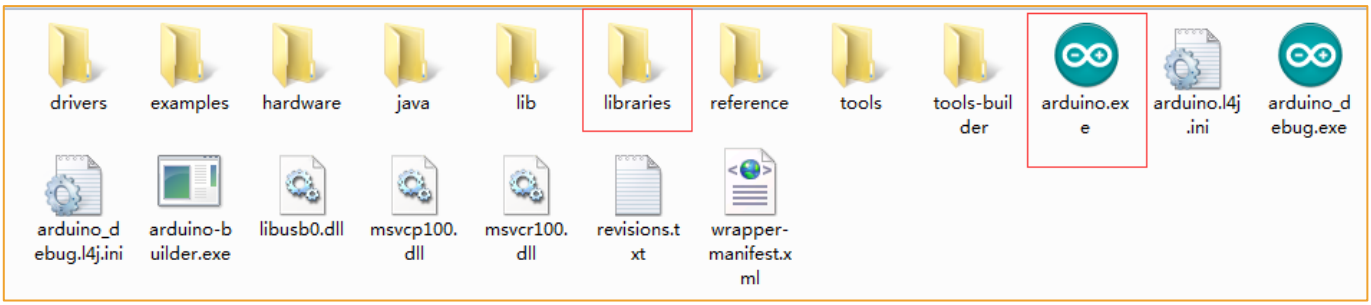


Figure 2-2-11 Software Catalog

1、The first thing you can see is the interface of the following figure. The functions of the toolbar buttons are "Compile" - "Upload" - "New Program" - "Open Program" - "Save Program" - "Serial Monitor" , as shown in Figure 2-2-12.

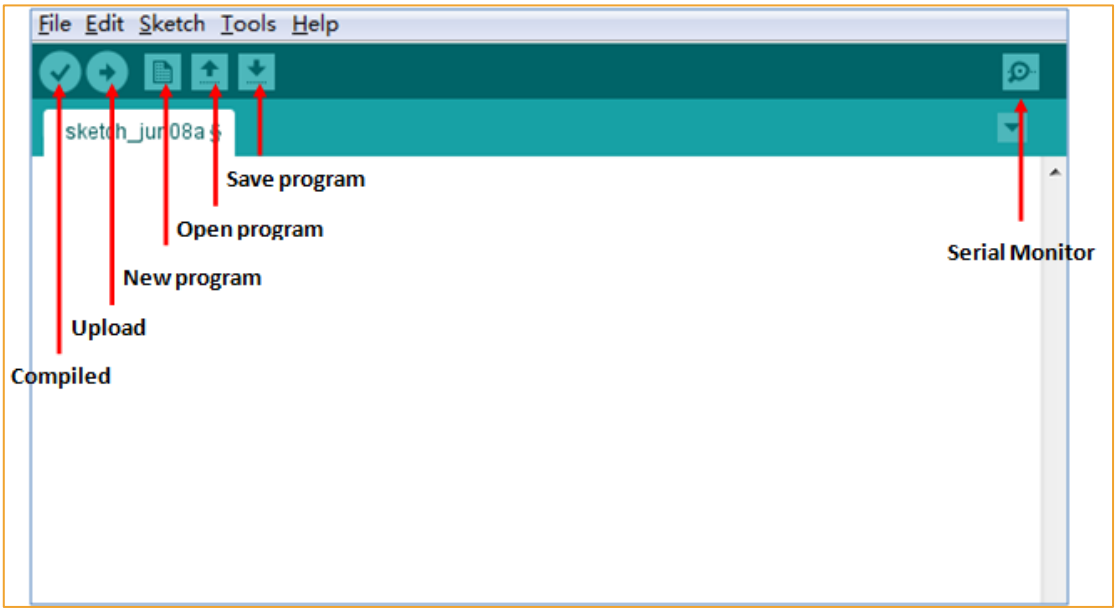


Figure 2-2-12 Arduino IDE Interface

2.There are 5 menus on the menu bar, but we mainly introduce File and Tools. Click File, the interface as shown in Figure 2-2-13 will be displayed, you can see the Examples and Preference options. The Examples are some of the Arduino's own programs, these are compiled without errors, the normal use of the program, a great help for beginners. The Preference option, It's mainly about the parameter settings, such as language, fonts and so on.

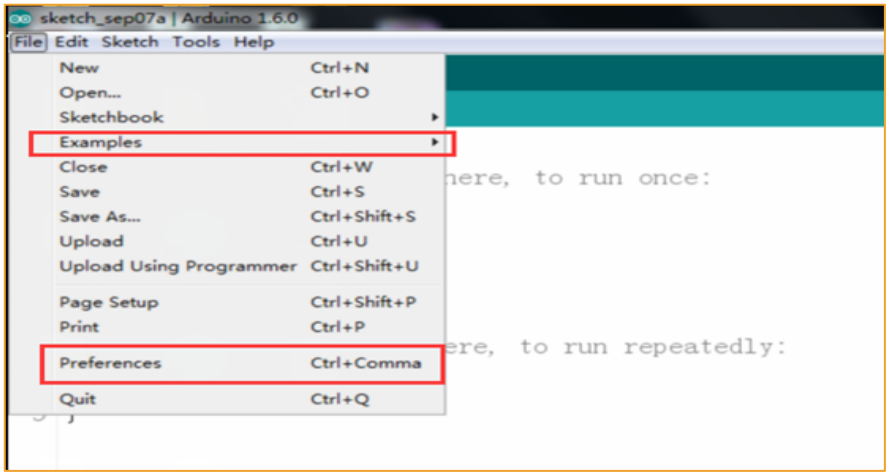


Figure 2-2-13 File Menu Bar Options

3、Click Tools, the interface shown in Figure 2-2-14 will pop up. Here we can see two options: Board and Port. In the board option, we can see the commonly used Arduino development board model, we only need to choose according to their own development board. In the Port option, the USB serial port is mainly selected, as shown in Figure 2-2-15. If you are not sure, you can check it in the "Device Manager" and select the corresponding COM port.

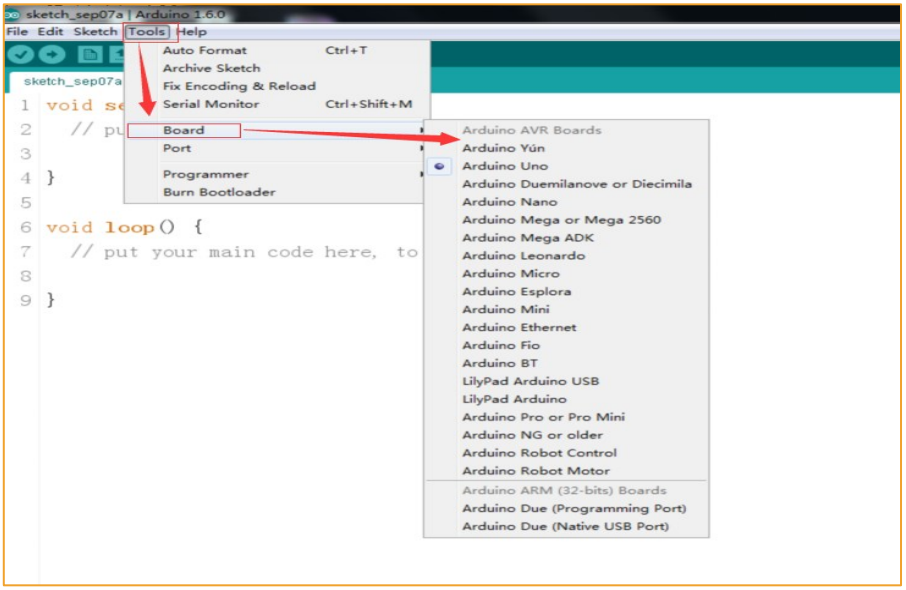


Figure 2-2-14 Tools interface

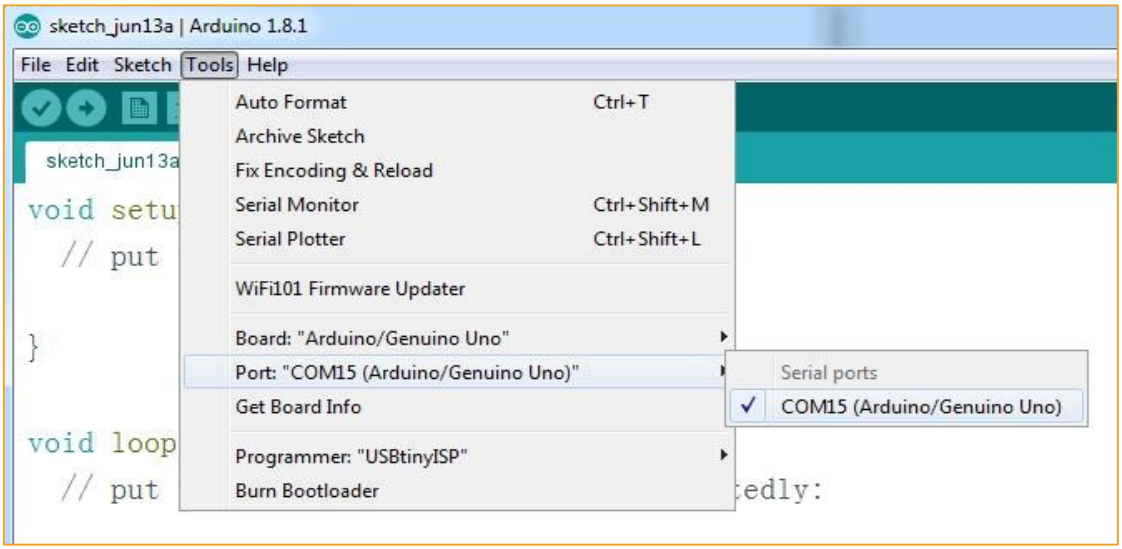


Figure 2-2-15 USB serial port selection

So far, we have basically completed all the work. The next step is actual experiments. Open any program in Examples. First compile the program. If it is compiled correctly, it can be directly downloaded to the development board and the corresponding device of the connection number. With wires, you can see the corresponding phenomenon.

Chapter 3 Installation

3.1 Tank Assembly

Firstly, we open the box, take out all the components and put it on the table lightly. (Note: There are many devices, be careful when installing to prevent some devices from being lost)

3.1.1 Lower metal backplane copper column and battery installation

Step 1: The installation of the copper plate under the metal bottom plate

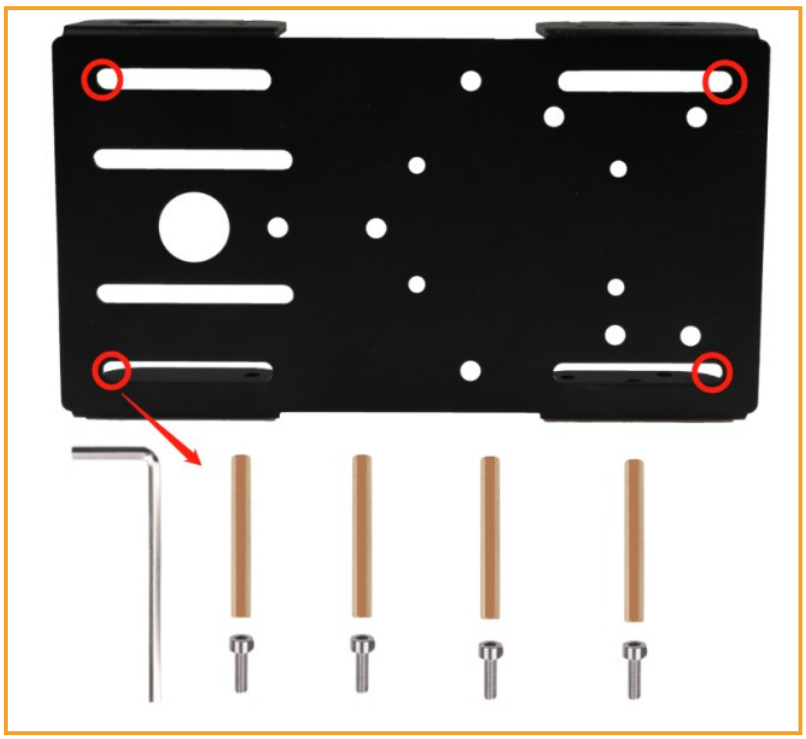


Figure 3-1-1 Schematic diagram of copper column installation on lower metal bottom plate

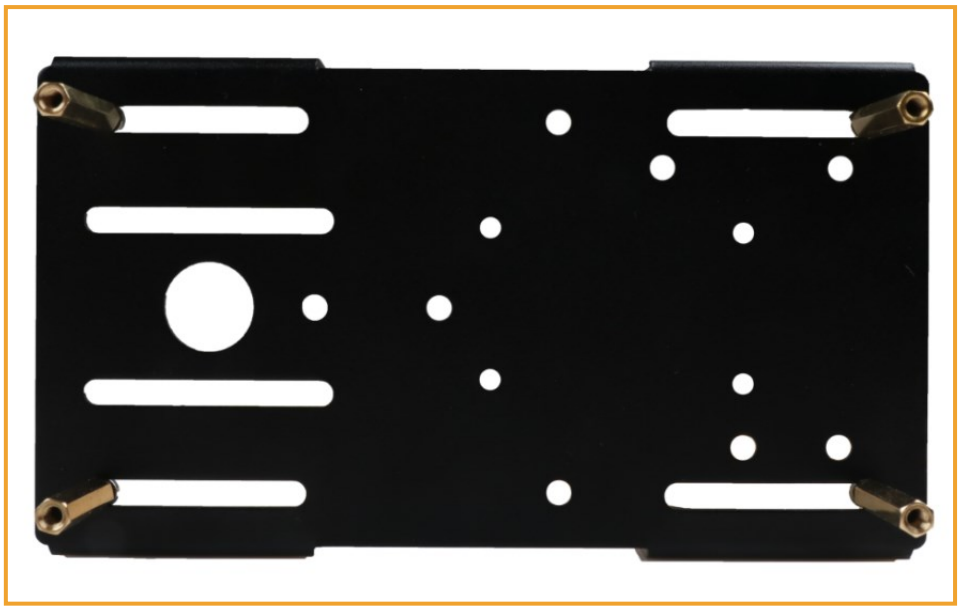
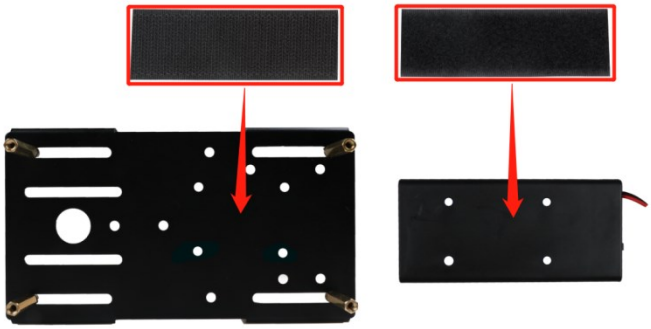


Figure 3-1-2 Under metal floor copper column installation effect diagram

Step 2：Battery installation



Note: Attach a piece of Velcro to the back of the battery and attach another to the top of the metal base plate.

Figure 3-1-3 Battery box installation diagram

Figure 3-1-4 Battery box installation effect diagram

Step 3：Welding DC head

Connect the power cable: Firstly, find the matching two power cables (the same as the wires used by the motor, one red and one black) and connect the two cables to the DC power head. The DC power connector is shown in Figure3-1-5The rubber ring marked as "1" can be removed to open the shell and then welding the wires to the +12V and GND, as shown in Fig.3-1-6

Figure 3-1-5 Power DC Head

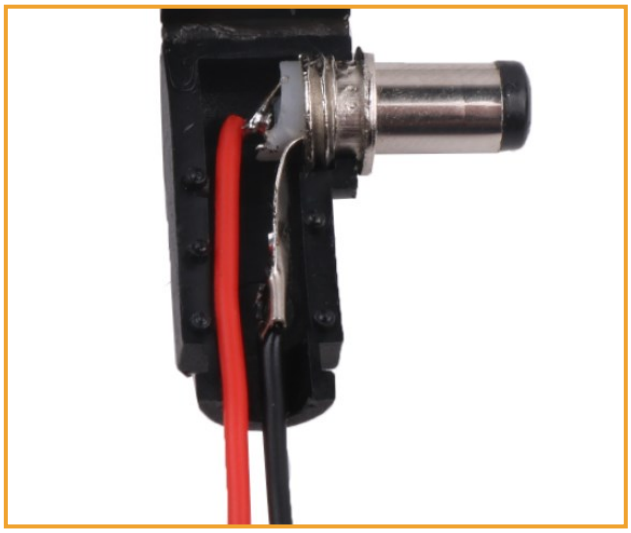


Figure 3-1-6 Schematic diagram of wire welding

3.1.2 load-bearing wheels、pedrail and motor installation

Step 1：Installation of load-bearing wheels

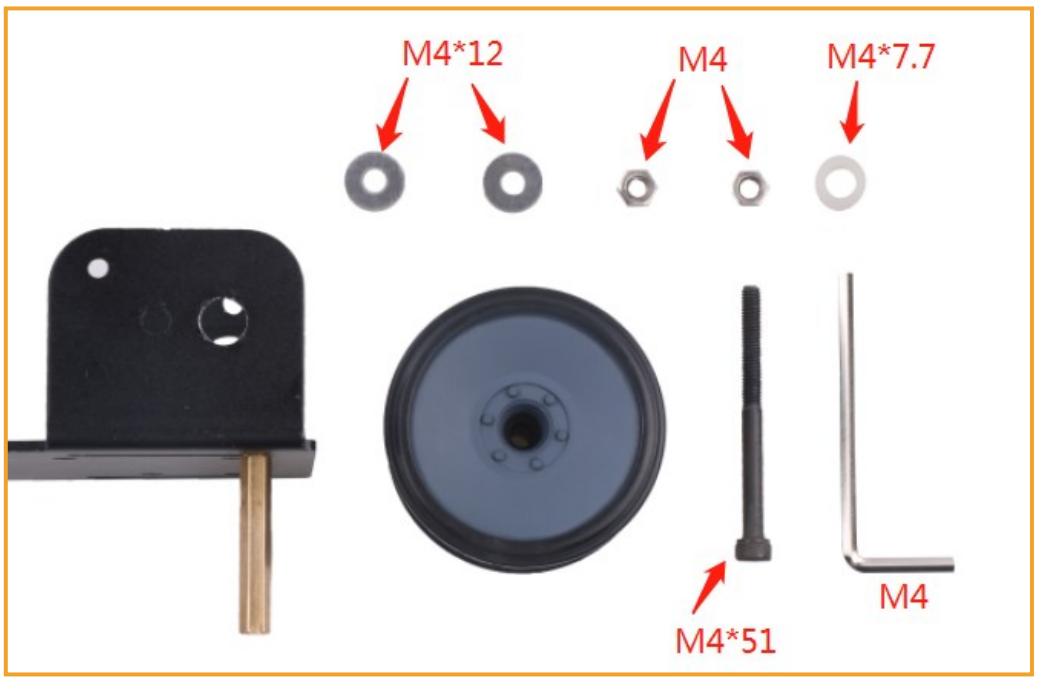


Figure 3-1-7 load-bearing wheels installation checklist

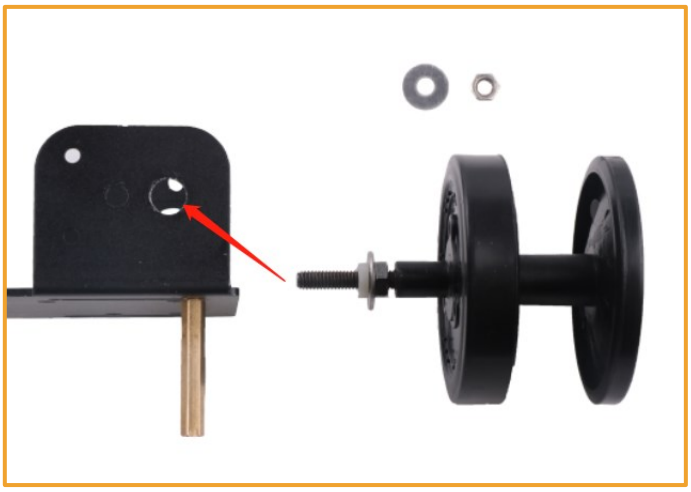
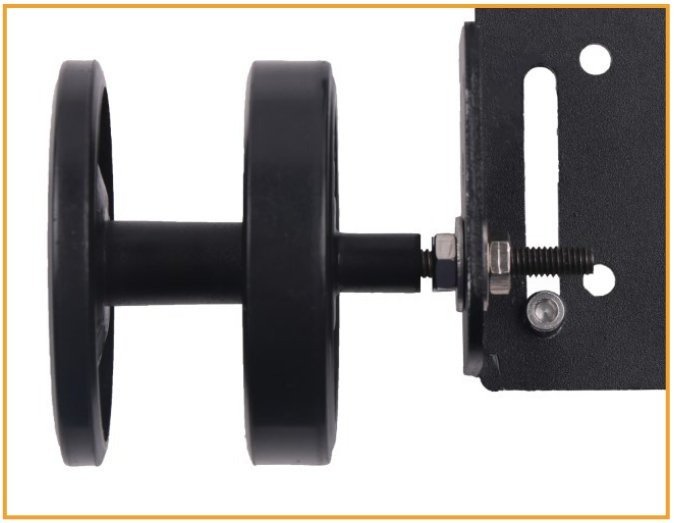


Figure 3-1-8 Schematic diagram of load-bearing wheels installation



Note: The load-bearing wheels must be reserved for 1mm from the inner nut.

