



Ubuntu 20.04

照祥同学

14-ESP32_Joystick_Control_Foxy

URL: https://github.com/ZhaoXiangBox/esp32_ros2_robot

Videos from Bilibili 照祥同学:[第十四节 PS4手柄通过micro ros控制ESP32舵机云台](#)

通过ROS2重定义游戏手柄功能

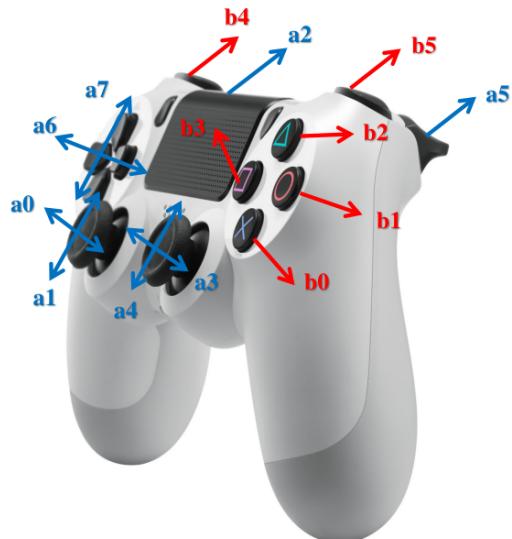
1. ROS2 Topic : joy

axes[] : 0 1 2 3 4 5 6 7 → 范围: [-1, 1]
buttons[] : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 → 状态: 0 or 1

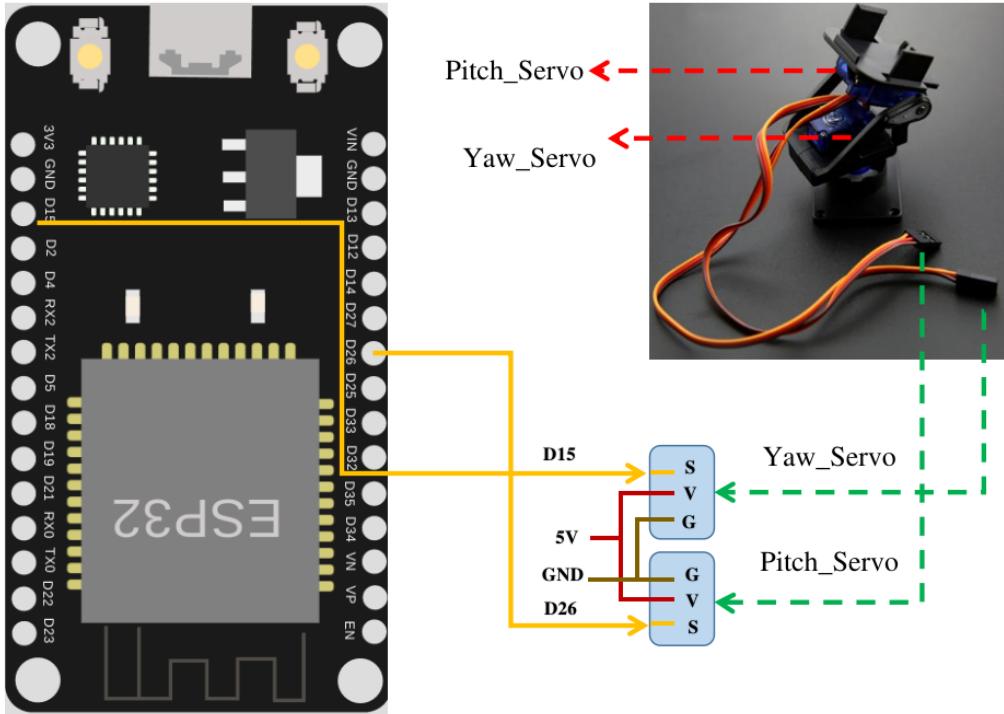
2. 定义

axes[i]: 0 1 2 3 4 5 6 7
 0: 角速度 [-1.0,1.0]; topic 名字: cmd_vel
 1: 线速度 [-1.0,1.0]; topic 名字: cmd_vel
 3: 舵机一的参考角度 [-90,90]; topic 名字: servo1
 4: 舵机二的参考角度 [-90,90]; topic 名字: servo2

buttons[i]: 0 1 2 3 4 5 6 7 8 9 10 11 12 13
 0: 关闭所有灯光; 3: 打开/关闭 后灯;
 1: 打开/关闭 前灯; 4: 打开/关闭 左转向灯;
 2: 打开/关闭 双闪; 5: 打开/关闭 右转向灯;



