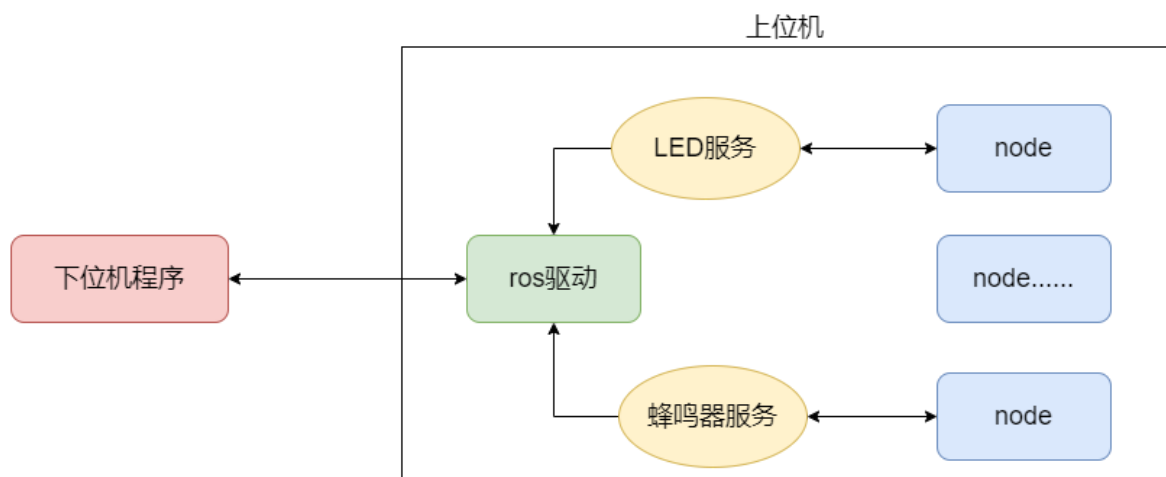


## 功能回顾

之前我们的下位机 `protocol.hex` 提供了简单的功能，我们上位机通过串口通信也实现这些功能。

功能	ROS驱动实现
LED控制	需要实现
蜂鸣器控制	需要实现
OLED显示	暂不实现
电机控制	已经实现
编码器	已经实现



驱动的作用就是作为上位机和下位机的统一桥梁。ros中，就是通过一个node来实现驱动的。

## LED和蜂鸣器控制

LED和蜂鸣器功能我们会集成到ros控制板驱动里，功能需要继续的对外开放，也就是谁想控制LED和蜂鸣器都可以，给我的驱动发消息就行。

那么，我的ros驱动中，负责LED和蜂鸣器控制的，就必须能够提供控制的服务，其实就是一个服务的Server端

## LED和蜂鸣器实现

### 实现分析

- 驱动负责和下位机通讯
- 驱动提供服务的server端

### 协议问题

- 下位机中，已有的数据反馈中，已经包含了编码器数据。
- 下位机中，现在需要反馈LED和蜂鸣器的数据。
- 下位机中，需要反馈的将数据类型进行区分
- 上位机读取过程中，读取的是固定长度，那么就要求下位机发送的也是固定长度

功能	上位机到下位机	下位机到上位机(旧)	下位机到上位机(新)
LED	0x01 + 0x01(开),0x02(关),0x03(切换)		0x01 + 0x01(开), 0x02(关) + 0x00(补位)
蜂鸣器	0x02 + 0x01(开),0x02(关),0x03(切换)		0x02 + 0x01(开), 0x02(关) + 0x00(补位)
电机	0x03 + 0xb8 0x0b(pwm值)	无	
OLED	0x04 + 0x69 0x74 0x63 0x61 0x73 0x74(显示的内容)	无	
编码器	无	0xee 0x11(每秒2次)	0x03 + 0xee 0x11(每秒2次)

新的协议中:

- 通过0x01,0x02,0x03来分辨下位机向上位机发送的数据类型
- 下位机每次发送的数据个数位3个（编码器有效数据为3个，其他的为2个，为了满足读取方便，不够3个的补一个）