水下机器人上位机的研究

集美大学海洋信息工程学院

通信工程专业　2018届 李昕凯　学号：201821112051

# 摘 要

随着计算机、微电子及控制等技术的快速发展，水下机器人在军事和民用领域方面得到了日益广泛的应用，其水下机器人控制技术和上位机相关问题也逐渐成为国内外研究的热点。作为水下机器人系统的指挥中心，研究上位机软件系统所涉及的关键技术和实现方法，对整个水下机器人系统的工程实现和应用有着重要的意义。

本文主要研究了水下机器人上位机软件系统的设计与实现。该软件为树下机器人提供了水下工作状态监控、航拍视频的实时播放、电子地图、航迹规划、串口通 信及数据存储和数据分析等功能，与水下机器人能进行实时交互。研究课题所涉及的技术主要有机器人参数调试、机器人状态监视、H.264视频推流、TCP通信技术及JSON协议等。论文首先介绍了水下机器人和上位机的组成结构，及国内外对水下机器人和 上位机的研究现状，总结出上位机软件系统的主要特点，并结合课题需求提出了 上位机软件系统所需要具备的功能。然后介绍了上位机软件系统开发过程中，实现所需功能所涉及到的关键技术，接着对水下机器人上位机软件系统的需求分析和总体设计作出了阐述，并对系统进行了功能层次划分，论述了每个功能层次上各个功能模块的设计。然后是根据系统的功能设计，详细地研究与实现系统的主要功能：一是用户操作层的功能实现，包括软件界面的设计与实现、参数调试及其功能操作的实现、运动状态的显示及控制指令的实现、通过播放器实时播放航拍视频的实现；二是数据通信层的实现，详细研究了TCP通信的实现过程；三是数据管理层中的数据存储与数据分析的实现。最后，将上位机软件系统进行了多次的测试，并且结合水下机器人进行了系统联调。测试结果表明：该上位机软件能实时接收和准确显示遥测数据，能实时播放航拍视频，能实时更改机器人参数并检测机器人各项状态信息，JSON协议功能能正确地响应对应的操作，能准确地完成上位机发送的各种控制命令，能快速地从存储文件中读取数据，并用图表显示各种遥测数据以进行数据分析。

关键词：水下机器人 ，上位机，TCP，参数调试，视频推流，JSON协议

Abstract

With the rapid development of computer, microelectronics and control technology, ROV has been widely used in military and civil fields. The ROV control technology and PC related issues have become a hot research topic. As the command center of the ROV system, it is of great significance to study the key technologies and implementation methods of the upper computer software system for the whole ROV system.

This paper mainly studies the design and implementation of the upper computer software system of underwater vehicle. The software provides the functions of underwater monitoring, real-time playing of aerial video, electronic map, route planning, serial communication, data storage and data analysis, etc. The technologies involved in the research mainly include robot parameter debugging, robot state monitoring, H. 264 video thruster, TCP communication technology and JSON protocol, etc. Firstly, this paper introduces the structure of ROV and PC, and the research status of ROV and PC at home and abroad, summarizes the main characteristics of PC software system, and puts forward the functions of PC software system. Then the paper introduces the key technologies involved in the development of the upper computer software system, and expounds the requirement analysis and overall design of the upper computer software system of the underwater vehicle, divides the functional levels of the system, and discusses the design of each functional module on each functional level. Then, according to the function design of the system, the main functions of the system are studied and realized in detail: first, the function realization of the user operation layer, including the design and realization of the software interface, the parameter debugging and the realization of the function operation, the display of the motion state and the realization of the control instruction, the realization of playing aerial video through the player in real time; second, the realization of the data communication layer, the realization of the TCP communication in detail; third, the realization of the data storage and data analysis in the data management layer. Finally, the software system of the upper computer is tested many times, and the system is debugged with the underwater vehicle. The test results show that the software can receive and display telemetry data in real time, play aerial video in real time, change the parameters of the robot in real time and detect the status information of the robot. The JSON protocol function can correctly respond to the corresponding operation, complete all kinds of control commands sent by the host computer, read the data from the storage file quickly, and display the telemetry data with chart for data analysis.

