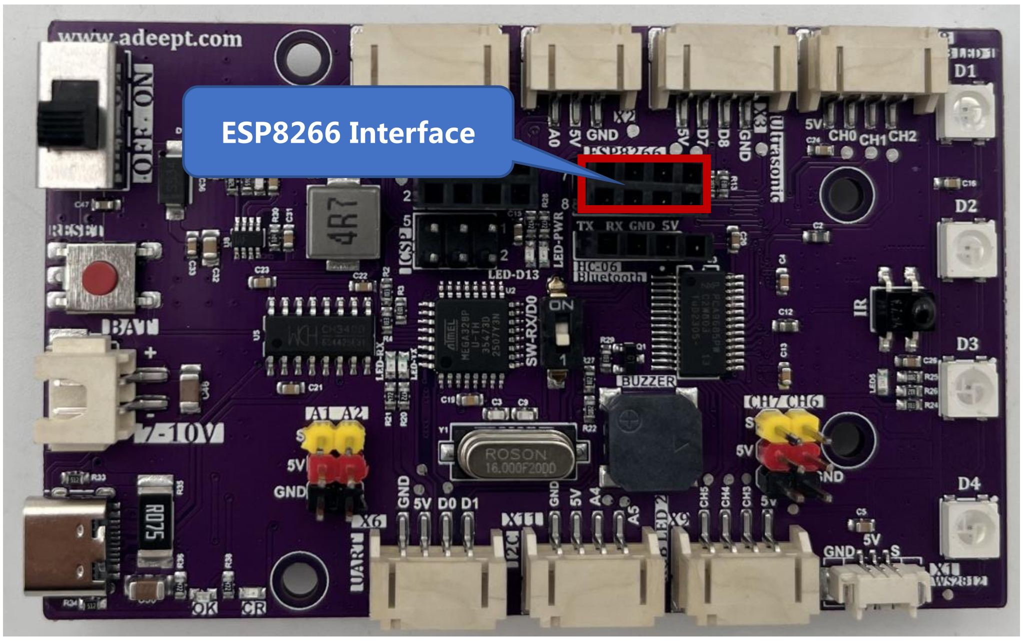
Lesson 23 Remote Control- Introduction to Processing

23.1 Overview

This lesson will introduce how to use the Processing programming environment to build a remote control interface for robot cars, and implement wireless communication with the ESP8266 module to control the robot's movement, obstacle avoidance, radar scanning, and other functions through a graphical user interface (GUI).

23.2 Install of ESP8266 Module



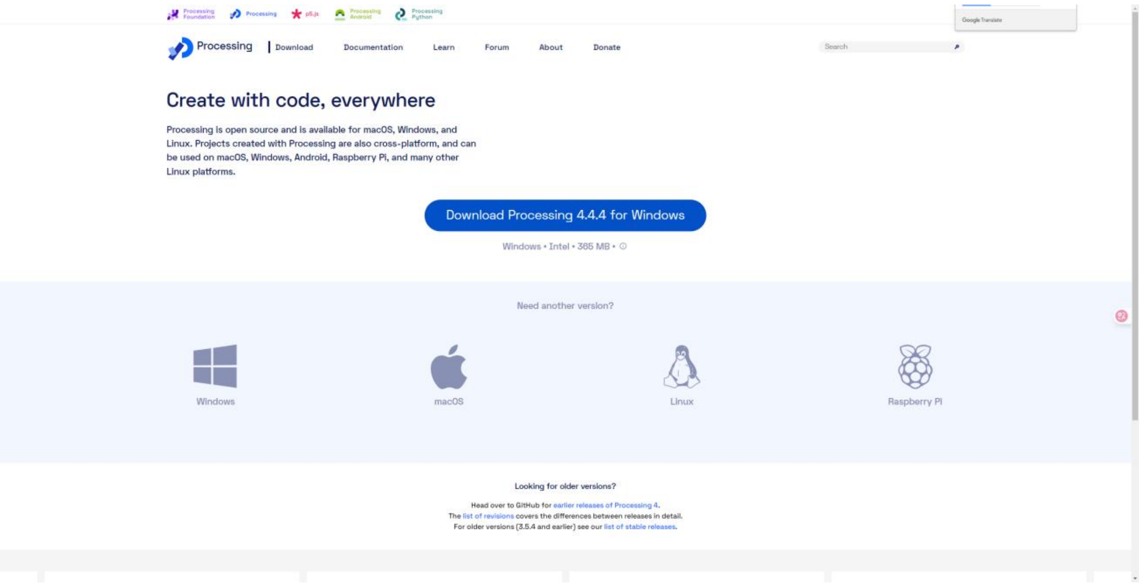
23.3 About Processing

Processing is a programming language, development environment, and online community. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Initially created to serve as a software sketchbook and to teach computer programming fundamentals within a visual context, Processing evolved into a development tool for professionals. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

23.3 Install Processing

You can download Processing via the link: https://processing.org/download/

You can choose an appropriate version to download according to your PC system.



