

Expand course ---8.IR control expand

Note:

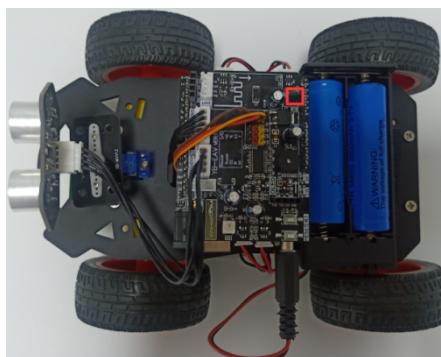
1. In order to avoid the interference of sunlight on infrared sensor, we need to carry out this experiment indoors.
2. If you switch between different modes will have a short delay, which is normal.
3. During remote control, the infrared remote control needs to face the infrared receiver on the expansion board.

1. Learning goal

In this lesson, we will learn how to control the car by infrared controller.

2. Preparation

2.1 The position of the infrared receiver on the robot car. As shown below.



2.2 The pin of UNO board is connected the infrared receiver.

3. Principle of experimental

3.1 From the hardware interface manual, we can know that infrared receiver are driven by Uno board.

Classification	Function	The number of Drive chip PCA9685	Drive Method	Connection with CPU	Uno board
Left Motor	Left front motor forward	LINB(13)	PCA9685	I2C_SDA/I2C_SCL	A4/A5
	Left front motor reverse	LINA(12)			
	Left rear motor forward	RINB(15)			
	Left rear motor reverse	RINA(14)			
Right Motor	Right front motor forward	LED10			
	Right front motor reverse	LED11			
	Right rear motor forward	LED8			
	Right rear motor reverse	LED9			
Servo	Control S1	LED0	Uno board drive directly		A0
	Control S2	LED1			A1
	Control S3	LED2			A2
	Control S4	S1 (3)			12
LOGO light	Control bluelight	LED7			11
Tracking sensor	Left tracking sensor				7
	Middle tracking sensor				IR
	Right tracking sensor				A3
Ultrasonic sensor	Ultrasonic Echo		Uno board drive directly		0
	Ultrasonic RGB light				1
Key	K1				6
IR	IR control				10
Bluetooth interface	RX				
	TX				
On board RGB Light	RGB Light on expansion board				
	Control buzzer				

3.2 Introduction of infrared remote controller :

This remote controller with a standard **38Khz** modulation frequency to accommodate a variety of infrared receivers on the market. It possess 0~9 number keys, 4 direction keys, acceleration and deceleration, left and right rotation, lighting, sound and other buttons, which can be easily applied to various smart cars and development boards. Built-in universal 3V button battery, long time and easy to replace.

3.3 Introduction of infrared receiver :

This infrared receiver comes with an iron-shell shield cover to prevent external interference(Some infrared receivers do not have). Built-in dedicated IC, with infrared remote control, can be applied to audio-visual equipment, home appliances, microcontroller learning, robot remote control and so on.

3.4 Introduction of infrared:

Electromagnetic waves having a wavelength from 760 nm to 400 um in the spectrum are called infrared rays, which is an invisible light. At present, almost all video and audio equipment can be remotely controlled by infrared remote

control, such as televisions, air conditioners, DVD players, etc., and the infrared remote control can be seen.

