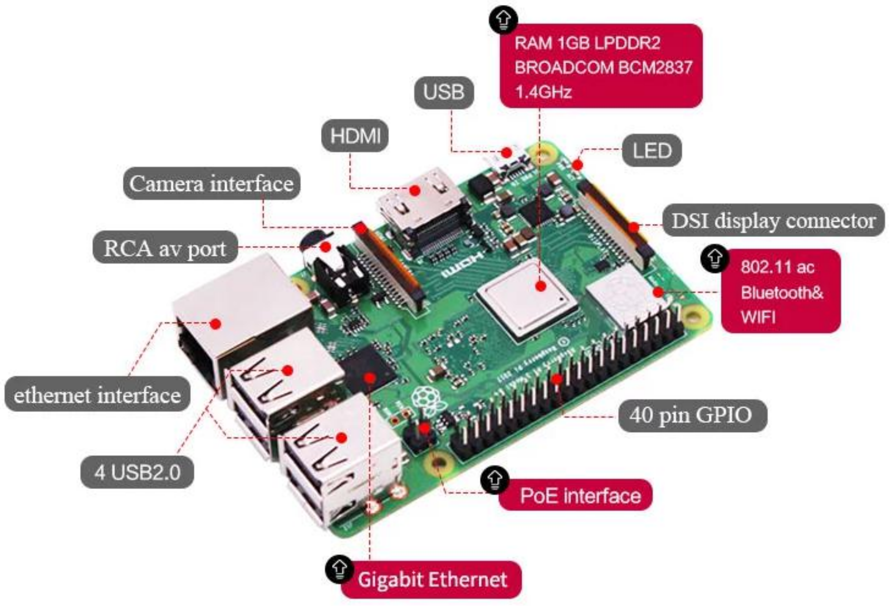
8.Raspberry Pi platform ------- servo_avoid_ultrasonic

Note:

A. This experiment needs to start the car by pressing the button KEY.

1)Preparation



1-1 Raspberry Pi board

1-2 Ultrasonic module

1-3 RGB module

1-4 SG90 servo

2)Purpose of Experimental

After running the servo_avoid_ultrasonic executable in the Raspberry Pi system, you need to press the K2 to start the car, and the ultrasonic obstacle avoidance with servo function is started.

When the distance on the front is more than 60cm, the colorful light module is green, the distance on the front is less than 60cm, then servo turn to 0 degree, ranging and recording, servo turn to 110 degree, ranging and recording, servo turn to 55 degree, ranging and recording. And the robot car will compare the left and right distance to determine whether to avoid the obstacle to the left or right. When the distance in three directions is less than 60, the robot car will turn round.

3)Principle of experimental

The working principle of the servo: the control signal enters the signal modulation chip from the channel of the receiver to obtain the bias voltage of the DC. It has a reference circuit inside, which generates a reference signal with a period of 20ms and a width of 1.5ms. It will compares the DC bias voltage with the voltage of the potentiometer to obtain a voltage difference and output. The positive and negative of the voltage difference is outputted to the motor drive chip to determine the forward and reverse of the motor.

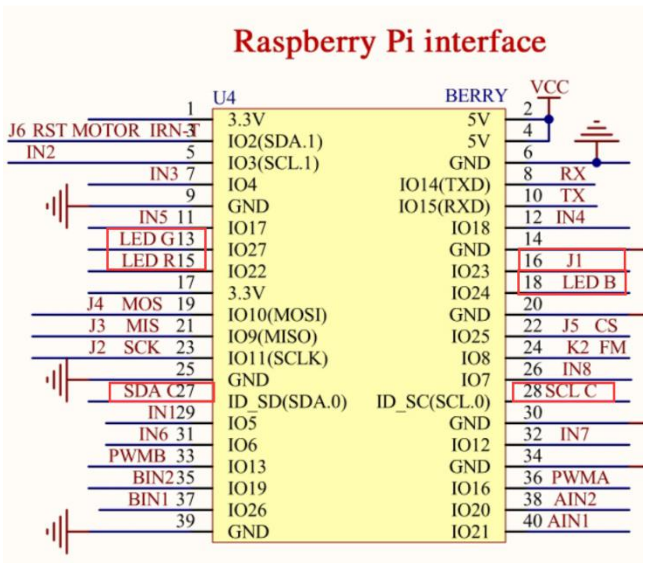
Servo rotation angle is by adjusting the duty ratios of PWM (pulse width modulation) signal. The standard PWM (pulse width modulation) signal has a fixed period of 20ms (50Hz). Theoretically, pulse width distribution should be between 1 ms to 2 ms, but in fact between pulse width can be 0.5 ms and 2.5 ms. Pulse width and the servo rotation angle 0°～180° corresponds, as shown below.

0.5ms-----------------	0°
1.0ms-----------------	45°
1.5ms-----------------	90°
2.0ms-----------------	135°
2.5ms-----------------	180°

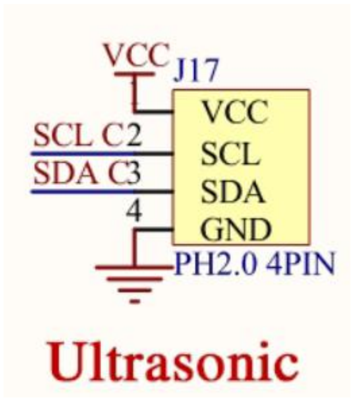
(Note:The TrikeBot car controls the direction of the front rubber wheel by controlling the servo. In this way, the direction of the car is controlled. In this experimental, we set the angle of the servo to 55 degrees according to the actual situation, indicating that the car is in the front. We set the angle of the servo to 110 degrees according to the actual situation, indicating that the car is turning left. We set the angle of the servo to 0 degrees according to the actual situation, indicating that the car is turning right.)

4)Experimental Steps

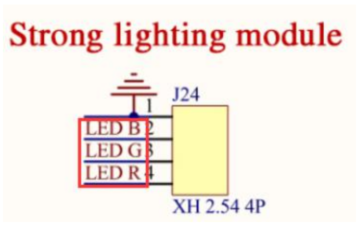
4-1 About the schematic



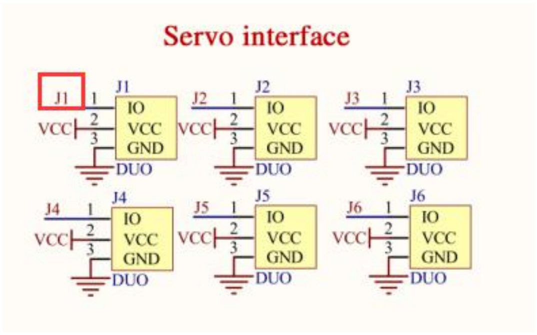
4-1 Raspberry Pi interface circuit diagram



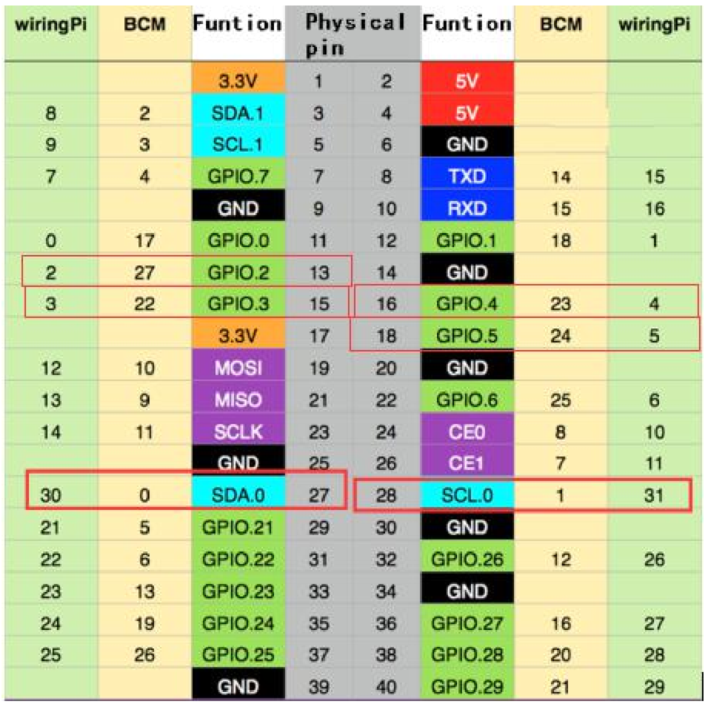
4-2 ultrasonic interface



4-3 RGB module interface



4-4 Servo 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)

4-3 About the code

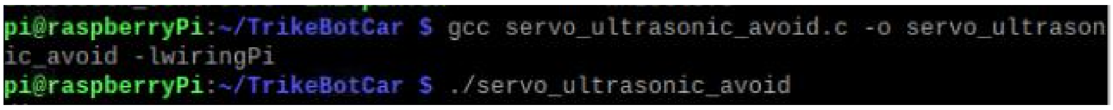
Please view .py and.c file

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 servo_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: ./servo_avoid_ultrasonic



```
pi@raspberryPi:~/TrikeBotCar $ gcc servo_ultrasonic_avoid.c -o servo_ultrasonic_avoid -lwiringPi
pi@raspberryPi:~/TrikeBotCar $ ./servo_ultrasonic_avoid
```

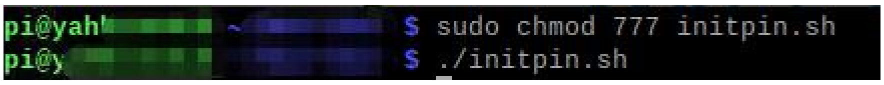
(3)We can input: ctrl+c to stop this process, which mean is send a signal to the linux kernel to terminate the current process, but the state of the relevant pin is uncertain at this time, we also need to run a script to initialize all pins.

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

You need to input: chmod 777 initpin.sh

./initpin.sh

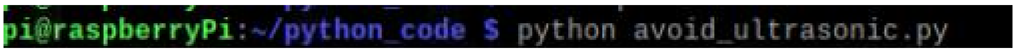
As shown in the figure below.



```
pi@yah[redacted] $ sudo chmod 777 initpin.sh
pi@y[redacted] $ ./initpin.sh
```

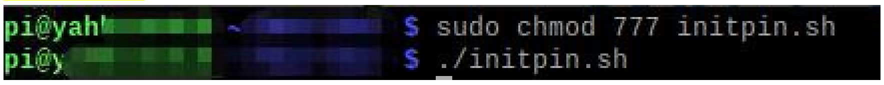
B. For python code

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



```
pi@raspberryPi:~/python_code $ python avoid_ultrasonic.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: sudo chmod 777 initpin.sh
./initpin.sh



```
pi@yah[redacted] $ sudo chmod 777 initpin.sh
pi@y[redacted] $ ./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 ultrasonic obstacle avoidance function is started. When there is an obstacle in front, the car can avoid the obstacle automatically.