Create with code, everywhere

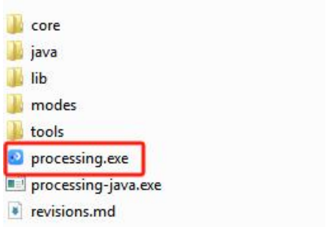
Processing is open source and is available for macOS, Windows, and Linux. Projects created with Processing are also cross-platform, and can be used on macOS, Windows, Android, Raspberry Pi, and many other Linux platforms.

Need another version?

Looking for older versions?

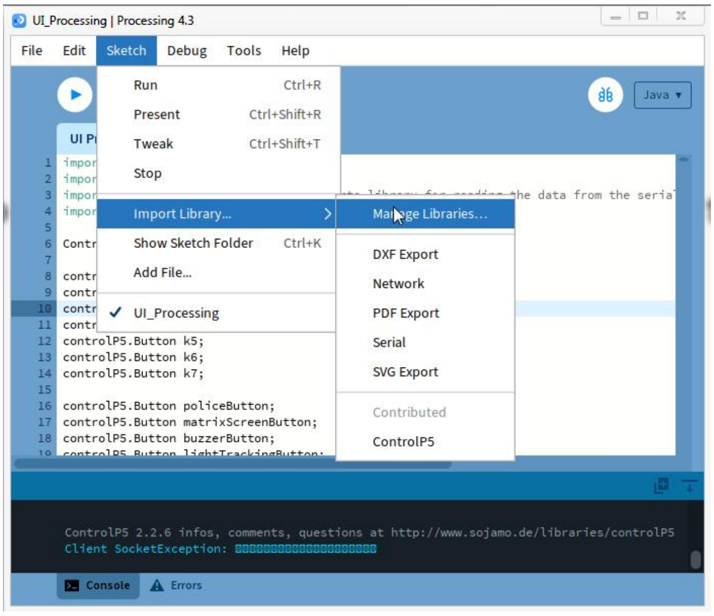
Head over to GitHub for earlier releases of Processing 4.
The list of revisions covers the differences between releases in detail.
For older versions (3.5.4 and earlier) see our list of stable releases.

Unzip the downloaded file to your computer. Click "**processing.exe**" as the figure below to run this software.



core
java
lib
modes
tools
processing.exe
processing-java.exe
revisions.md

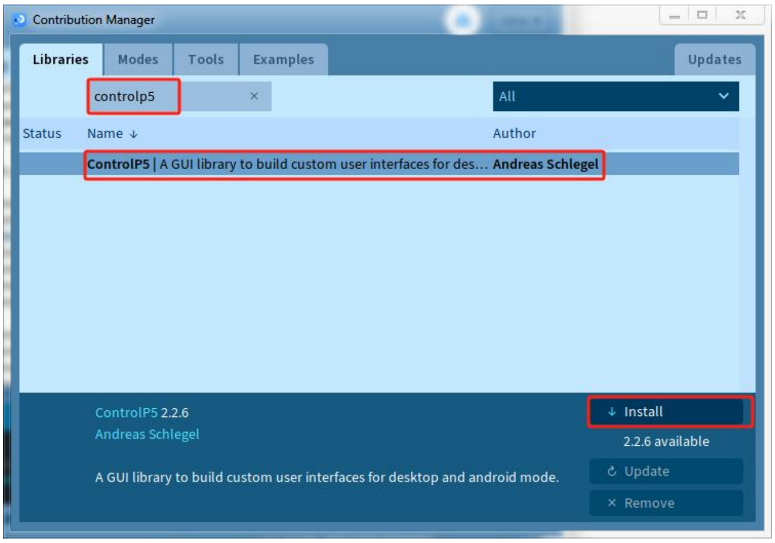
In the interface of Processing, click **Sketch** on Menu bar, select “**Import Library...**” and then click “**Manage Libraries...**”



```
controlP5.Button k5;
controlP5.Button k6;
controlP5.Button k7;

controlP5.Button policeButton;
controlP5.Button matrixScreenButton;
controlP5.Button buzzerButton;
```

```
ControlP5 2.2.6 infos, comments, questions at http://www.sojamo.de/libraries/controlP5
Client SocketException: ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
```

Enter “**ControlP5**” in the input field of the pop-up window. Click the searching result and then click



controlp5

Status	Name ↓	Author
	ControlP5 | A GUI library to build custom user interfaces for des...	**Andreas Schlegel**