Figure 3-1-9 Effect diagram of load-bearing wheels installation

Step 2: Install the motor

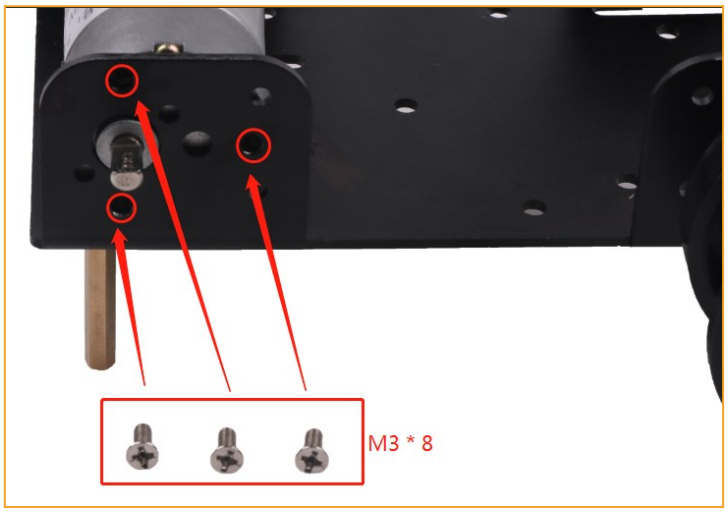


Figure 3-1-10 Schematic diagram of motor installation

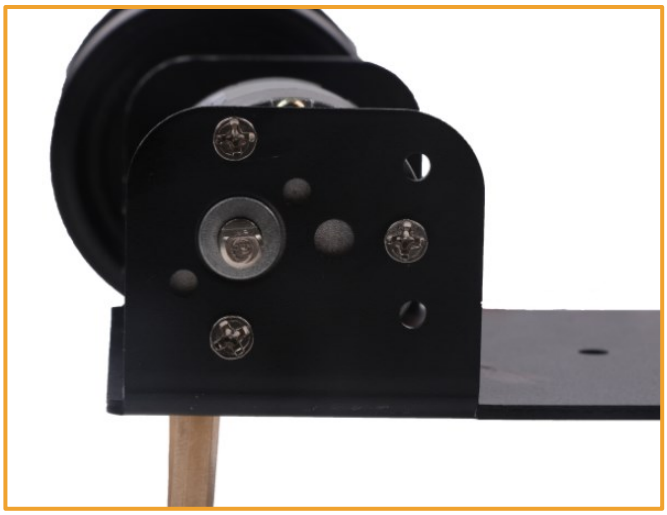


Figure 3-1-11 Effect diagram of motor installation

Step 3: Install the coupling

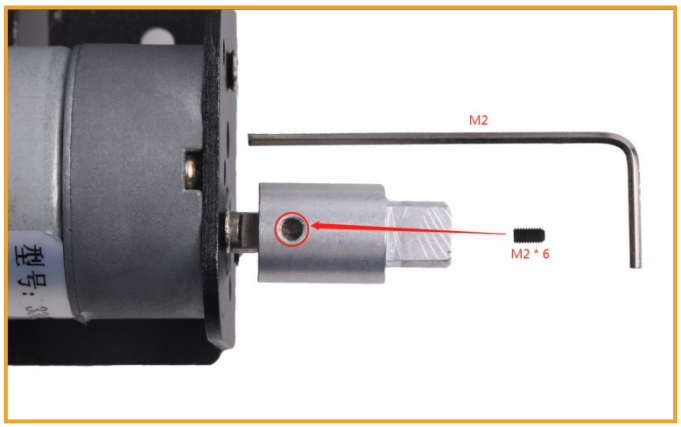
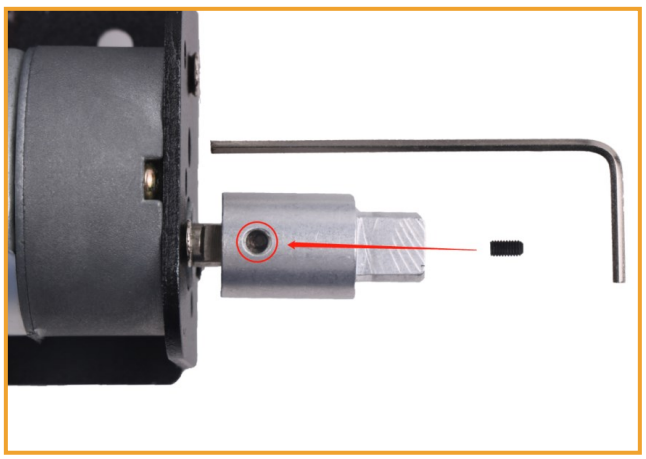


Figure 3-1-12 Schematic diagram of coupling installation

Firstly, insert the motor into the circular hole of the coupling, make sure that the hole of the coupling just reaches the flat position of the motor shaft, and screw the black top wire into the hole of the coupling as shown in Figure 3-1-13



Note: Do not move the coupling when twisting the black set screw

Figure 3-1-13

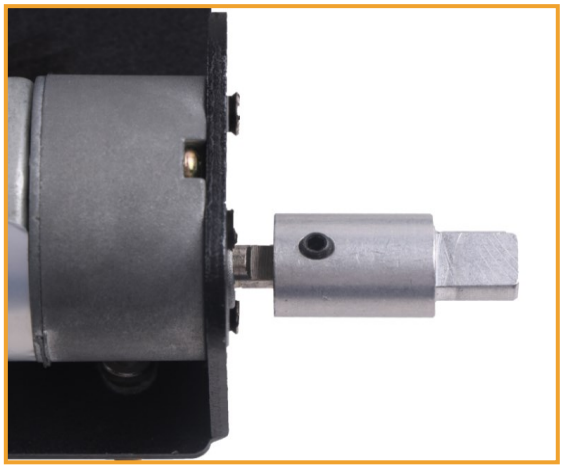


Figure 3-1-14 Effect diagram of coupling installation

Step 4: Install the drive wheel

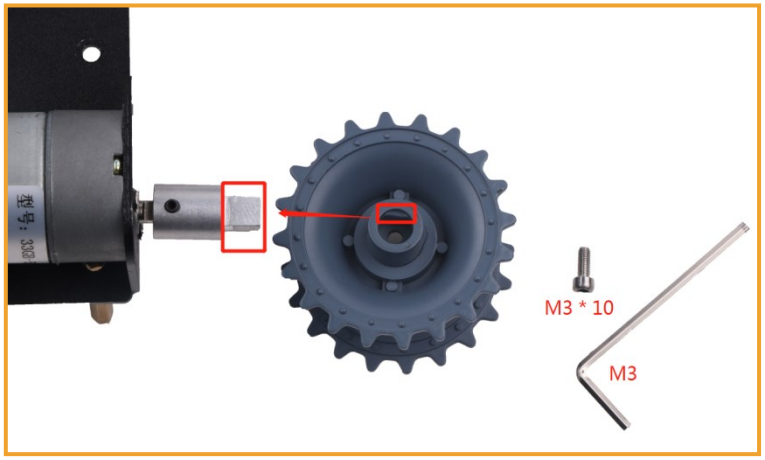


Figure 3-1-15 Schematic diagram of drive wheel installation

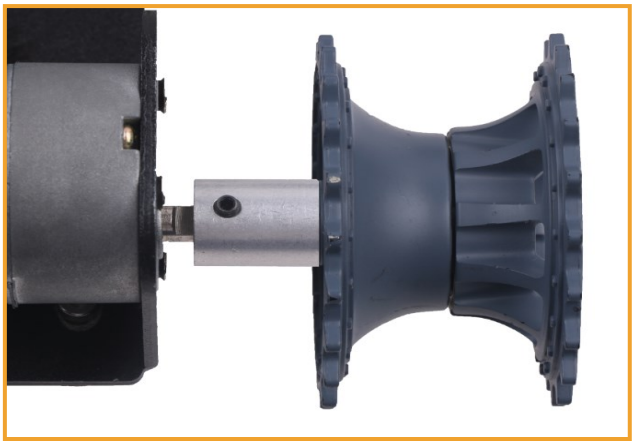


Figure 3-1-16 Effect diagram of drive wheel installation

Step 5: Install the pedrail

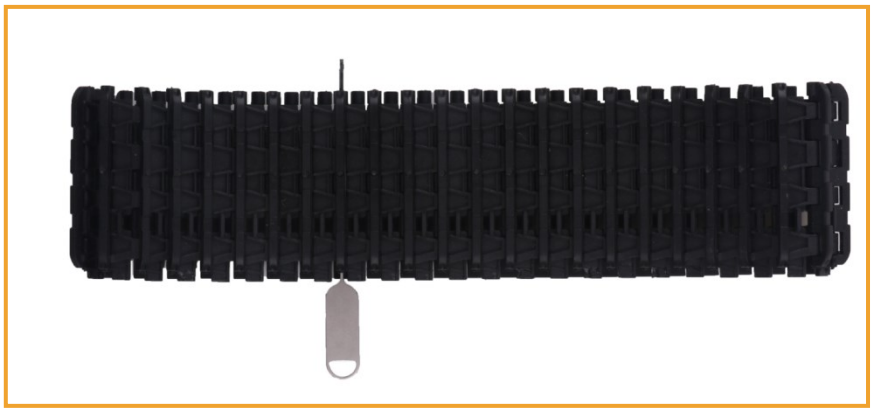
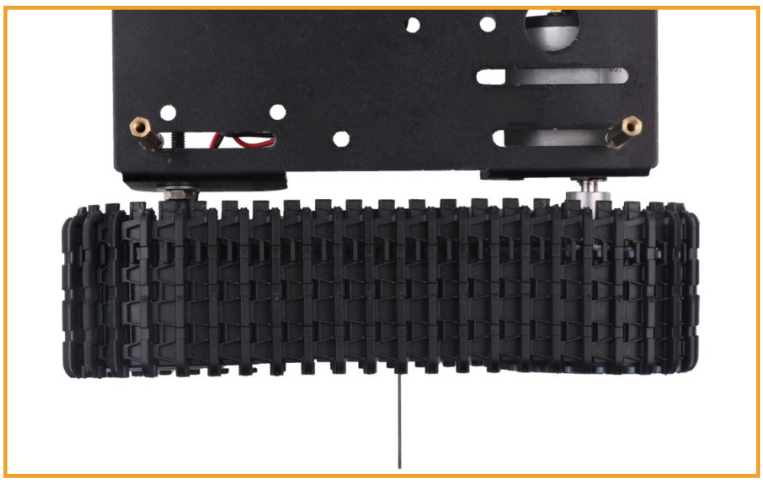


Figure 3-1-17 Schematic diagram of the pedrail disassembly



Note: Disassemble the proper size of the pedrai for installation

Figure 3-1-18 Schematic diagram of the pedrail installation

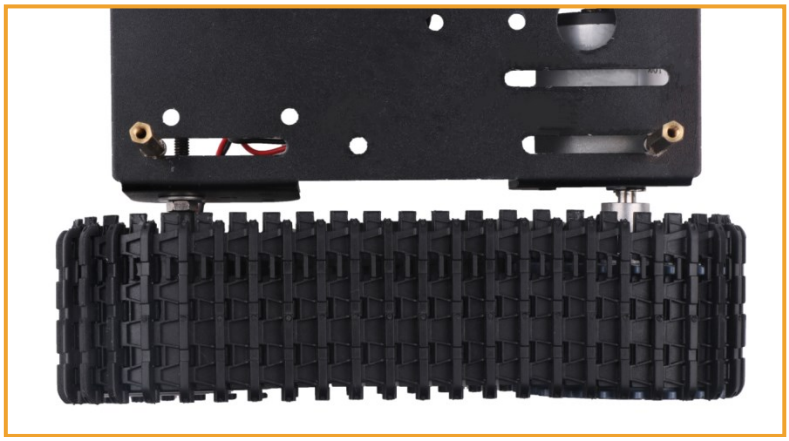


Figure 3-1-19 Effect diagram of the pedrail installation

3.1.3 Servo and UNO board installation

Step1: Install UNO board

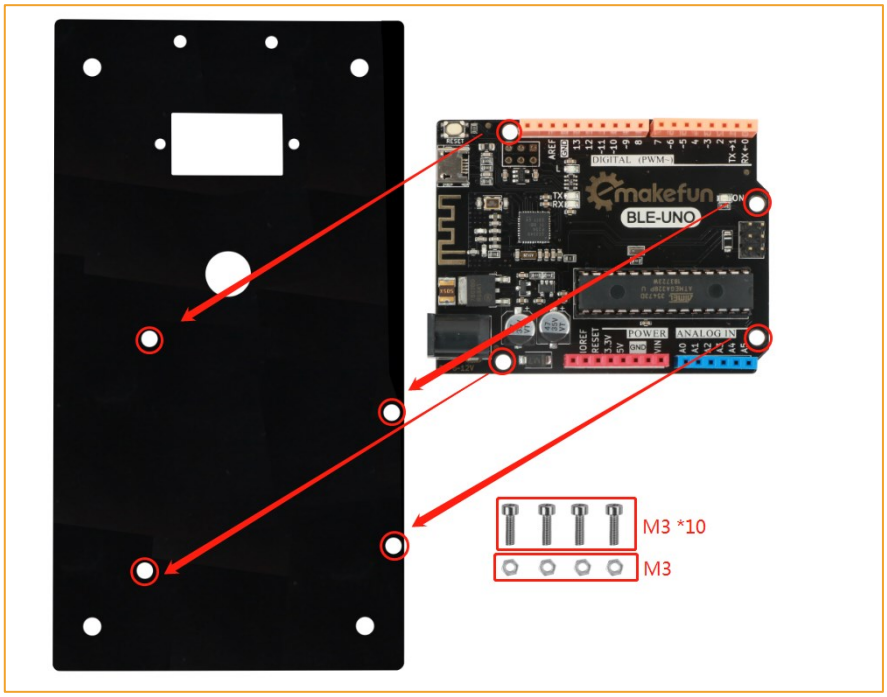


Figure 3-1-20 Schematic diagram of UNO board installation

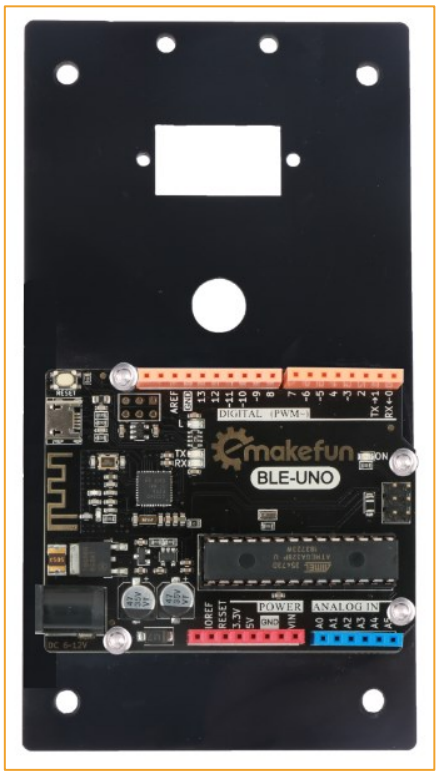


Figure 3-1-21 Effect diagram of UNO board installation

Step 2：Install servo

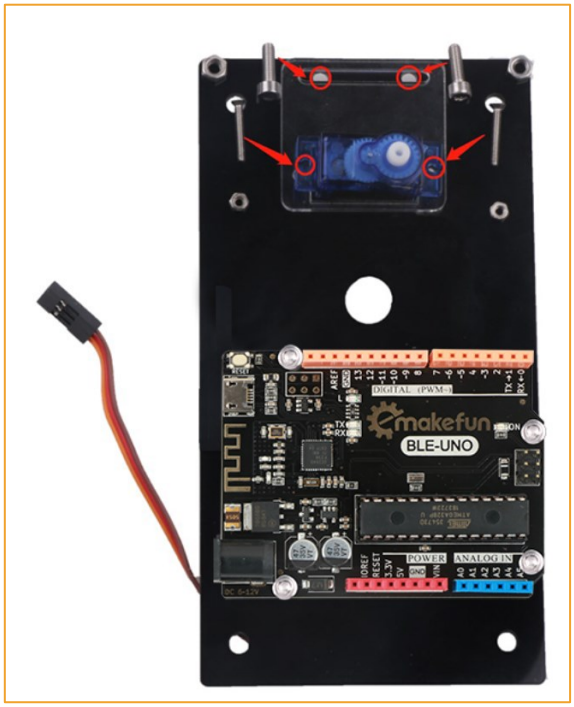


Figure 3-1-22 Schematic diagram of servo installation

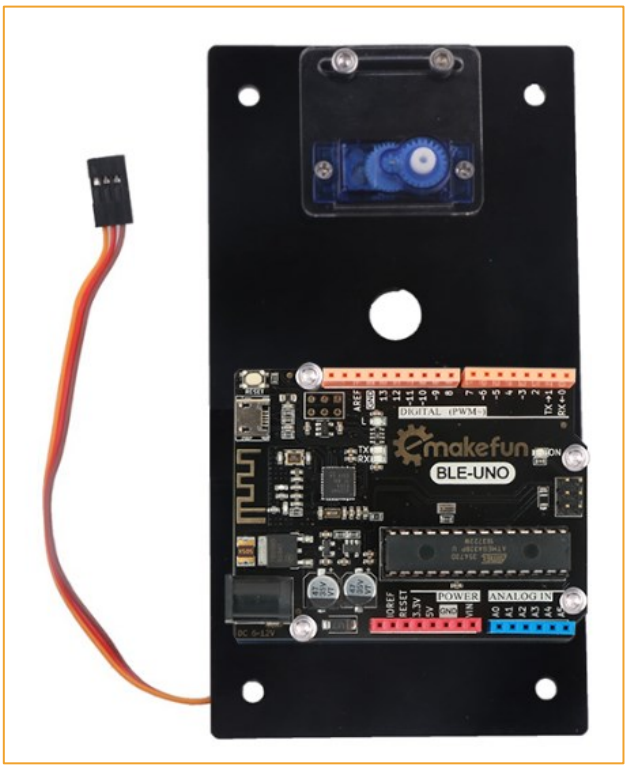


Figure 3-1-23 Effect diagram of servo installation

In order to reduce the angle adjustment of the rear steering gear, we need to download the following program ("Lesson\Module_Test\Servo_Test\Servo_Test.ino") to the control panel. The three wires of the servo are in turn the signal lines. (orange), power cord (red), ground (brown), then connect the servo signal cable (orange) to the I2C servo 1, and install the steering pad without first fixing the screws. See Figure 3-1-24. Show.

