Lesson 17 Remote Control- Introduction to IR

17.1 Overview

In the last lesson we know how to use IR module to pass data with Adeept Robot Control Board. In this course we learn how to use infrared remote control to control the car.

17.2 Principle Introduction

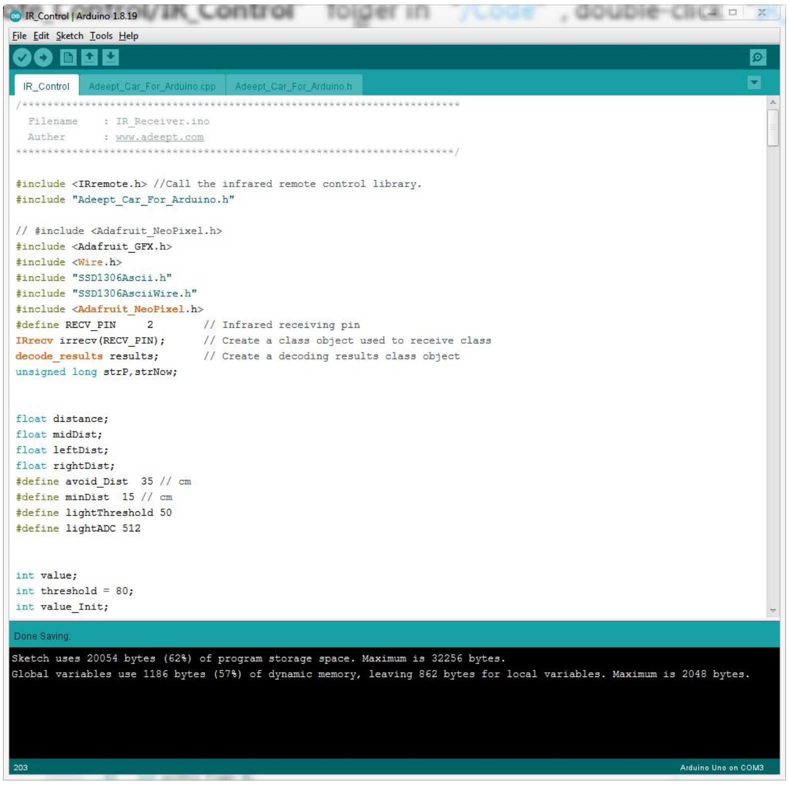
The corresponding relationship between the buttons of the infrared remote control and the functions of the car is as follows:

Button	Function	Button	Function
⇧	Forward	5	
⇩	Backward	6	Matrix screen lights up
⇦	Left	7	
⇨	Right	8	Line Tracking Function
OK	Stop Function	9	
1	Servo 1 turn left	*	Avoid Obstacles Function
2	Servo 1 turn to 90	0	Keep distance
3	Servo 1 turn right	#	Light Tracking Function
4	Alarm Light		

17.3 Running Infrared Control

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

2. Open “ **13_Remote_Control_Ordinary/IR_Control** ” folder in “ /Code ” , double-click “IR_Control.ino” .



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

```
/**********************************************************************
  Filename    : IR_Receiver.ino
  Auther      : www.adeept.com
**********************************************************************/

#include <IRremote.h> //Call the infrared remote control library.
#include "Adeept_Car_For_Arduino.h"

// #include <Adafruit_NeoPixel.h>
#include <Adafruit_GFX.h>
#include <Wire.h>
#include "SSD1306Ascii.h"
#include "SSD1306AsciiWire.h"
#include <Adafruit_NeoPixel.h>
#define RECV_PIN     2        // Infrared receiving pin
IRrecv irrecv(RECV_PIN);      // Create a class object used to receive class
decode_results results;       // Create a decoding results class object
unsigned long strP,strNow;


float distance;
float midDist;
float leftDist;
float rightDist;
#define avoid_Dist  35 // cm
#define minDist  15 // cm
#define lightThreshold 50
#define lightADC 512


int value;
int threshold = 80;
int value_Init;
```

Done Saving.