将右侧摇杆的两个自由度重映射为两个舵机转动的角度，并编写 ROS2 节点，通过Topic 的通讯方式予以发布。如上图所示，摇杆 a3 作为1号舵机的信号，摇杆 a4 作为2号舵机的信号，分别为: servo1, servo2。



如上图所示，ESP32 使用 **D15** 和 **D26** 控制云台的两个舵机。ESP32 使用的是 Arduino IDE 下的 **ESP32Servo** 这个库来驱动舵机。该库的使用分两步：

First: Initial Servo

```
#include <ESP32Servo.h>

// 定义相关的舵机引脚
#define Yaw_Servo_Pin 15
#define Pitch_Servo_Pin 23

// 定义舵机的PWM所对应的角度范围
#define Min_Servo_Value 0
#define Max_Servo_Value 180

// 声明两个舵机对象
Servo Yaw_Servo;
Servo Pitch_Servo;

int minUs = 1000;
int maxUs = 2000;

// 初始化两个舵机对象
// 安装舵机的时候，需要让其处在正中间的位置后再固定拉杆，
void Init_Servos_Drivers()
{
    Yaw_Servo.setPeriodHertz(50);          // Standard 50hz servo
    Pitch_Servo.setPeriodHertz(50);         // Standard 50hz servo

    Yaw_Servo.attach(Yaw_Servo_Pin, minUs, maxUs);
    Pitch_Servo.attach(Pitch_Servo_Pin, minUs, maxUs);

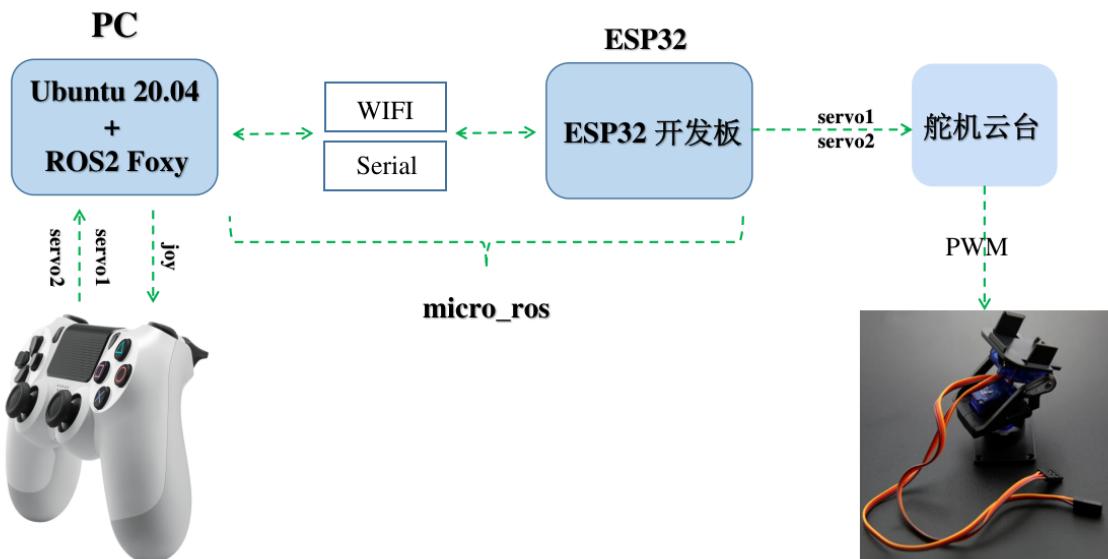
    Servo_control(Init_Angle_Value, Init_Angle_Value);           // 初始舵机
    角度居中
}
```

```
// 可以手动释放舵机寄存资源
void Detach_Servos()
{
    Yaw_Servo.detach();
    Pitch_Servo.detach();
}
```

Second : Drive Servo

```
Yaw_Servo.write(Yaw_Degree);      // Yaw_Degree   : Yaw_Servo 的目标角度
Pitch_Servo.write(Pitch_degree);  // Pitch_degree : Pitch_degree 的目标角度
```

PS4通过micro_ros控制ESP32驱动的舵机云台



本实验的整个架构图如上，详细的操作视频请看视频，代码已开源，链接如视频详情。

update by zhaoxiangli 23.09.24