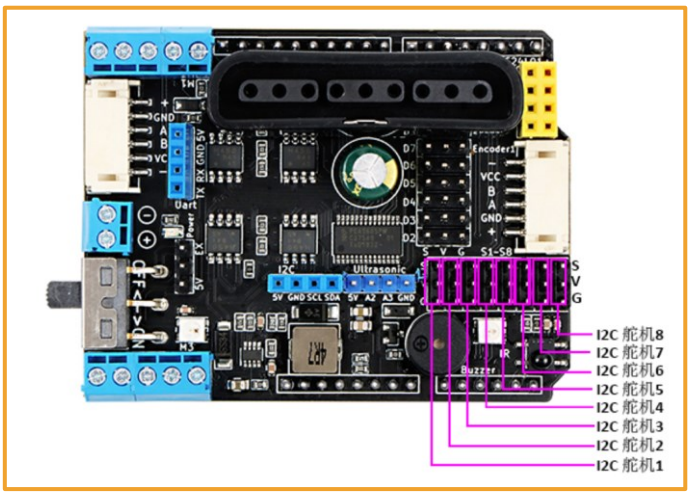


Figure. 3-1-24

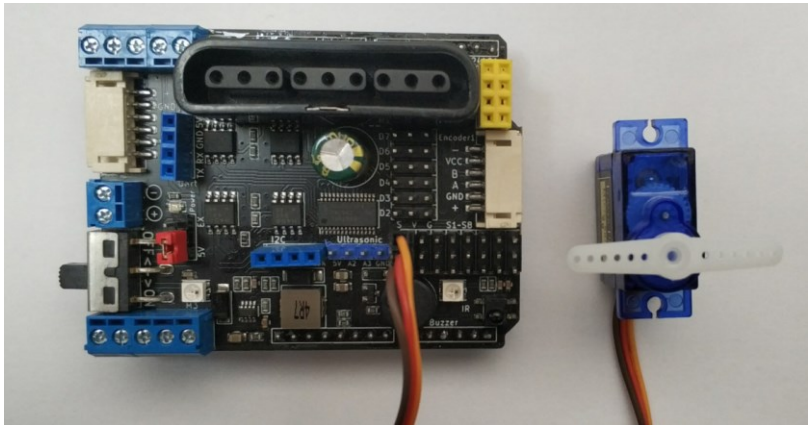


Figure 3-1-24 Steering gear connection diagram

After burning the program to the UNO board, do not unplug the USB cable and plug the battery into the UNO board to power it. Then open the serial monitor of the Arduino IED, as shown in Figure 3-1-25.

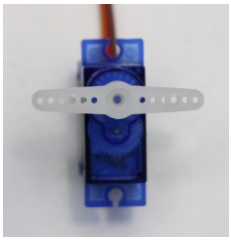


Figure 3-1-27 Steering gear calibration diagram

```
#include<Arduino.h>
#include<Wire.h>
#include "Emakefun_MotorDriver.h"

Emakefun_MotorDriver mMotorDriver = Emakefun_MotorDriver(0x60, MOTOR_DRIVER_BOARD_V5);
Emakefun_Servo *mServo1 = mMotorDriver.getServo(1);

char inByte = 0; //Serial port to receive data
int angle = 0;  //Angle value
String temp = "";//Temporary character variables, or use it for the cache

void setup()
{
  Serial.begin(9600);
  mMotorDriver.begin(50);
}

void loop()
{
  while (Serial.available() > 0) //Determine whether the serial data
  {
    inByte = Serial.read();//Read data, the serial port can only read 1 character
    temp += inByte;//The characters read into temporary variables inside the cache,
    //Continue to determine the serial port there is no data, know all the data read out
  }
  //Determine whether the temporary variable is empty
  if (temp != "") {
    angle = temp.toInt();    //Convert variable string type to integer
    Serial.print("Servo degree: ");
    Serial.println(angle);  //Output data to the serial port for observation
    mServo1->writeServo(angle);  //Control the servo to rotate the corresponding angle.
  }
  temp = "";//Please see temporary variables
  delay(100);//Delayed 100 milliseconds
}
```

Step 3: Install the ultrasonic bracket

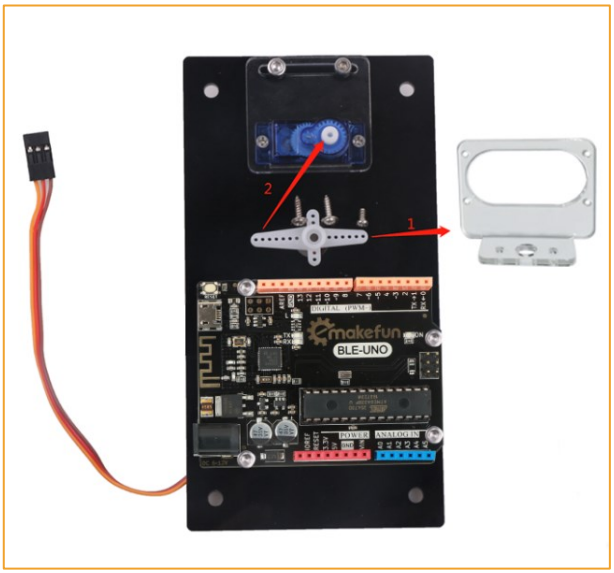


Figure 3-1-28 Schematic diagram of ultrasonic bracket installation Figure 3-1-29

First fix the rudder paddle and the ultrasonic bracket, as shown in Figure 3-1-30.

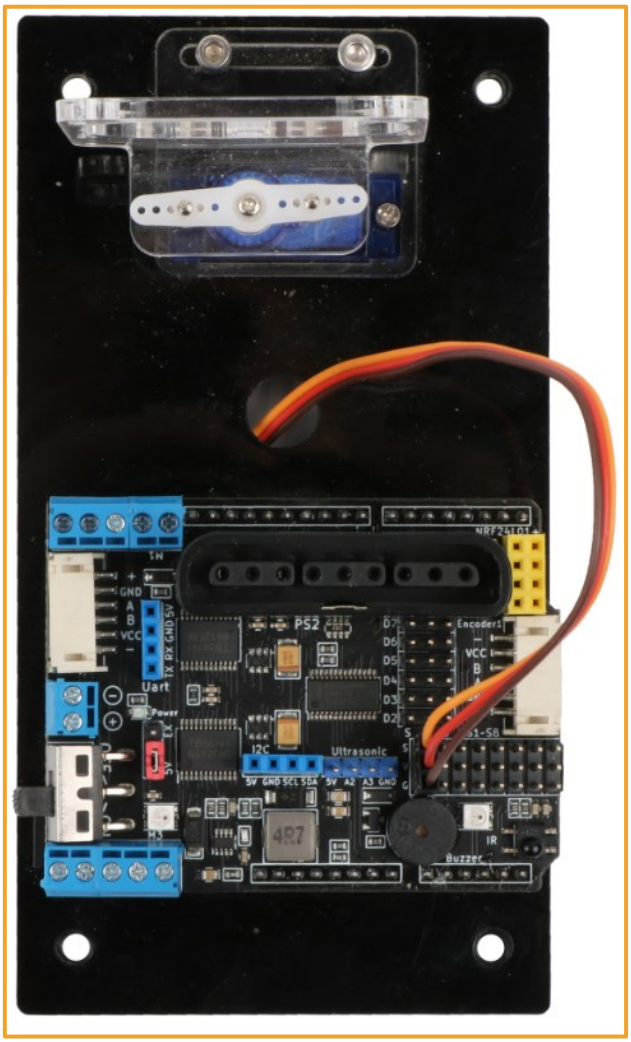


Figure 3-1-30 Effect diagram of ultrasonic bracket installation

Step 4: Ultrasonic module and uno expansion board installation

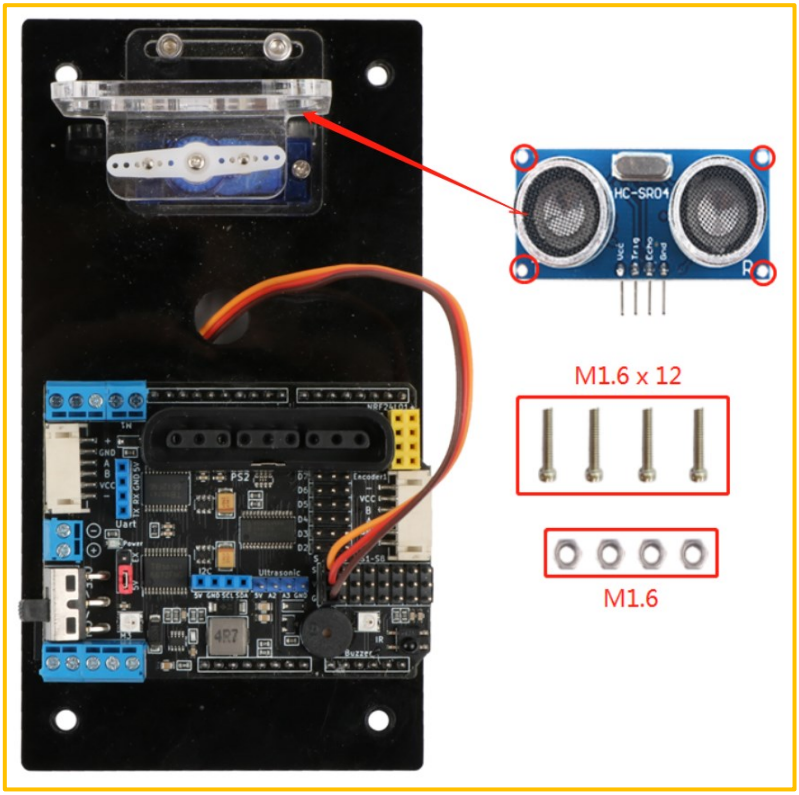


Figure 3-1-31 Schematic diagram of ultrasonic module and uno expansion board installation

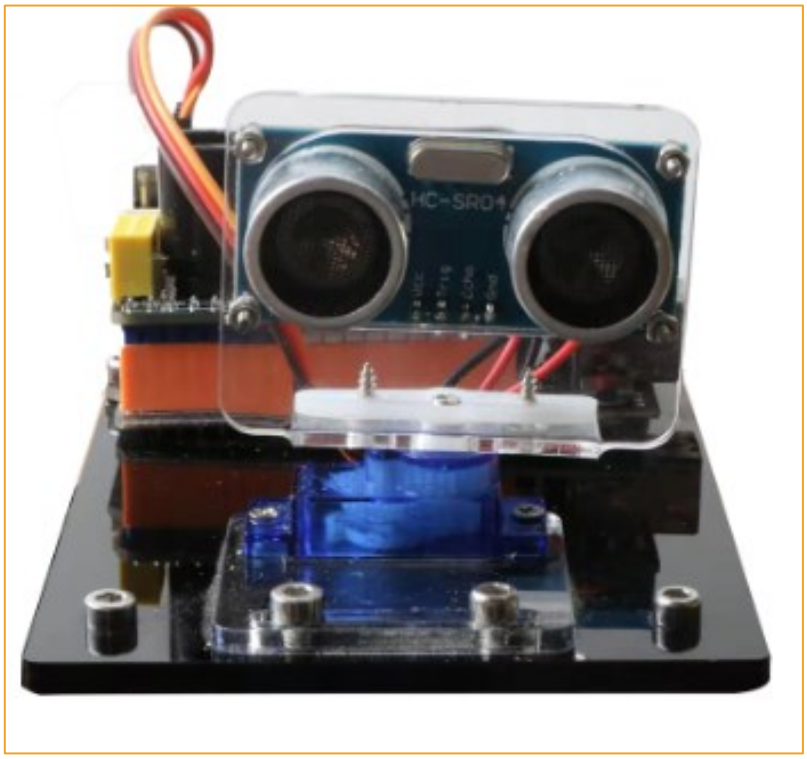


Figure 3-1-32 Effect diagram of ultrasonic module and uno expansion board installation

3.1.4 Upper acrylic floor mounting

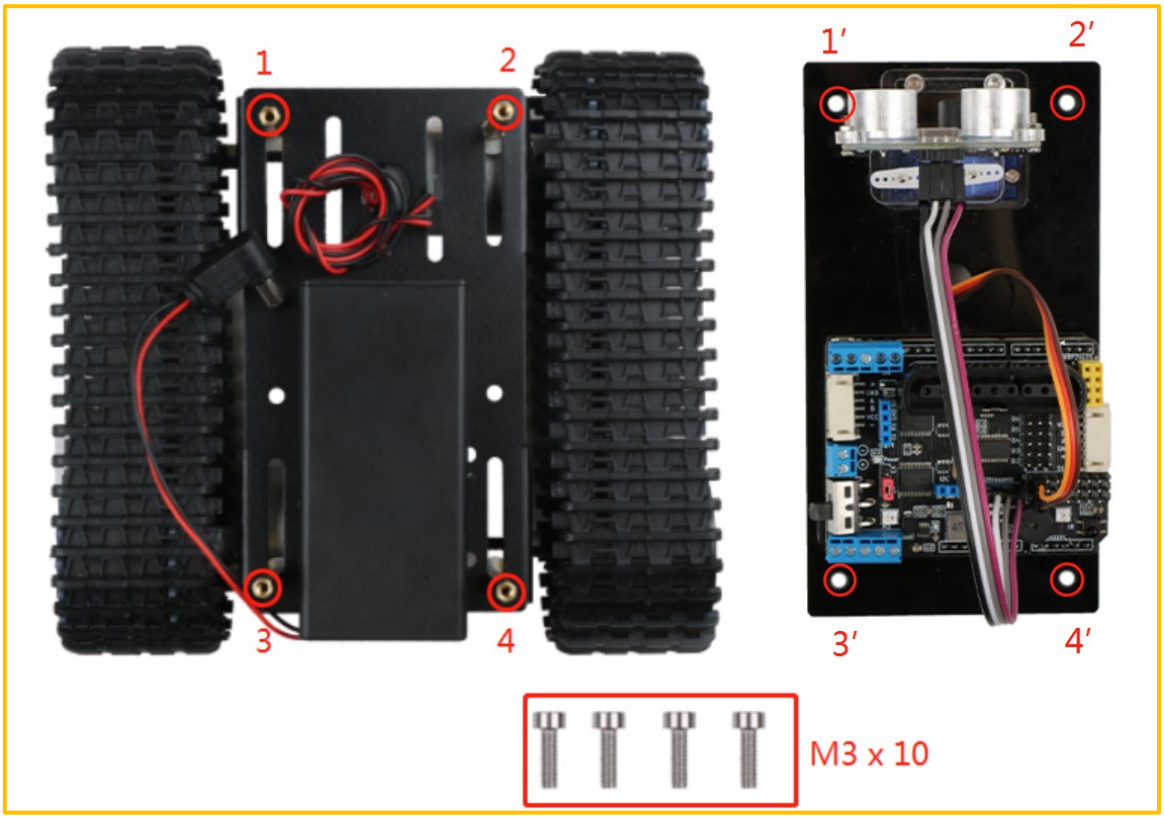
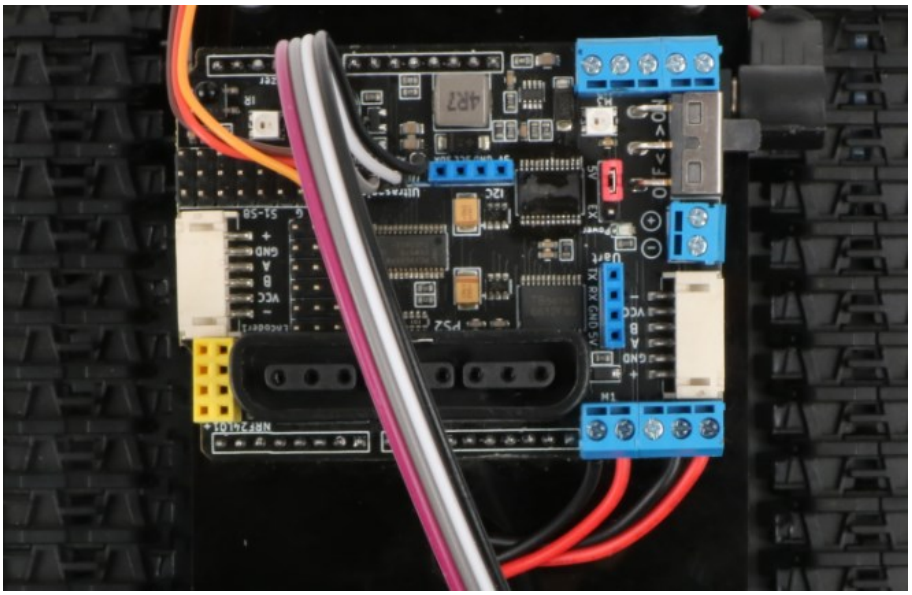


Figure 3-1-33 Schematic diagram of upper acrylic baseboard installation



Note: The left motor is connected to the M1 binding post, the right motor is connected to the M2 binding post, and the steering gear is connected to the IIC steering gear.

