

一个名为“tun0”的虚拟网卡. 直接用程序读写该设备, tun0 建立在数据链路层, 接入模块将边界路由节点上递的数据, 通过 slip 串口与 IP 协议, 将其封装为 IP 数据包, 再递交至 UDP 服务端. UDP 服务端将执行节点的最新 IP 地址存入地址池, 为反向控制做好准备, 同时将传感数据封装为 ros 消息, 以规定的话题发布至 ROS 网络.

反之, 接入模块 (见图 5) 通过 ros_node 订阅规定话题, 接收 ROS 网络传来的数据 (多为控制指令). 通过比对地址池, 匹配目标执行节点, tunslip6 程序将数据转换为串口数据通过串口递交至边界路由设备.

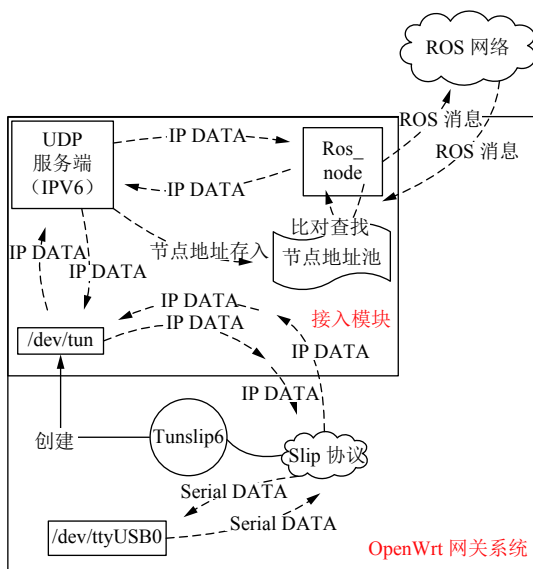


图 5 接入模块结构设计

2.3 应用层

提供针对感知层数据监控与反向控制和针对机器人控制的 web 前端页面. web 服务与 ROS 网络的通讯, 采用 rosbridge 协议规范下的 rosbridge_suite 软件包进行 web 服务端与客户端的开发.

rosbridge 主要包含两个部分, Rosbridge Protocol 和 Rosbridge Implementation. 其中 Protocol 部分提供了非 ROS 系统和 ROS 系统通信的具体格式, 包括话题的订阅, 消息的发布等. Implementation 部分是 rosbridge 的具体实现, 包含 rosbridge_server 等包. rosbridge_server 负责通信的传输层, 包括 websocket, tcp, udp 等格式.

用户使用 PC 或移动终端采取相关操作, 将 UI 操作绑定为一种指定话题下消息的发布或订阅, 以 websocket 的方式与 rosbridge 节点通讯, 该节点将收到的数据封装为 ROS 规范的 ROS 消息, 并递交至 ROS 网络, 实

现 Web 服务与 ROS 网络的结合, 如图 6.

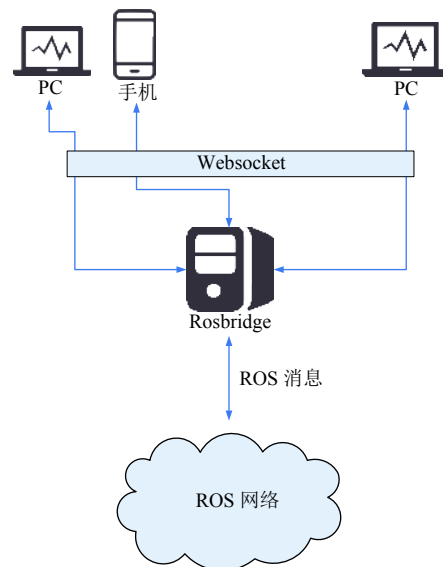


图 6 Web 服务与 ROS 网络

3 机器人设计

3.1 硬件设计

机器人硬件框架如图 7 所示.

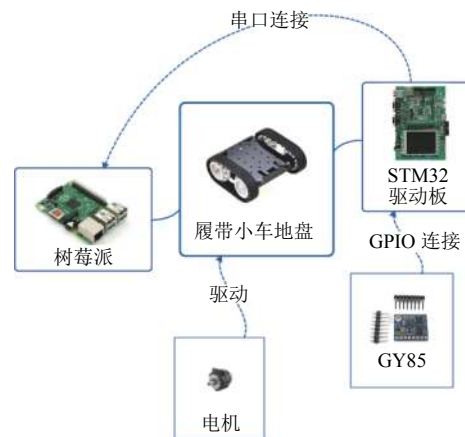


图 7 机器人硬件部件与架构

履带小车底盘作为机器人移动的基本部件, 通过装载的电机实现驱动. 底盘搭载了 STM32 核心驱动板和作为主控设备的树莓派. 核心驱动板上连接了三轴陀螺仪 GY-85, 如图 8.

- (1) STM32 核心驱动板
运行一个基本的 ROS 节点.
- (2) 树莓派
运行 ROS-Master, 承担 ROS 网络的牵头组建的功

能.同时可协同多个 ROS 节点.同时提供多个外设接口,为机器人的服务拓展提供硬件条件.



