Lesson 2 - LK COKOINO Balance Car Drive Shield

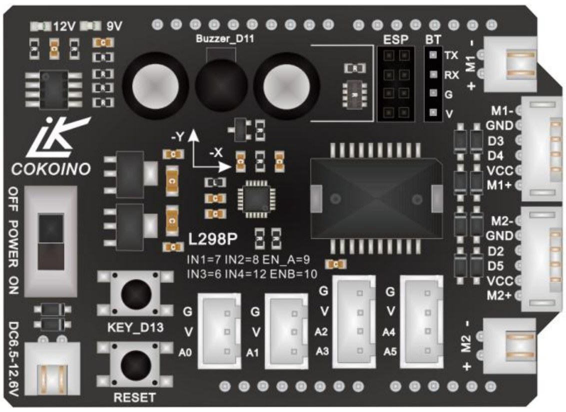


Table of Contents

1、Overview

LK COKOINO balance car drive shield can be directly inserted into UNO R3 board;
The balance car drive shield has HX2.54 interfaces, with a 5V\1A voltage regulator chip, it can quickly connect the sensor module and the servos.
Reserve a serial port interface, which can be directly plugged into a Bluetooth module;
Reserved a WiFi interface, you can directly insert an esp-01 module;
Integrated L298P motor drive system which can drive 2 motors;
Integrated a MPU6050 6-axis motion component, a gyroscope and a accelerometer;
The battery indicator and buzzer are integrated on the board to make it easy to understand the external battery status and make sounds.
Combined with a UNO R3 board and other modules, it can be combined into a variety of interesting products, such as: intelligent Bluetooth robots, smart WiFi robots, etc., which we mainly use in smart balance car.

2、Specifications

(1) External voltage: DC6.5-12.6V, anti-reverse connection

(2) 5V power supply: use AMS1117-5V, maximum 1A current

(3) Motor drive chip: L298P, input DC6.5-12.6V, output total current up to 4A

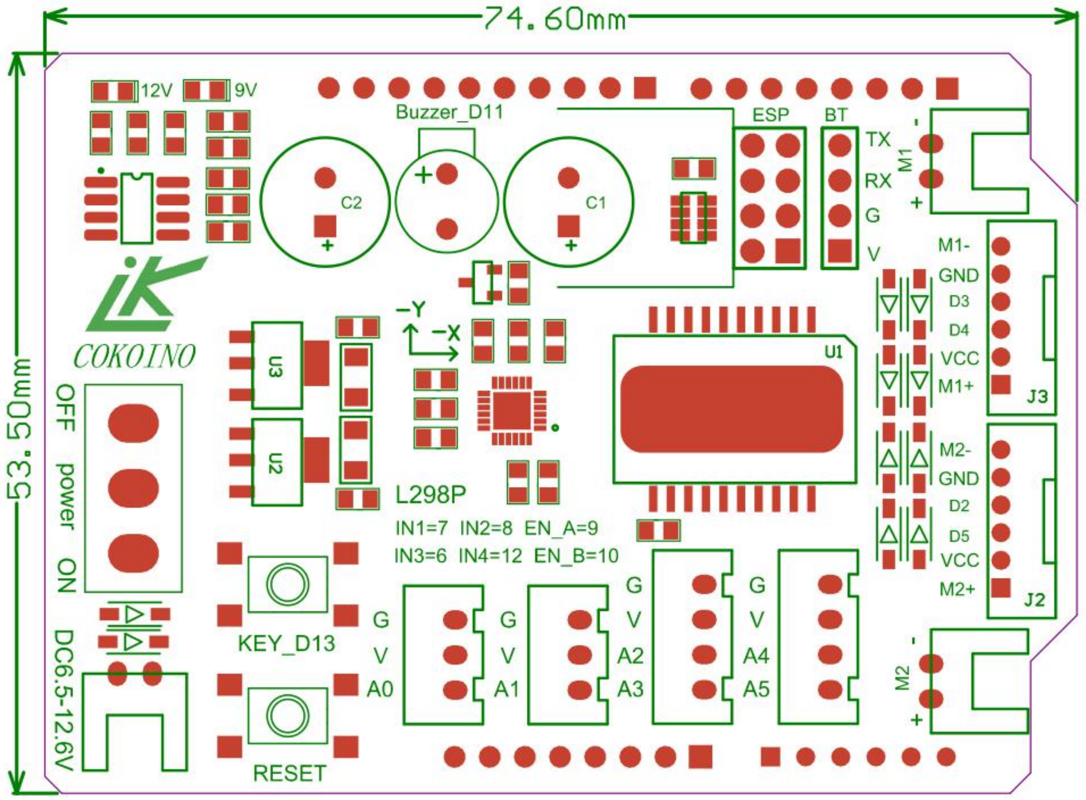
(4) 6-axis sensor: MPU6050 gyroscope and acceleration chip

(5) Interface: XH2.54 connector, 2.54mm pin header

(6) wifi interface power supply: use AMS1117-3.3V, maximum 1A current

(7) Compatible motherboard: can be directly inserted into arduino UNO R3

3、Size（unit: mm）



4、Description of Silkscreen

M1\ M2 interface： Can be connected to DC6.5-12.6V motor, commonly used in ordinary robot motors.

J2 interface：J2 interface is used for the motor with the code wheel. It is often used to drive the motor of the balance car, as defined below.
M2+\- is the A-channel motor drive output of L298P, GND is logic ground, D2\D5 is connected to 2\5 pin of UNO R3, VCC is connected to 5V power supply of balance car drive shield.

J3 interface：J2 interface is used for the motor with the code wheel. It is often used to drive the motor of the balance car, as defined below.
M1+\-is the B-channel motor drive output of L298P, GND is logic ground, D3\D4 is connected to 2\5 pin of UNO R3, VCC is connected to 5V power supply of balance car drive shield.

G\V\A0 interface： Equal to UNO R3 A0 pin, can be connected to single analog signal, digital signal module.

G\V\A1 interface： Equal to UNO R3 A1 pin， Can be connected to single analog signal, digital signal module.

G\V\A2\A3 interface：Equal to UNO R3 A2-3 pin，can be connected to an ultrasonic module.

G\V\A4\A5 interface： Equal to UNO R3 A4-5 pin（IIC）, can be connected to IIC 1602LCD module.

V\G\TX\RX interface： Equal to UNO R3 serial port, which can be connected to the Bluetooth module.

ESP-01 interface： Equal to UNO R3 serial port, which can be connected to ESP-01 wifi module.

Buzzer_D11： The buzzer is controlled by the UNO R3 11 pin.

RESET： UNO R3 reset button.

KEY_D13： Connect the UNO R3 13 pin.

9V： battery voltage 9V, 12V: battery voltage 12V

Power： Power switch, ON to turn on the power, OFF to turn off the power

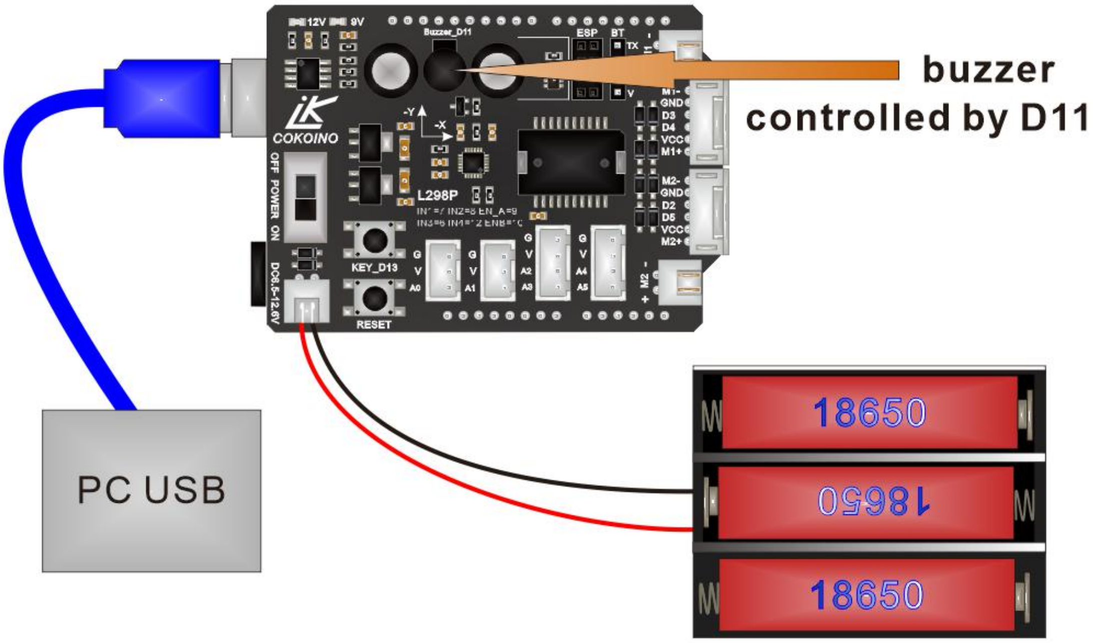
V、G：The positive pole of the DC5V/1A power supply, and the negative pole of the DC5V/1A power supply.