Figure 3-1-34 Motor wiring

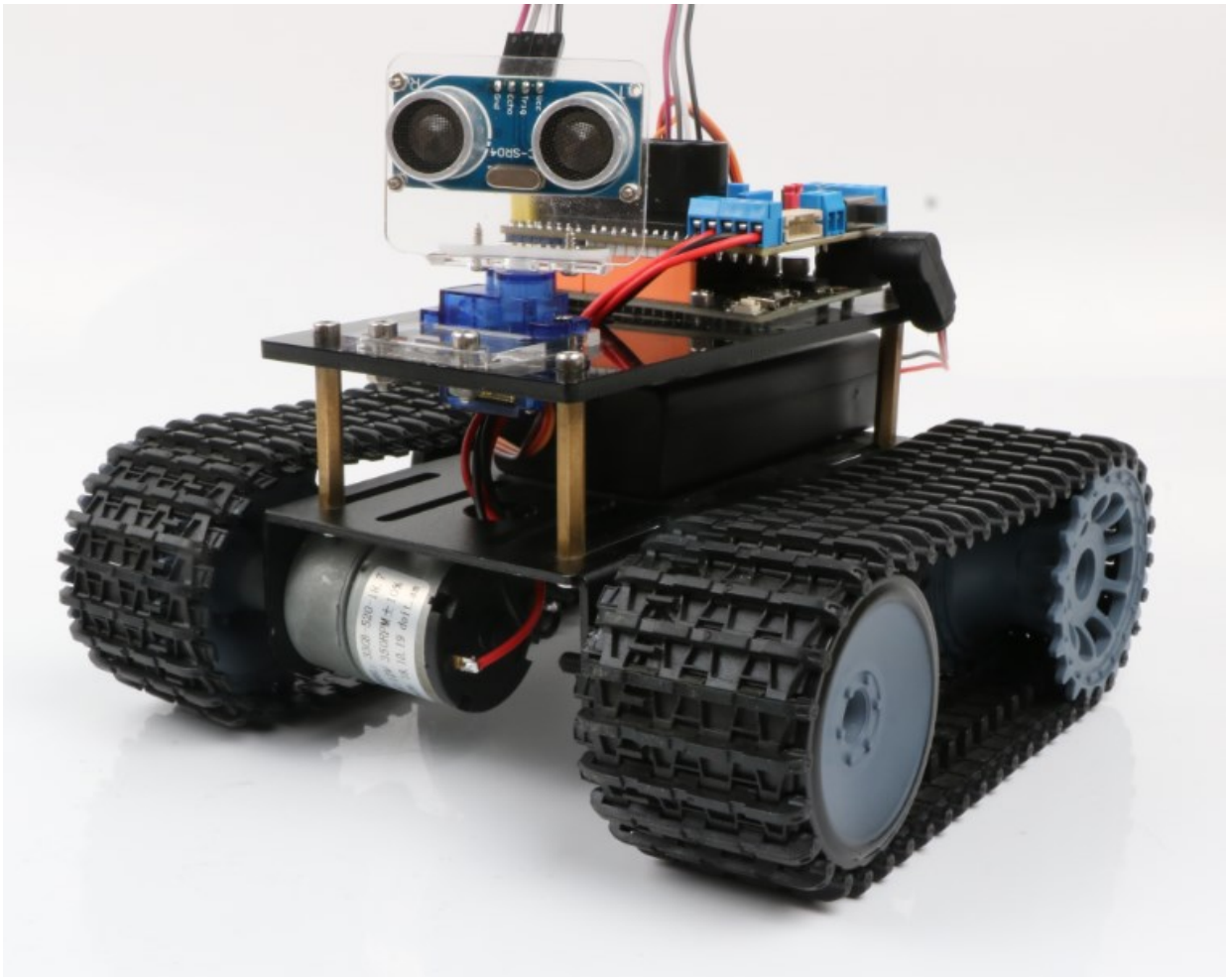


Figure 3-1-34 Effect diagram of upper acrylic baseboard installation

Charging instructions: As shown in Figure 3-1-3, please turn the battery switch on after plugging in the charger.

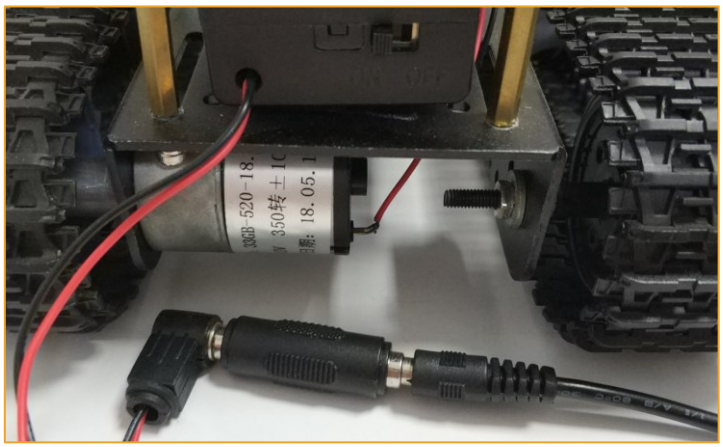


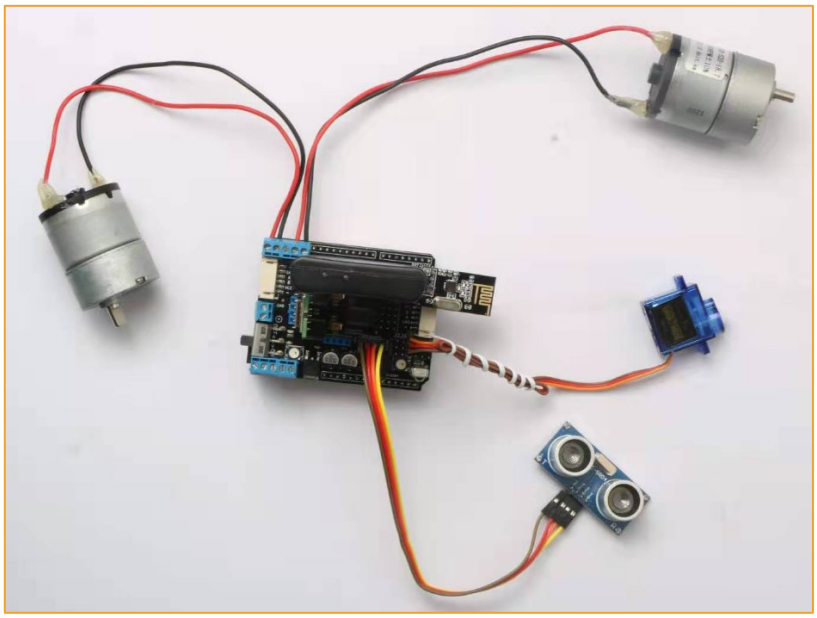
Figure 3-1-35

So far, the basic assembly of the tank has been completed. We believe you have some basic knowledge of your car's structure, function and some modules through a short period of time, then you can achieve the corresponding functions only by downloading the program to the development board, each function has a

corresponding program in CD, so please enjoy playing. However, if you can read the program and write your own program, there will be more fun, now let's go to the software section!

Wiring diagram:

The motor on the left side of the tank is connected to M1, and the motor on the right side is connected to M2. Steering gear connected to I2C servo 1



Chapter 4 Experiment

4.1 Servo

4.1.1 Servo Introduction

The steering gear is also called servo motor. It was first used to realize its steering function on the ship. Because it can continuously control its rotation angle through the program, it is widely used to realize the steering and the various joint movements of the robot. The steering gear is the control of the steering of the trolley. The mechanism has the characteristics of small size, large torque, simple external mechanical design and high stability. Whether in hardware design or software design, the steering gear design is an important part of the car control. Generally speaking, the steering gear is mainly composed of the following The components are composed of steering wheel, reduction gear set, position feedback potentiometer, DC motor, control circuit, etc., as shown in Figure 4-1-1. The tank uses a 180 degree SG90 (180 degrees) 9g steering gear.

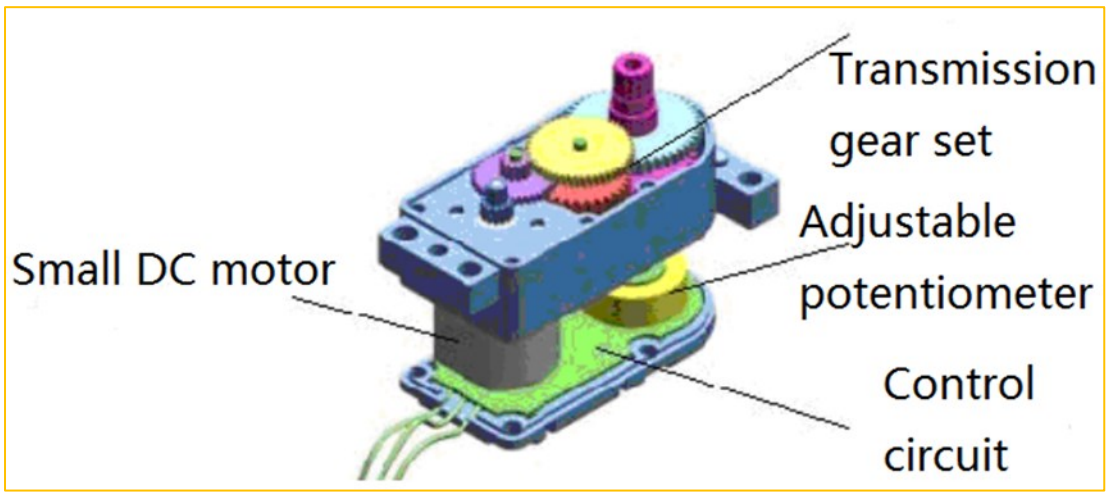


Figure 4-1-1 Schematic diagram of the steering gear

4.1.2 Working principle

The control signal enters the signal modulation chip by the receiver channel, gets the DC bias voltage. The steering gear has a reference circuit which generates a reference signal with a period of 20ms and a width of 1.5ms. Comparing the obtained DC bias voltage with the voltage of the potentiometer and obtaining the output voltage difference. Finally, the positive and negative output voltage difference in the motor driver chip decide the positive and negative rotation of motor. When the speed of motor is certain, the cascade reducer gear will drive potentiometer to rotate so that the voltage difference is reducing to 0, the motor will stop rotating.

When the control circuit receives the control signal, the motor will rotate and drive a series of gear sets, the signal will move to the output steering wheel when the motor decelerates. The the output shaft of

steering gear is connected with the position feedback potentiometer, the potentiometer will output a voltage signal to the control circuit board to feedback when the steering gear rotates, then the control circuit board decides the rotation direction and speed of the motor according to the position, so as to achieve the goal. The working process is as follows: control signal→control circuit board→motor rotation→gear sets deceleration →steering wheel rotation→position feedback potentiometer→control circuit board feedback.

The control signal is 20MS pulse width modulation (PWM), in which the pulse width varies linearly from 0.5-2.5MS, the corresponding steering wheel position varies from 0-180 degrees, which means the output shaft will maintain certain corresponding degrees if providing the steering gear with certain pulse width. No matter how the external torque changes, it only changes position until a signal with different is provided as shown in Fig.3.2.4.4. The steering gear has an internal reference circuit which can produce reference signal with 20MS period and 1.5MS width, there is a comparator which can detect the magnitude and direction of the external signal and the reference signal, thereby produce the motor rotation signal.

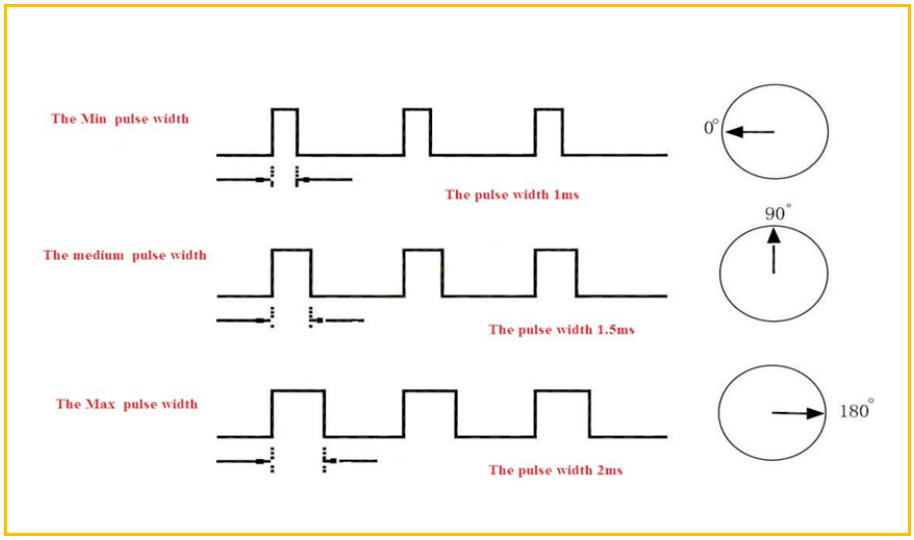


Figure 4-1-5 Relationship between the Motor Output Angle and Input Pulse

4.1.3 steering test test

The PS2X&Motor Driver Board_V5.0 driver board can drive the servo, the servo pin position on the driver board (red pin), with ground pin (GND), power pin (VCC) and signal pin (S), three The pins are respectively connected to the corresponding pins of the servo. The servo can be controlled by I2C communication. The position of the servo test program is**"courseware code\Lesson\Module_Test\Servo_Test\Servo_Test.ino"**. The foot definition is shown in Figure 4-1-6.

```
Emakefun_MotorDriver mMotorDriver =Emakefun_MotorDriver(0x60,MOTOR_DRIVER_BOARD_V5);
```

// Initialization library, 0x60 is the I2C address of PCA9685, MOTOR_DRIVER_BOARD_V5 is MotorDrvierBoar_V5, if your MotorDriverBoard is V4, please change MOTOR_DRIVER_BOARD_V5 to MOTOR_DRIVER_BOARD_V4.

```
Emakefun_Servo *mServo1 = mMotorDriver.getServo(1);
```

//Initialize I2C servo, 1-8 are I2C servo interface.

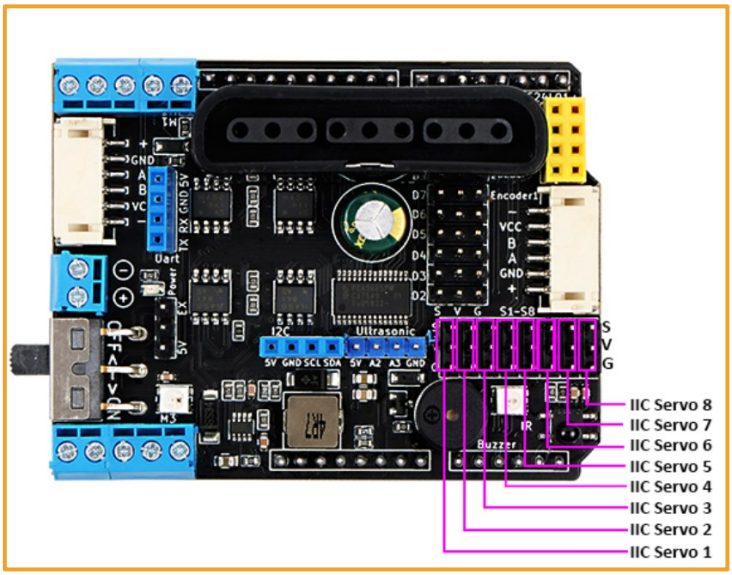


Figure 4-1-6 Steering gear pin definition

4.2 RGB WS2812B Experiment

4.2.1 RGB WS2812B Description

The WS2812B is 3 output channels special for LED driver circuit. Its internal includes intelligent digital port data latch and signal reshaping amplification drive circuit. Also includes a precision internal oscillator and a 12V voltage programmable constant current output drive. In the purpose of reduce power supply ripple, the 3 output channels designed to delay turn-on function.

Unlike the traditional RGB, WS281B **is integrated with a WS281B LED drive control special chip, which requires a single signal line to control a LED lamp or multiple LED modules. Features as below:**

- Output port compression 12V.
- Built-in voltage-regulator tube, only a resistance needed to add to IC VDD feet when under 24V power supply.
- 256 Gray-scale adjustable and scan frequency is more than 2KHz.
- Built in signal reshaping circuit, to ensure waveform distortion do not accumulate after wave reshaping to the next driver
- Built-in electrify reset circuit and power-down reset circuit.
- Cascading port transmission signal by single line
- Any two point the distance less than 5 Meters transmission signal without any increase circuit.
- When the refresh rate is 30fps, the cascade number is at least 1024 pixels.
- Send data at speed of 800Kbps.

For more parameters of WS2812,please check the file in CD:"Document\WS281B.pdf"

4.2.2 RGB WS281B working principle

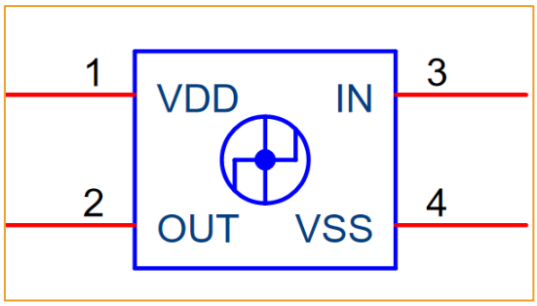
IC uses single NZR communication mode. After the chip power-on reset, the DIN port receive data from controlle,the first IC collect initial 24bit data then sent to the internal data latch,the other data which reshaping by the internal signal reshaping amplification circuit sent to the next cascade IC through the DO port. After transmission for each chip, the signal to reduce 24bit. IC adopt auto reshaping transmit technology,making the chip cascade number is not limited the signal transmission,only depend on the speed of signal transmission.

The data latch of IC depend on the received 24bit data produce different duty ratio signal at OUTR,OUTG,OUTB port.All chip synchronous send the received data to each segment when the DIN port input a reset signal. It will receive new data again after the reset signal finished. Before a new reset signal received, the control signal of OUTR ,OUTG, OUTB port unchanged. IC sent PWM data that received justly to OUTR, OUTG, OUTB port, after receive a low voltage reset signal the time retain over 280μs.