Keywords: ROV, PC, TCP, Parameter Debugging, Video Push Stream, JSON Protocol

# 引言

浩瀚的海洋不仅孕育了包括人类在内的许多物种的生命，而且还蕴藏着大量的优质矿产资源和能源。 特别是在过去20年中，深海领域活动的频繁发展和海底资源新品种的发现，使人们更深刻地认识到，海洋高新技术是确保保护国家海洋、资源安全权益乃至国家安全的重要保障。 因此，在未来相当长的时间内，发展海洋工程装备和关键技术将成为各国抢占21世纪海洋经济制高点的战略选择。而其中，开发利用深海矿产资源则是一项十分重要且紧迫的任务。 世界上绝大多数临海国家都把开发海洋作为基本国策，抓紧制定海洋科技规划，投入大量人力、财力和物力进行深海探测技术。 在国际上， 一项深海探测计划正在紧锣密鼓地进行， 例如，19921992启动的国际中洋InterRidge方案现已扩大到27个国家，国际海洋全球变化计划(IMAGES)和国际大陆边方案(InterMARGIN)，这些方案以深海海底为目标，在深海物理、化学、生物和地质方面开展多学科研究。 我国也将建设“21世纪海上丝绸之路”列入国民经济与社会发展第十一个五年规划纲要（2010—2020年），以加快推进经济全球化进程，促进区域经济社会协调发展。 为了实施这些研究项目，我们必须利用大量的深海探测设备和作业装备技术，而潜水器技术作为重要的技术载体之一，在海底发挥着重要的作用，海底是人类未来最实际、最可发展的空间。 其中，潜水器作为实现这些目标最有效、最有潜力的装备之一，已经成为各国发展深海研究与开发利用所必需的基础支撑性设施。 潜水器， 也被称为水下机器人，水下运载器或水下航行器， 它被定义为可以在海底活动或海面物种， 具备一定的视觉功能和感知功能，并通过遥控方式或自主操作方式使用相关作业工具； 替代或辅助水下作业科研工作者的设备是机械设计、流体力学、控制理论、机器人学、电子学、水声学、网络通讯等学科的综合应用技术，涵盖潜水器水动力特性、操纵控制、导航通讯、传感数据处理等。

在水下机器人设计中，嵌入式系统在与其他平台设备进行通信的过程中，往往采用串口或其他标准接口直接传输数据的方式。 但这两种方法都有一定的局限性，需要对数据进行加密处理后再传送至其它嵌入式设备上。而在一些特殊场合下，如磁场较强或感染较大的环境下等。则不能采用上述方法。 传统的传输方式不仅存在传输效率低的问题，而且在调试的过程中，调试人员一定要在嵌入式设备的工作现场，而且由于协议的设计比较简陋，使用效率和灵活性都比较低。 这些情况都会给嵌入式系统调试的调试带来很大的麻烦。 在嵌入式系统实际应用中，嵌入式设备通常处于各种复杂的网络环境中，支持一些网络协议，带宽有限。 因此，本文将以可靠的TCP通信协议与JSON 基于QT编程开发框架，设计了一种通过物联网向嵌入式设备数据传输的上位机软件。