3.5 Introduction to the principle of infrared communication

User code:0XFF

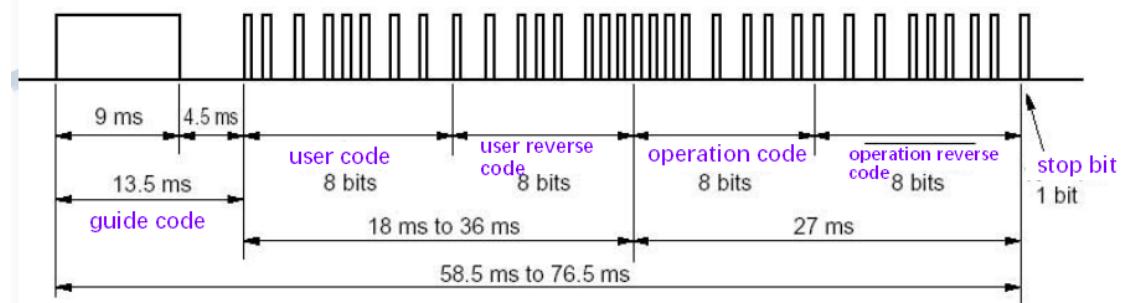


Infrared sending and receiving codes are divided into: **guide code**, **user code**, **user reverse code**, **operation code(data code)**, **operation reverse code(data reverse code)**.

Below we will introduce these kinds of encoded waveform:

For the part of infrared transmission, the infrared remote control we provided has been set up. Therefore, we do not explain the knowledge about infrared transmission here, but mainly explain some points of infrared reception.

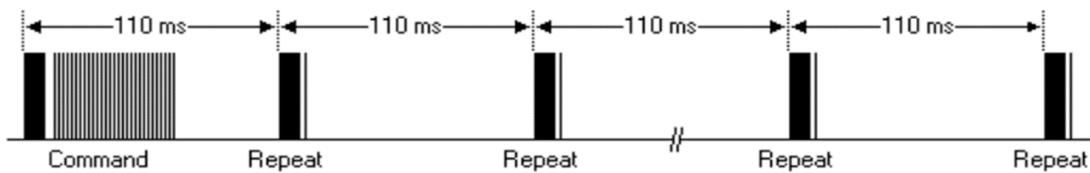
We mainly use the **NCE** protocol:



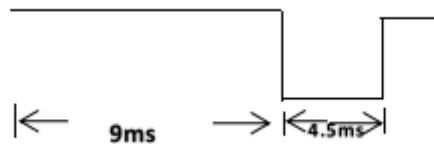
The protocol stipulates that the lower bits are transmitted first.

A string of information first sends a 9ms high pulse of AGC (Automatic Gain Control) . Then, send a 4.5ms low level. Next, send four-byte address code and a command code. These four bytes are : Address code, address reverse code; command code; command reverse code.

Note: If you keep pressing that button, a series of messages can only be sent once. If you keep pressing, the repeating code is sent with a period of 110ms. Waveform as shown below.



1) Guide code: The guide code defined by the uPD6121G of the NEC protocol is 9ms high level + 4.5ms low level, waveform as shown in the figure below:

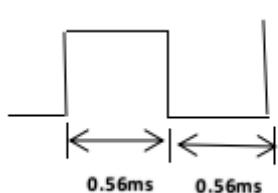


2) User code and operation code: The user code and operation code defined by the uPD6121G of the NEC protocol is:

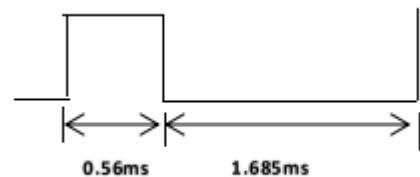
"Logic 0" : 0.56ms high level + 0.565ms low level;

"Logic1" : 0.56ms high level + 1.685ms low level;

Waveform as shown below :

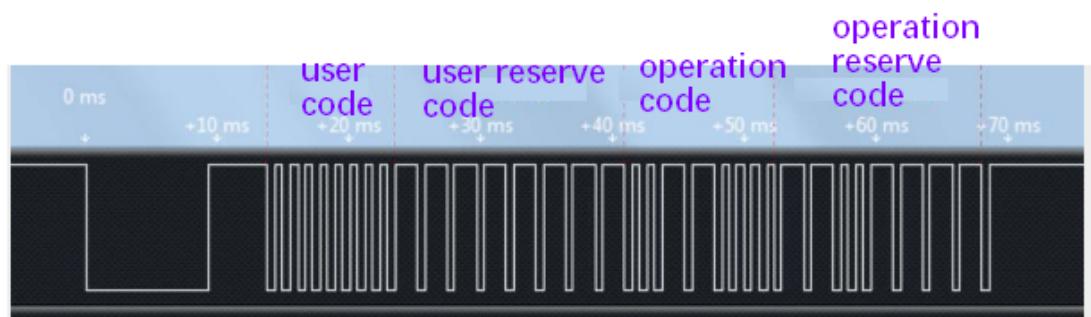


Logic 0 --waveform



Logic 1--waveform

The wave output after passing through the infrared receiver, as shown below:



The process of infrared receiving and controlling the car is as follows:

1) Generate a falling edge, enter the interrupt function of external interrupt 0, and check whether the IO port is still low level after delay time. If it is low level, wait for the low level of 9ms to pass. Wait for the 9ms low level to pass, and

then wait for the 4.5ms high level to pass.

- 2) Start receiving the 4 sets of data transmitted, wait for the low level of 560us to pass, and detect the duration of the high level. If it exceeds 1.12ms, it is high level (the duration of the high level is 1.69ms, the low level Duration is 5.65ms.)
- 3) Detect whether the received data is the same as the data reverse code (operation reverse code), and wait for the same data. If it is the same, it is proved that the corresponding data sent by the infrared remote control is received. Then, the car moves according to the function set in the program.

4. About code

For the code of this course, please refer to: [IR_Control_expand.ino](#) in the **IR_Control_expand** folder.

5. Compiling and downloading code

- 5.1 We need to open the **IR_Control_expand .ino** file by Arduino IDE software. Then click“√”under the menu bar to compile the code, and wait for the word "**Done compiling**" in the lower left corner, as shown in the figure below.

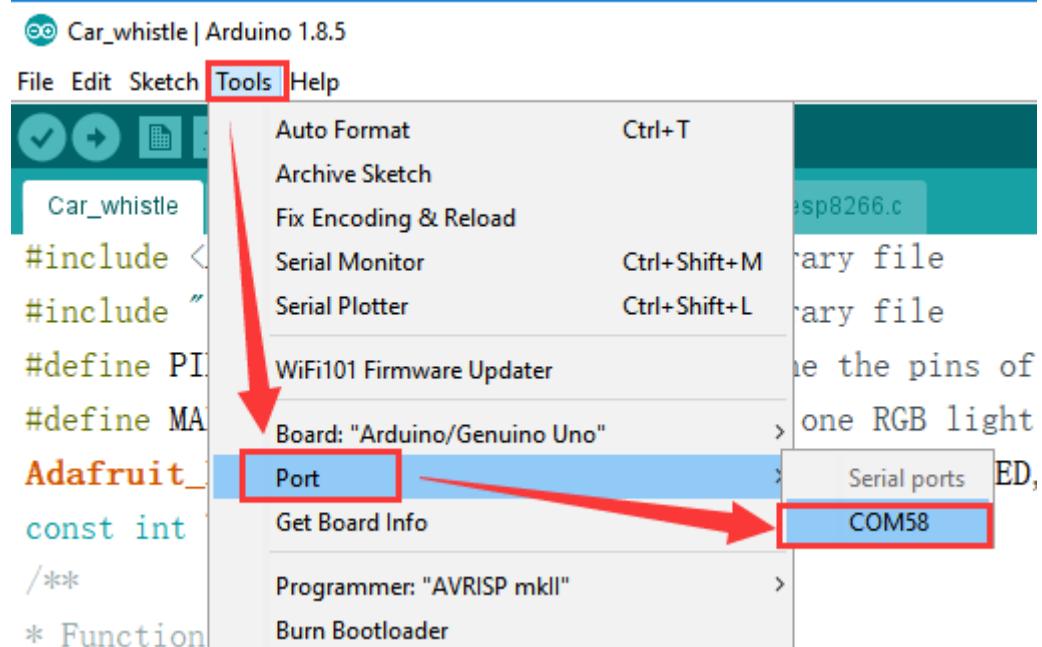
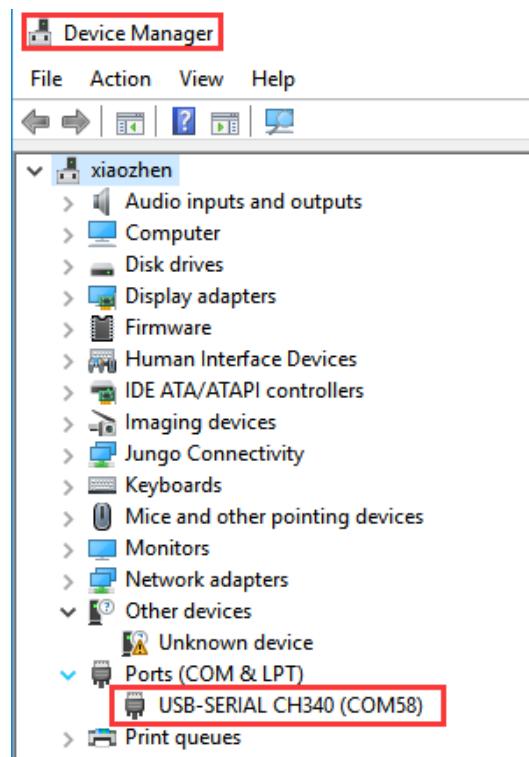
```

IR_Control_expand | Arduino 1.8.5
File Edit Sketch Tools Help
✓ ↻ ⌂ ⌄ ⌅ ⌆
IR_Control_expand Adafruit_NeoPixel.cpp Adafruit_NeoPixel.h Adafruit_PWMServoD
* @brief
* @details
* @par History NO
*/
#include <Arduino.h>
#include "Adafruit_PWMServoDriver.h"
#include "Adafruit_NeoPixel.h"
#include "IRremote.h"
#include "RGBLed.h"

#define RGB_GREEN 0xFF0000 //Define different color(gr
#define RGB_RED 0x00FF00
#define RGB_BLUE 0x0000FF
<
Done compiling.