Pin function and Pin configuration as picture 4-2-1

NO	Symbol	Function description
1	VDD	Power supply voltage
2	OUT	Data signal cascade output
3	IN	Data signal input
4	VSS	Ground

WS281B Pin function



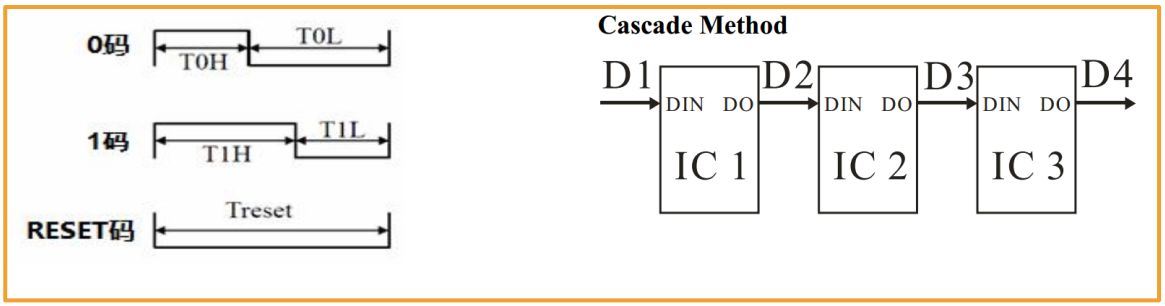
Picture 4-2-1 WS281B Pin configuration

4.2.3 WS281B drive principle

WS281B The low level is represented by T0, consists of a 0.5μs high level and 2μs low level. The high level is represented by T1, consists of a 2μs high level 0.5μs low level. When low level last more than 50μs there will be a reset signal.

T0H	0 code, high voltage time	0.5μs
T1H	1 code, high voltage time	2μs
T0L	0 code, low voltage time	2μs
T1L	1 code, low voltage time	0.5μs
RES	Frame unit, low voltage time	>50μs

Sequence Chart



Picture 4-2-2Waveform sequence diagram

Composition of 24bit Data:

R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

Note: Data transmit in order of GRB, high bit data at first.

4.2.4 RGB WS281B Test Program

Open the supporting information "Lesson\Module_Test\RGB_Test\RGB_Test.ino ",
The first parameter sets the RGB sequence number, and the second parameter hexadecimal color code:

```
rgbled.setColor(index, RGB_RED);
```

The first parameter is set to illuminate the RGB sequence number. The next three parameters are Red, Green, and Blue. The three color value components are in the range of 0 to 255. A 24-bit value is synthesized.

```
rgbled.setColor(index, R, G, B);
```

4.3 Passive Buzzer

3.2.2.1 Description

The buzzer is an integrated electronic alarm device that uses DC voltage to power electronic products for sound devices.Buzzers are mainly divided into active buzzer and passive buzzer.The main difference between the two type device is as follow :

The ideal signal for active buzzer operation is direct current,usually labeled at VDC, VDD, etc. There is a simple oscillating circuit inside the buzzer. As long as it is energized, it can motivate the molybdenum plate to vibrate. But some active buzzer can work under specific AC signal, while it has high requires of AC signal voltage and frequency, this situation is very rarely .

The passive buzzer does not include internal oscillation circuit, the working signal is a certain frequency of pulse signal .If we give DC signal, the passive buzzer will not work, because the magnetic circuit is constant, the molybdenum sheet cannot vibrate. They are just as shown as follow picture 4-3-1:

Active Buzzer Passive Buzzer

Picturec 4-3-1 Ative Buzzer and Passive Buzzer

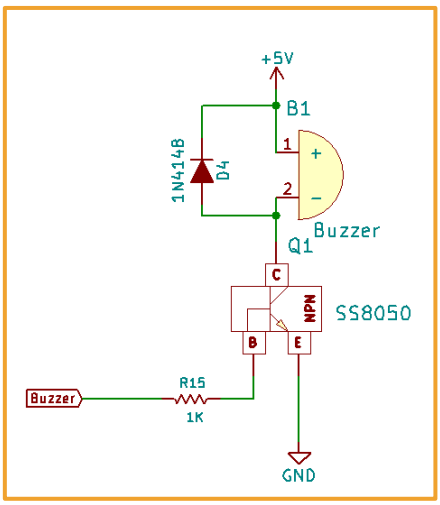
4.3.2 Buzzer working principle

The passive buzzer generates music mainly through the I/O port of the single-chip microcomputer to output different pulse signals of different levels to control the buzzer pronunciation.

For example,if Arduino use12MHzs crystal oscillator, to produce middle tune “Re” sound, it needs 587Hzs audio pulse frequency output .The audio signal pulse cycle is T=1/587=1703.5775us, half cycle time is 852us,the pulse timer needs always count at =852us/1us=852,when it count at 852,the I/O port will reverse the direction, then it get the “Re” sounds in C major scale.

In addition to that, the passive buzzer sound principle is the current through the electromagnetic coil, making the electromagnetic coil generated magnetic field to drive the vibration film audible. Therefore,a

certain amount of current is required to drive it, while the ARDUINOI/O pin outputs a lower voltage. Arduino output level can hardly drive the buzzer, so it needs to add an amplifier circuit. And transistor S8050 is used here as an amplification circuit. As shown in picture 4-3-2 is the onboard buzzer schematic diagram:



Picture 4-3-2 Schematic diagram of onboard buzzer

4.3.3 Alarm Test

Experimental purposes:

Make the buzzer simulate the alarm sound

Experimental principle:

The sound starts with the frequency increasing from 200HZ to 800HZ, then stops for a period of time from 800HZ to 200HZ, and loops experimental code location.

"Lesson\Module_Test\Buzzer_Test\AlarmSound\AlarmSound.ino"

```
void setup()
{
    pinMode(A0,OUTPUT);
}
void loop()
{
    for(int i = 200; i <= 800; i++)   // 200HZ ~ 800HZ
    {
        tone(A0,i);
    }
    delay(1000);                  //Max Frequency hold 1s
    for(int i= 800; i >= 200; i--)   // 800HZ ~ 200HZ
    {
        tone(A0,i);
        delay(10);
    }
}
```

Firstly,we use a simple procedure to understand how to use the buzzer, and its sound principle. And to drive a buzzer like singing sound,we need make the buzzer issued frequency and duration of the different sound . Cycle is equal to the reciprocal of the frequency, so you can know the time by frequency, and then by calling the delay function or timer to achieve.Similarly the sound duration can also be achieved through the delay function. So the key factor to make the buzzer sing is to know how much time to prolong! Play music with Arduino, you just need to understand the two concepts of "tone" and "beat" .

The tone means the frequency at which a note should be sung.

The beat means how long a note should be sung.

The commonly used method is "look-up table method", this method is complex where you have to find the corresponding frequency of each note (according to the note, the frequency comparison), and then according to the formula converted to the corresponding time (take half cycle), and then through the delay function implementation. Finally by programming to achieve.

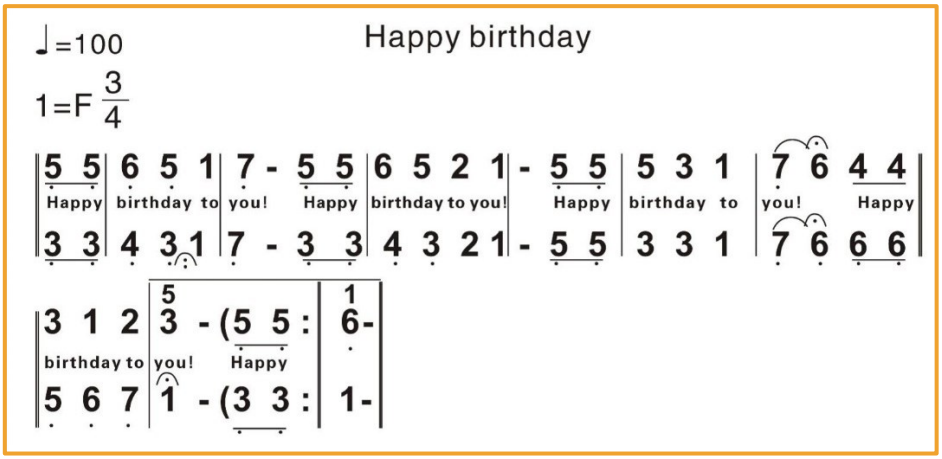
The whole process is like this:

Firstly, according to the score of Happy Birthday song, convert each tone to the corresponding frequency.

For example: picture 4-3-3 is the note frequency conversion table, picture 4-3-4 is the Happy Birthday song score.

	Musical notes	Corresponding frequency (Hz)	Half cycle(us)
Bass	1	261.63	1911.13
	1.5	277.18	1803.86
	2	293.66	1702.62
	2.5	311.13	1607.06
	3	329.63	1516.86
	4	349.23	1431.73
	4.5	369.99	1351.37
	5	392.00	1275.53
	5.5	415.30	1203.94
	6	440.00	1136.36
	6.5	446.16	1120.66
	7	493.88	1012.38
Alto	1	523.25	955.56
	1.5	554.37	901.93
	2	587.33	851.31
	2.5	622.25	803.53
	3	659.26	758.43
	4	698.46	715.86
	4.5	739.99	675.69
	5	783.99	637.76
	5.5	830.61	601.97
	6	880.00	568.18
	6.5	932.33	536.29
	7	987.77	506.19
Treble	1	1046.50	477.78
	1.5	1108.73	450.97
	2	1174.66	425.66
	2.5	1244.51	401.77
	3	1318.51	379.22
	4	1396.91	357.93
	4.5	1479.98	337.84
	5	1567.98	318.88
	5.5	1661.22	300.98
	6	1760.00	284.09
	6.5	1864.66	268.15
	7	1975.53	253.10

Picture 4-3-3 The note frequency conversion table



Picture 4-3-4 The score of Happy birthday song

Here to understand a little knowledge of the score, look at the above score, the score below the score is a bit of bass, no point is normal, the number is a bit high, and the bass of 5 is 4.5, the treble is 5.5, other notes are also The corresponding reason.

At the top left of the score, there is "1 = F", and the general score is Ç, which is "1 = C". Note that the 1234567 in the score corresponds to ABCDEFG. It is CDEFGAB! Therefore, if the rule here is Ƒ, then it means that 2 will sing G, 3 will sing A, ... 7 will sing E, so the bass 5 here should correspond to the bass 1.5!!!! The so-called corresponding left shift or right shift, if you still don't understand, look at the following:

1 The original corresponding should be C, 4 should correspond to F and then 1 corresponds to F, which is equivalent to 4, then 1.5 corresponds to 4.5, 2 corresponds to 5.

In this case, the bass 5 is actually 4.5, so the half cycle is 1803 μs.

As for why it is based on the half-cycle, it is because the microcontroller is set by the loop to the port connected to the buzzer, reset to make the sound, so it is a half cycle. Because I use a passive buzzer, the active buzzer is full cycle.

Then follow the above principle, one by one, and use the delay function to achieve, because each note has a different conversion frequency, or use multiple delay functions to achieve accurate pitch frequency one by one, but this is too annoying, and the microcontroller itself It is not specifically for singing. We shouldn't be embarrassed about them, so we will just be able to do it. Therefore, the delay function has a similar frequency in order to adapt to each tone. This is calculated by itself, and different songs have different values, so this is a troublesome problem.

After we know the frequency of the tone, the next step is to control the playing time of the note. Each note will play for a certain period of time, in order to form a beautiful piece of music, instead of a blunt tone, all the notes are played in one brain, and the note rhythm is divided into one beat, half beat, quarter beat. For one-eighth shot, we specify a time for a note to be 1; a half beat is 0.5; a 1/4 beat is 0.25; a 1/8 beat is 0.125...

First, ordinary notes. Take 1 shot.

Second, underlined notes indicate 0.5 beats; two underscores are quarter beats (0.25)

Third, some notes are followed by a dot, which means that 0.5 more shots are added, that is, 1 + 0.5.

Fourth, some notes are followed by a "-", which means that 1 more shot is added, that is, 1 + 1.

So we can give each note a play like this, and the music becomes. Basic music knowledge is all right, and many don't understand it. After all, it is a music idiot. So I understand that. As for the conversion of the beat to the frequency, there is also a corresponding table, just follow table two:

Music beat	1/4 beat delay time	Music	1/8 beat delay time
4/4	125ms	4/4	62ms
3/4	187ms	3/4	94ms
2/4	250ms	2/4	125ms

Table 2: Beat and frequency correspondence table

It is also achieved through the delay function,of course there will be errors. The idea of programming is very simple, firstly convert the note frequency and the time you want to sing into the two arrays. Then in the main programming, through the delay function to reach the corresponding frequency . sing it over, stop for a while, and then sing it, all the conversion is complete, we get the following frequency (Table 3) and beat:

Do 262	Re 294	Mi 330	Fa 349	Sol 392	La 440	Si 494	Do_h 523
Si_h 988	Mi_h 659		La_h 880	Sol_h 784	Fa_h698	Re_h 587	

Table 3: Happy Birthday Song beat table

According to the music score, we can get the frequency of the birthday song：

Sol, Sol, La, Sol, Do_h, Si, Sol, Sol, La, Sol, Re_h, Do_h, Sol, Sol, Sol_h, Mi_h, Do_h, Si, La, Fa_h, Fa_h , Mi_h, Do_h, Re_h, Do_hfloat

The beat is as follows:

0.5, 0.5, 1, 1, 1, 1+1, 0.5, 0.5, 1, 1, 1, 1+1, 0.5, 0.5, 1, 1, 1, 1, 1, 0.5, 0.5, 1, 1, 1, 1+1,

Add beats and frequency to the program and download it to Arduino to play.

Happy Birthday music score beat, view table two rhythm and frequency corresponding table 1 beats time is 187*4 = 748ms

Note: The procedure is shown in the:

“Lesson\Module_Test\Buzzer_Test\Happy_Birthday\Happy_Birthday.ino”

```
#define BUZZER_PIN 9 //buzzer pin 9
#define RGB A1     //RGB pin A1
#define Do 262
#define Re 294
#define Mi 330
#define Fa 349
#define Sol 392
```

```
#define La 440
#define Si 494
#define Do_h 523
#define Re_h 587
#define Mi_h 659
#define Fa_h 698
#define Sol_h 784
#define La_h 880
#define Si_h 988
#include "RGBLed.h"
RGBLed rgbled_A3(RGB, 2);
int length;
// happy birthday Music score
int scale[] = {Sol, Sol, La, Sol, Do_h, Si, Sol, Sol,
             La, Sol, Re_h, Do_h, Sol, Sol, Sol_h, Mi_h,
             Do_h, Si, La, Fa_h, Fa_h, Mi_h, Do_h, Re_h, Do_h
            };
// Beats time
float durt[] ={0.5, 0.5, 1, 1, 1, 1 + 1, 0.5, 0.5,
              1, 1, 1, 1 + 1, 0.5, 0.5, 1, 1,
              1, 1, 1, 0.5, 0.5, 1, 1, 1, 1 + 1
              };
void setup()
{
  pinMode(BUZZER_PIN, OUTPUT);
  // get scale length
  length = sizeof(scale) / sizeof(scale[0]);
  Serial.begin(9600);
}
void loop()
{
  for (int x = 0; x < length; x++) {
    tone(BUZZER_PIN, scale[x]);
    rgbled_A3.setColor(1, scale[x] - 425, scale[x] - 500, scale[x] - 95);
    rgbled_A3.setColor(2, scale[x] - 425, scale[x] - 500, scale[x] - 95);
    rgbled_A3.show();
    // 1= 3/4F so one Beats is  187*4 = 748ms
    delay(748 * durt[x]);
    noTone(BUZZER_PIN);
  }
  delay(3000);
}
```

4.4 DC motor drive principle

4.4.1 Motor control principle

The PS2X&Motor Driver Board uses the PCA9685 to output the PWM control motor driver chip H450. We will now briefly introduce the two chips.

The main parameters of PCA9685 are as follows:

- IIC interface control can control 16-channel PWM support up to 16 servos or PWM output, 12-bit resolution per channel (4096 levels)
- Built-in 25MHz crystal oscillator, can be connected to external crystal oscillator, can also be connected to external crystal oscillator, up to 50MHz
- Support 2.3V-5.5V voltage, maximum withstand voltage 5.5V
- With power-on reset, software reset and other functions

The device address of PCA9685 is determined by pins A0, A1, A2, A3, A4, A5, and the pin cannot be left floating. Since there are 6 pins that jointly determine the device address, there can be 64 device addresses due to The IC is powered on to retain the LED All Call address (E0h, 1110 000) and Software Reset address (06h, 0000 0110). There are only 62 available device addresses. Therefore, in theory, one I2C interface can control the channel 16*. 62=992 PWM, the pin control device address is shown in the figure below:

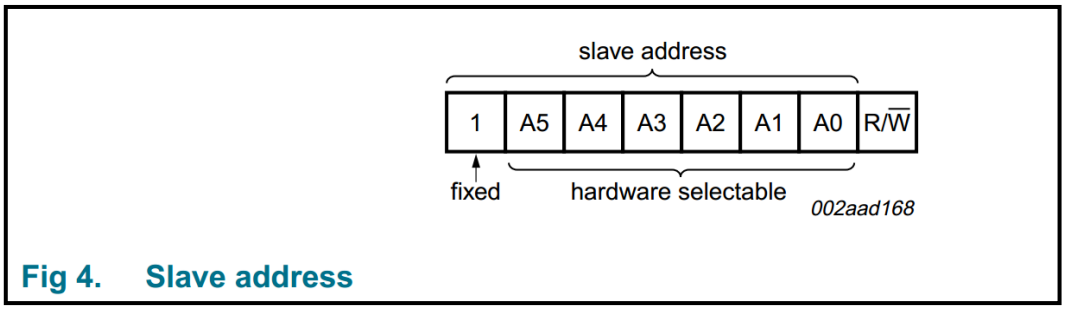


Figure 4-4-1 Schematic diagram of device address

For detailed use of the chip, please refer to "**Document\PCA9685.pdf**"

Motor drive chip introduction

PS2X & Motor Driver Board_V5.0 uses four high-current driver chips to drive the motor. It is a PWM chopper DC brush motor driver. One channel of the motor output block is embedded, manufactured using BiCD process, rated output voltage 50V, maximum current 3A, built-in output MOSFET with low on-resistance (high side + low side = 0.6Ω (typical)), multiple built-in Error detection functions (thermal shutdown (TSD), overcurrent detection (ISD), and undervoltage lockout (UVLO)).