ControlP5 2.2.6
Andreas Schlegel

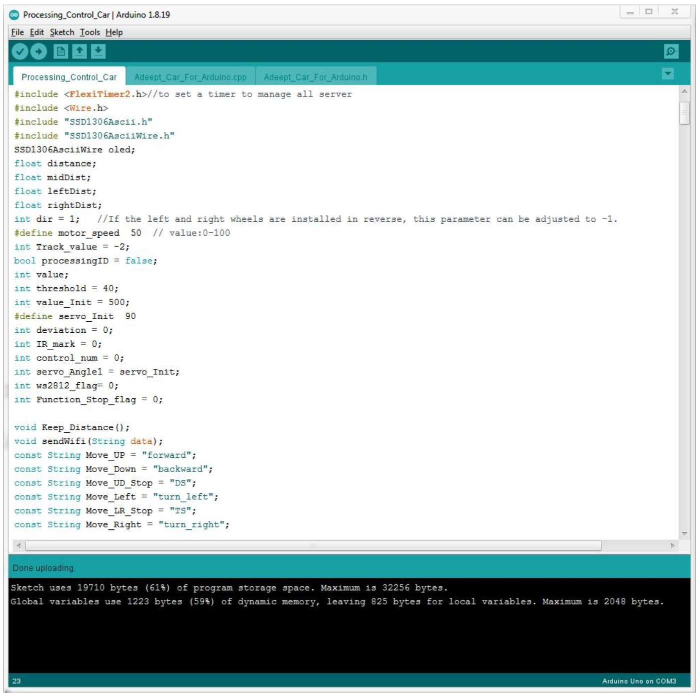
A GUI library to build custom user interfaces for desktop and android mode.

2.2.6 available

23.4 Upload the Program to the Arduino Car

1. Connect your computer and Adeept Robot Control Board (Arduino Board) with a USB cable.

2. Open ” **13_Remote_Control_Mecanum\Processing_Control\Processing_Control_Car”** folder in “/Code”, double-click “Processing_Control_Car.ino” .


Processing_Control_Car | Arduino 1.8.19
File Edit Sketch Tools Help
Processing_Control_Car
Adeept_Car_For_Arduino.cpp
Adeept_Car_For_Arduino.h
Done uploading.
23
Arduino Uno on COM3

```
#include <FlexiTimer2.h>//to set a timer to manage all server
#include <Wire.h>
#include "SSD1306Ascii.h"
#include "SSD1306AsciiWire.h"
SSD1306AsciiWire oled;
float distance;
float midDist;
float leftDist;
float rightDist;
int dir = 1;   //If the left and right wheels are installed in reverse, this parameter can be adjusted to -1.
#define motor_speed  50  // value:0-100
int Track_value = -2;
bool processingID = false;
int value;
int threshold = 40;
int value_Init = 500;
#define servo_Init  90
int deviation = 0;
int IR_mark = 0;
int control_num = 0;
int servo_Angle1 = servo_Init;
int ws2812_flag= 0;
int Function_Stop_flag = 0;

void Keep_Distance();
void sendWifi(String data);
const String Move_UP = "forward";
const String Move_Down = "backward";
const String Move_UD_Stop = "DS";
const String Move_Left = "turn_left";
const String Move_LR_Stop = "TS";
const String Move_Right = "turn_right";
```

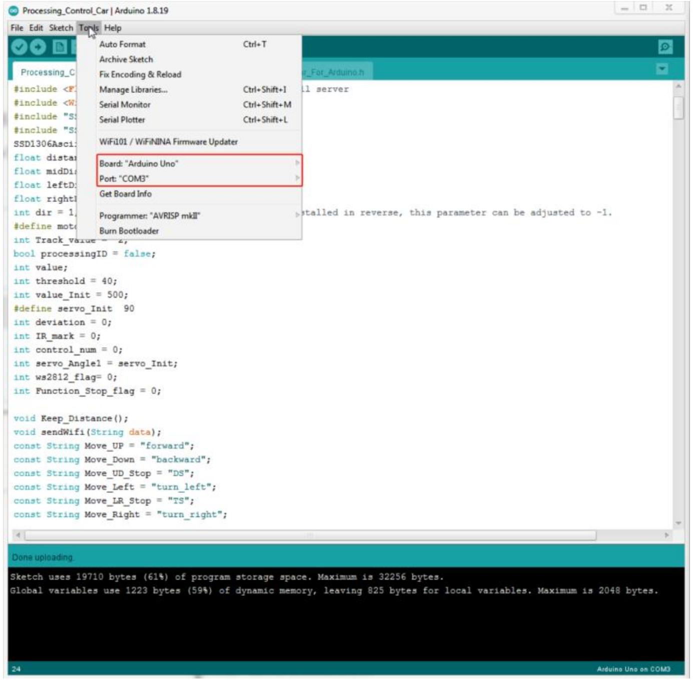
```
Sketch uses 19710 bytes (61%) of program storage space. Maximum is 32256 bytes.
Global variables use 1223 bytes (59%) of dynamic memory, leaving 825 bytes for local variables. Maximum is 2048 bytes.
```