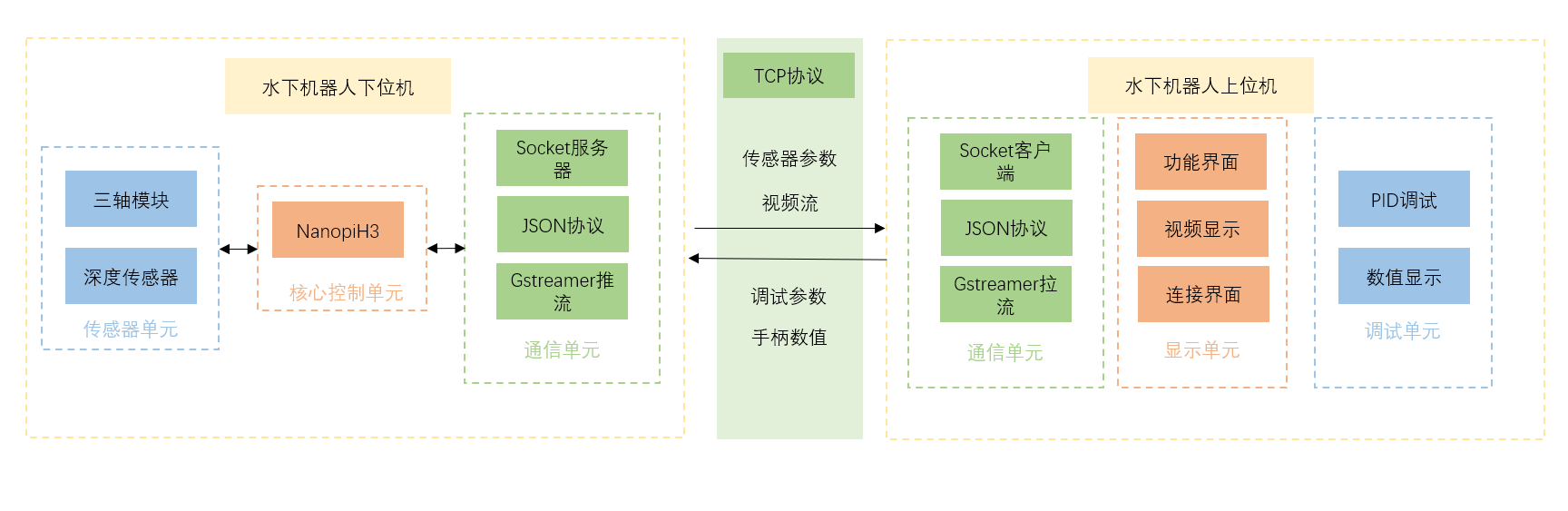
2.1 总体方案设计

1. 以QT的图形控件为显示界面框架而开发的，它为具有技术专业背景的程序提供GUI组件和一组实用类，其目标是以基于2D方式的窗体部件来显示数据， 数据源以数值，数组或一组浮点数等方式提供，提供美观，简介图形组件的同时，使用上手门槛也比较低。

2.基于JSON协议（是一种用于数据交换的轻量级数据结构）。它基于ECMAScript的子集，以完全独立于编程语言的一种文本数据格式存储和代表数据。 这种设计方法能够很好地处理复杂且庞大的数据结构以及对时间要求非常高的应用需求，并且具有良好的扩展性。此外，由于使用了简单直观的数据类型描述方式，所以更容易理解和操作。 层次结构的简单和清晰使JSON成为数据交换的理想语言。 它完全独立于任何一种语言，不论C还是Python都能高效的解析与申明该数据类型；还具有良好的扩展性和可配置性。它允许用户在不需要任何编程基础的情况下直接调用。 便于开发者阅读，易于解析和生成，有效提高开发者同时分组不同部分工作协同的效率，尤其在不同设备上的协同开发。 该协议通常建立在TCP/IP协议上，因为Json协议具有轻量级的特性，为带宽有限或需要随时更改协议定义的远程设备提供实时、可靠、可扩展的消息服务。 因此JSON协议常常被应用于各类物联网设备及嵌入式系统中。TCP协议是一种非常有效的应用层传输协议你，除开少量功能位保证通信建可靠性，每个包可含括1460个字节。 TCP数据采用端到端的校验方法，保证了接收端接收到的数据的准确性。对于每一组数据，它等待接收端对ACK信号作出响应，接收到ACK的确认信号才继续接收数据，确保下位机能够正确地接收到所有正确数据。 JSON通信协议可以保证上位机和下位机之间通信的灵活性，TCP校验机制可以保证数据包传输的正确性，并避免传输中的错误或更改。