5、Make the onboard buzzer sound

5.1、Need to prepare：

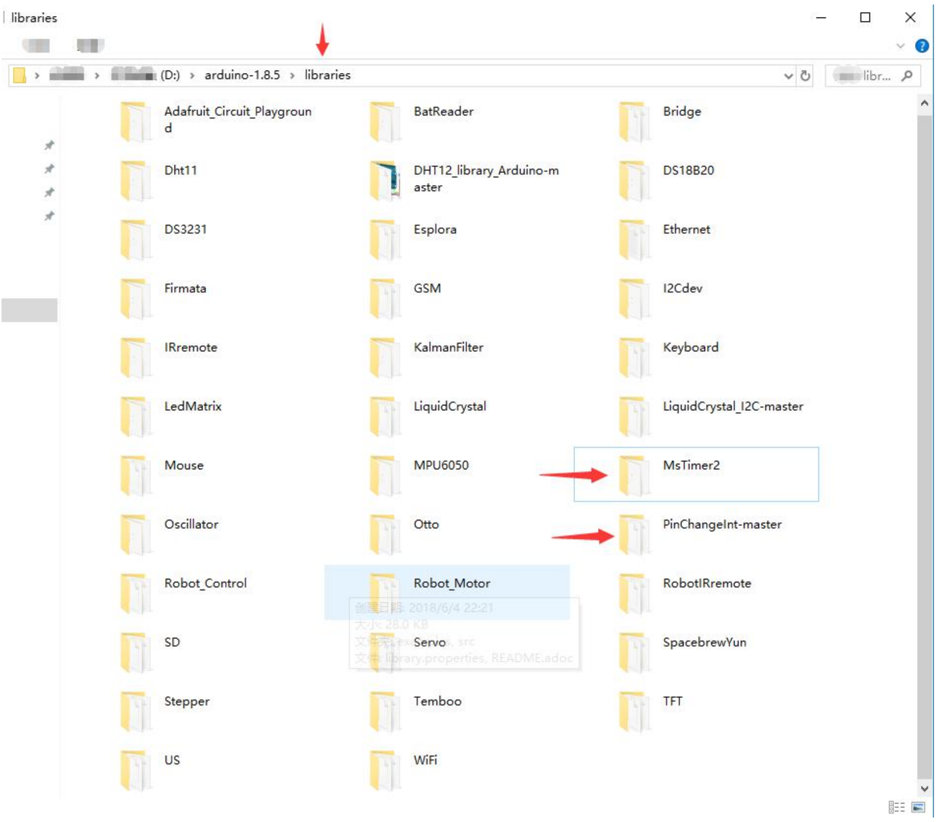
A uno r3 board
A computer with reliable Internet
A battery case with two 18650 batteries
A usb cable

5.2、Wiring diagram:



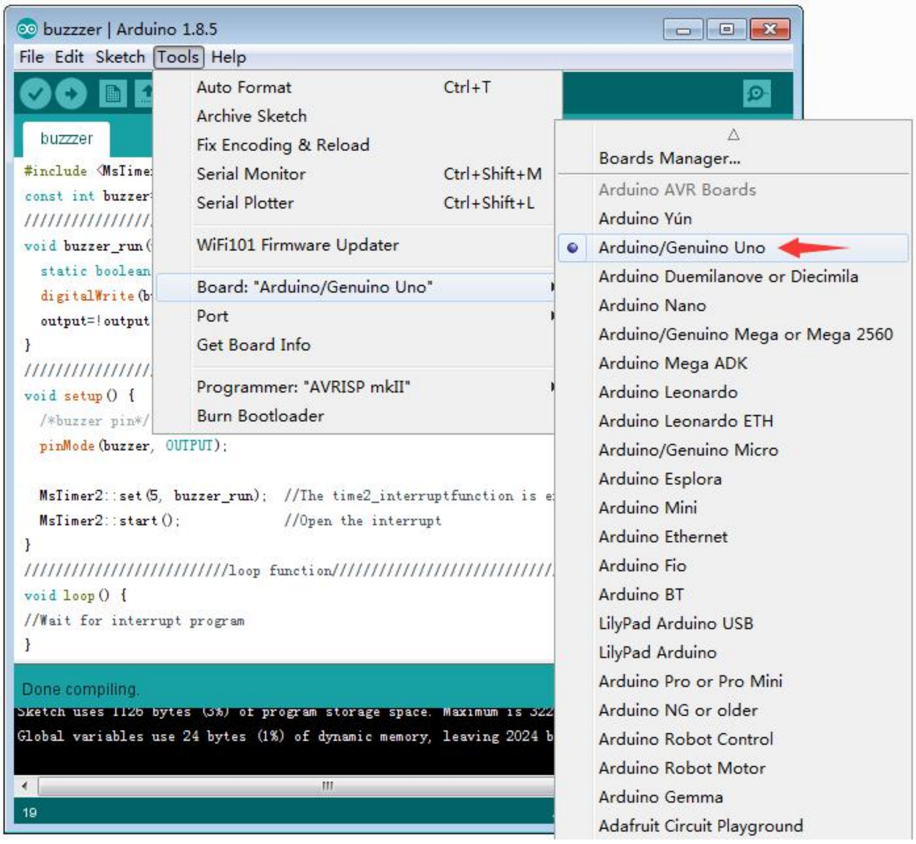
5.3、Upload the Code：

Copy the MsTimer2 and PinChangeInt library files to the library folder of the Arduino IDE, as shown below:



Adafruit_Circuit_Playground
BatReader
Bridge
Dht11
DHT12_library_Arduino-master
DS18B20
DS3231
Esplora
Ethernet
Firmata
GSM
I2Cdev
IRremote
KalmanFilter
Keyboard
LedMatrix
LiquidCrystal
LiquidCrystal_I2C-master
Mouse
MPU6050
MsTimer2
Oscillator
Otto
PinChangeInt-master
Robot_Control
Robot_Motor
RobotIRremote
SD
Servo
SpacebrewYun
Stepper
Temboo
TFT
US
WiFi

Select board:



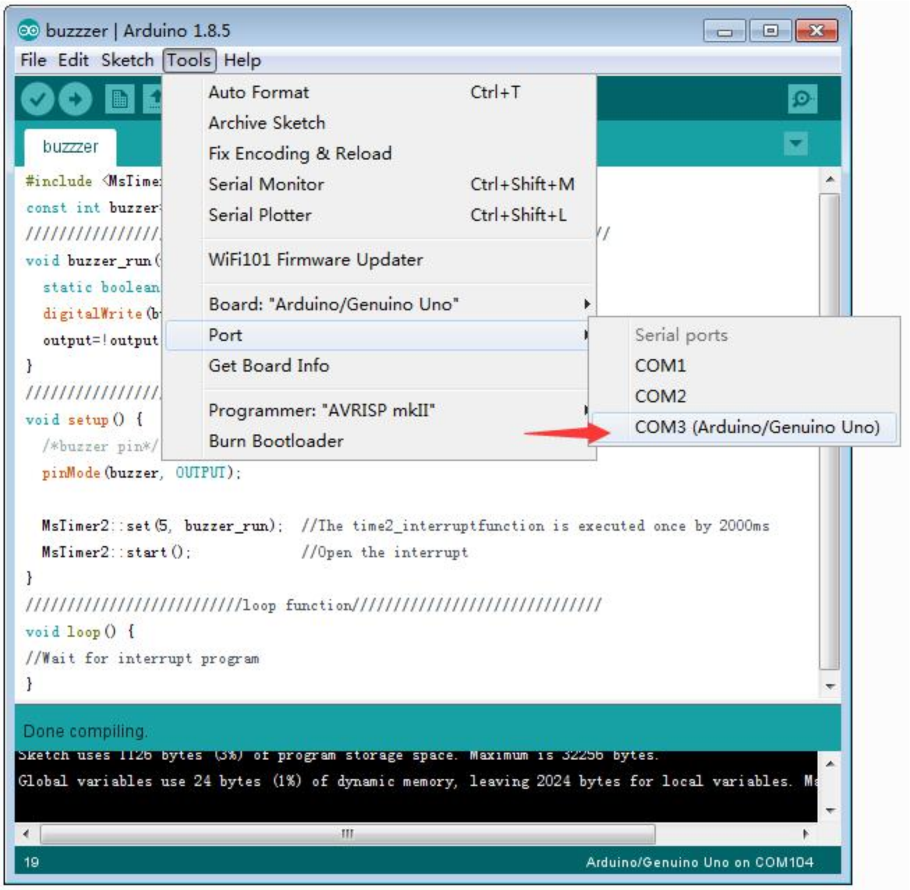
```
#include <MsTime
const int buzzer
///////////////
void buzzer_run(
  static boolean
  digitalWrite(b
  output=!output
}
///////////////
void setup() {
  /*buzzer pin*/
  pinMode(buzzer, OUTPUT);

  MsTimer2::set(5, buzzer_run);  //The time2_interruptfunction is e
  MsTimer2::start();             //Open the interrupt
}
//////////////////////////loop function///////////////////////////
void loop() {
//Wait for interrupt program
}
```

Done compiling.

```
Sketch uses 1126 bytes (3%) of program storage space. Maximum is 322
Global variables use 24 bytes (1%) of dynamic memory, leaving 2024 b
```

Select the USB port:


buzzzer | Arduino 1.8.5
File Edit Sketch Tools Help
buzzer
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM2
COM3 (Arduino/Genuino Uno)
19
Arduino/Genuino Uno on COM104

```
#include <MsTime
const int buzzer
///////////////
void buzzer_run(
  static boolean
  digitalWrite(b
  output=!output
}
///////////////
void setup() {
  /*buzzer pin*/
  pinMode(buzzer, OUTPUT);

  MsTimer2::set(5, buzzer_run);  //The time2_interruptfunction is executed once by 2000ms
  MsTimer2::start();             //Open the interrupt
}
//////////////////////////loop function/////////////////////////////
void loop() {
//Wait for interrupt program
}
```

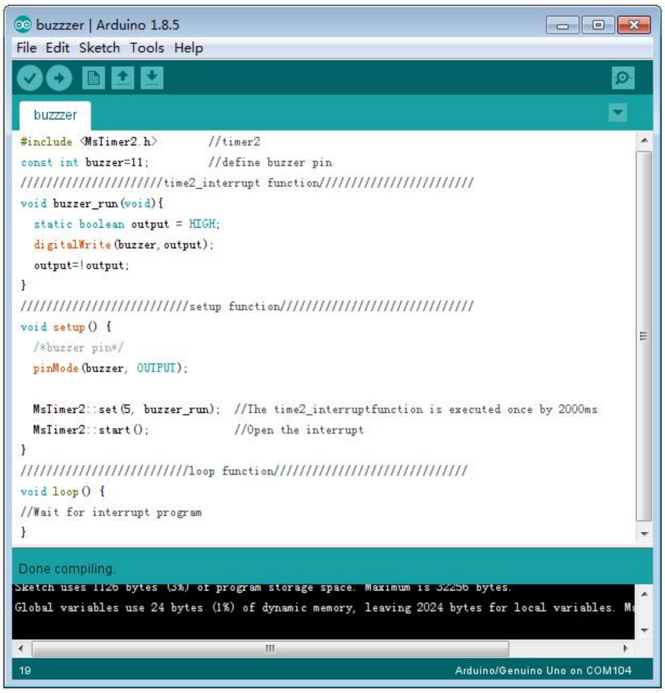
Done compiling.

```
Sketch uses 1126 bytes (3%) of program storage space. Maximum is 32256 bytes.
Global variables use 24 bytes (1%) of dynamic memory, leaving 2024 bytes for local variables. Ma
```

Use the IDE to open the code in this folder code / buzzer and upload it）

```
#include <MsTimer2.h>              //timer2
const int buzzer=11;               //define buzzer pin
////////////////////time2_interrupt function//////////////////////
void buzzer_run(void){
   static boolean output = HIGH;
   digitalWrite(buzzer,output);
   output=!output;
}
////////////////////////setup function///////////////////////////
void setup() {
   /*buzzer pin*/
   pinMode(buzzer, OUTPUT);

   MsTimer2::set(5, buzzer_run);    //The time2_interruptfunction is executed once by 2000ms
   MsTimer2::start();               //Open the interrupt
}
////////////////////////loop function///////////////////////////
void loop() {
//Wait for interrupt program
}
```



```
#include <MsTimer2.h>        //timer2
const int buzzer=11;         //define buzzer pin
////////////////////time2_interrupt function//////////////////////
void buzzer_run(void){
  static boolean output = HIGH;
  digitalWrite(buzzer,output);
  output=!output;
}
////////////////////////setup function///////////////////////////
void setup() {
  /*buzzer pin*/
  pinMode(buzzer, OUTPUT);

  MsTimer2::set(5, buzzer_run);  //The time2_interruptfunction is executed once by 2000ms
  MsTimer2::start();             //Open the interrupt
}
////////////////////////loop function///////////////////////////
void loop() {
//Wait for interrupt program
}
```

Done compiling.

```
Sketch uses 1126 bytes (3%) of program storage space. Maximum is 32256 bytes.
Global variables use 24 bytes (1%) of dynamic memory, leaving 2024 bytes for local variables. M
```

19 Arduino/Genuino Uno on COM104

5.4、result：

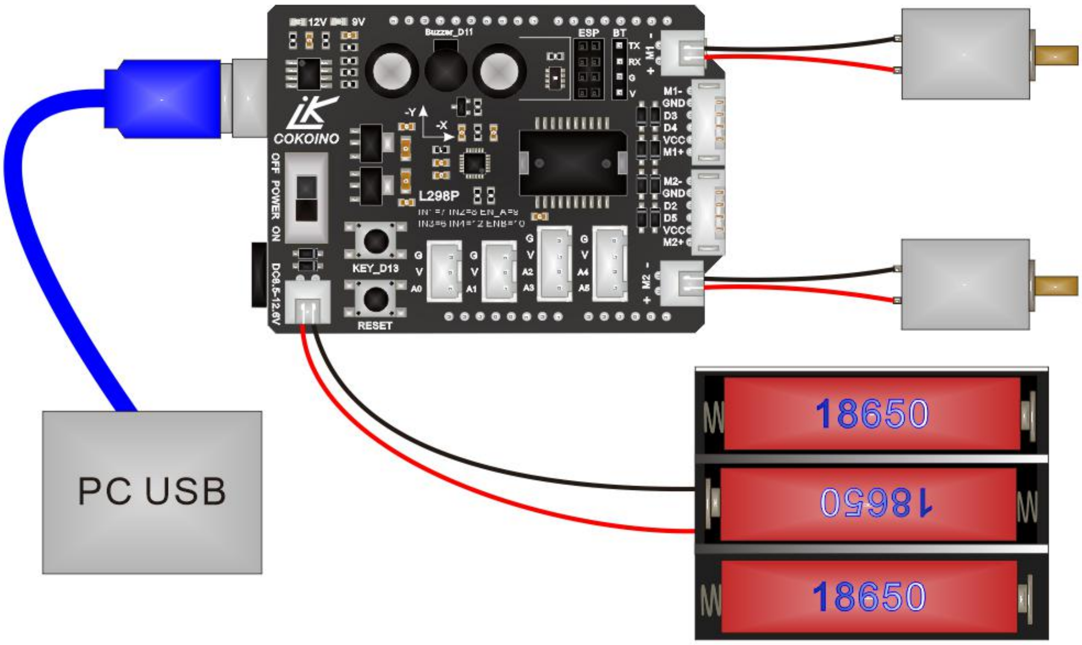
Turn on the power switch on the balance car shield and the buzzer will sound.