3. Select development board and serial port.

Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

Port: Tools --->Port--->COMx

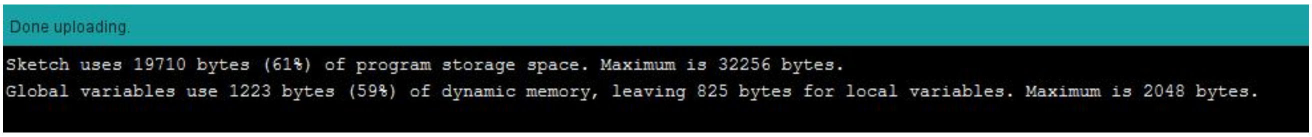
Note: The port number will be different in different computers.


Processing_Control_Car | Arduino 1.8.19
File Edit Sketch Tools Help
Processing_C
_For_Arduino.h
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM3"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Done uploading.
24
Arduino Uno on COM3

```
#include <F                         ll server
#include <W
#include "S
#include "S
SSD1306Asci
float distai
float midDi
float leftD
float right
int dir = 1,                        stalled in reverse, this parameter can be adjusted to -1.
#define mot
int Track_value = -2;
bool processingID = false;
int value;
int threshold = 40;
int value_Init = 500;
#define servo_Init  90
int deviation = 0;
int IR_mark = 0;
int control_num = 0;
int servo_Angle1 = servo_Init;
int ws2812_flag= 0;
int Function_Stop_flag = 0;

void Keep_Distance();
void sendWifi(String data);
const String Move_UP = "forward";
const String Move_Down = "backward";
const String Move_UD_Stop = "DS";
const String Move_Left = "turn_left";
const String Move_LR_Stop = "TS";
const String Move_Right = "turn_right";
```

```
Sketch uses 19710 bytes (61%) of program storage space. Maximum is 32256 bytes.
Global variables use 1223 bytes (59%) of dynamic memory, leaving 825 bytes for local variables. Maximum is 2048 bytes.
```

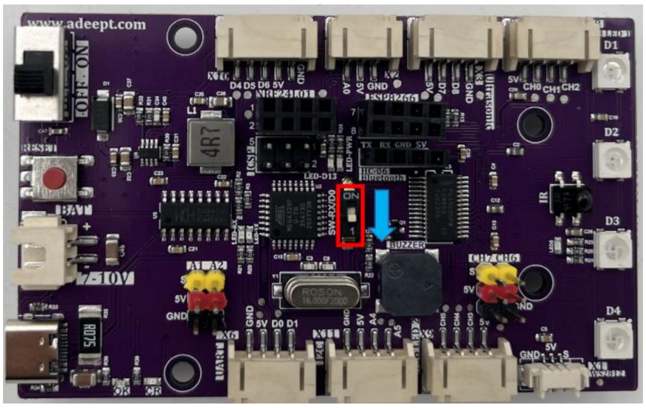
4. After opening, click → to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



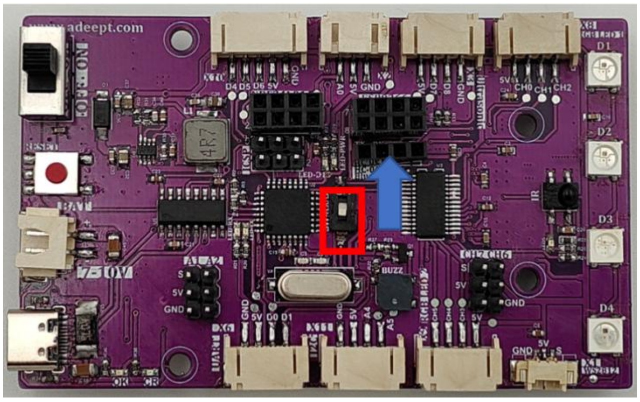
```
Done uploading.
Sketch uses 19710 bytes (61%) of program storage space. Maximum is 32256 bytes.
Global variables use 1223 bytes (59%) of dynamic memory, leaving 825 bytes for local variables. Maximum is 2048 bytes.
```

Note: When the ESP8266 module transmits data to the Arduino, it needs to occupy the RX interface of the Arduino, and when the Arduino uploads the program, it also needs to occupy the RX interface. The RX interface cannot satisfy both functions at the same time, so a switch is needed to distinguish them.