2.GStreamer 是一款用于构建流媒体应用的[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90/246339)[多媒体框架](https://baike.baidu.com/item/%E5%A4%9A%E5%AA%92%E4%BD%93%E6%A1%86%E6%9E%B6/4639266)，其目标是要简化音频与视频应用程序的开发，通过 [GStreamer](https://gstreamer.freedesktop.org/) 对 [RTP](https://wikipedia.org/wiki/Real-time_Transport_Protocol) 视频流进行多种格式的实时编解码，减小视频传输对带宽的要求，以增加视频流的显示质量与完整性。

3.上位机系统方案包括通信功能、下位机数据解析功能，实时播放视频功能，调试功能等。



**2.2程序UI开发**

用户交互界面（简称UserInterface，简称UI），界面设计家有美感之称，通过软件界面美观度的设计等，可以为人们提供一个界面简洁、动态美观、大方、使用方便的窗口。 漂亮的用户UI设计往往会创造舒适的感官体验，部分原因是缩短了人与机器之间的距离，人们更有可能使用。 因此，一个优秀的上位机UI界面需要根据工科要求来设计界面，作为一款电子控制类软件，界面需要简介明了，易于使用，提示清晰，功能效果突，“傻瓜式”使用体验。

**2.2.1用控件应用**

QT出于为开发者提供上手难度简单，门槛低的图形控件。为开发者提供了大量可以直接调用，操作方便的函数接口。只需要在进行项目开发前对这些控件做一定的了解，稍作练习熟悉接口的效果，理解每个API的属性、显示效果、使用方法。根据应用场合调整参数及如何添加事件等，就能在设计过程中能够按照自己需要熟练的操作和使用它们，减少复杂BUG的数量。

QT是一个跨平台的图像库，内含有可以直接使用window的函数库，对于一个window上位机程序。经常使用的API接口有 文本框-TextBox（）函数，按钮-Button（）API函数，标签-Label（）函数，面板-Panel（）函数，报表-CrystalReport（）函数等。

以下会详细介绍四类常用的API，具体介绍API的属性，用法以及效果，并加以自己开法过程中的一些理解，一些不常用的API接口便没有介绍。

窗体 Form 控件：

下表介绍了窗体Form控件中常用的一些属性和方法：

表 2.1 Form 窗体主要属性及方法

|  |  |
| --- | --- |
| 属性或方法 | 用途 |
| Name | 窗体的名称 |
| WindowState | 表示 Windows 窗口打开后呈现的状态，默认 Normal |
| Text | 窗体标题栏中显示的文本 |
| BackColor | 背景颜色 |
| Fond | 用于选择窗体内要显示的内容的字体类型[13] |
| Location | 用于设定窗体的左上角相比于整个界面的左上角的坐标点[14] |
| Visible | 用于设定能否让用户看到该窗体，若值等于 false，则用户不能看  到该窗体 |
| Hight | 获取或设置窗体的高度 |
| Width | 获取或设置窗体的宽度 |
| Start Position | 用于设定当我们执行程序时该窗体在屏幕上的相对位置，可以选择居中，系统默认，或者当前居中等 |
| Minimize Box | 用于设定该窗体是否能够以最小的形式显示，如果可以，那么在窗体的右上角出的最小化图标可以点击；如果不能够以最小化形  式显示，那最小化的按钮为灰色显示，处于不可用状态 |
| Maximize Box | 类似于最小化窗口属性，该属性用于设定该窗体是否能够以设计时的最大窗口显示，如果可以，那么在窗体的右上角出的最大化图标可以点击；如果不能够以最大化形式显示，那最大化的按钮  为灰色显示，处于不可用状态[15] |

2.Button 控件：

Button API控件是图形API控件最常使用的按钮API，在window上位机的界面的开发中经常使用的Button控件，常常使用Button去实现一些具体点击反馈的操作。其中最长使用API控件的方法和属性定义如下表所示。

|  |  |
| --- | --- |
| 属性 | 说明 |
| Enabled | 使按钮处于能够点击使用或者不可用无法点击的状态 |
| Click(方法) | 单击时用以实现一定的指令操作 |