6 、 the onboard L298P motor chip drives ordinary DC motor

6.1、Need to prepare：

A uno r3 board
Two ordinary DC motor(not included, we only provide tutorial)
A computer with reliable Internet
A battery case with two 18650 batteries
A usb cable

6.2、Wiring diagram：



UNO R3 PIN	L298P PIN	M1 DC Motor
9=PWM	8（EnableA）	Control motor speed
8=HIGH	9(INPUT1)=HIGH	Motor forward
7=LOW	7(INPUT2)=LOW	
8=LOW	9(INPUT1)=LOW	Motor reversal
7=HIGH	7(INPUT1)=HIGH	

UNO R3 PIN	L298P PIN	M2 DC Motor

10=PWM	14（EnableB）	Control motor speed
12=HIGH	15(INPUT4)=HIGH	Motor forward
6=LOW	13(INPUT3)=LOW	
12=LOW	15(INPUT1)=LOW	Motor reversal
6=HIGH	13(INPUT1)=HIGH	

Remarks: Provide L298P specification。

6.3、Upload code：

Use the IDE to open the code in this folder code / DC_motor and upload it）

```
void setup(){            //Set parameter function
pinMode(6, OUTPUT);      //Set pin6 to output mode
pinMode(7, OUTPUT);      //Set pin7 to output mode
pinMode(8, OUTPUT);      //Set pin8 to output mode
pinMode(9, OUTPUT);      //Set pin9 to output mode
pinMode(10, OUTPUT);     //Set pin10 to output mode
pinMode(12, OUTPUT);     //Set pin12 to output mode
}
void loop(){              //The main loop function
analogWrite(10,200);      //pin10 output duty ratio of 200:55 of the PWM signal
digitalWrite(12,HIGH);    //pin12 output high level
digitalWrite(6,LOW);      //pin6 output low level
analogWrite(9,200);       //pin9 output duty ratio of 200:55 of the PWM signal
digitalWrite(8,HIGH);     //pin2 output high level
digitalWrite(7,LOW);      //pin7 output low level
delay(2000);              //delay 2000ms
analogWrite(10,200);      //pin10 output duty ratio of 200:55 of the PWM signal
digitalWrite(12,LOW);     //pin12 output low level
digitalWrite(6,HIGH);     //pin6 output high level
analogWrite(9,200);       //pin9 output duty ratio of 200:55 of the PWM signal
digitalWrite(8,LOW);      //pin8 output low level
digitalWrite(7,HIGH);     //pin7 output high level
delay(2000);              //delay 2000ms
}
```

6.2、Result：

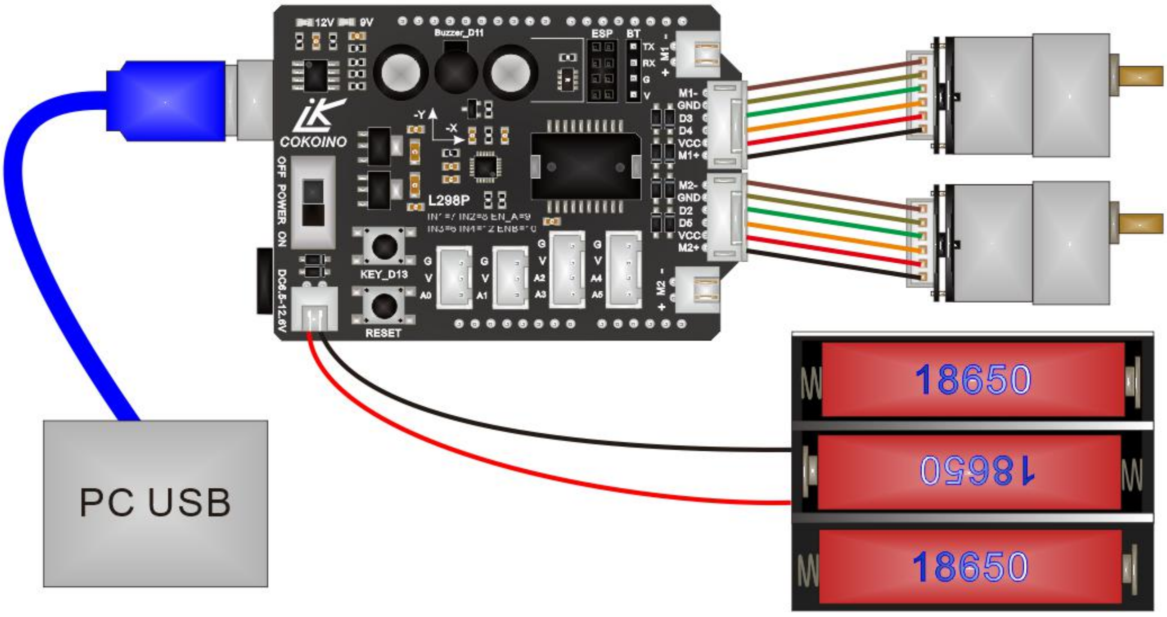
The two motors are cyclically rotated forward for 2 seconds and then reversed for two seconds.

7、On-board L298P motor chip drives DC motor with encoder

7.1、Need to prepare：

A uno r3 board
A computer with reliable Internet
A battery case with two 18650 batteries
A usb cable
Two DC motor with encoder

7.2、Wiring diagram：

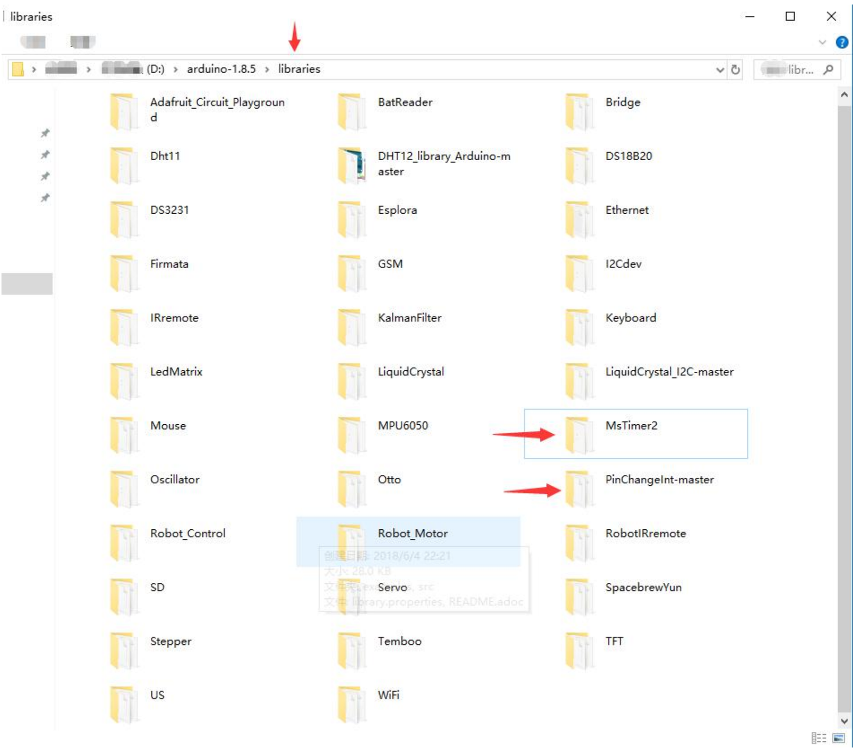


UNO R3 PIN	L298P PIN	M1 encoder motor
9=PWM	8（EnableA）	control motor speed
8=HIGH	9(INPUT1)=HIGH	Motor forward
7=LOW	7(INPUT2)=LOW	
8=LOW	9(INPUT1)=LOW	Motor reversal
7=HIGH	7(INPUT1)=HIGH	
3		hall C1 PIN
4		hall C2 PIN

UNO R3 PIN	L298P PIN	M2 encoder motor
10=PWM	14（EnableB）	control motor speed
12=HIGH	15(INPUT4)=HIGH	Motor forward
6=LOW	13(INPUT3)=LOW	
12=LOW	15(INPUT1)=LOW	Motor reversal
6=HIGH	13(INPUT1)=HIGH	
2		hall C1 PIN
5		hall C2 PIN

7.3、Upload code：

Copy the MsTimer2 and PinChangeInt library files to the library file of the Arduino IDE, as shown below:



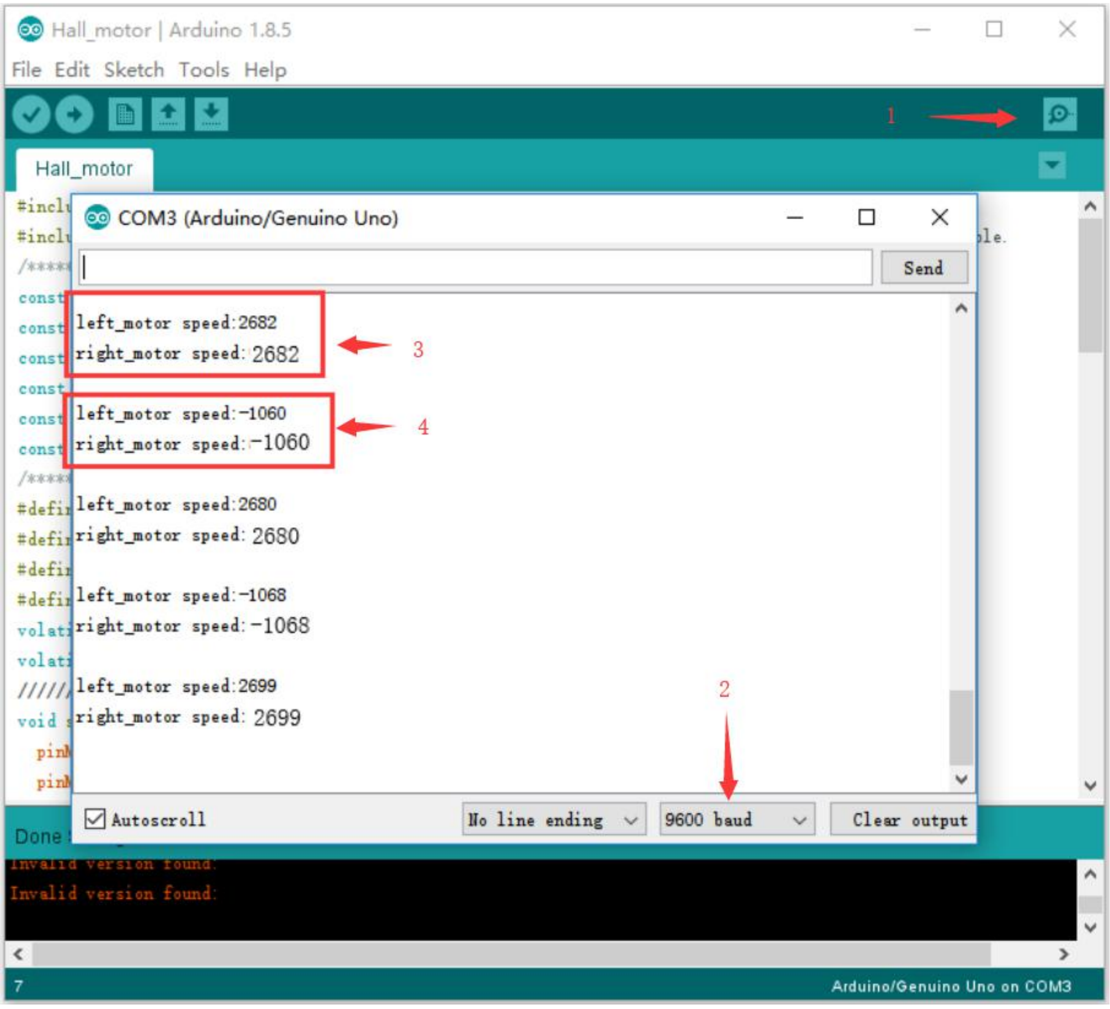
Use the IDE to open the code in this folder code / Hall_motor and upload it）

```
#include <MsTimer2.h>        //timer2
#include <PinChangeInt.h>    //The pin external interruption library.All arduino
UNO pins are available.
/***************************define L298P pin***************************/
```

```
const int right_R1=7;
const int right_R2=8;
const int PWM_R=9;
const int left_L1=12;
const int left_L2=6;
const int PWM_L=10;
/**************************Interrupt speed count***********************/
#define Hall_left 2             //interrupt pin
#define Direction_left 5
#define Hall_right 3            //interrupt pin
#define Direction_right 4
volatile long count_right = 0;
volatile long count_left = 0;
//////////////////////////setup function//////////////////////////////
void setup() {
  pinMode(right_R1,OUTPUT);      //motor control pin
  pinMode(right_R2,OUTPUT);
  pinMode(left_L1,OUTPUT);
  pinMode(left_L2,OUTPUT);
  pinMode(PWM_R,OUTPUT);
  pinMode(PWM_L,OUTPUT);
  pinMode(Hall_left, INPUT);     //hall pin
  pinMode(Direction_left,INPUT);
  pinMode(Hall_right, INPUT);
  pinMode(Direction_right,INPUT);
  Serial.begin(9600);                       //Opening a serial port,Set the baud
rate to 9600
  MsTimer2::set(2000, time2_interrupt);      //The time2_interruptfunction is
executed once by 2000ms
  MsTimer2::start();                         //Open the interrupt
  attachPinChangeInterrupt(Hall_left, Encoder_left, CHANGE);          //define
interrupt function
  attachPinChangeInterrupt(Hall_right, Encoder_right, CHANGE);        //define
interrupt function
}
//////////////////////////loop function//////////////////////////////
void loop() {
  digitalWrite(left_L1,LOW);
  digitalWrite(left_L2,HIGH);
  analogWrite(PWM_L,160);
  digitalWrite(right_R1,LOW);
  digitalWrite(right_R2,HIGH);
  analogWrite(PWM_R,160);
  delay(2000);
  digitalWrite(left_L1,HIGH);
  digitalWrite(left_L2,LOW);
  analogWrite(PWM_L,80);
  digitalWrite(right_R1,HIGH);
  digitalWrite(right_R2,LOW);
  analogWrite(PWM_R,80);
  delay(2000);
}
//////////////////////time2_interrupt function////////////////////////
void time2_interrupt(void){
  Serial.print(″left_motor speed:″);
  Serial.println(count_left);
  Serial.print(″right_motor speed:″);
  Serial.println(count_right);
  Serial.println(″″);
  count_left=count_right=0;
}
///////////////////////hall of the left motor/////////////////////////
void Encoder_left() {
  if(digitalRead(Hall_left)==LOW){
    if(digitalRead(Direction_left)==LOW){count_left ++;}
    else {count_left --;}
    }
  else{
    if(digitalRead(Direction_left)==LOW){count_left --;}
    else {count_left ++;}
    }
}
///////////////////////hall of the right motor/////////////////////////
void Encoder_right() {
  if(digitalRead(Hall_right)==LOW){
    if(digitalRead(Direction_right)==LOW){count_right --;}
    else {count_right ++;}
    }
  else{
    if(digitalRead(Direction_right)==LOW){count_right ++;}
    else {count_right --;}
    }
}
```

7.4、 Result：

After the code upload is successful, turn on the power switch on the shield, the motor rotates forward for two seconds, then reverses for two seconds, cyclically move like this;
Open the serial monitor of the arduino IDE, then adjust the baud rate to 9600. The monitor will automatically print the number of times the Hall component outputs the signal on the motor.
As shown in the figure:



Hall_motor | Arduino 1.8.5
File Edit Sketch Tools Help
1
Hall_motor
COM3 (Arduino/Genuino Uno)
Send
3
4
2
☑ Autoscroll
No line ending
Clear output
Done
Arduino/Genuino Uno on COM3

```
left_motor speed:2682
right_motor speed: 2682

left_motor speed:-1060
right_motor speed:-1060

left_motor speed:2680
right_motor speed: 2680

left_motor speed:-1068
right_motor speed:-1068

left_motor speed:2699
right_motor speed: 2699
```

9600 baud

```
Invalid version found.
Invalid version found.
```

8、On-board MPU6050

8.1、Need to prepare：

A uno r3 board
A computer with reliable Internet
A battery case with two 18650 batteries
A usb cable
Two DC motors