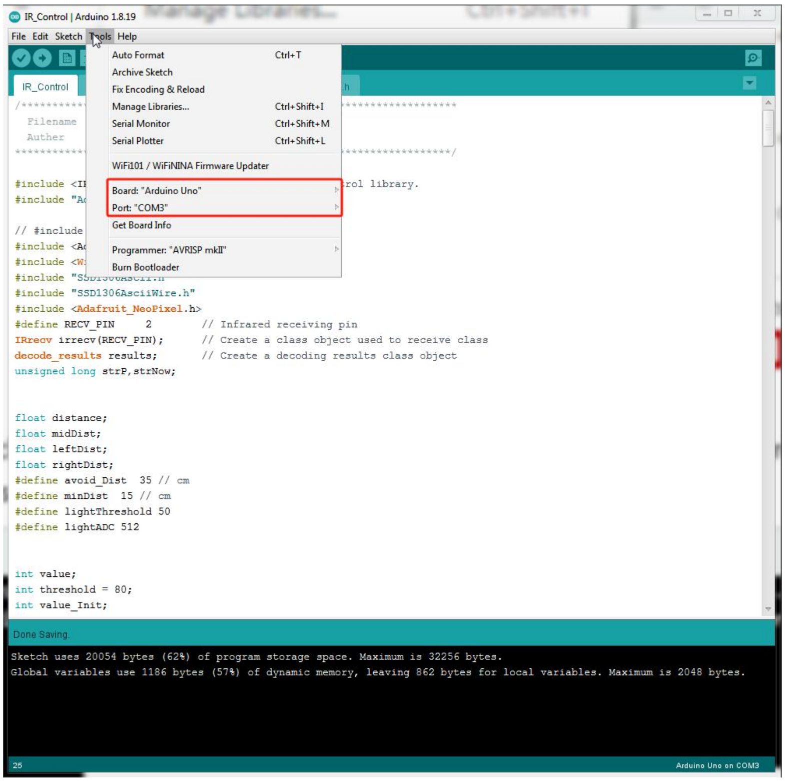
```
Sketch uses 20054 bytes (62%) of program storage space. Maximum is 32256 bytes.
Global variables use 1186 bytes (57%) of dynamic memory, leaving 862 bytes for local variables. Maximum is 2048 bytes.
```

3. Select development board and serial port.

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

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

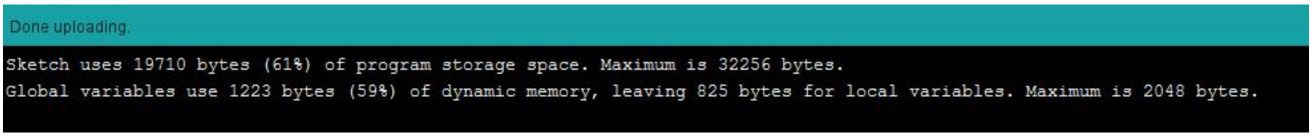
Note: The port number will be different in different computers.


IR_Control | Arduino 1.8.19
File Edit Sketch Tools Help
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
IR_Control
25
Arduino Uno on COM3

```
/***********                                        *******************
  Filename
  Auther
************                                        ******************/

#include <I                                         rol library.
#include "A

// #include
#include <A
#include <W
#include "S
#include "SSD1306AsciiWire.h"
#include <Adafruit_NeoPixel.h>
#define RECV_PIN     2        // Infrared receiving pin
IRrecv irrecv(RECV_PIN);      // Create a class object used to receive class
decode_results results;       // Create a decoding results class object
unsigned long strP,strNow;


float distance;
float midDist;
float leftDist;
float rightDist;
#define avoid_Dist  35 // cm
#define minDist  15 // cm
#define lightThreshold 50
#define lightADC 512


int value;
int threshold = 80;
int value_Init;
```

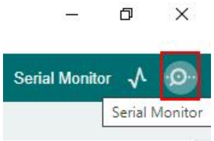
```
Done Saving.
Sketch uses 20054 bytes (62%) of program storage space. Maximum is 32256 bytes.
Global variables use 1186 bytes (57%) of dynamic memory, leaving 862 bytes for local variables. Maximum is 2048 bytes.
```

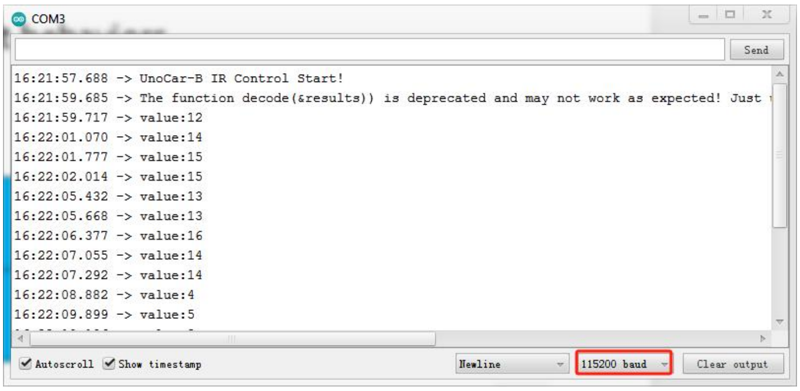
4. After opening, click → to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



```
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.
```

5. Click Serial Monitor, Set the baud rate as 115200.Then operate the car, and you can see the commands received by the car in this window.





```
16:21:57.688 -> UnoCar-B IR Control Start!
16:21:59.685 -> The function decode(&results)) is deprecated and may not work as expected! Just
16:21:59.717 -> value:12
16:22:01.070 -> value:14
16:22:01.777 -> value:15
16:22:02.014 -> value:15
16:22:05.432 -> value:13
16:22:05.668 -> value:13
16:22:06.377 -> value:16
16:22:07.055 -> value:14
16:22:07.292 -> value:14
16:22:08.882 -> value:4
16:22:09.899 -> value:5
```

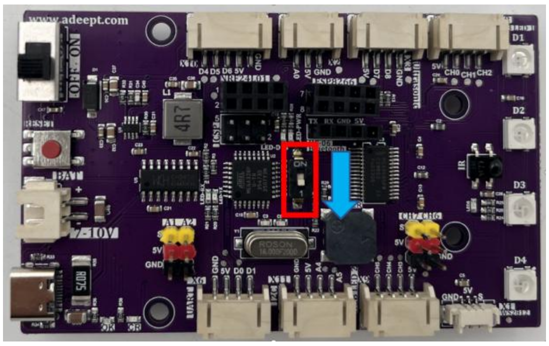
6. Use the IR remote to aim at the IR receiver on the expansion board. Press different buttons on the infrared remote control, you can see that the car has different behaviors.