按钮API库中经常用是 Click 方法，该方法能够在点击按钮之后进行相应反馈，在点击反馈中加入写好的函数，从而满足功能开关的效果。

3.TextBox 控件：

TextBox 是QT图新控件最经常使用的API控件，通常用于获取输入变量和显示提示信息以及文本信息，QT提供的TextBox 控件可以满足绝大部分场景，合理搭配使用可实现大部分需求。TextBox 控件的常见API和方法参照下表的说明。

表 TextBox 类基本功能和方法

|  |  |
| --- | --- |
| 属性 | 说明 |
| Text | 检索在控件中输入的内容 |
| MaxLength | 可在文本框中输入的最大字符个数[17] |
| Multiline | 表示能否在该控件中显示若干行的文本 |
| ReadOnly | 文本框的文本为只读 |
| AppendText | 用于追加字符。用法同 Text.Add(str)类似，但两者并  不完全相同 |
| Clear（方法） | 删除文本框内现有的所有文本 |
| Show（方法） | 用于显示一个控件 |
| KeyPress（事件） | 当我们按任意一个键时会执行该事件函数内的一些  操作 |
| TextChanged（事件） | 在文本框内的内容产生变化的情况下会执行一定的  操作 |

基本属性：

Text 属性：

Text属性是文本框函数最基本的属性之一，它用于显示用户可以看到的文本框内容。 编程人员可以通过使用Text Bow控件来实现对多个文件进行操作。 但有时由于一些原因，如用户对文档的要求不同等因素,Text属性可能会被修改或取消。需要根据具体情况具体设置。 如果要增加或删除某个文本框控例中的字符，则需要先将该控件添加到Multi Lce属性上，然后再把它放入true,此时就能实现32KB以上的字符了。

Text属性通常有三种设置方式：一种是在编写代码时修改名称，也可以直接在文本框的“属性”列中输入内容，可以根据输入的变量更改属性中内容；或者用户可以直接按键输入字符串，并通过在程序初始化阶段或其他触发事件中获取Text中的字符串。

MaxLength 属性：

MaxLenth用于设置文本框限制的输入的最大字符数，合理的设置可以避免许多不必要的字体错位，格式不统一的问题。将MaxLenth设置成0，即是解除了输入长度的限制。

MultiLine 属性：

MultiLIne用于设置文本框多行输入开关，在标题或提示文本中经常进行更改设置，为了减少显示bug或者错位的问题。

基本方法：

(1)AppendText 方法：

AppendText用于文本拼接。与.add()函数用法相似，但实现的效果不完全相同，该方法会在原有文本字符串的后面直接添加文本数据，并且可以任意次数使用。

(2)Clear方法：将控件内的内容全部删除。

该方法调用的一般方式如下：

在需要删除的文本函数后，使用Clear（），该函数没有参数。

Focus方法：

为文本框函数在特定像素位置设置一个焦点，常用于文本框的定位，是文本框根据界面的大小试试调整自己的位置，可自适应不同分辨率的设备一般以文本对象.Focus()的格式进行使用;

TextChanged 事件：

程接口（如Foundry,JAVA等）访问文件，就可以通过调用该接口来实现这种功能。这就是我们所熟悉的Text Box技术。 当开发人员可在更改文本框的内容时触发此事件，更改TextBox的Text属性或添加其他函数达到效果。

Show 方法：

通过Show函数，我们调整控件的显示属性。但在使用此方法的名称作为控件名时。 Label函数可以作为实现某些操作的提示功能的最简单控件提供对Text值的访问，并用于显示用户无法编辑的图像和文本。  在许多应用系统中,Label2是很重要的工具之一。它能使应用程序更加简洁和美观，并且可以为用户带来更多的便利。因此，使用非常广泛。 Label控件中的常见属性与方法参照表 2.4 的说明。

表 2.4Label 控件主要属性和方法

|  |  |
| --- | --- |
| 属性 | 说明 |
| BackColor | 获取或设置控件的背景颜色 |
| BorderStyle | 用来设置或返回边框 |
| Enabled | 设置或返回控件的状态 |