8.2、Specification of the On-board MPU6050：

(1) Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a

(2) user-programmable fullscale range of ±250, ±500, ±1000, and ±2000°/sec

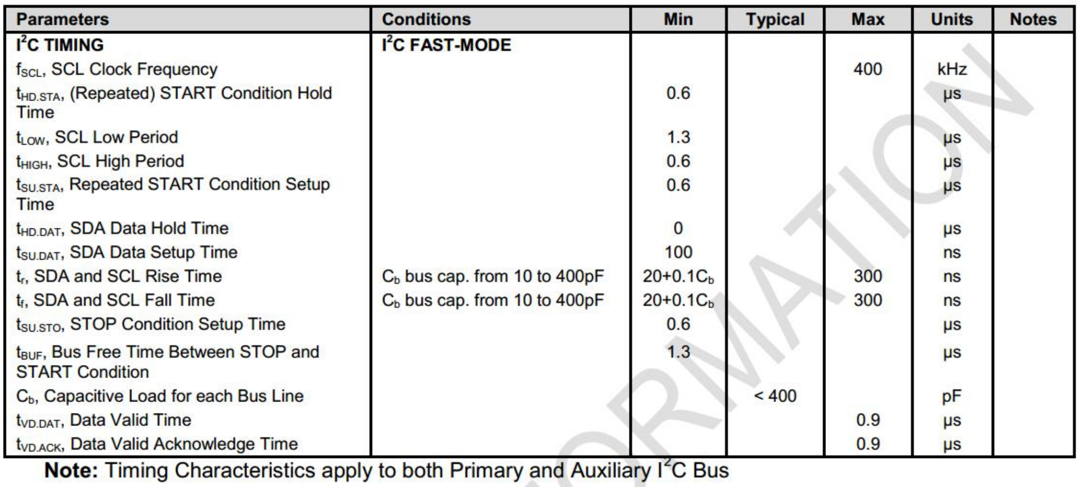
(2) Integrated 16-bit ADCs enable simultaneous sampling of gyros
(3) Enhanced bias and sensitivity temperature stability reduces the need for user
(3) calibration
(4) Improved low-frequency noise performance
(5) Digitally-programmable low-pass filter
(6) Gyroscope operating current: 3.6mA
(7) Standby current: 5µA
(4) (8) Digital-output triple-axis accelerometer with a programmable full scale range

(5) of ±2g, ±4g, ±8g and ±16g

(6) Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while
(7) requiring no external multiplexer
(8) (10) 400kHz Fast Mode I2C for communicating with all registers

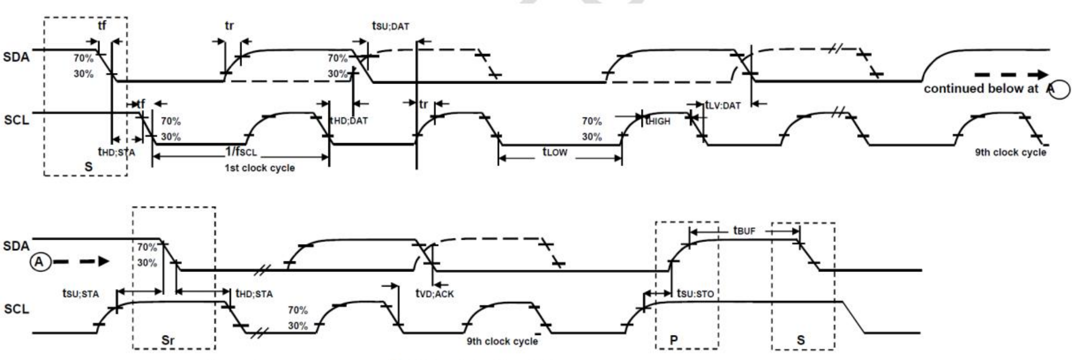
8.3、IIC communication：

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25° C



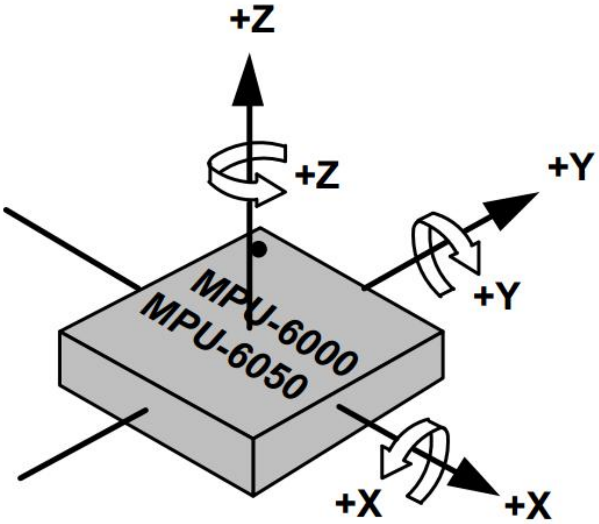
Parameters	Conditions	Min	Typical	Max	Units	Notes
I^{2}C TIMING	**I^{2}C FAST-MODE**					
f_{SCL}, SCL Clock Frequency				400	kHz	
$t_{HD.STA}$, (Repeated) START Condition Hold Time		0.6			µs	
t_{LOW}, SCL Low Period		1.3			µs	
t_{HIGH}, SCL High Period		0.6			µs	
$t_{SU.STA}$, Repeated START Condition Setup Time		0.6			µs	
$t_{HD.DAT}$, SDA Data Hold Time		0			µs	
$t_{SU.DAT}$, SDA Data Setup Time		100			ns	
t_r, SDA and SCL Rise Time	C_b bus cap. from 10 to 400pF	20+0.1C_b		300	ns	
t_f, SDA and SCL Fall Time	C_b bus cap. from 10 to 400pF	20+0.1C_b		300	ns	
$t_{SU.STO}$, STOP Condition Setup Time		0.6			µs	
t_{BUF}, Bus Free Time Between STOP and START Condition		1.3			µs	
C_b, Capacitive Load for each Bus Line			< 400		pF	
$t_{VD.DAT}$, Data Valid Time				0.9	µs	
$t_{VD.ACK}$, Data Valid Acknowledge Time				0.9	µs	

Note: Timing Characteristics apply to both Primary and Auxiliary I^{2}C Bus



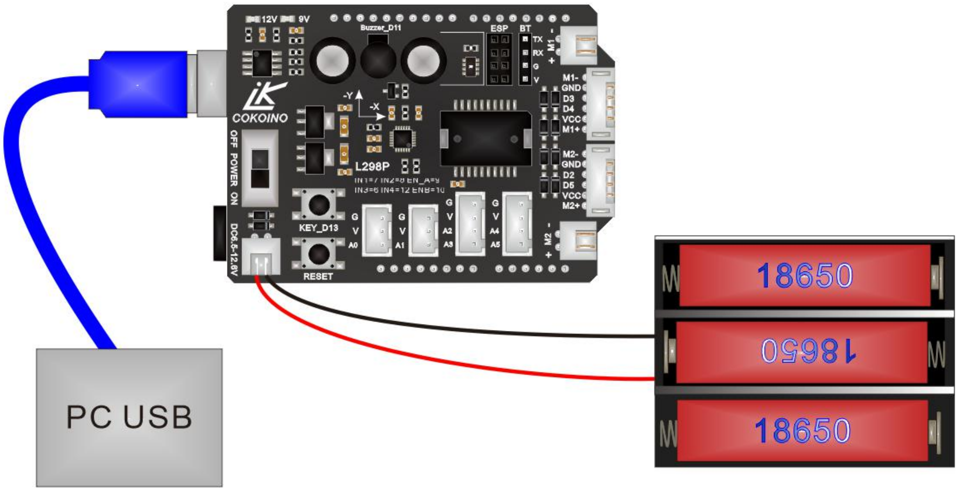
I^{2}C Bus Timing Diagram

Axial diagram：



8.4、Wiring diagram

The MPU6050 gyroscope and accelerometer are integrated on the shield. When the expansion board is directly inserted into UNO R3, the SCL\SDA of the MPU6050 is connected to A4\A5 of UNO R3, and communicates by IIC.

