

语音交互模块识别播报

1. 学习目标

本次课程我们主要学习使用树莓派和语音交互模块实现识别播报。

2. 课前准备

语音交互模块采用的是 UART 通讯或 USB 通讯，这里以 USB 通讯为例。

使用 type-c 线连接树莓派和语音模块，运行命令 `ls /dev | grep 'ttyUSB'`，可以看到识别到语音模块为 USB0

```
pi@yahboomtrikebot:~ $ ls /dev | grep 'ttyUSB'
ttyUSB0
```

3. 程序

本次课程的程序请参考：`speech_moudle.py`

初始化 USB:

```
ser = serial.Serial("/dev/ttyUSB0", 115200)
```

语音播报函数

```
def void_write(void_data):
    void_data1 = int(void_data/100)+48
    void_data2 = int(void_data%100/10)+48
    void_data3 = int(void_data%10)+48
    cmd = [0x24, 0x41, void_data1, void_data2, void_data3, 0x23]
    #print(cmd)
    ser.write(cmd)
    time.sleep(0.005)
    #ser.flushInput()
```

语音识别函数

```
def speech_read():
    count = ser.inWaiting()
    if count:
        speech_data = ser.read(count)
        speech_data1 = int(str(speech_data)[4:5])
        speech_data2 = int(str(speech_data)[5:6])
        speech_data3 = int(str(speech_data)[6:7])
        ser.flushInput()
        time.sleep(0.005)
        return int(speech_data1*100+speech_data2*10+speech_data3)
    else:
        return 999
```

语音处理主循环

```
while True:
    speech_r = speech_read()
    if speech_r != 999:
        if speech_r == 0:
            print("hi")
        elif speech_r == 10:
            print("OK")
            time.sleep(0.1)
            void_write(10)
```

4. 运行程序

终端输入 `python3 speech_moudle.py` 运行程序。

5. 实验现象

程序运行以后，开始初始化 USB，初始化成功显示“Speech Serial Opened! Baudrate=115200”，否则显示“Speech Serial Open Failed!”，如果错误需要检查接线或 USB 端口，之后开始识别播报。

例程中说：你好，小亚，扬声器播报：在的，终端打印：hi。

唤醒之后说：关灯，扬声器播报：好的，已关灯，终端打印：OK。