```

- 5.2 In the menu bar of Arduino IDE, we need to select **【Tools】---【Port】** --- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



5.3 After the selection is completed, you need to click “→”under the menu bar to upload the code to the UNO board. When the word “**Done uploading**” appears in the lower left corner, the code has been successfully uploaded to the UNO board, as shown in the figure below.

IR_Control_expand | Arduino 1.8.5

File Edit Sketch Tools Help

IR_Control_expand Adafruit_NeoPixel.cpp Adafruit_NeoPixel.h Adafruit_PWMservoDriver.cpp Adafruit_I2CInterrupt.h

```
* @brief
* @details
* @par History NO
*/
#include <Arduino.h>
#include "Adafruit_PWMservoDriver.h"
#include "Adafruit_NeoPixel.h"
#include "IRremote.h"
#include "RGBLed.h"

#define RGB_GREEN 0xFF0000 //Define different color(green, red, blue)
#define RGB_RED 0x00FF00
#define RGB_BLUE 0x0000FF

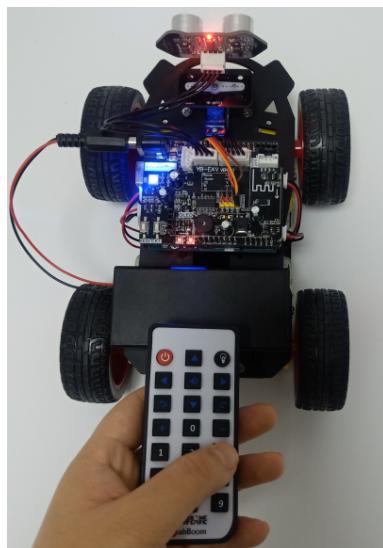
<<
```

Done uploading.

Sketch uses 14314 bytes (44%) of program storage space. Maximum is 32256 bytes.

6. Experimental phenomena

After the program is downloaded, we can control robot car by infrared controller. As shown below.



The following functions are controlled by the infrared remote controller.

