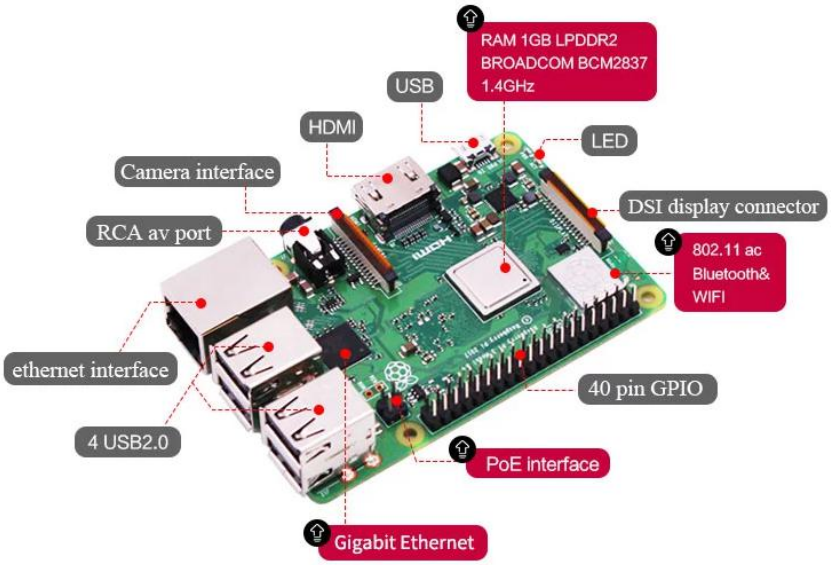
10.Raspberry Pi platform ------- avoid_ultrasonic

1)Preparation



1-1 Raspberry Pi board

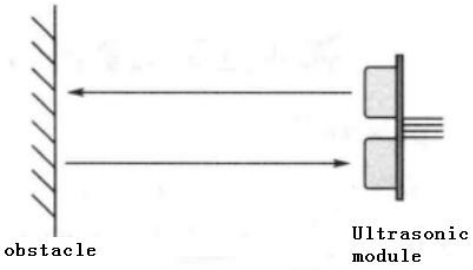
1-2 Ultrasonic module

2)Purpose of Experimental

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

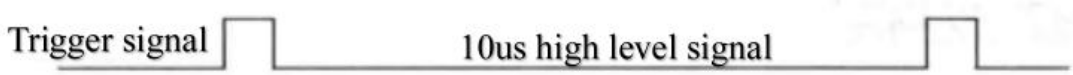
3)Principle of experimental

The ultrasonic module is a sensor that uses ultrasonic characteristics to detect the distance. It has two ultrasonic probes for transmitting and receiving ultrasonic waves. The range of measurement is 3-450 cm.



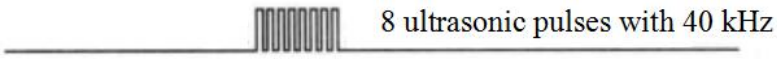
3-1 Ultrasonic emission and reception schematic

(1) You need to input a high level signal of at least 10us to the Trig pin to trigger the ranging function of the ultrasonic module.



3-2 Ultrasonic module sends trigger signal

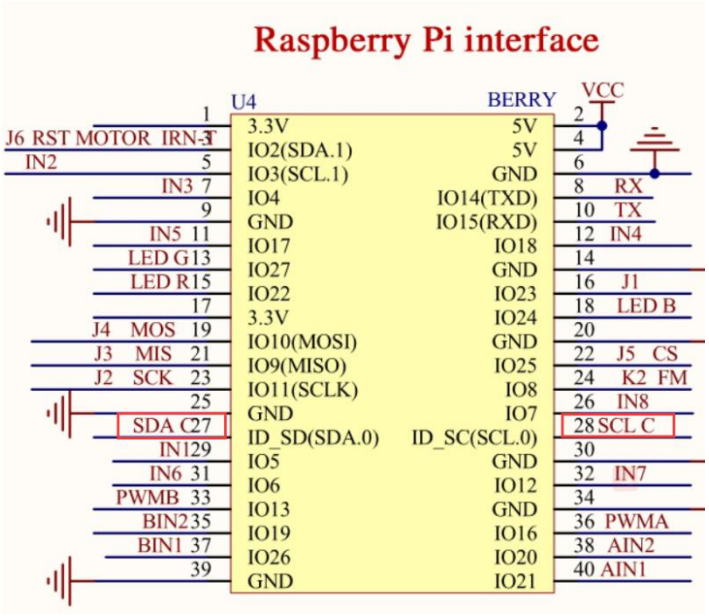
(2) After the ranging function is triggered, the module will automatically send out 8 ultrasonic pulses with 40 kHz and automatically detect whether there is a signal return. This step is done internally by the module.



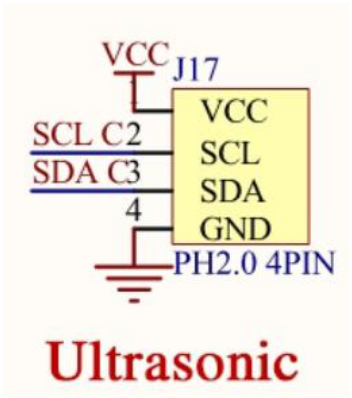
(3) When the module detects an echo signal, the ECHO pin will output a high level. The high level duration is the time from when the ultrasonic wave is sent to when it returns. You can calculate the distance by using the time function to calculate the high level duration. Formula: Distance = High level duration * Speed of sound(340M/S)/2.

4)Experimental Steps

4-1 About the schematic



4-1 Raspberry Pi interface circuit diagram



4-2 ultrasonic 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:

Trig-----28(Physical pin)----- 31(wiringPi)

Echo-----27(Physical pin)----- 30(wiringPi)

(Note: We use the wiringPi library to write code.)

4-3 About the 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 avoid_ultrasonic.c -o avoid_ultrasonic -lwiringPi

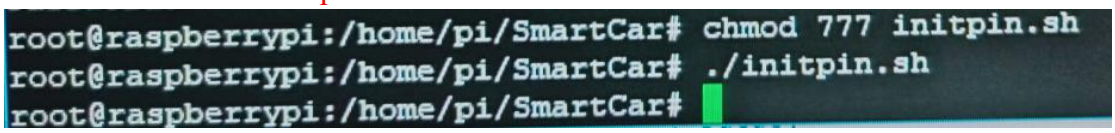
(2)We need to run the compiled executable file in the Raspberry Pi system.We need to input: ./avoid_ultrasonic

As shown in the figure below.

```
root@raspberrypi:/home/pi/SmartCar# gcc avoid_ultrasonic.c -o avoid_ultrasonic -
lwiringPi
root@raspberrypi:/home/pi/SmartCar# ./avoid_ultrasonic
^C
```

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



After completing the above steps, the experiment is over.