IN1	IN2	OUT1	OUT2	Mode
L	L	OFF (Hi-Z)	OFF (Hi-Z)	Stop
				Standby mode after 1 ms
H	L	H	L	Forward
L	H	L	H	Reverse
H	H	L	L	Brake

Figure 4-4-2 Function description diagram

4.4.2 Motor Test Procedure

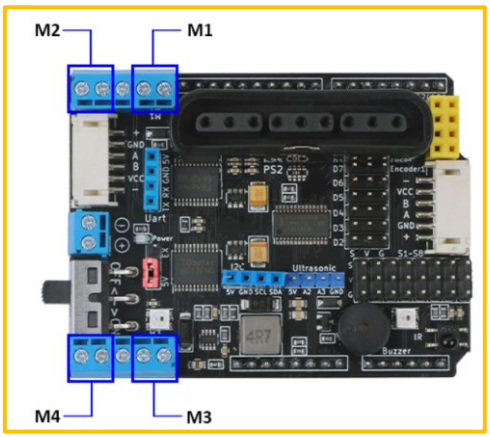


Figure 4-4-3

As shown in Figure 4-4-3, I only used the M1 and M2 motor interfaces in the car. So just use the motor 1, 2 in the program. Num represents the motor serial number

```
Emakefun_MotorDriver mMotorDriver =Emakefun_MotorDriver(0x60,MOTOR_DRIVER_BOARD_V5);
```

// Initialization library, 0x60 is the I2C address of PCA9685, MOTOR_DRIVER_BOARD_V5 is MotorDrvierBoar_V5, if your MotorDriverBoard is V4, please change MOTOR_DRIVER_BOARD_V5 to MOTOR_DRIVER_BOARD_V4.

```
Emakefun_DCMotor *DCMotor_1 = mMotorDriver. getMotor(num);
```

// Initialize the motor, num represents the motor label (M1, M2, M3, M4).

```
DCMotor_1->setSpeed(200)                    // Set the motor speed. The speed is 0-255.
DCMotor_1->run(FORWARD);          // Set the motor status, FORWARD forward, BACKWARD reverse, BRAKE stop.
```

In the supporting information, we provide a small motor test program (the file name is "Lesson\Module_Test\Motor_Test\Motor_Test.ino"), of course you can also write the program yourself, very simple, download the program After the Arduino, we can see the motor: one second forward -> one second back -> one second turn right -> one second left -> one second stop, this cycle has been repeated, indicating that the motor can work normally. Then just mix the other modules together and you can easily control the car.

The test procedure is as follows:

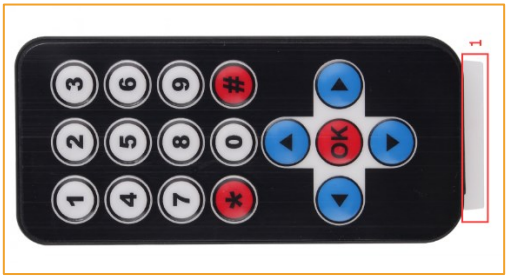
```
#include<Arduino.h>
#include<Wire.h>
#include "Emakefun_MotorDriver.h"
Emakefun_MotorDriver mMotorDriver = Emakefun_MotorDriver(0x60,MOTOR_DRIVER_BOARD_V5);
Emakefun_DCMotor *DCMotor_1 = mMotorDriver.getMotor(1);
Emakefun_DCMotor *DCMotor_2 = mMotorDriver.getMotor(2);
void setup()
{
  Serial.begin(9600);
  mMotorDriver.begin(50);
}
void loop()
{
  DCMotor_1->setSpeed(100);
  DCMotor_2->setSpeed(100);
  DCMotor_1->run(FORWARD);
  DCMotor_2->run(FORWARD);
  delay(1000);
  DCMotor_1->run(BACKWARD);
  DCMotor_2->run(BACKWARD);
  delay(1000);
  DCMotor_1->run(FORWARD);
  DCMotor_2->run(BACKWARD);
  delay(1000);
  DCMotor_1->run(BACKWARD);
  DCMotor_2->run(FORWARD);
  delay(1000);
  DCMotor_1->setSpeed(0);
  DCMotor_2->setSpeed(0);
  DCMotor_1->run(RELEASE);
  DCMotor_2->run(RELEASE);
  delay(1000);
}
```

4.5 Infrared Remote Control

4.5.1 Introduction

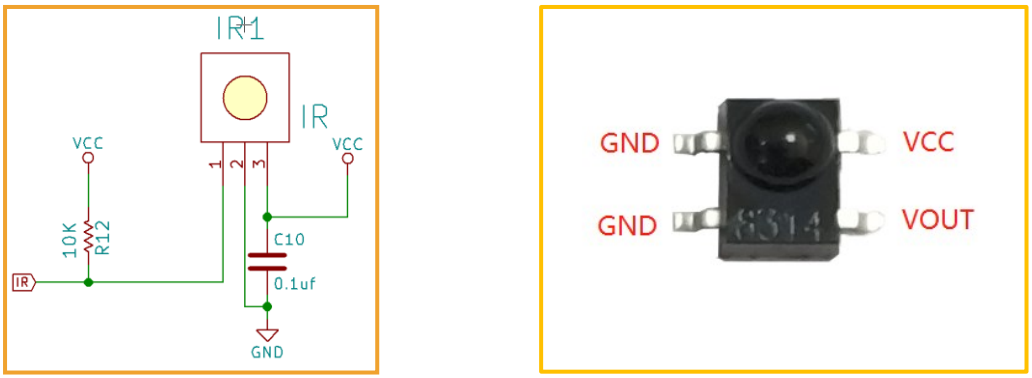
Infrared remote control is widely used in every field at present.Infrared wireless remote control consists of Mini ultra-thin infrared remote controller (physical map shown in the picture 4-5-1) and integrated

38KHz infrared receiver. Mini ultra-thin infrared remote controller has 17 function keys, and the launch distance is up to 8 meters. Suitable for indoor control of various devices.



Picture 4-5-1 Infrared remote Control physical map

In the " Panther-Tank " car, integrated IR receiver head has been added to the expansion board, just need to plug the expansion board into the Arduino, and in the program defined pins (8th number IO), the IR receiver head has three pins, including power supply feet, grounding and signal output feet. The circuit is shown in the picture 4-5-2. The ceramic capacitor 0.1uf is a decoupling capacitor, which filters out the interference from the output signal. The 1-terminal is the output of the demodulation signal, which is directly connected with the Arduino 8th of the single-chip microcomputer. When the infrared coded signal is emitted, the output of the square wave signal after the infrared joint is processed, and is provided directly to the Single-chip microcomputer, and the corresponding operation is carried out to achieve the purpose of controlling the motor.

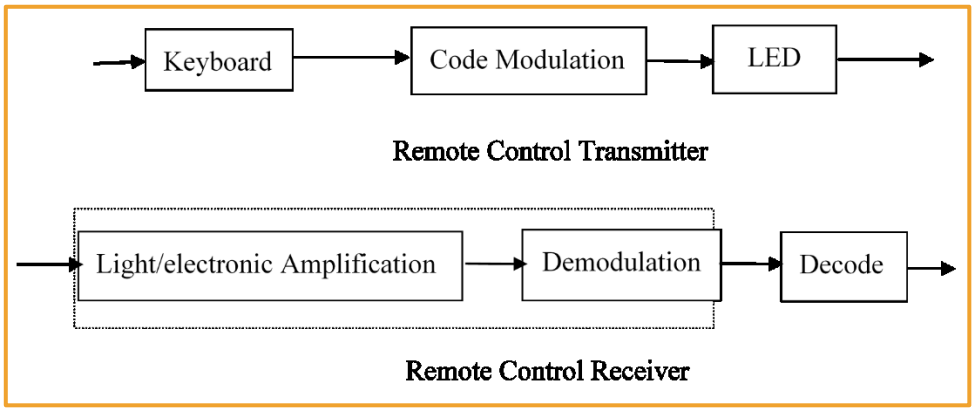


Picture 4-5-2 Infrared receiver Head circuit diagram and physical map

4.5.2 Working Principle

Remote control system is generally composed of remote control (transmitter), receiver, when you press any button on the remote control, the remote will produce the corresponding coded pulse, output a variety of infrared as the medium of control pulse signal, these pulses are computer instruction code, infrared monitoring diode monitoring to infrared signals, The signal is then sent to the amplifier and the limiter, which controls the pulse amplitude at a certain level, regardless of the distance between the IR transmitter and the receiver. The AC signal enters the bandpass filter, the bandpass filter can pass the load wave of 30KHZ to 60KHZ, through the demodulation circuit and the integral circuit to enter the comparator, the

comparator outputs the high and low level, restores the signal waveform of the transmitting end. As shown in the 4-5-3.

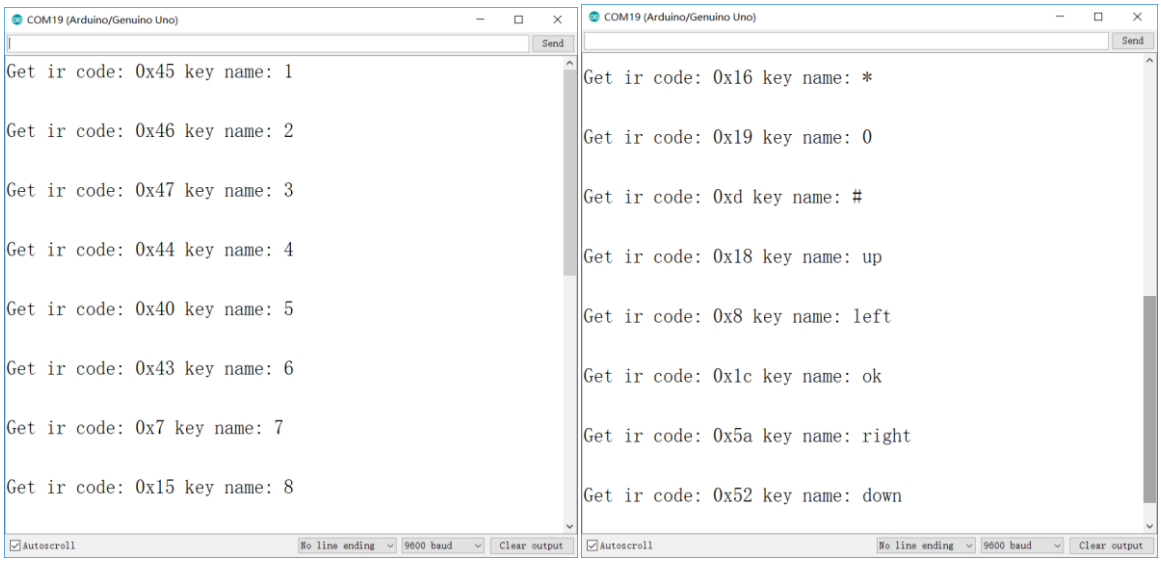


Picture 4-5-3 Infrared emitter and receiver system block diagram

4.5.3 Acquiring Infrared remote value

Open the program named "Lesson\Module_Test\IrkeyPressed_Test\IrkeyPressed_Test.ino" in the program is opened and downloaded to the development board, and the transparent plastic sheet labeled "1" in Figure 4-5-1 is removed. Then open the "serial monitor" and use the remote control to align the receiving head and press any button to observe the value displayed in the "serial monitor" and record it for later development and use, as shown in Figure 4-5-4.

```
#include "IRremote.h"
IRremote ir(12);
unsigned char keycode;
char str[128];
void setup() {
    Serial.begin(9600);
    ir.begin();
}
void loop()
{
    if (keycode = ir.getCode()) {
        String key_name = ir.getKeyMap(keycode);
        sprintf(str, "Get ir code: 0x%x key name: %s \n", keycode, (char
*)key_name.c_str());
        Serial.println(str);
    } else {
      //   Serial.println("no key");
    }
    delay(110);
}
```



Picture 4-5-4 Remote coded query

n Figure 4-5-4, we can see the values of Ir Code “0x45” and keyname “1”, where “0x45” is the code of a button on the remote control and “1” is the name of the button function of the remote control. The total encoding value of the supporting remote control is shown in Figure 4-5-4.

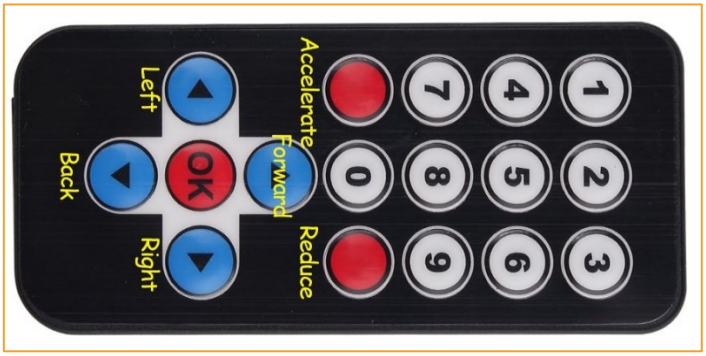


Figure 4-5-5 Infrared remote control instructions

4.6 Ultrasonic obstacle avoidance

4.6.1 Introduction to Ultrasound

In the "Panther-Tank" car, we use the HC-SR04 ultrasonic module to measure distance and obstacles. The module can provide 2cm-400cm non-contact distance sensing function, ranging accuracy up to 3mm; The temperature sensor corrects the ranging result and uses GPIO communication mode with a watchdog inside, which is stable and reliable. The module includes an ultrasonic transmitter, a receiver, and a control circuit. Figure 4-6-1 is a relatively common ultrasonic sensor.

Figure 4-6-1 Ultrasonic module physical map

4.6.2 Module parameter

- Working voltage: 4.5V~5.5V. In particular, it is absolutely not allowed to exceed 5.5V.
- Power consumption current: minimum 1mA, maximum 20mA
- Resonant frequency: 40KHz；
- Detection range: 4 mm to 4 m. Error: 4%；
- Dimensions: 48mm * 39mm * 22mm (H）
- Fixed hole size 3*Φ3mm spacing: 10mm3.2.4.2 working principle

4.6.3trasonic principle

The most commonly used method of ultrasonic distance measurement is echo detection method, the ultrasonic transmitter launches ultrasonic toward a direction and starting the time counter at the same time, the ultrasonic will reflect back immediately when encountering a blocking obstacle and stopping the counter immediately as soon as the reflected ultrasonic is received by the receiver. The working sequence diagram is shown in Fig 4-6-2 . The velocity of the ultrasonic in the air is 340m/s, we can calculate the distance between the transmitting position and the blocking obstacle according to the time t recorded by the time counters, that is: s=340*t/2.

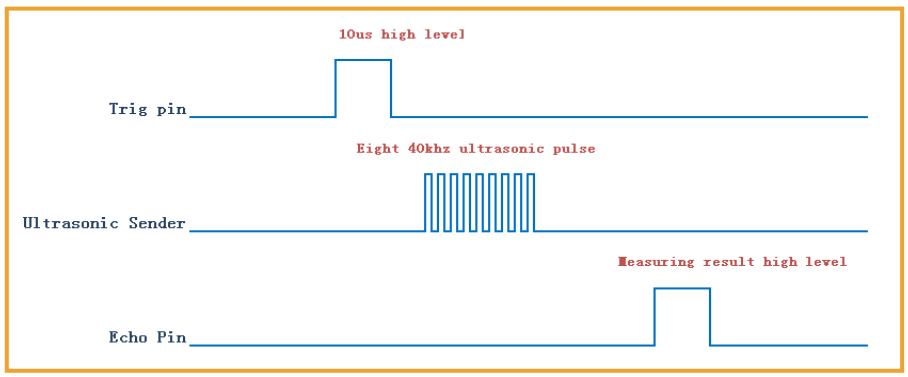


Figure 4-6-2 Ultrasonic working sequence diagram

Let us analyze the working sequence, first the trigger signal starts the HC-RS04 distance measurement module, which means the MCU sends an at least 10us high level to trigger the HC-RS04,the signal sent inside of the module is responded automatically by the module, so we do not have to manage it, the output

signal is what we need to pay attention to. The output high level of the signal is the transmitting and receiving time interval of the ultrasonic, which can be recorded with the time counter, and don't forget to divided it with 2.

The ultrasonic is a sound wave which will be influenced by temperature. If the temperature changes little, it can be approximately considered that the ultrasonic velocity is almost unchanged in the transmission process. If the required accuracy of measurement is very high, the measurement results should be to corrected with the temperature compensation. Once the velocity is determined, the distance can be obtained. This is the basic principle of ultrasonic distance measurement module which is shown in Fig 4-6-3:

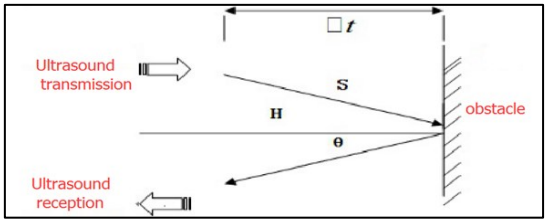


Figure 4-6-3 Principle of measuring the distance of ultrasonic waves

The ultrasonic is mainly divided into two parts, one is the transmitting circuit and the other is the receiving circuit, as shown in Fig 4-6-4. The transmitting circuit is mainly composed of by the inverter 74LS04 and ultrasonic transducer T40, the first 40kHz square wave from the Arduino port is transmitted through the reverser to the one electrode on the ultrasonic transducer, the second wave is transmitted to another electrode on ultrasonic transducer, this will enhance the ultrasonic emission intensity. The output end adopts two parallel inverters in order to improve the driving ability. the resistance R1 and R2 on the one hand can improve the drive ability of the 74LS04 outputting high level, on the other hand, it can increase the damping effect of the ultrasonic transducer and shorten the free oscillation time.

The receiving circuit is composed of the ultrasonic sensor, two-stage amplifier circuit and a PLL circuit. The reflected signal received by the ultrasonic sensor is very weak, which can be and amplified by the two-stage amplifier. PLL circuit will send the interrupt request to the microcontroller when receiving the signal with required frequency. The center frequency of internal VCO in the PLL LM567 is $f_0 = 1/(1.1R_P1C2)$, the locking bandwidth is associated with C3. Because the transmitted ultrasonic frequency is 40kHz, the center frequency of the PLL is 40kHz, which only respond to the frequency of the signal, so that the interference of other frequency signals can be avoided.

The ultrasonic sensor will send the received the signal to the two-stage amplifier, the amplified signal will be sent into the PLL for demodulation, if the frequency is 40kHz, then the 8 pins will send low level interrupt request signal to the microcontroller P3.3, the Arduino will stop the time counter when detecting low level.

