

5、路标指示动作

5、路标指示动作

- 5.1、实验说明
- 5.2、实验目标
- 5.3、实验操作
- 5.4、实验效果
- 5.5、实验总结

5.1、实验说明

本实验是属于拓展类实验，需要搭配其他外部设备使用，这里用到了小车底盘和ROS扩展板都不属于K210开发板套件内容，所以本实验效果仅供参考，如果没有对应的设备是不能直接使用本例程代码。

ROS扩展板需要提前烧录固件：ROS-CAR.hex

由于本次使用的电机电压为8.4V，所以ROS扩展板的电池不可以插入12.6V电池，需要插入8.4V电池。

小车电机连接线如下图所示：

电机Motor 1连接左前轮，电机Motor 2连接左后轮，电机Motor 3连接右前轮，电机Motor 4连接右后轮。



K210开发板与ROS扩展板连接的线序如下图所示：

白色连接GND，黄色连接VCC，黑色连接SCL，红色连接SDA。

这里需注意一下，图示中的标识为I2C线序标识，但是K210使用的是串口通讯，由于烧录的ROS-CAR.hex文件已经把这个接口修改为串口信号，所以实际上ROS扩展板上的接口对应关系是：SCL实际上是TX，SDA实际上RX。



5.2、实验目标

本节课主要学习K210开发板搭配小车底盘做视觉巡线的功能。

本次实验的参考代码路径为：06-export\sign_motion.py

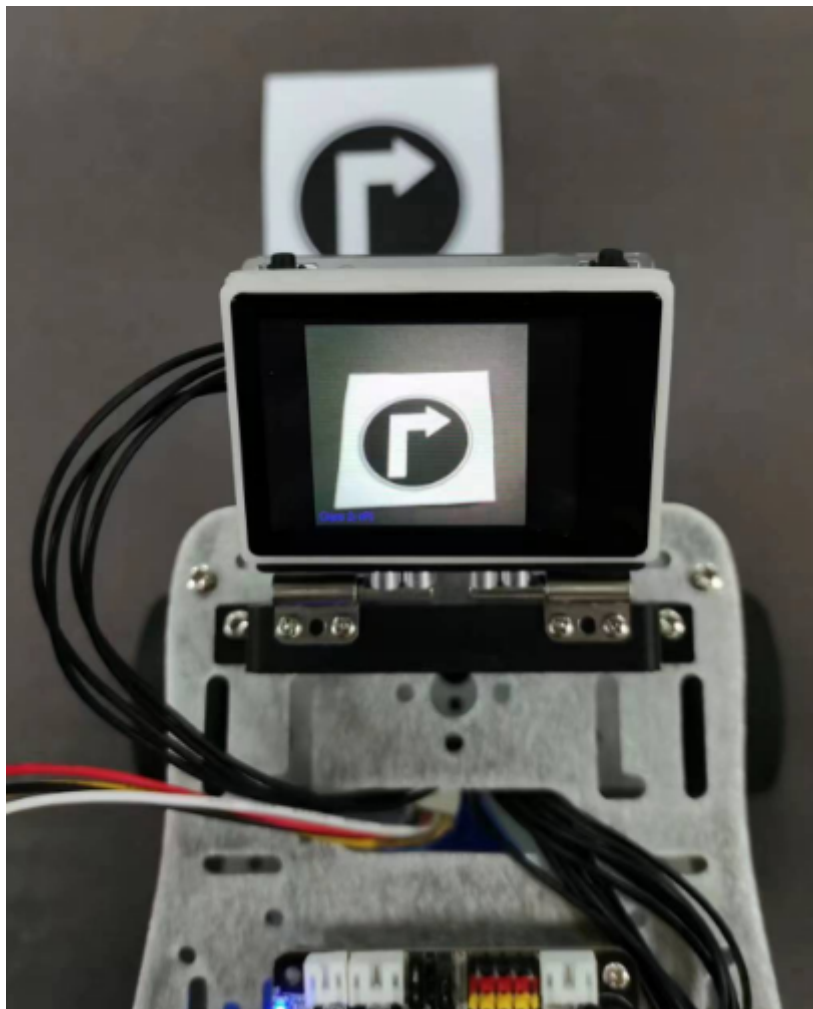
5.3、实验操作

1. ROS扩展板烧录固件：ROS-CAR.hex
2. 将RGB灯条插入ROS扩展板的RGB灯接口。
3. 请将06-export\library目录下的小车驱动库和PID控制库提前下载到内存卡的根目录上。
4. 打开CanMV IDE打开sign_motion.py代码并下载到K210开发板上。

5. 将K210开发板通过4PIN排线连接到ROS扩展板上。
6. 将小车放入白色或黑色背景中，掰动K210开发板支架到合适的角度，打开小车的开关。
7. 首先需要学习一下左转的图标，操作步骤与自主学习的方式一样，根据屏幕提示拍摄五张左转图标的图片。



8. 然后学习右转的图标，根据屏幕提示拍摄五张右转图标的图片。



9. 接着学习停止的图标，根据屏幕提示拍摄五张停止图标的图片。
10. 学习完成后小车开始向前运动，当检测到图像中有对应的图标，就执行对应的动作。

5.4、实验效果

等待系统初始化完成后，根据上面操作步骤完成学习路标后，小车开始往前走，当检测到摄像头图像中有左转路标时，执行左转功能，然后再直行；当检测到右转路标时，执行右转功能，然后再直行；当检测到停止路标时，小车停止。

小车转弯的幅度和功能可以在 `car_control` 函数中修改，`car_count`表示转弯的时间，`set_car_motion` 设置转弯的速度。

```
def car_control(class_id):
    global car_count, car_state
    car_state = 1
    if class_id == 1: # 向左转 turn left
        car_state = 1
        car_count = 10
        bot.set_car_motion(0, 0, 3)
    elif class_id == 2: # 向右转 turn right
        car_state = 1
        car_count = 10
        bot.set_car_motion(0, 0, -3)
    elif class_id == 3: # 停止 stop
        car_state = 0
        car_count = 0
        bot.set_car_motion(0, 0, 0)
```

5.5、实验总结

小车路标识别功能是基于K210开发板自主学习功能修改而来的，学习步骤与自主学习的步骤类似，先学习左转图标，再学习右转图标，最后学习停车图标，由于不同小车不同地图的运动效果不同，可修改小车 `car_control` 函数里的内容来优化运行效果。