BackColor 属性：

BackColor（）函数用于获取或设置Label的背景颜色。上位机中自定义系统的背景颜色的方法有三种。 其中，系统对用户输入的参数进行分析并根据分析结果来自动地对环境的背景颜色进行设定；而系统则是通过手动方式来实现。系统中有很多选项，可以根据功能或美感设置任何颜色，选择阴影等效果。

BorderStyle 属性：

用于设置 Label 的边框。此时有几种边框风格： 系统默认情况下为

BorderStyle.None，没有边框。 通过使用该方法可将任意大小的图像都显示出来，同时还能在屏幕上自由地移动鼠标和键盘等操作。用户只需选择需要查看的内容就能得到想要的结果。 固定的固定单边框,以设置属性值FixedSingle.

(3)Enabled 属性：

Enabled可设置控件的点击属性，当值设置为false时,Label控件为暗灰色,无法直接边境。合理设置该属性能减少大量不必要的bug，增加开发的效率。

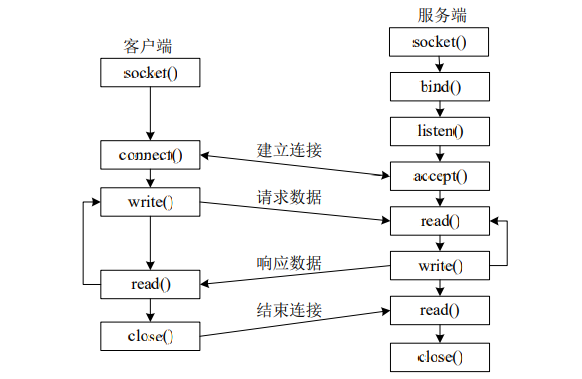
如果要修改Enabled属性，可以通过调用如下方法：

nameLab.Enabled="字符串";//设置显示内容

**2.3.1 Socket 技术**

Socket套接字技术是连接两个或两个以上终端通信链路的基石。它是一种基于TCP/IP协议的通信技术，位于传输层和应用层之间，类似于为会话层网络通讯提供的接口。 计算机作为网络通讯中最重要的设备之一，其IP地址非常多，每个IP地址对应着上千个端口，因此我们通常使用“IP+端喊”方式来实现套接字的传输。 当一台主机发生故障时，其他主机能够迅速地从其它主机获取该主机的地址信息来恢复其正常运行状态。本文主要研究如何利用套接字机制提高网络通信效率。 两个套接字之间可以是TCP连接，也可以是两个独立的套接字：服务端套接字及客户端套接字； 客户端套接字会在空闲端口上发送，而服务端套接字会同时向两台主机发送信息。 这种数据交互方式通过套接字区分不同的应用进程，一张网卡就可以实现多个TCP连接的数据并发服务，节约了计算机资源，降低了硬件成本。

如下图所示所示，一个 Socket 连接由服务端和客户端两个组成部分，.NET 控件提供了标准 Socket 类库，采用QT实现 Socket 通信的示意图



服务端不会主动发起连接，而是作为监听端，监听特定端口，等待客户端的socket连接请求。 服务端生命周期为：

1）socket()创建套接字；

2）bind()绑定端口；

3）listen()侦听队列；

4）accept()接受连接；

5）close()关闭连接。

客户端是socket网络连接的发起者，通过非指定的空闲端口发送连接请求。客户端生命周期为：

1）socket()创建套接字；

2）connect() 发起连接；

3）close()关闭连接。

在Socket连接中， 数据传输可以作为数据流来完成， 数据是按信道顺序传输的，就像水流过管道一样。它的优点是不会重复和没有错误。为了防止过多的数据造成通讯堵塞，Socket限制数据流的大小，并提供可靠的数据传输。此外，Socket是面向连接的数据交互方式，在服务端处理客户请求之后保持连接打开。 最后，本文介绍了嵌入式系统的设计与开发。 Socket是指通过建立通讯连接来实现数据传输的方法，即两台以上的主机之间可以相互通信并完成数据交互，它不需要额外的通讯规约和接口。 基于上述特点,Socket可以应用于多种工业环境下的实时测控领域。本文介绍的水下机器人监控系统就是一个典型的远程控制和监测平台。它由单片机控制器作为主控部分。 由于嵌入式系统中没有串口,IIC等标准的数据传输协议，在传输过程中易出现误码或数字丢失的现象。 因此，该系统将采用Socket和JSON数据格式实现水下机器人下位机之间的数据交互。