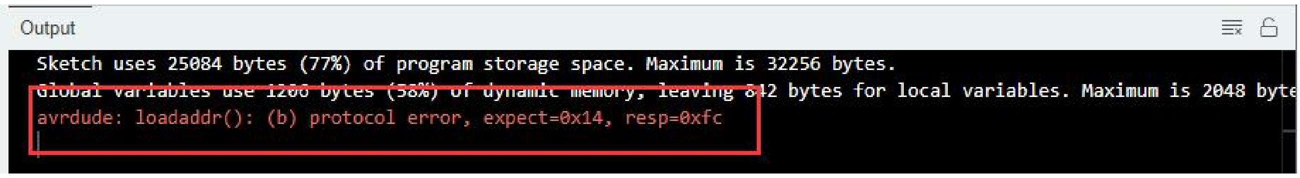
When the switch is flipped downward, the ESP8266 module is disconnected from the RX interface, and the program can be uploaded normally.



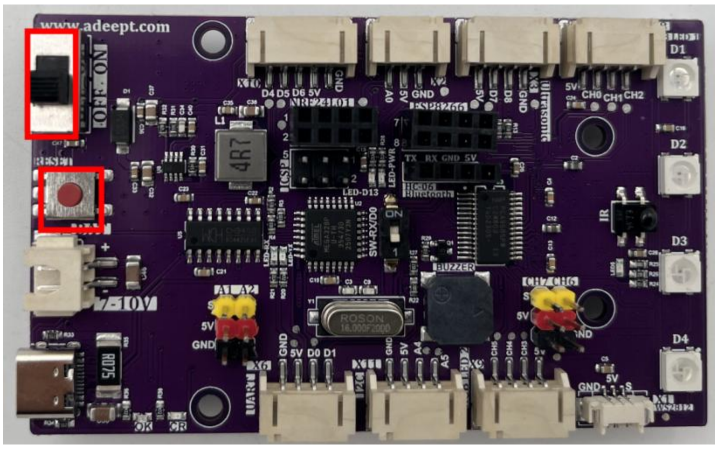
When the switch is flipped upward, the RX interface is connected to the ESP8266, and the ESP8266 module will continue to occupy the RX interface. At this time, the program cannot be uploaded normally.



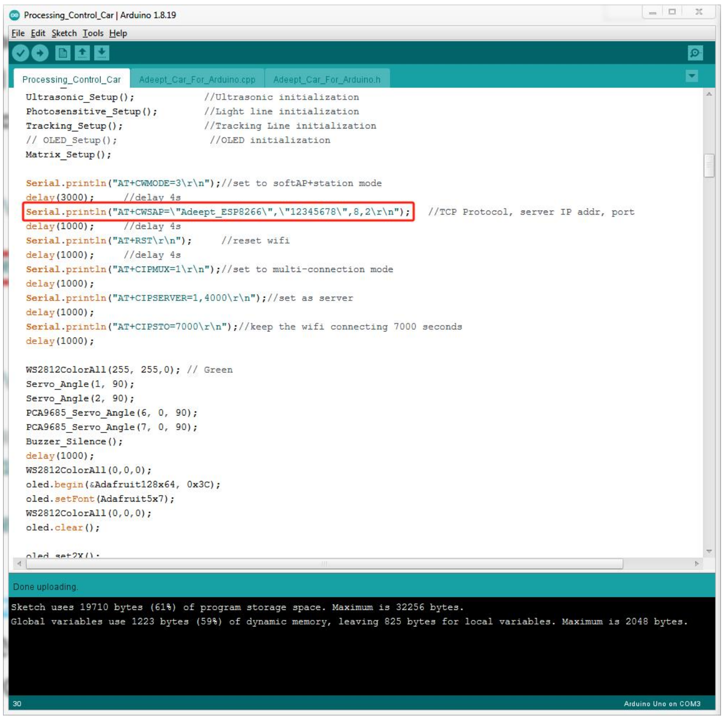
If the following error messages also appear, this may cause the program upload to fail. Please try to press the "RESET" button, or try to turn off the power switch and then turn on the power switch.



```
Sketch uses 25084 bytes (77%) of program storage space. Maximum is 32256 bytes.
Global variables use 1206 bytes (58%) of dynamic memory, leaving 842 bytes for local variables. Maximum is 2048 byte
avrdude: loadaddr(): (b) protocol error, expect=0x14, resp=0xfc
```



5. After uploading successfully, the mobile phone can detect a WiFi name named "Adeept_ESP8266", and the WiFi password is "12345678". The WiFi name and password can be modified through the procedure below. Modifications to the program are not recommended for initial use.