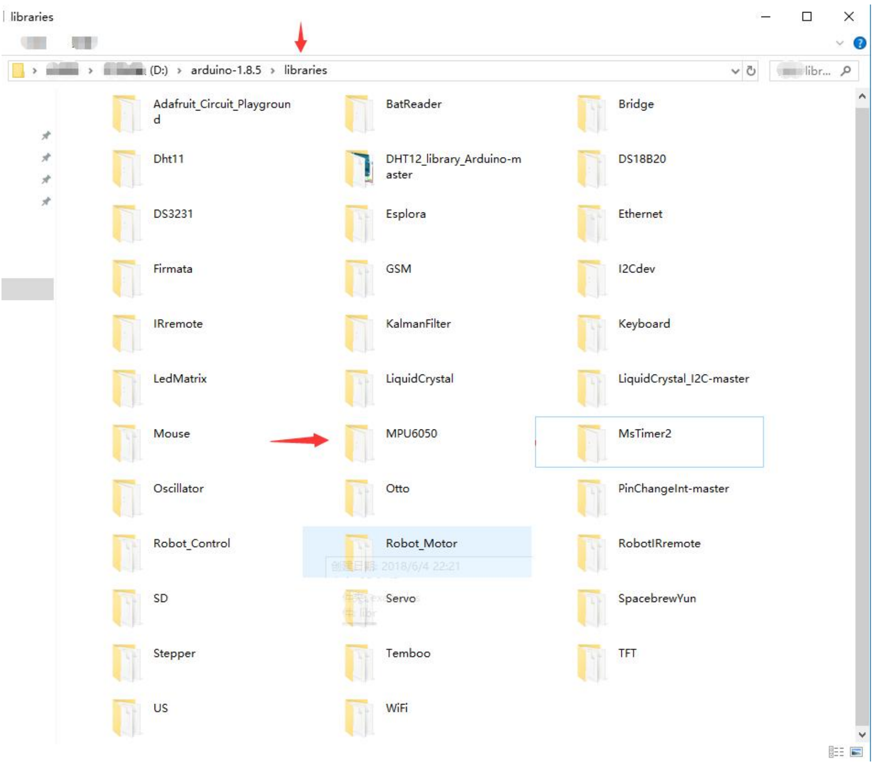


UNO R3	MPU6050
A4	SDA
A5	SCL
NC	On-board 3.3V
GND	GND

Remarks: Provide MPU6050 specification

8.5、Upload code：

Copy the MPU6050 library file to the library file of the Arduino IDE, as shown below:



Use the IDE to open the code in this folder code / MPU6050_test and upload it）

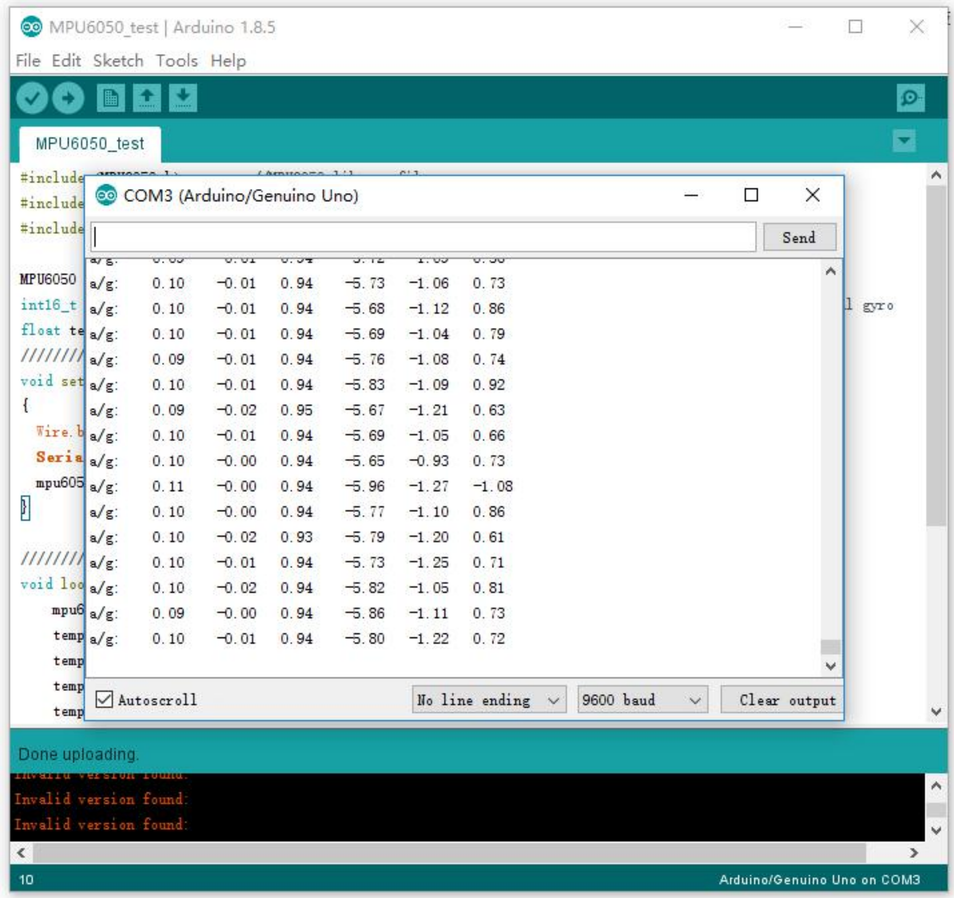
```
#include <MPU6050.h>          //MPU6050 library file
#include <Wire.h>             //IIC library file
#include "I2Cdev.h"

MPU6050 mpu6050;                       //define MPU6050 class
int16_t ax, ay, az, gx, gy, gz;     //define Variable of triaxial acceleration and
variables of Triaxial gyro
float temp_ax,temp_ay,temp_az,temp_gx,temp_gy,temp_gz;
//////////////////////////////////////////////////////////////////
void setup() {
  Wire.begin();                               //init the IIC bus
  Serial.begin(9600);                        //Opening a serial port,Set the baud
rate to 9600
  mpu6050.initialize();                       //init MPU6050
}
/////////////////////////loop function////////////////////////////
void loop() {
    mpu6050.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);     //get MPU6050 ax ay az
gx gy gz data
```

```
    temp_ax=ax;
    temp_ay=ay;
    temp_az=az;
    temp_gx=gx;
    temp_gy=gy;
    temp_gz=gz;
    Serial.print("a/g:\t");
    Serial.print(temp_ax/16384); Serial.print("\t");
    Serial.print(temp_ay/16384); Serial.print("\t");
    Serial.print(temp_az/16384); Serial.print("\t");
    Serial.print(temp_gx/131); Serial.print("\t");
    Serial.print(temp_gy/131); Serial.print("\t");
    Serial.println(temp_gz/131);
    delay(200);
}
```

8.6、Result：

The following is the value output by the MUPU6050 when the shield is placed flat on the desktop. The value is the raw data that has not been processed, and there will be some errors.