Button	Function	Button	Function
⇧	Forward	5	
⇩	Backward	6	Matrix screen lights up
⇦	Left	7	
⇨	Right	8	Line Tracking Function
OK	Stop Function	9	
1	Servo 1 turn left	*	Avoid Obstacles Function
2	Servo 1 turn to 90	0	Keep distance
3	Servo 1 turn right	#	Light Tracking Function
4	Alarm Light		
Some Functions may require multiple clicks to stop.			

Please familiarize yourself with the button functions of the infrared remote control first. Please see the list above for the button functions.

It is recommended to raise the vehicle body during testing so that the wheels are suspended in the air.

Note: If the upload program fails, a red error message appears. Please confirm whether the RX/D0 switch is in the correct position. Toggle down the paddle (white) of the switch in the picture below.

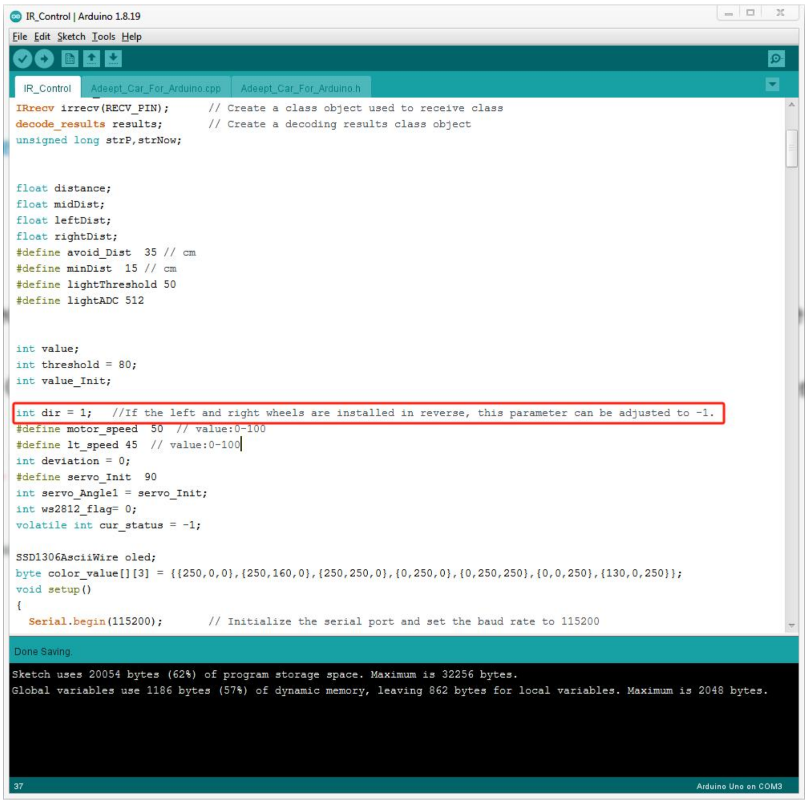


After assembling the car, please use the 18650 battery to provide power when uploading the program, otherwise the program may not be uploaded successfully due to excessive load.

Q&A

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_Ordinary\IR_Control**’ folder, open the ‘**IR_Control.ino**’ file, and modify the parameter ‘**dir=-1**’ .


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

```
IRrecv irrecv(RECV_PIN);      // Create a class object used to receive class
decode_results results;       // Create a decoding results class object
unsigned long strP,strNow;


float distance;
float midDist;
float leftDist;
float rightDist;
#define avoid_Dist  35 // cm
#define minDist  15 // cm
#define lightThreshold 50
#define lightADC 512


int value;
int threshold = 80;
int value_Init;

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
#define lt_speed 45  // value:0-100
int deviation = 0;
#define servo_Init  90
int servo_Angle1 = servo_Init;
int ws2812_flag= 0;
volatile int cur_status = -1;

SSD1306AsciiWire oled;
byte color_value[][3] = {{250,0,0},{250,160,0},{250,250,0},{0,250,0},{0,250,250},{0,0,250},{130,0,250}};
void setup()
{
  Serial.begin(115200);       // Initialize the serial port and set the baud rate to 115200
```

Done Saving.

```
Sketch uses 20054 bytes (62%) of program storage space. Maximum is 32256 bytes.
Global variables use 1186 bytes (57%) of dynamic memory, leaving 862 bytes for local variables. Maximum is 2048 bytes.
```