图8 机器人实物图

3.2 软件设计

机器人的软件设计主要基于 ROS 规范的 ROS 节点开发.机器人运行的 ROS 节点如图9所示.

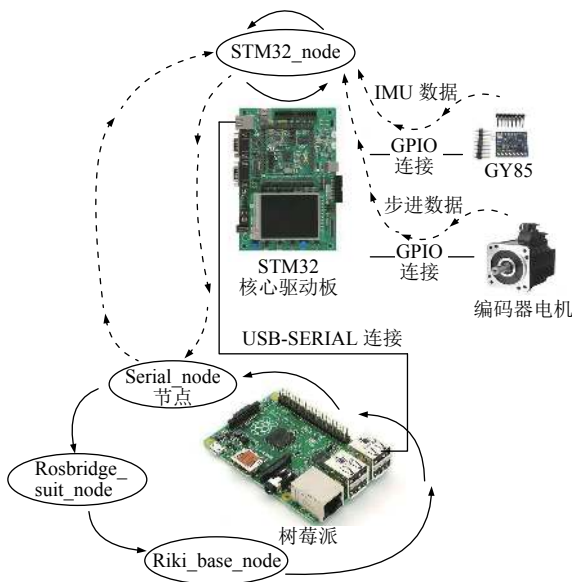


图9 机器人运行节点

STM32_node: 读取机器人的电机里程数、电量、速度、Imu 数据等机器人状态数据并以 ROS 规范封装,上递至 ROS 网络;同时可监控传感数据,并监听用户执行指令.

serial_node: 将 STM32 核心驱动板通过串口连接至 ROS 网络.

riki_base_node: 描述机器人基本信息的节点,接受来自驱动板经过过滤的 imu 融合数据,速度数据等,同时可与本文物联网系统下数据实现交互.

rosbridge_suit_node: 实现 ROS 网络与非 ROS 网络的应用层应用之间的通信.

4 系统运作

系统整体运作形式下,感知层部署感知节点和执行节点,执行节点集成相应的执行器,感知节点将感知数据递交至边界路由节点,由边界路由节点上传至 ROS-Contiki 网关.网关记录节点所分配地址,并将数据封装为 ROS 消息,发布至指定的话题,供机器人与应用层服务节点订阅.机器人时刻订阅机器人指令话题,监听机器人指令,作出反馈.

同时,web 页面提供了反向控制的接口,将其和规定的 ROS 消息绑定,发送至约定好的 ROS 话题,ROS-Contiki 网关订阅该话题,解析 ROS 消息获取命令,并根据地址池将控制指令下发至执行节点.

5 测试效果

(1) 启动机器人

为机器人上电,通过远程登录软件进入树莓派.输入如下命令:

```
roslaunch rikirobot bringup.launch
roslaunch rosbridge_suit rosbridge_websocket.launch
```

(2) 部署节点,启动网关

将边界路由节点通过 USB 线接入巴法络路由器.输入如下指令:

```
tunslip6 -s /dev/ttyUSB0 aaaa::1/64 &
再执行接入模块程序:
./pc-server /dev/USB0
```

(3) 打开浏览器,进入 Web 服务(如图10).



图10 Web 服务页面

6 总结

本文分为两个方面: 基于 ROS 的物联网系统的三层设计和该系统下机器人的设计.

物联网系统分为三层: 感知层、接入层、应用层. 本文中感知层节点运行 Contiki 操作系统, 被分配不同的 IPv6 地址, 以 mesh 的方式组建传感网络. ROS-Contiki 网关则基于 OpenWrt 与 Rosserial 协议, 将底层上报的数据以 ROS 消息规范封装上递至 ROS 网络. 应用层服务节点通过 ros_bridge 进行 Web 前端的服务集成, 搭建一套基于 ROS 网络的物联网系统.

机器人以树莓派做主控, STM32 核心驱动板作为硬件驱动. 机器人的核心系统框架采用当下流行的开源的机器人操作系统——ROS. 并结合当下优秀的嵌入式系统 OpenWrt 项目, 依托优秀的 ROS 通讯框架,

用较低的成本, 开发出本文物联网系统下智能机器人.

参考文献

- 1 Grieco LA, Rizzo A, Colucci S, *et al.* IoT-aided robotics applications: Technological implications, target domains and open issues. *Computer Communications*, 2014, 54: 32–47. [doi: [10.1016/j.comcom.2014.07.013](https://doi.org/10.1016/j.comcom.2014.07.013)]
- 2 孔令富, 吴培良. 物联网机器人系统研究进展. *燕山大学学报*, 2013, 37(6): 471–479. [doi: [10.3969/j.issn.1007-791X.2013.06.001](https://doi.org/10.3969/j.issn.1007-791X.2013.06.001)]
- 3 汤莉莉, 王金勇, 黄伟. 物联网智能机器人设计. *现代电子技术*, 2017, 40(8): 73–76.
- 4 张美平, 丁文才, 许友泽. IPv6 物联网接入网关的设计实践. *计算机系统应用*, 2018, 27(2): 112–116. [doi: [10.3969/j.issn.1003-3254.2018.02.019](https://doi.org/10.3969/j.issn.1003-3254.2018.02.019)]