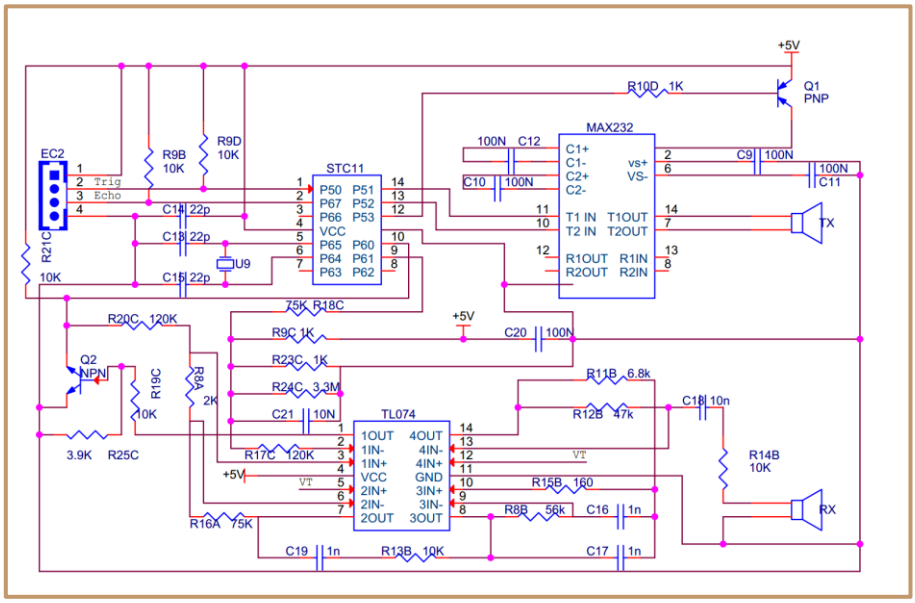
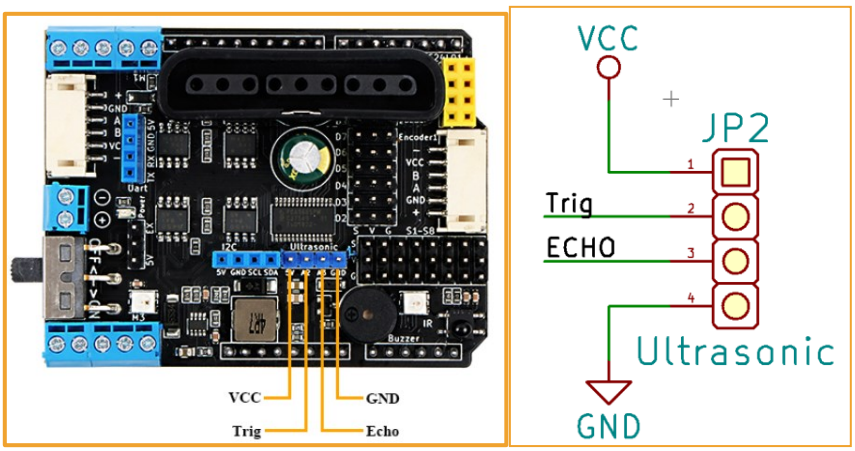


Figure 4-6-4 Schematic diagram of ultrasonic transmission and reception

4.6.4 Experimental procedure



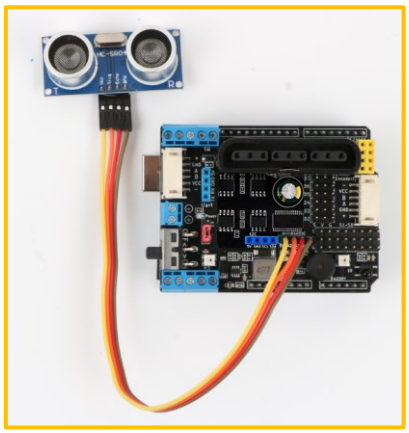


Figure 4-6-5 Ultrasonic wiring

Test program:

Program location:“Lesson\Module_Test\Ultrasonic_Test\Ultrasonic_Test.ino”

4.7 PS2 Wireless Control (optional)

4.7.1 Introduction to the kit

The PS2 handle is the remote control handle of the Sony game console. Sony's series of game consoles are very popular all over the world. I don't know when someone got the idea of the PS2 controller and cracked the communication protocol, so that the handle can be connected to other devices for remote control, such as remote control of the familiar four-wheelers and robots. The outstanding feature is that this handle is now very cost-effective. The buttons are rich and easy to expand to other applications, as shown in Figure 4-7-1 is the commonly used PS2 wireless controller.

Figure 4-7-1 PS2 wireless controller

PS2 handle is composed of two parts, the handle and receiver, the handle needs two section 7th 1.5V batteries, the receiver and arduino control board use the same power supply, the voltage is 3~5v, can not be reversed, can not exceed , overvoltage and reverse connection will cause the receiver to burn out. There is a power switch on the handle,turn handle switch to on, the lamp on the handle will flash if didn't search the receiver, the handle will enter Standby mode if this station last for some time,the light will go out. At this time, you need to press the "START" button to wake-up handle.

The receiver is connected to the Arduino, and powered by Arduino, such as the picture 3.2.40, in an unpaired condition,a green light will flash.If the handle is open, the receiver is powered, the handle and receiver will automatically pair, then the lamp is always bright, the handle pair succeeds. The key "mode" (handle batch is different, the identification may be "analog", but will not affect the use), you can choose "Red light Mode", "Green mode".

Some users report that the handle and receiver are not properly paired! Most problems are that the receiver is not wired correctly or there is a problem with the program.

Solution: We have already made the receiver on the driver board as shown in Figure 4-7-2.

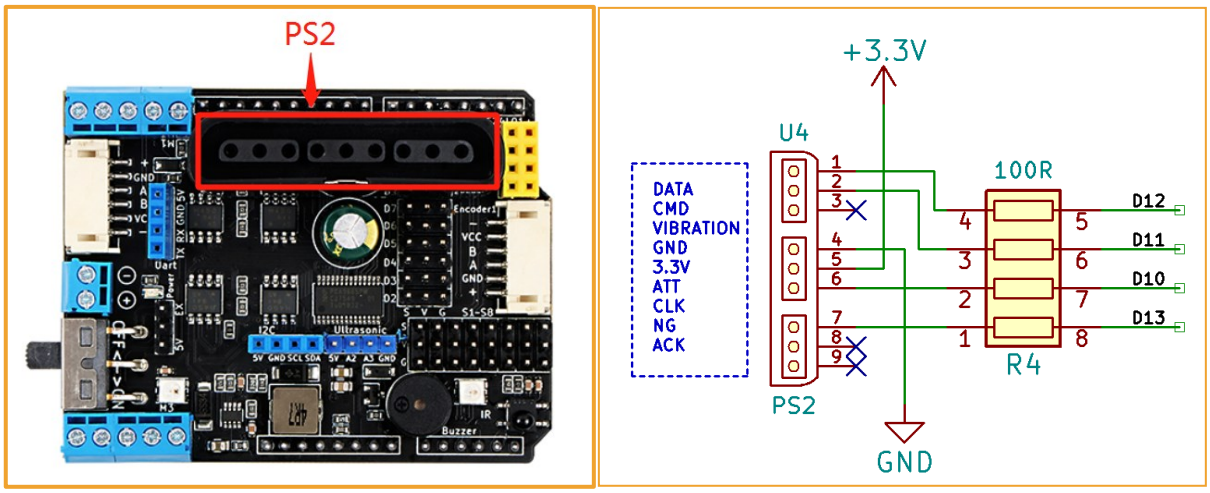


Figure 4-7-2 Remote Receiver Module

There are 9 interfaces on the the receiver, shown in the following table:

1	2	3	4	5	6	7	8	9
DI/DAT	DO/CMD	NC	GND	VDD	CS/SEL	CLK	NC	ACK

Note: Different batches, the appearance of the receiver will be some different, but the pin definition is the same,so don't worry about the use.

Di/dat: Signal flow, from the handle to the host, this signal is a 8bit serial data, synchronous transmission in the clock down the edge. The reading of the signal is done in the process of changing the clock from high to low.

Do/CMD: Signal flow, from the host to the handle, this signal is relative to DI, the signal is a 8bit serial data, synchronously transmitted to the clock down the edge.

NC: Empty port; .

GND: Ground;

VDD: The 3~5V power supply;

CS/SEL: Providing trigger signals for handles, the level is low during communication;

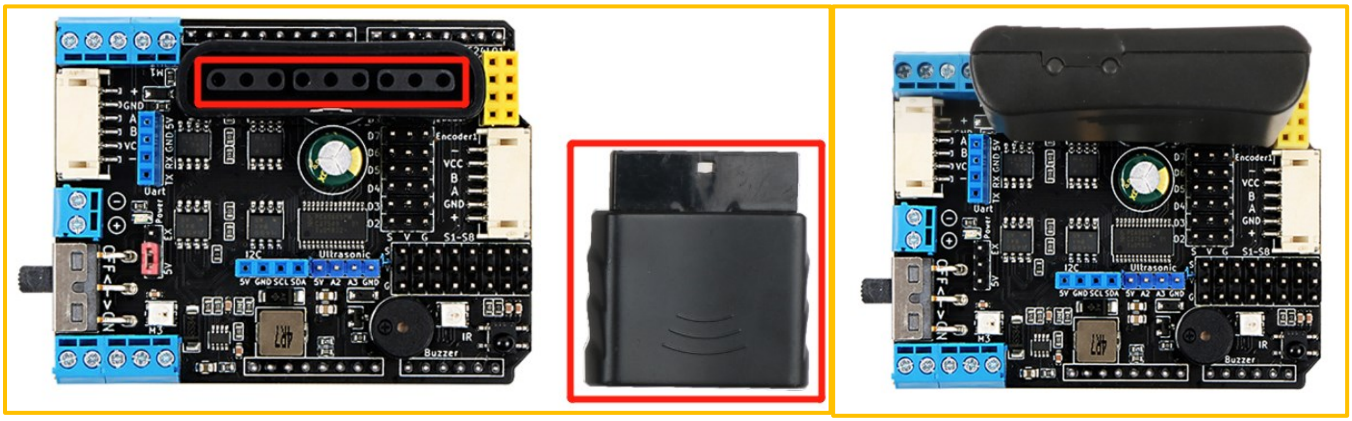
CLK: The clock signal is sent by the host to maintain data synchronization;

NC: Empty port;

ACK: the response signal from the handle to the host. This signal changes to low in the last cycle of each 8-bit data sending, and the CS remains low. If the CS signal do not remain low, the PS host will try another device in about 60 microseconds. The ACK port is not used in programming.

4.7.2 Experimental steps

1. For the sake of simple wiring, the PS2 we use can directly insert the PS2 receiver into the socket. As shown in Figure 4-7-3



4-7-3 Installation of Receiving Head

2、open the "Lesson\Module_Test\PS2X_Test\PS2X_Test.ino" in the supporting courseware Finally, download the program to the development board of Arduino and open the PS2 remote control. If the receiving head is connected to the remote control (or the pairing is successful), the indicator light on the receiving head is long, and vice versa, the LED light flashes continuously. Finally, we open the "serial monitor", press any button on the remote control, you can see the corresponding data on the "serial monitor", as shown in Figure 4-7-4.

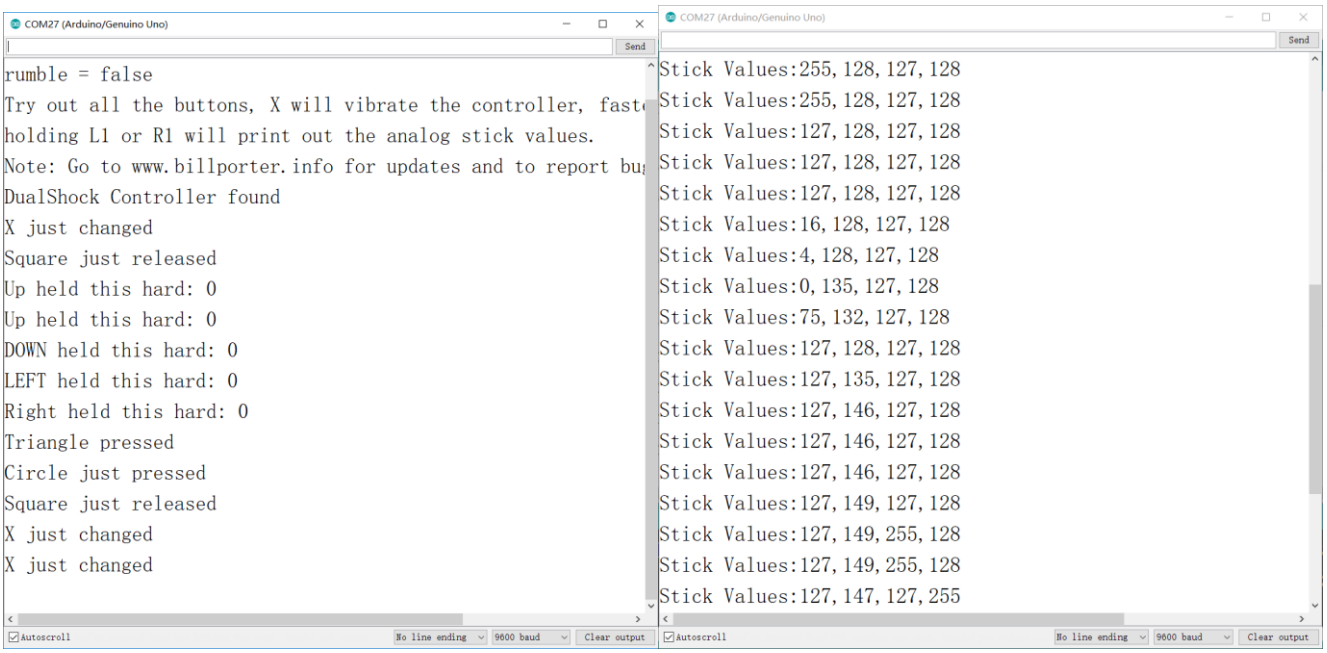


Figure 4-7-4 “Serial Monitor” data display

4.7.3 Software Design

Ps2 Control program in

“Lesson\Comprehensive Experiment\PantherTank_PS2X\PantherTank_PS2X.ino”

In above programs,we just finished the experiment of testing the button. In this experiment we want to implement the PS2 remote control car function. We first define all the button functions as follows:

Figure 4-7-5 PS2 handle function button

Logo UP: Advance

Logo DOWN: Back

Logo LEFT: Turn left

Logo RIGHT: Turn right

Logo A: Acceleration

Logo B: Left spin

Logo C: Deceleration

Mark D: right spin

Mark 3: Control the front of the robot arm to grab the servo and open. (manipulator arm is optional)

Mark 4: Control the front of the robot arm to grab the servo and close. (manipulator arm is optional)

Joystick Left: Control the left and right rotation of the base of the robot arm (the arm is optional), 0-89 degrees or 270-360 degrees to control the bottom steering right turn, 90-269 degrees to control the bottom steering left turn.

Joystick Right: control the left and right steering of the robot arm (the arm is optional), 44-135 degrees to control the right steering right turn, 224-315 degrees to control the right steering left turn, 136-223 degrees to control the left rudder Turn left, 0-43 degrees or 315-360 degrees to control the left steering right turn.

4.8 CC2540 Bluetooth Module Test Experiment

4.8.1 Introduction to Bluetooth Module

Ble - UNO bluetooth 4.0 protocol is based on the perfect combination Arduino UNO by emakefun to create customer research and development of a revolutionary product, function and the pin is fully compatible with traditional Arduino UNO motherboard, scope of work frequency of 2.4 GHZ, modulation mode for the GFSK, maximum transmitted power of 0 db, the largest launch distance of 50 meters, USES the import original TI CC2540 chip design, support user through AT commands to modify view device name service UUID transmitted power matching password instructions, such as convenient and quick to use flexibleHowever, the product is very small in size, which is suitable for many applications with severe restrictions on the size. We provide Android and IOS mobile phone demo.You can quickly develop a communication with mobile phone hardware equipment as now very popular wearable mobile phone peripheral equipment, can use Ble - Nano this development platform, you can use a Ble - UNO is connected, with bluetooth 4.0 wireless transmission between two bluetooth devices, master-slave machine Settings. Even with PC bluetooth HID connections AT the same time, we provide developers with a great degree of freedom and support, the user can not only by the AT command debugging Ble - UNO, you can also add on Ble - UNO controller board sensor Arduino compatible extensionEmakefun's exclusive bluetooth host mode automatically connects to the slave, and supports sending over 20 bytes, making it easier to use

4.8.2 TI CC2540 Bluetooth Module Parameters

- BLE chip :TI CC2540
- Work channel : 2.4G
- Transmission distance: 50m from the open space
- Supports AT directive to configure BLE
- Support USB virtual serial port, hardware serial port,BLE three-way transparent transmission

- Support master and slave switch
- Support bluetooth automatic connection of slave in host mode
- Supports sending over 20 byte
- Support iBeacons
- Interface: Mircor – Usb
- Input voltage :Usb power supply,Vin6~12V,5V

4.8.3 Bluetooth module test experiment steps

1、Open the Arduino IDE, connect the serial port as shown in Figure 4-8-1, open the serial port monitor as shown in Figure 3-2-8-2

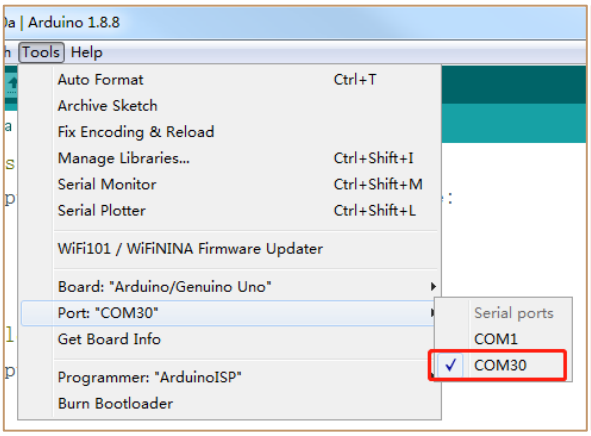


figure 4-8-1

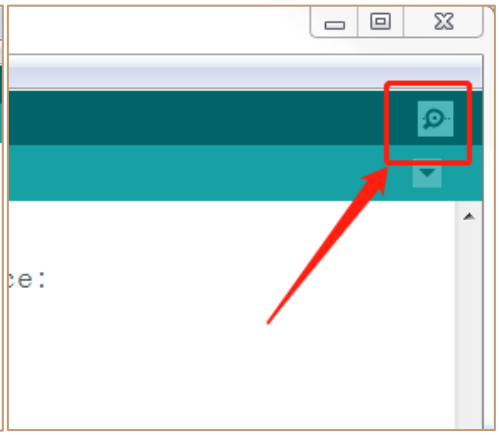


figure 4-8-2

2、The test AT instruction is shown in figure 4-8-3. Set the USB and bluetooth data transmission mode of BLE-UNO to USB serial port data and BLE transmission as shown in figure 4-8-4