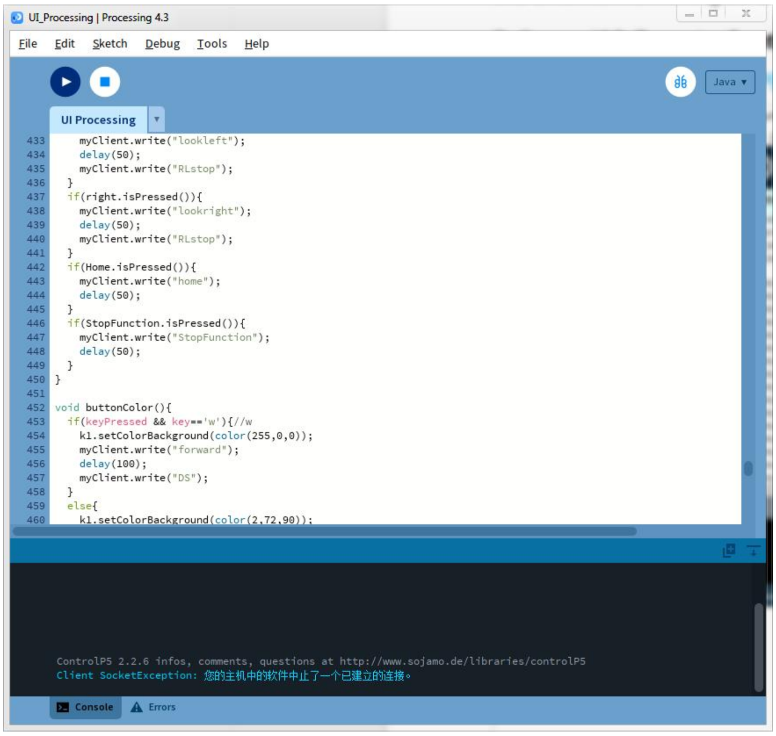

Processing_Control_Car | Arduino 1.8.19
File Edit Sketch Tools Help
Processing_Control_Car
Adeept_Car_For_Arduino.cpp
Adeept_Car_For_Arduino.h
Done uploading.
30
Arduino Uno on COM3

```
  Ultrasonic_Setup();          //Ultrasonic initialization
  Photosensitive_Setup();      //Light line initialization
  Tracking_Setup();            //Tracking Line initialization
  // OLED_Setup();              //OLED initialization
  Matrix_Setup();

  Serial.println("AT+CWMODE=3\r\n");//set to softAP+station mode
  delay(3000);     //delay 4s
  Serial.println("AT+CWSAP=\"Adeept_ESP8266\",\"12345678\",8,2\r\n");   //TCP Protocol, server IP addr, port
  delay(1000);     //delay 4s
  Serial.println("AT+RST\r\n");     //reset wifi
  delay(1000);     //delay 4s
  Serial.println("AT+CIPMUX=1\r\n");//set to multi-connection mode
  delay(1000);
  Serial.println("AT+CIPSERVER=1,4000\r\n");//set as server
  delay(1000);
  Serial.println("AT+CIPSTO=7000\r\n");//keep the wifi connecting 7000 seconds
  delay(1000);

  WS2812ColorAll(255, 255,0); // Green
  Servo_Angle(1, 90);
  Servo_Angle(2, 90);
  PCA9685_Servo_Angle(6, 0, 90);
  PCA9685_Servo_Angle(7, 0, 90);
  Buzzer_Silence();
  delay(1000);
  WS2812ColorAll(0,0,0);
  oled.begin(&Adafruit128x64, 0x3C);
  oled.setFont(Adafruit5x7);
  WS2812ColorAll(0,0,0);
  oled.clear();

  oled.set2X();
```

```
Sketch uses 19710 bytes (61%) of program storage space. Maximum is 32256 bytes.
Global variables use 1223 bytes (59%) of dynamic memory, leaving 825 bytes for local variables. Maximum is 2048 bytes.
```

6. Use PC to connect to "Adeept_ESP8266" WiFi. Since this WiFi can only be used for communication between the PC and ESP8266, after the PC is connected to WiFi, it cannot access the external network (you cannot use the PC to access the Internet).