**2.4 JSON 数据格式**

JSON协议是一种轻量级的数据结构，为 JavaScript 的中一个子集，是一种独立于语言之外的数据机构，其具有许多优点：

1）信息描述十分简洁，体量轻量级的数据交换格式；

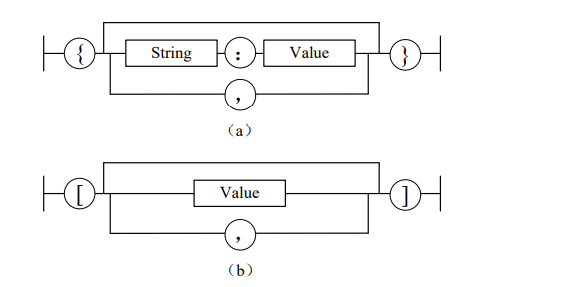
2）信息直观，十分易于开发者进行阅读、编写和维护；

3）易于计算机生成和解析；

4）独立于编程语言，不论是C， Java、 Python还是go语言都能十分便捷的使用JSON协议等。

5）JSON语法结构

JSON数据格式有对象和数组两种结构类型，JSON数据的书写格式是键（名称）/值对。JSON 键值对是用来保存JSON对象的一种方式。JSON 对象和数组的语法结构分别如图



“{”表示开始读取对象；

“}”表示结束读取对象；

“[”表示开始读取数组；

“]”表示结束读取数组；

“：”表示分隔“名称-值”中的名称和值

“，”表示分隔对象中的“名称-值”元素或数组中的“值”元素。 例如，JSON 对象：{ “电站”:“龙首” , “instantVeocity”: 12 , “instantFlow”: 34}； JSON 数组：{ 12 , 56 , 34}。

（2）JSON 序列化 JSON 序列化是将对象转换为字符串的过程，服务端对象信息经过序列化才能够被存储和传输。将信息按照 JSON 语法编写其实就是一种序列化的过程。除 此之外，JSON 还可以引用类库函数直接序列化，这种方式的优点是能够验证语法正误。序列化功能函数主要有 JavaScriptSerializer 和 DataContractJsonSerializer 两个.NET 原生类的 Serialize()和 WriteObject()，以及第三方类库 JSON.NET的SerializeObject()等。

（3）JSON 反序列化 JSON 反序列化是将字符串转换为对象的过程。客户端将接收的 JSON 格式字符串通过 JSON.parse()、eval()、DeserializeObject()等反序列化函数转化为对象后， 就可以直接提取对象中的元素，对传输数据进行解析。 JSON 虽然只是一种文本格式，与 Modbus 等通讯协议有本质区别，但是通过服务端和客户端程序支持，同样可以实现数据交互过程中的数据解析功能。采用 JSON 数据格式进行数据交互，在不影响传输速率的基础上大大简化了数据解析流程，开发人员不需要理解通讯协议，只需要提取 JSON 对象元素就能解析传输的数据内容。基于JSON数据格式的特点以及能够完成的功能，本系统将在多处不同的数据交互中应用。

**2.5 视频编码标准**

**2.5.1 主流视频编码标准介绍**

视频内部是由一连串的图像连续播放而构成的，因此视频常常会产生冗余信息。例如在一连串图像中，前后几帧之间的图像数据几乎完全相同，这些几乎完全相同的图像使得视频产生了时间冗余信息，可以通过对图像帧进行预测来消除这些冗余。另一方面，图像中相邻像素点的亮度和色度分量都是非常接近的，也可以通过邻近的像素点来预测从而消除这种空间冗余。视频编码技术是利用特定的编码方案，去除视频数据内部的时间冗余和空间冗余信息。 在 ISO 和 ITU-T 的努力和推动下，视频编码技术已经经历了数个发展阶段，被大量应用 于数字电话、会议电视、媒体转播以及实时监控等应用中。目前使用较多的视频编码标准主 要为 H.264、H.265 以及 MPEG-4。