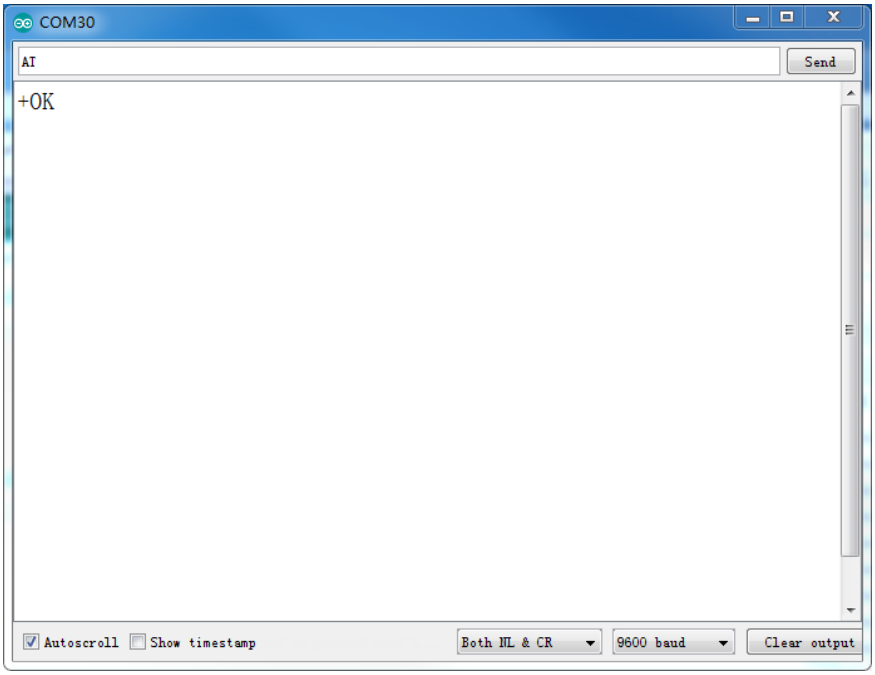


figure 4-8-3

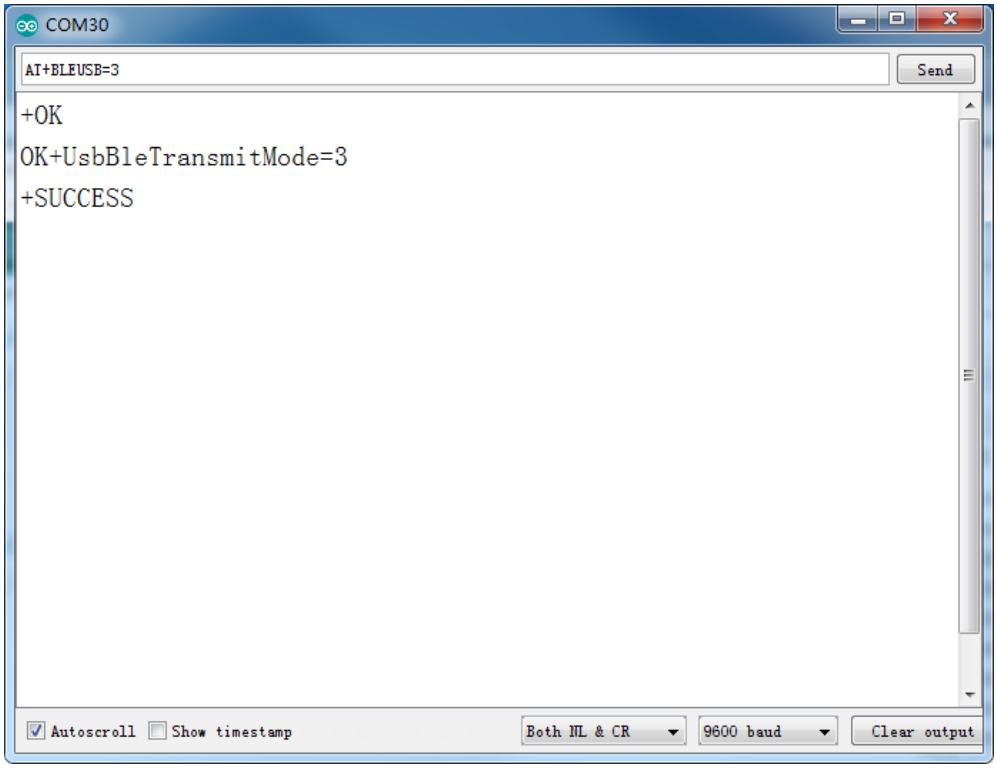


figure 4-8-4

3、Install BLE_TOOL.apk on the phone and open the test app. The interface is shown in Figure 4-8-5. Find the corresponding Bluetooth name (Ble-Nano) and click to connect. As shown in Figure 4-8-6, there will be 4 options for testing different functions, because here we only test whether Bluetooth can Normally send and receive data, so we choose SK Service into Figure 4-8-6 in the selection of SK_KEYPRESSED as shown in Figure 4-8-7

Figure 4-8-5 Figure 4-8-6

Figure 4-8-7 Figure 4-8-8

4、We select "SK-KEYPRESSED", click on it as shown in Figure 4-8-9. We can see that there is a "write" button, click to enter the interface shown in Figure 4-8-10, in Figure 4-8-10, We click on the "red box" to enter the data you want to send. After the input is completed, click "send" to send the data, as shown in Figure 4-8-10.

Figure 4-8-9 Figure 4-8-10

5、After clicking Send, we can see that the content sent by the mobile phone is printed on the serial monitor, as shown in Figure 4-8-11, indicating that the Bluetooth module can send data normally. Of course, in order to test the accuracy, you can Test a few more times and try to test in different environments.

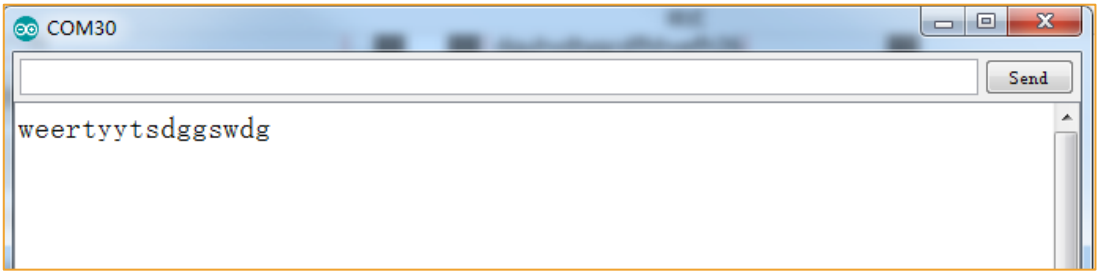


Figure 4-8-11

6、 As shown in Figure 3-2-8-12, we can input the content you want to send on the serial monitor. After clicking “Send”, you can send the data to the mobile APP via Bluetooth, as shown in Figure 4-8-13 is shown.

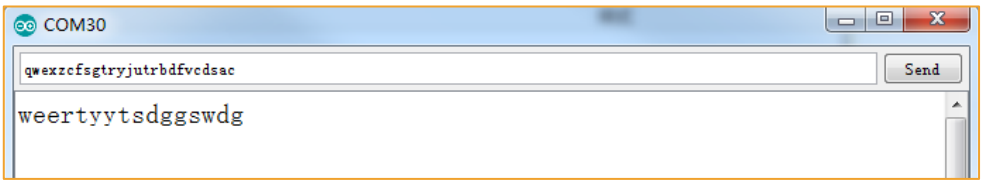


Figure 4-8-12

Figure 4-8-13

4.9 ESP-M2 Wifi Module Test Experiment (Optional)

4.9.1 Introduction to Wifi Module

Pather-Tank supports mobile phone APP Wifi remote control function uses the Wifi module is ESP-M2.

ESP-8285 Wifi transparent transmission module is based on TCP/IP protocol standard, working frequency range is 2.4GHZ range, maximum transmission power is 20dBm, adopts imported original chip design, supports standard IEEE802.11 b/g/n protocol, complete TCP /IP protocol stack. Users can use this module to add networking capabilities to existing devices or to build separate network controllers. The module data can be found in the ESP-M2-Wifi module\ ESP-M2 product manual. The physical picture of the Wifi module is shown in Figure 4-9-1.

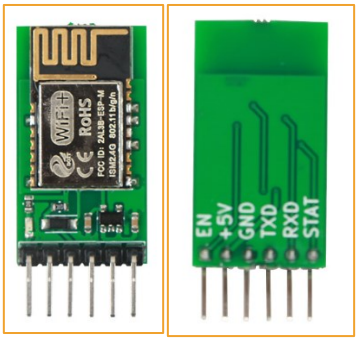


Figure 4-9-1 ESP-M2 Wifi module physical map

4.9.2 Wifi Module Features

1) Built-in Tensilica L106 ultra-low power 32-bit microprocessor, supporting 80MHz and 160MHz, supporting RTOS

2) Built-in TCP/IP protocol stack

3) Built-in 10 bit high precision ADC

4) Built-in TR switch, balun, LNA, power amplifier and matching network

5) Built-in PLL, voltage regulator and power management components, +20dBm output power in 802.11b mode

6) A-MODU\A-MSDU aggregation and 0.4s guard interval

7) Wifi@2.4GHz, zhichi WPA/WPA2 security mode

8) Support Smart Config function (including Android and iOS devices)

9) HSPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO

10) Deep sleep keeps current at 10uA and shutdown current is less than 5uA

11) Wake up, connect and transfer packets within 2 ms

12) Standby power consumption is less than 1.0mW (DTIM3)

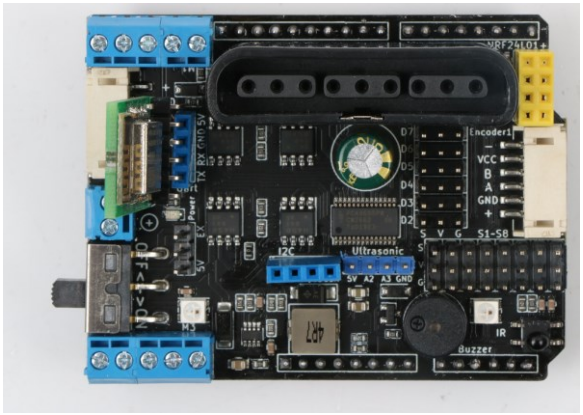
13) Operating temperature range: -20 ° C -85 ° C

4.9.3 Wifi Module Test Experiment Steps

1) Open the supporting information, find the folder "ESP-M2-Wifi Module\Wifi module test APP installation package" installation package and open it. Install the test software "TCP-Test.apk" in the directory to the mobile phone (currently only supported in Android system) On the phone).

2) Open data "ESP-M2-Wifi Module\WifiModule_Test\WifiModule_Test.ino"

3) Burn the WifiModule_Test.ino program to BLE-UNO R3; align the serial port pins of the Wifi module with the mother socket on the Arduino-UNO R3 main board and insert it; connect the Bluetooth module to the picture shown in Figure 4-9-2. Yes, the connection method is as follows: the VCC port of the ESP-M2 Wifi module is connected to the positive pole of the 5V DC power supply, the GND port is connected to the negative pole of the power supply, the RXD port is connected to the TXD port of the Arduino expansion board, and the TXD port is connected to the RXD port of the Arduino expansion board.



ESP-M2	Arduino UNO
VCC	VCC
GND	GND
TXD	RXD
RXD	TXD

Figure 4-9-2 Wifi module connection location

4) Turn on the power and test that the red indicator light on the Wifi module is on;

5) The Wifi name of the Wifi module connected to the phone is similar to "Doit_WIFI_C4D02B".

6) Open the test app, as shown in Figure 4-9-3. Click on the connection, click on the connection as shown in Figure 4-9-4, enter "192.168.4.1" in the address bar and then enter "9000" in the port field. Just connect the point. We only test whether the Wifi module can send and receive data normally. Click to enter the Figure 4-9-5, we click on the "red box" to enter the data you want to send, click "blue box" after the input is complete, the data will be sent out, as shown in Figure 4-9-6 Show.

Figure 4-9-3

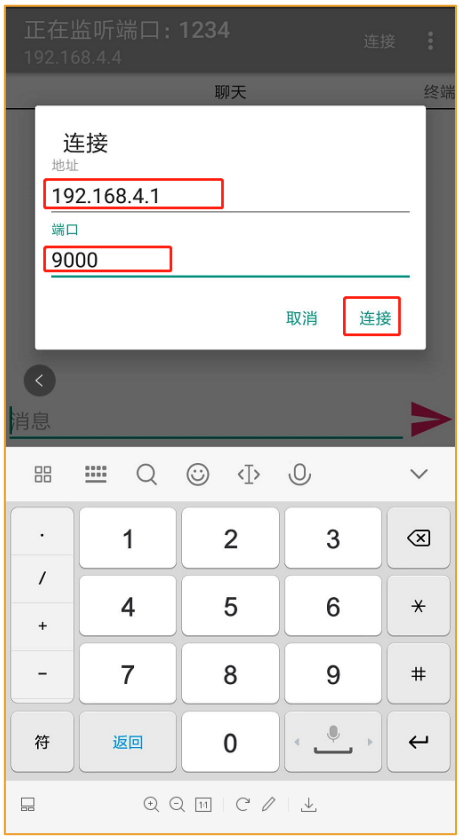


Figure 4-9-4

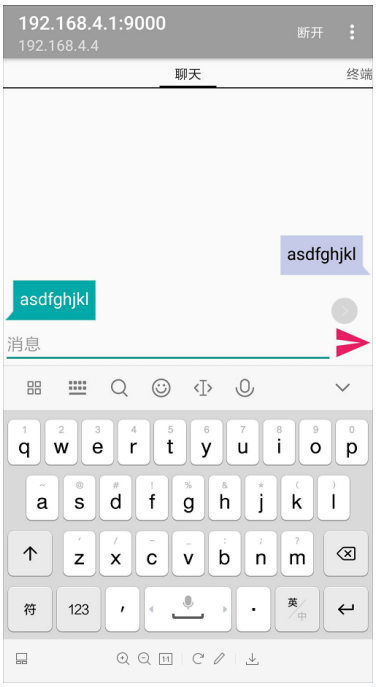


Figure 4-9-5

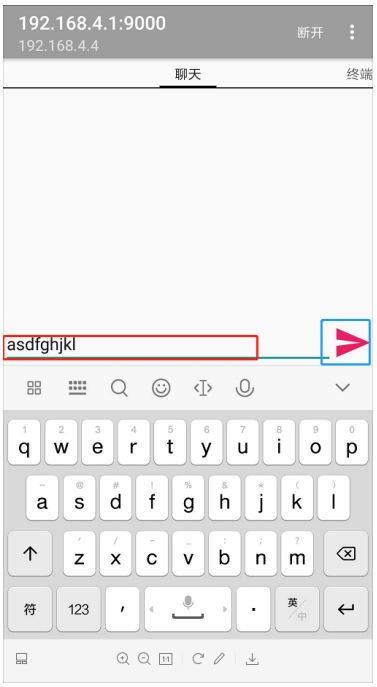


Figure 4-9-6

7) After clicking Send, we can see that the content sent by the mobile phone is printed on the serial monitor, as shown in Figure 4-9-7, indicating that the Bluetooth module can send data normally. Of course, for the accuracy of the test. High, you can test more times and try to test in different environments. You can also

send data from the serial monitor to the mobile phone, and the mobile terminal will also display the data sent by the serial port.

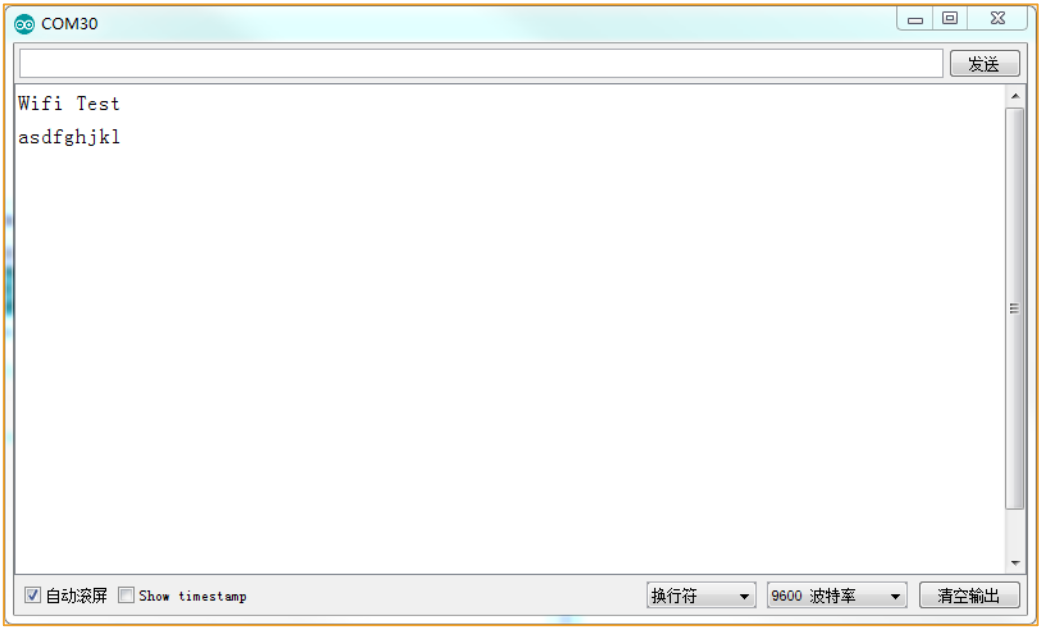


Figure 4-9-7

4.10 Robot arm (optional)

4.10.1 Robot arm wiring

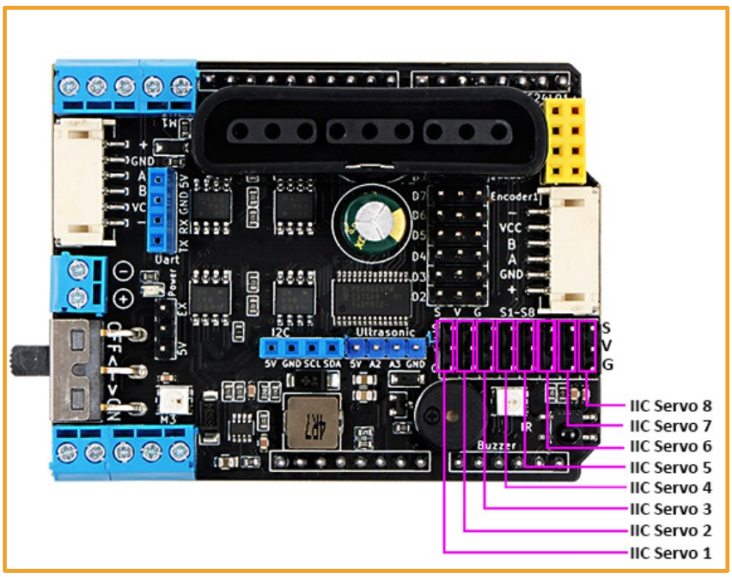


Figure 4-10-1

The arm of the arm is connected to the IIC servo 1, the left steering gear of the robot arm is connected to the IIC steering gear 2, the right steering gear is connected to the IIC steering gear 3, and the clip steering gear is connected to the IIC steering gear 4.

4.10.1 Robotic arm PS2 control

Figure 4-10-2

Left rocker 0 – 90°, 270 - 360° to control the I2C servo 1 The angle is slowly reduced, and 91–269° is used to control the I2C steering angle slowly.

Right rocker 0 – 44°, 315 - 360° for controlling I2C steering gear 2 angle slowly decreasing, 135 – 225 degrees for controlling I2C steering gear 2 angle slowly increasing, 45 – 134° for controlling I2C steering gear 3 angle Slowly increase, 225 – 314° control I2C servo 3 angle slowly decreases.

L1 is to increase the control I2C servo 4, and R1 is to control the I2C servo 4 to slowly decrease.

Note: Because of the mechanical arm structure problem, the left and right steering gears have a travel limit. If you are not using well, you can ask the technical support personnel how to adjust. If you understand, you can try it yourself.