Infrared remote control

This tutorial is a comprehensive experiment combining multiple peripherals. You can understand individual peripherals before conducting this experiment.

1. Software-Hardware

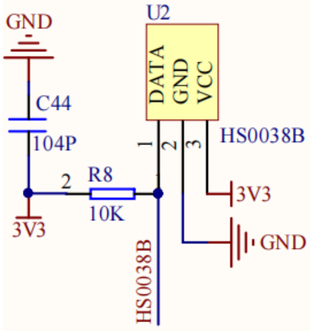
- **KEIL**
- **MSPM0G3507 Development Board**
- **Infrared Remote Control**
- **Type-C data cable or DAP-Link**

  For program download or simulation to the development board

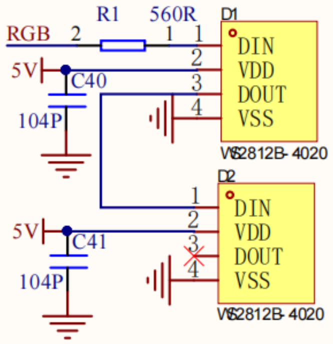
2. Brief Principle

2.1 Hardware Schematic Diagram

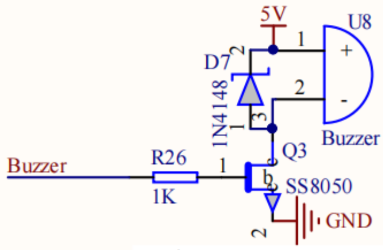
Infrared remote control



RGB light

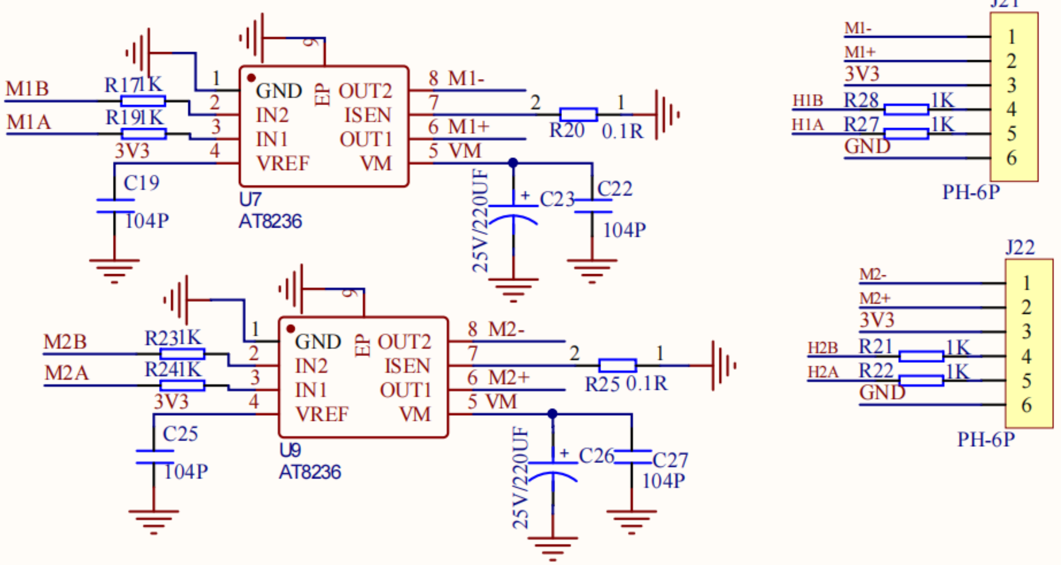


Buzzer

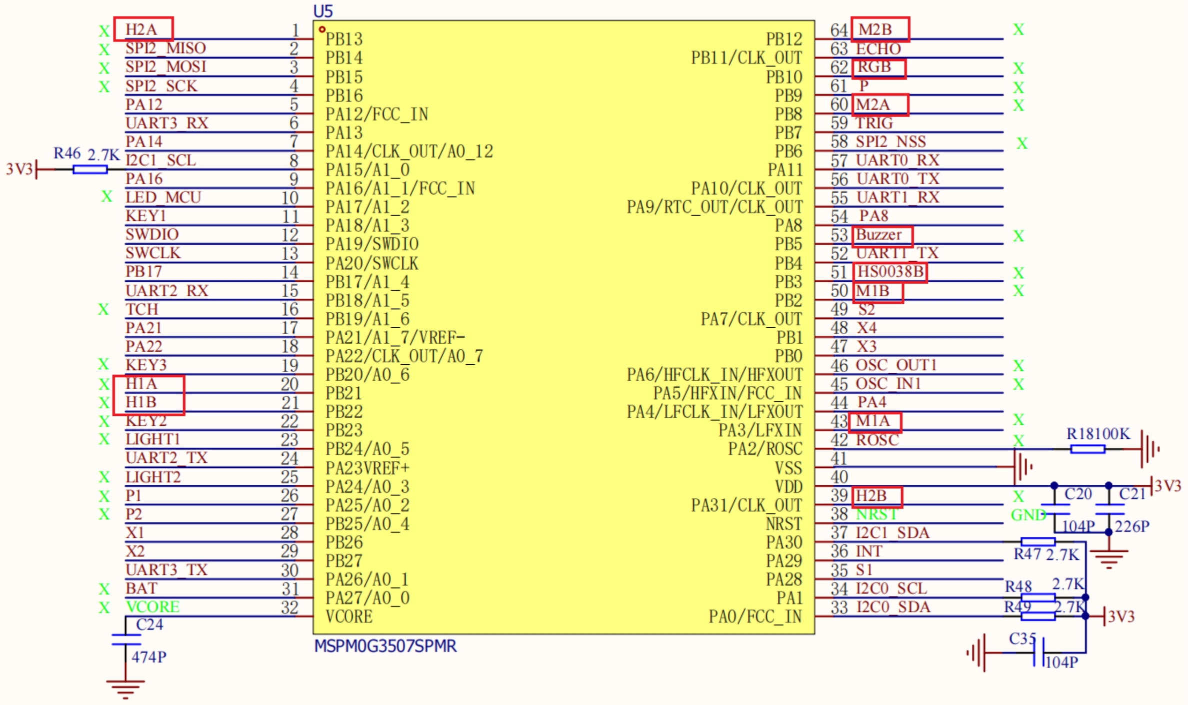


5V Active Buzzer
(High and Low Level Control)

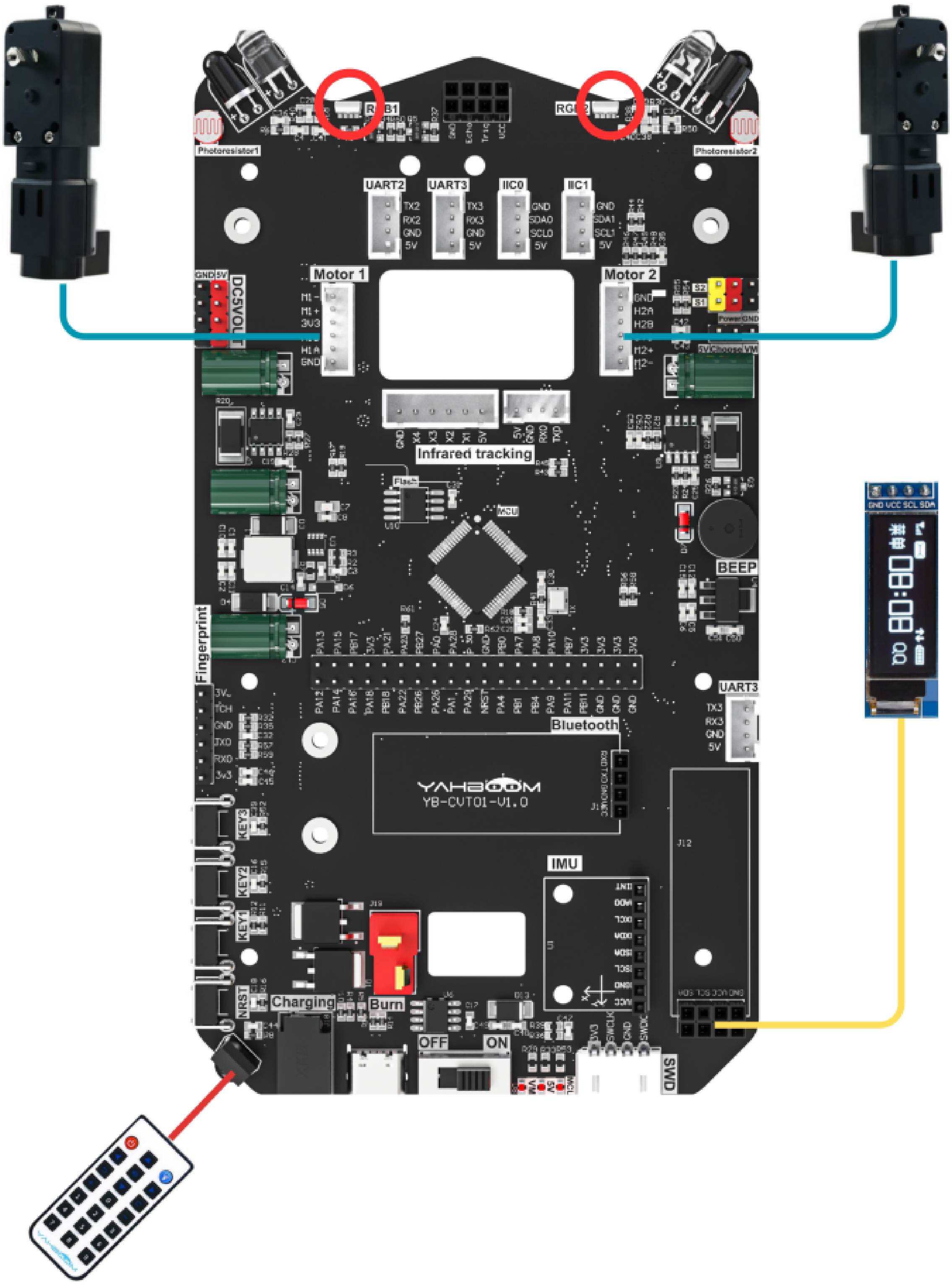
Motor Drive



MCU



2.2 Physical Connection Diagram

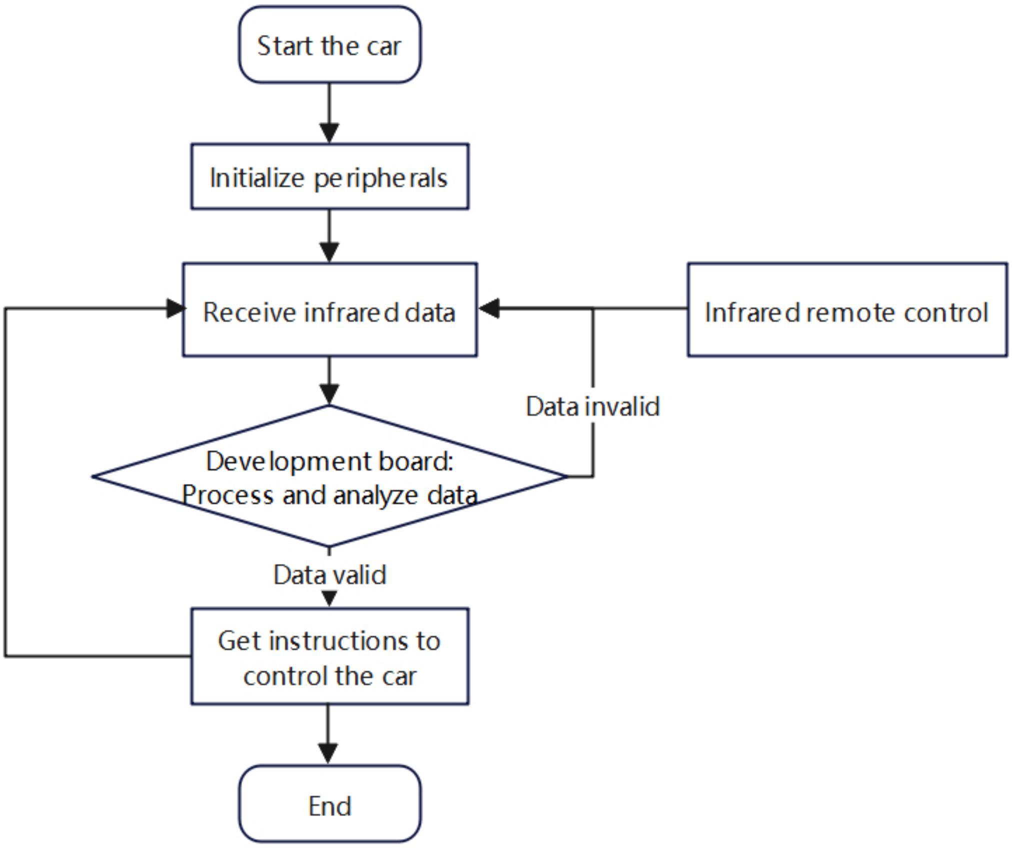


2.3 Control Principle

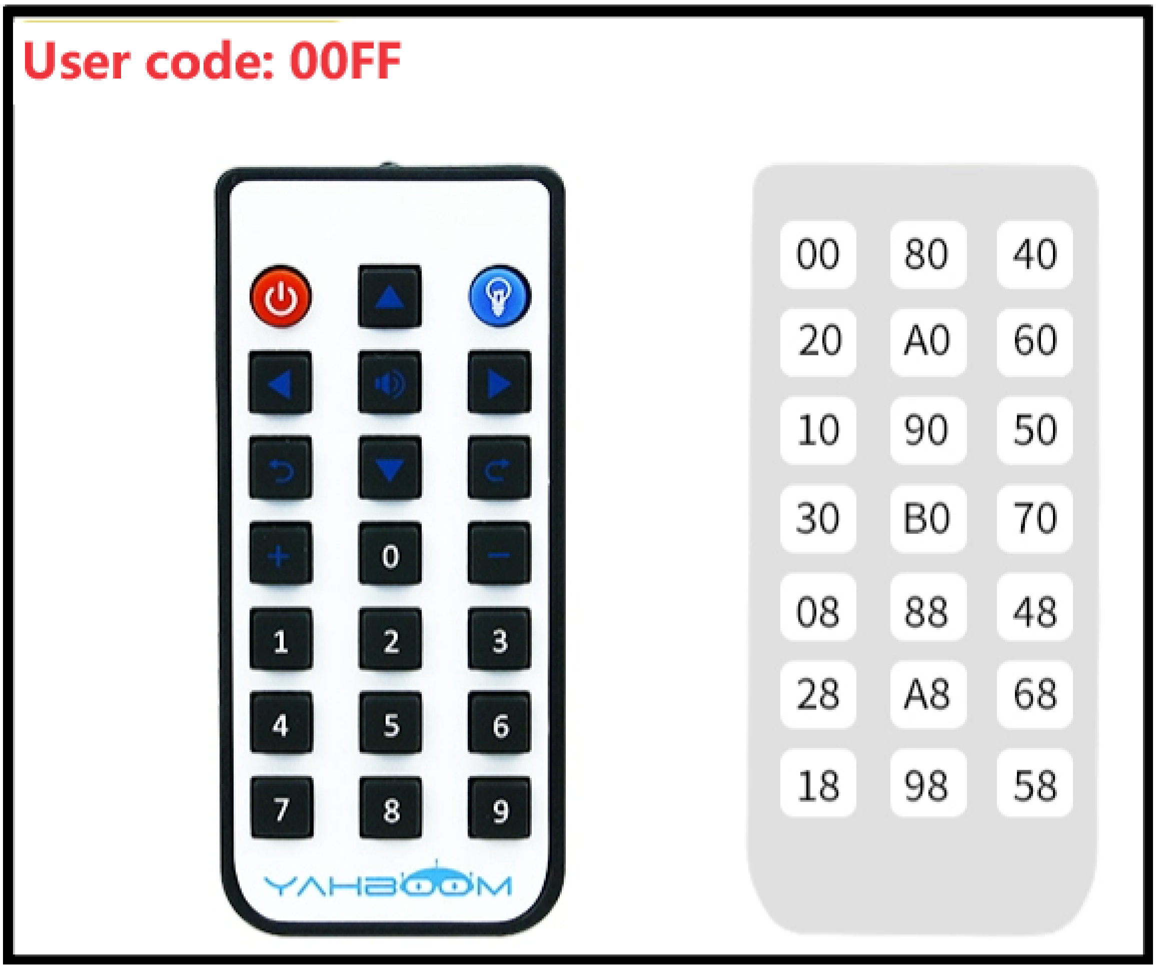
Control different car states based on key values pressed on the remote control (key values).

Use external interrupt to receive data sent by infrared, and parse this data through function: Pressing different keys on the infrared remote control will receive different data, we control different functions of the car based on this data (Switch statement is used in the program); Different key values correspond to different functions, in addition there are some undefined keys that can be defined by yourself.

- **Program Flowchart**



- **Infrared Remote Control**



User code: 00FF

00	80	40
20	A0	60
10	90	50
30	B0	70
08	88	48
28	A8	68
18	98	58

Remote Control (Key Value)	Function (Customizable)	Actual Function in This Tutorial Code
0x00	IR_POWER	Turn off motors and lights
0x80	IR_UP	Control car to move forward
0x40	IR_LIGHT	Control blue RGB light on (press once to turn on, press again to turn off)
0x20	IR_LEFT	Control car to turn left
0xA0	IR_BEEP	Control buzzer (press once to turn on, press again to turn off)
0x60	IR_RIGHT	Control car to turn right
0x10	IR_LEFT_SPIN	Control car to spin left
0x90	IR_DOWN	Control car to move backward
0x50	IR_RIHGT_SPIN	Control car to spin right
0x30	IR_ADD	Control car to accelerate
0x70	IR_SUB	Control car to decelerate
0xB0	IR_0	Control RGB light rotation
0x08	IR_1	Control RGB breathing light rotation
0x88	IR_2	Control red RGB light on
0x48	IR_3	Control orange RGB light on
0x28	IR_4	Control yellow RGB light on
0xA8	IR_5	Control green RGB light on
0x68	IR_6	Control cyan RGB light on
0x18	IR_7	Control blue RGB light on
0x98	IR_8	Control purple RGB light on
0x58	IR_9	Turn off light
0xFF	CLR_CLR	Default value

3. Main Functions

Mainly introduces the functional code written by users, **for detailed code, you can open the project files we provide and view the source code in the Bsp folder**.

> **Function: IR_Control_Car**

Function Prototype	void IR_Control_Car(void)
Function Description	Infrared remote control function
Input Parameters	None
Output Parameters	None

> **Function: Motor_IR_Control**

Function Prototype	void Motor_IR_Control()
Function Description	Control motors
Input Parameters	None
Output Parameters	None

> **Function: Control_BEEP_IR**

Function Prototype	void Control_BEEP_IR()
Function Description	Control buzzer
Input Parameters	None
Output Parameters	None

> **Function: Control_Green_RGB_IR**

Function Prototype	void Control_Green_RGB_IR()
Function Description	Control RGB lights
Input Parameters	None
Output Parameters	None

> **Function: IR_Control_Car**

Function Prototype	void IR_Control_Car()	
Function Description	Main program combining key values and functions	
Input Parameters	None	
Output Parameters	None	

> **Function: Printf_Irremote**

Function Prototype	**void Printf_Irremote()**
Function Description	Use serial port to print key values
Input Parameters	None
Output Parameters	None

4. Experimental Phenomenon

After successfully downloading the program, press the RESET button on the development board and use the infrared remote control to control the car!

```
For program download, refer to [3. Development Environment Setup and Usage:
3.Uniflash Programming]
```

Phenomenon:

According to different buttons pressed on the remote control, different functions of the car will be activated.