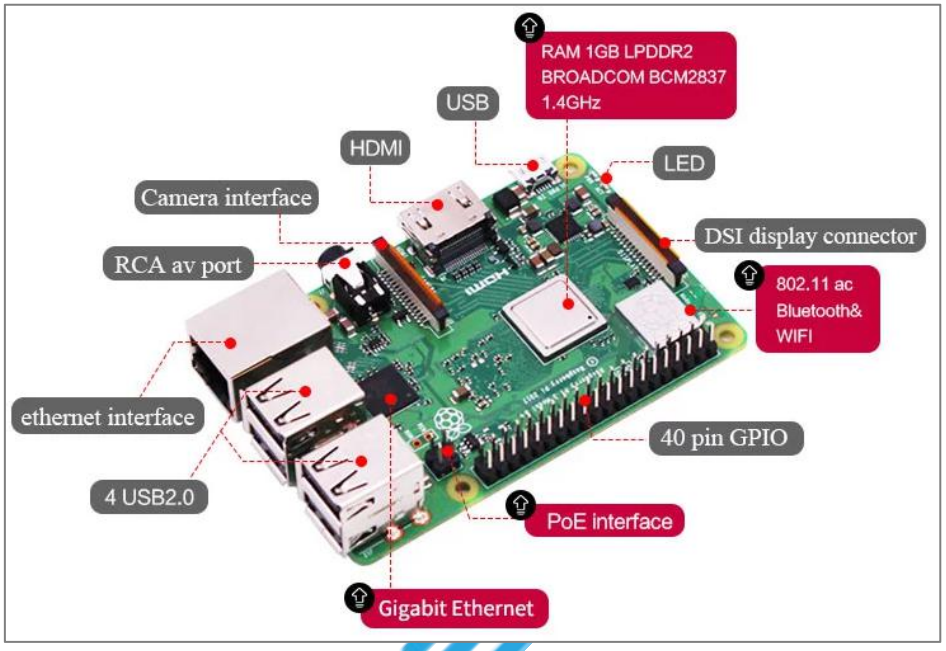
6.Raspberry Pi platform ------- infrared_avoid

Note:

A. Before this experiment, we can adjust the sensitivity of the infrared obstacle avoidance module by rotating the potentiometer on the infrared module to achieve better experimental results.

B. This experiment needs to be done indoors to reduce the interference of sunlight on the infrared receiver.

1)Preparation



1-1 Raspberry Pi board

1-2 Infrared obstacle avoidance module

2)Purpose of Experimental

After running the infrared_avoid executable in the Raspberry Pi system. You need to press the K2 to start the car, and the infrared obstacle avoidance function is started. When there is an obstacle in front, the car can avoid the obstacle automatically.

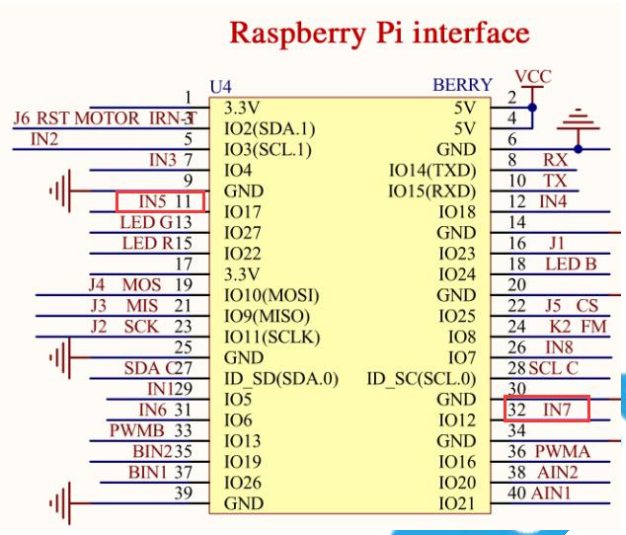
3)Principle of experimental

The basic principle of the infrared sensor to avoid obstacles is to use the reflective nature of the object.

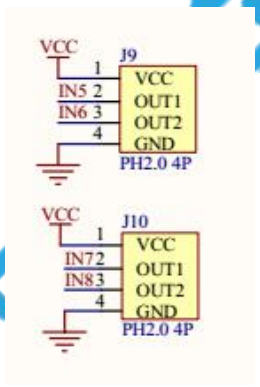
Within a certain range, if there is an obstacle, the infrared rays will encounter obstacle and will be reflected to reach the sensor receiving pin. In this experiment, we used 2 infrared sensors connected to the Raspberry Pi board to detect the obstacles by detecting the electrical level of the two ports, and robot car will make corresponding obstacle avoidance actions.

4)Experimental Steps

4-1 About the schematic



4-1 Raspberry Pi interface circuit diagram



4-2 Left and right infrared sensor interface

wiringPi	BCM	Funtion	Physical pin		Funtion	BCM	wiringPi
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

4-3 Raspberry Pi 40 pins comparison table

4-2 According to the circuit schematic:

Left infrared sensor-----32(Physical pin)----- 26(wiringPi)

Right infrared sensor-----11(Physical pin)----- 0(wiringPi)

4-3 About the code

Please view .py and.c file

A. For .c code

(1) We need to compile this file in the Raspberry Pi system. (Note: we need to add -lwiringPi to the library file.)

We need to input: gcc infrared_avoid.c -o infrared_avoid -lwiringPi

(1)We need to run the compiled executable file in the Raspberry Pi system.We need to input: ./infrared_avoid

```
pi@yahboom4wd:~/SmartCar $ gcc infrared_avoid.c -o infrared_avoid -lwiringPi
pi@yahboom4wd:~/SmartCar $ ./infrared_avoid
```

(3)We can input: ctrl+c to stop this process, which mean is send a signal to the linux kernel to terminate the current process, but the state of the relevant pin is uncertain at this time, we also need to run a script to initialize all pins.

(Note:The initpin.sh script file is included in the SmartCar/python directory.)

You need to input: chmod 777 initpin.sh

./initpin.sh

```
pi@yahboom4wd:~/SmartCar $ sudo chmod 777 initpin.sh
pi@yahboom4wd:~/SmartCar $ ./initpin.sh
```

B. For python code

(1) We need to input following command to run python code.

python infrared_avoid.py

```
pi@yahboom4wd:~/python $ python infrared_avoid.py
```

(2) We can input: ctrl+c to stop this process, which mean is send a signal to the linux kernel to terminate the current process, but the state of the relevant pin is uncertain at this time, we also need to run a script to initialize all pins.

(3) You need to input: chmod 777 initpin.sh

./initpin.sh

```
pi@yahboom4wd:~/SmartCar $ sudo chmod 777 initpin.sh
pi@yahboom4wd:~/SmartCar $ ./initpin.sh
```

After completing the above steps, the experiment is over.

5) Experimental phenomenon

After running the programs. **You need to press the K2 to start the car,** and the infrared obstacle avoidance function is started. When there is an obstacle in front, the car can avoid the obstacle automatically.