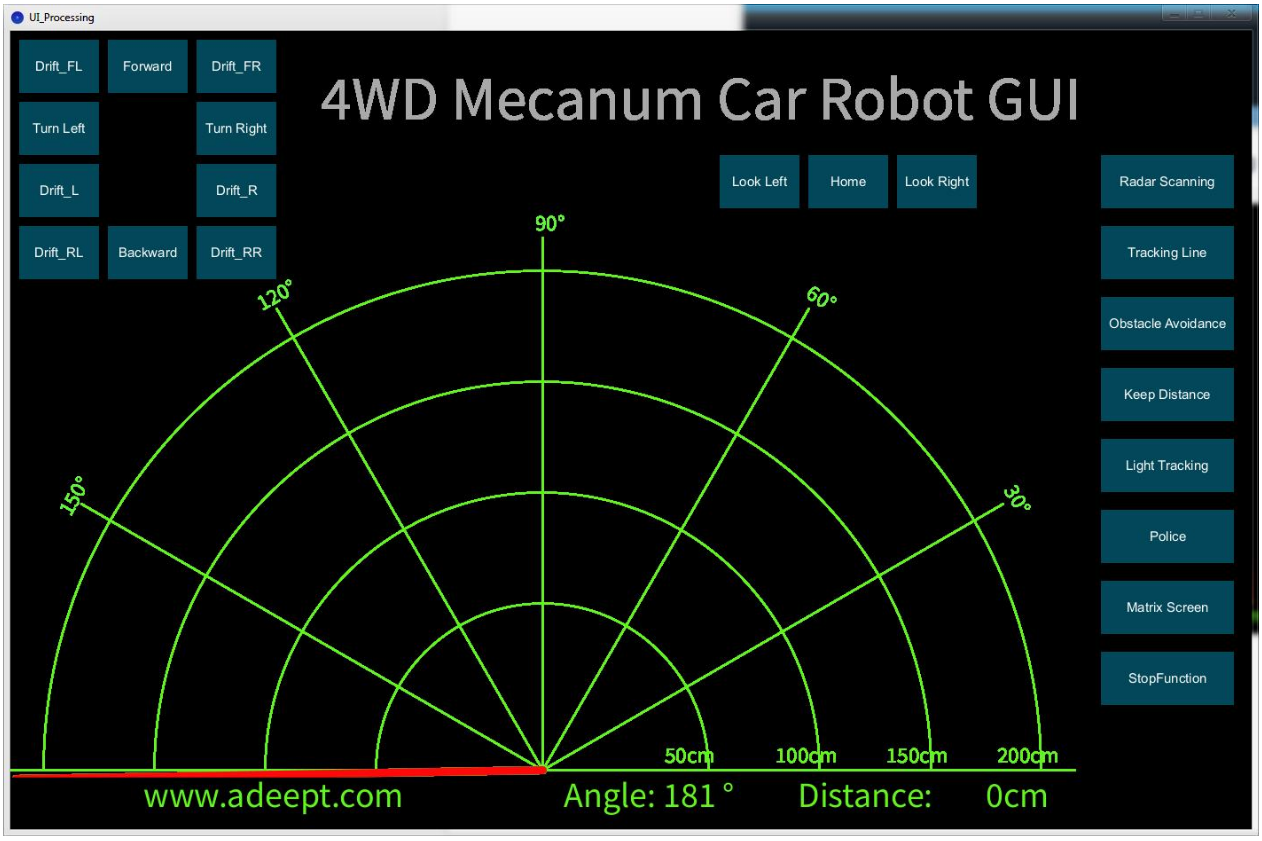
7. Open "**13_Remote_Control_Mecanum\Processing_Control\UI_Processing**" folder in "**/Code**", double-click "UI_Processing.pde"



```
    myClient.write("lookleft");
    delay(50);
    myClient.write("RLstop");
  }
  if(right.isPressed()){
    myClient.write("lookright");
    delay(50);
    myClient.write("RLstop");
  }
  if(Home.isPressed()){
    myClient.write("home");
    delay(50);
  }
  if(StopFunction.isPressed()){
    myClient.write("StopFunction");
    delay(50);
  }
}

void buttonColor(){
  if(keyPressed && key=='w'){//w
    k1.setColorBackground(color(255,0,0));
    myClient.write("forward");
    delay(100);
    myClient.write("DS");
  }
  else{
    k1.setColorBackground(color(2,72,90));
```

```
ControlP5 2.2.6 infos, comments, questions at http://www.sojamo.de/libraries/controlP5
Client SocketException: 您的主机中的软件中止了一个已建立的连接。
```

8. Click '**Run**' and you will see the information displayed on the Processing control interface.



The operation instructions are as follows:

Button	Instruction	Describe
Forward/The **W** key on the keyboard	forward	Car moving forward
Back/The **S** key on the keyboard	backward	Car backwards
Turn Left/The **A** key on the keyboard	turn_left	Turn left in the car
Turn right/The **D** key on the keyboard	turn_right	Turn right in the car
LEFT	lookleft/RLstop	Head left turn
RIGHT	lookright/RLstop	Head to the right

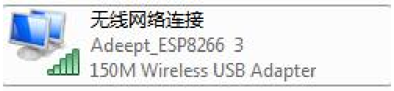
Drift_FL/The **Z** key on the keyboard	drift_front_left/DRS	The car drifts to the left front.
Drift_FR/The **X** key on the keyboard	drift_front_right/DRS	The car drifts to the right front.
Drift_L/The **C** key on the keyboard	drift_left/DRS	The car drifts to the left.
Drift_R/The **V** key on the keyboard	drift_right/DRS	The car drifts to the right.
Drift_RL/The **B** key on the keyboard	drift_rear_left/DRS	The car drifts to the left rear.
Drift_RR/The **N** key on the keyboard	drift_rear_right/DRS	The car drifts to the right rear.
Home	home	Head back to the middle position
Radar Scanning	scan	Used to perform the ultrasound scan function and display the scan results
Obstacle Avoidance	automatic	Switch to automatic obstacle avoidance mode
Police	police	Make the WS2812 LED lights on the robot flash alternately in red and blue.
Tracking Line	trackLine	Implement line tracking function using a 3-channel infrared module.
Light Tracking	lightTracking	Light Tracking Function
Matrix Screen	matrix	Matrix screen lights up

Keep Distance	KD	Keep distance
Stop Function	StopFunction	Stop Function

Q&A

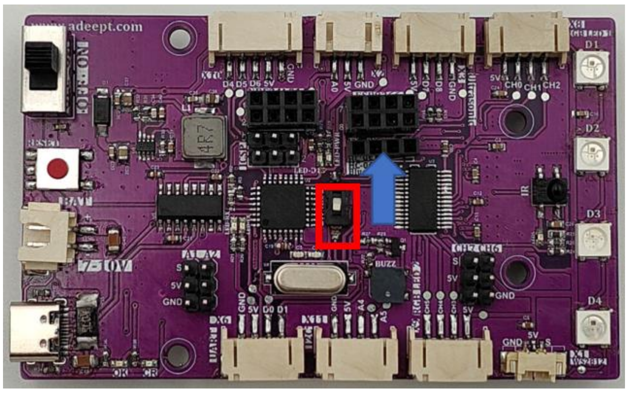
If the user interface cannot be opened or operated:

1. Please check whether you are connected to the WiFi named "**Adeept_ESP8266**".



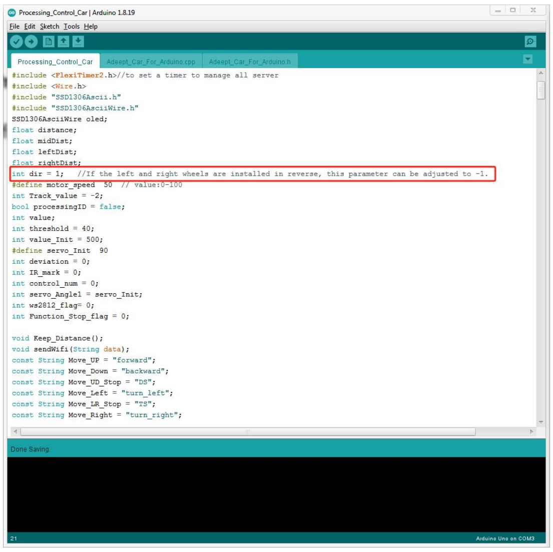
无线网络连接
Adeept_ESP8266 3
150M Wireless USB Adapter

2. Check whether the RX/D0 switch is correctly turned on.



If the rotation direction of the wheels is opposite to the actual operation, please perform the following steps:

Go to the ‘**Code\13_Remote_Control\Processing_Control\Processing_Control_Car**’ folder, open the ‘**Processing_Control_Car.ino**’ file, and modify the parameter ‘**dir=-1**’ .



Processing_Control_Car | Arduino 1.8.19
File Edit Sketch Tools Help
Processing_Control_Car
Adeept_Car_For_Arduino.cpp
Adeept_Car_For_Arduino.h

```
#include <FlexiTimer2.h>//to set a timer to manage all server
#include <Wire.h>
#include "SSD1306Ascii.h"
#include "SSD1306AsciiWire.h"
SSD1306AsciiWire oled;
float distance;
float midDist;
float leftDist;
float rightDist;
int dir = 1;   //If the left and right wheels are installed in reverse, this parameter can be adjusted to -1.
#define motor_speed  50  // value:0-100
int Track_value = -2;
bool processingID = false;
int value;
int threshold = 40;
int value_Init = 500;
#define servo_Init  90
int deviation = 0;
int IR_mark = 0;
int control_num = 0;
int servo_Angle1 = servo_Init;
int ws2812_flag= 0;
int Function_Stop_flag = 0;

void Keep_Distance();
void sendWifi(String data);
const String Move_UP = "forward";
const String Move_Down = "backward";
const String Move_UD_Stop = "DS";
const String Move_Left = "turn_left";
const String Move_LR_Stop = "TS";
const String Move_Right = "turn_right";
```

Done Saving.