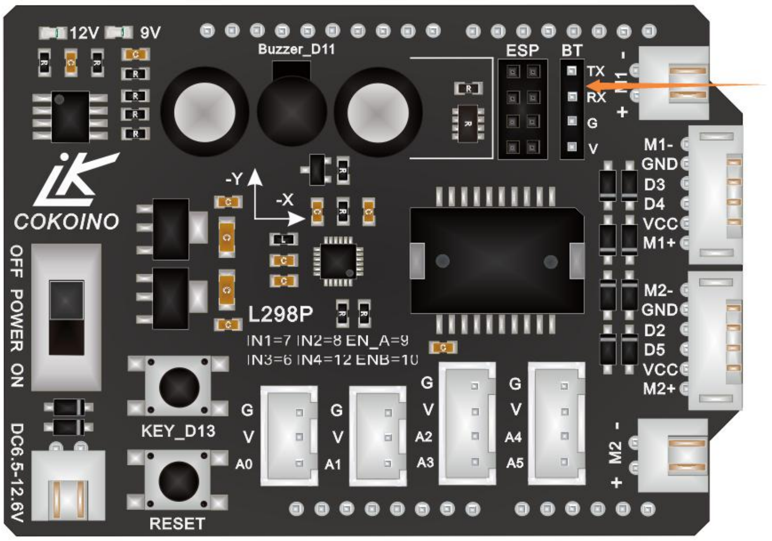

MPU6050_test | Arduino 1.8.5
File Edit Sketch Tools Help
MPU6050_test
COM3 (Arduino/Genuino Uno)
Send
☑ Autoscroll
No line ending
9600 baud
Clear output
Arduino/Genuino Uno on COM3

```
a/g:    0.10    -0.01   0.94    -5.73   -1.06   0.73
a/g:    0.10    -0.01   0.94    -5.68   -1.12   0.86
a/g:    0.10    -0.01   0.94    -5.69   -1.04   0.79
a/g:    0.09    -0.01   0.94    -5.76   -1.08   0.74
a/g:    0.10    -0.01   0.94    -5.83   -1.09   0.92
a/g:    0.09    -0.02   0.95    -5.67   -1.21   0.63
a/g:    0.10    -0.01   0.94    -5.69   -1.05   0.66
a/g:    0.10    -0.00   0.94    -5.65   -0.93   0.73
a/g:    0.11    -0.00   0.94    -5.96   -1.27   -1.08
a/g:    0.10    -0.00   0.94    -5.77   -1.10   0.86
a/g:    0.10    -0.02   0.93    -5.79   -1.20   0.61
a/g:    0.10    -0.01   0.94    -5.73   -1.25   0.71
a/g:    0.10    -0.02   0.94    -5.82   -1.05   0.81
a/g:    0.09    -0.00   0.94    -5.86   -1.11   0.73
a/g:    0.10    -0.01   0.94    -5.80   -1.22   0.72
```

```
Done uploading.
Invalid version found.
Invalid version found.
```

9、Onboard Bluetooth interface

When the shield is directly plugged into the UNO R3 board, the Bluetooth interface is directly connected to the serial port of UNO R3. Please do not connect the Bluetooth module to this interface when uploading the code, otherwise the code cannot be uploaded.

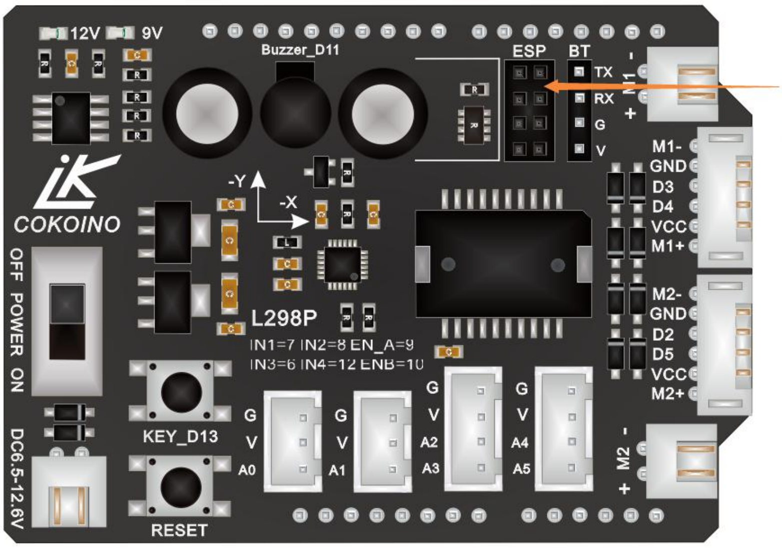


bluetooth port

10、Onboard wifi interface

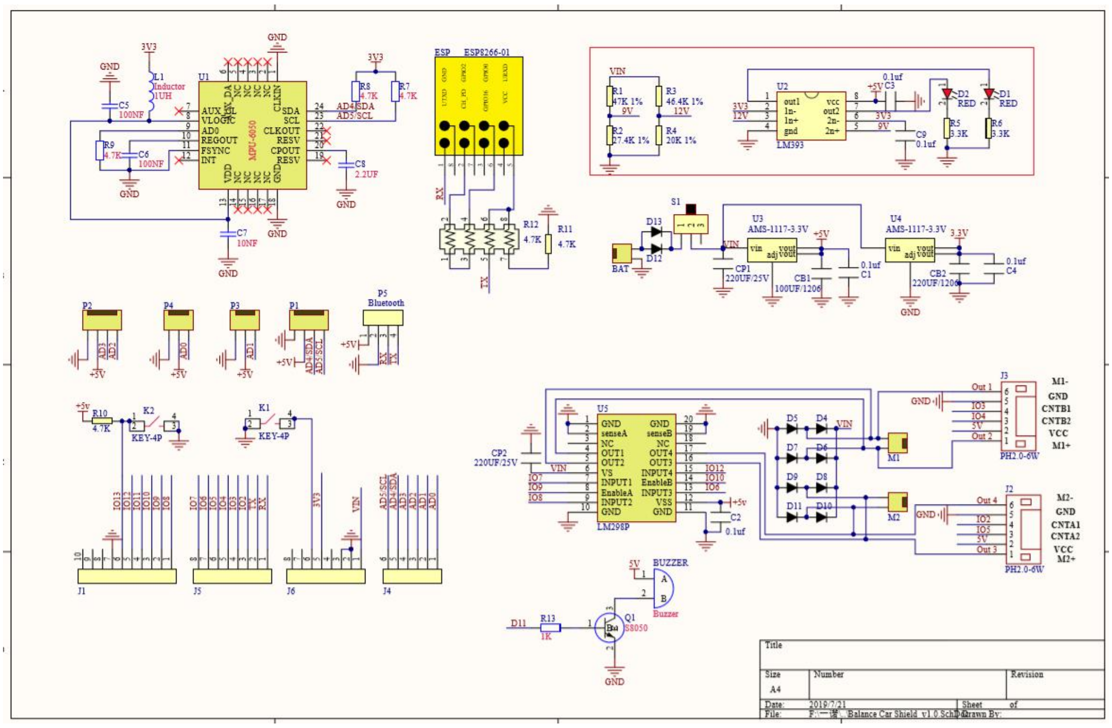
This Wifi interface can be plugged into an esp-01 module.

When the shield is directly plugged into the UNO R3 motherboard, and the esp-01 module is plugged into thewifi interface, the esp-01 module has been enabled by the hardware on the shield to the flash boot mode. The esp-01 interface is directly connected to the serial port of UNO R3. Please do not connect the esp-01 module to this interface when uploading the code. Otherwise, the code cannot be uploaded



ESP wifi port

10、Schematic Diagram



U1
MPU-6050
AUX_DA
AUX_CL
VLOGIC
AD0
REGOUT
FSYNC
INT
SDA
SCL
CLKOUT
RESV
CPOUT
CLKIN
VDD
GND
3V3
L1 Inductor 1UH
C5 100NF
C6 100NF
C7 10NF
C8 2.2UF
R9 4.7K
R8 4.7K
R7 4.7K
AD4/SDA
AD5/SCL
ESP
ESP8266-01
UTXD
CH_PD
GPIO0
GPIO2
GPIO16
URXD
VCC
GND
RX
TX
R12 4.7K
R11 4.7K
VIN
R1 47K 1%
R2 27.4K 1%
R3 46.4K 1%
R4 20K 1%
9V
12V
U2
LM393
out1
In-
In+
gnd
vcc
out2
2n-
2n+
+5V
C3 0.1uf
C9 0.1uf
D1 RED
D2 RED
R5 3.3K
R6 3.3K
S1
D13
D12
BAT
U3 AMS-1117-3.3V
U4 AMS-1117-3.3V
vin
adj
vout
CP1 220UF/25V
CB1 100UF/1206
C1 0.1uf
CB2 220UF/1206
C4 0.1uf
3.3V
P2
P4
P3
P1
P5 Bluetooth
AD3
AD2
AD0
AD1
+5V
R10 4.7K
K2 KEY-4P
K1 KEY-4P
J1
J5
J6
J4
IO13
IO12
IO11
IO10
IO9
IO8
IO7
IO6
IO5
IO4
IO3
IO2
U5
LM298P
senseA
senseB
OUT1
OUT2
OUT3
OUT4
VS
VSS
INPUT1
INPUT2
INPUT3
INPUT4
EnableA
EnableB
NC
CP2 220UF/25V
C2 0.1uf
D4
D5
D6
D7
D8
D9
D10
D11
M1
M2
J3
J2
Out 1
Out 2
Out 3
Out 4
PH2.0-6W
M1-
CNTB1
CNTB2
VCC
M1+
M2-
CNTA1
CNTA2
M2+
BUZZER
Buzzer
A
B
Q1 S8050
R13 1K
D11
5V
GND
Title
Size
A4
Number
Revision
Date:
2019/7/21
Sheet
of
File:
Balance Car Shield_v1.0.SchDoc
Drawn By: