Voice control autonomous driving

Experimental Objective

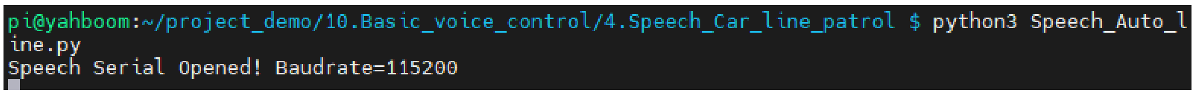
Understand and learn to use fixed basic semantics to enable the car to patrol the line according to the color of the semantics.

Experimental Procedure and Results

1. First, enter the following command in the terminal:

```
cd /home/pi/project_demo/10.Basic_voice_control/4.Speech_Car_line_patrol/
python3 Speech_Auto_line.py
```

2. After entering this interface, wake up the car using the wake-up phrase: Hi, Yahboom in English.



```
pi@yahboom:~/project_demo/10.Basic_voice_control/4.Speech_Car_line_patrol $ python3 Speech_Auto_l
ine.py
Speech Serial Opened! Baudrate=115200
```

3. After successfully waking up, the car responds: "Hi, I'm here" in English.
4. Then, control the car using fixed commands to patrol the line according to the corresponding color.

Fixed statement table

Wake-up phrase (international users)	Operation phenomenon	Answers by Car (English version)
Close tracking mode	The car stops	OK, tracking mode is closed
Tracking the red line	The car will follow the red patrol line	OK, I will track the red line
Tracking the green line	The car will follow the green patrol line	OK, I will track the green line
Tracking the blue line	The car will follow the blue patrol line	OK, I will track the blue line
Tracking the yellow line	The car will follow the yellow patrol line	OK, I will track the yellow line

Main source code analysis

```python
if __name__ == "__main__":
    tracker = ColorLineTracker()
    mySpeech = Speech()

    try:
        while True:
            time.sleep(0.2)

            num = mySpeech.speech_read()
            if num !=999 :
                #print(num)
                if num == 0:
                    mySpeech.void_write(num)
                if num == 22:
                    tracker.stop()
                    mySpeech.void_write(num)
                elif num == 23:
                    mySpeech.void_write(num)
                    print('red')
                    tracker.start('red')


                elif num == 24:
                    mySpeech.void_write(num)
                    print('green')
                    tracker.start('green')

                elif num == 25:
                    mySpeech.void_write(num)
                    print('blue')
                    tracker.start('blue')


                elif num == 26:
                    mySpeech.void_write(num)
                    print('yellow')
                    tracker.start('yellow')


    except KeyboardInterrupt:
        tracker.stop()
        print('Speech end!')
```

speech_read: Identifies the color of the line being tracked based on fixed semantics.
ColorLineTracker: Starts tracking the line.