（1）H.264 H.264 作为视频编码应用的第二代解决方案，显著提高了编码效率。H.264 带来了全新的 技术特征：4×4 的 DCT 变换降低了算法复杂度；多模式高精度帧间预测和多种帧内预测相结合，提高了编码压缩比；码流结构的改善获得了更好的信道适应性。较 H.263 而言，同等条件下能够节约 50%左右的比特率，并且对流媒体的传输的支持更好。这些优势使 H.264 被 广泛应用在视频通信领域。

**2.5.2 H.264 的技术特点和选择说明**

H.264作为目前市场份额最大的编码标准，兼具理想编码效率和优秀的网络传输适应能力。这主要得益于H.264在技术特点上的创新，引入了新的系统设计方式和编码特性。 在系统设计方式上，H.264创造性地将编码系统分割成两层，分别为视频编码层（VLC）和网络传输层（NAL），前者用于对视频核心内容信息进行压缩编码，包括了基于块的运动补 偿混合编码和一些新的编码特性，定义了宏块层及其以下的语法。后者则根据不同类型的网 络进行信息分段封装递送，支持RTP、UDP、IP、电路交换传输格式等多种传输格式，定义 了片层及其之上的语法。 编码特性上，H.264在采用基于块的混合编码方案，通过帧间预测和运动补偿消除时间冗余， 利用变换编码消除空间冗余，显著的编码特性[31]主要包括：

（1）帧内预测编码 H.264通过利用当前像素块左侧和上侧的像素对当前像素块进行预测并对实际值和预测 值的差值编码，可以有效减少空间冗余。H.264针对宏块大小提供了灵活的帧内预测编码模式， 具备很强的应用适应性。

（2）帧间预测编码 H.264利用连续图像序列前后帧的时间相关性来进行帧间预测编码。除了应用传统I、B、 P帧预测方案，还使用了高精度的运动补偿算法，引入了多参考帧帧间预测和加权预测等方法。

（3）量化与整数DCT变化 在进行变换编码时，H.264使用了整数DCT变换来近似离散余弦变换特性，以整数为基础 变化运算不但降低了计算量和复杂度，而且反变换没有偏差，物体运动的划分也更为精确， 抑制了块效应和振铃效应。

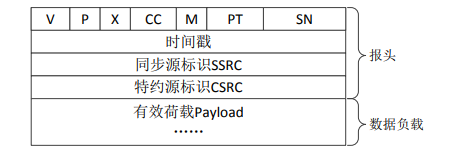
（4）去块滤波 H.264采用了去块效应滤波器来消除由高压缩比而产生的块失真效应，所有宏块按扫描顺 序进行有条件滤波。根据块编码模式、运动矢量和残差值自适应的调整块边界的滤波强度， 有效地改善了图像的视觉质量。

（5）熵编码熵编码在统计意义上对数据的冗余进行了压缩。H.264使用了两种可选的熵编码模式，基 于上下文自适应的可变长编码CAVLC以及基于内容自适应的二进制算数编码CABAC。前者 根据已编码语法元素的情况动态调整编码中使用的码表，自适应能力强较强。然而由于使用 静态概率来估计码表，限制了CAVLC的编码效率。CABAC克服了这一缺点，能够实时更新概率，更适合高码率的情况。 与其他视频编码方案相比，H.264无论是在复杂度，编码效率，还是网络状况的适用性上， 都十分出色。尽管HEVC作为新一代的编码标准，压缩性能优于H.264，但是HEVC更复杂的 算法会降低编解码的计算速度，在强调实时性的视频监控系统中，H.264作为压缩性能和算法复杂度的折衷考虑，无疑是更好的选择。并且目前而言，H.264的软硬件支持都优于HEVC。 因此本文选择了H.264编码作为本文系统的视频编码方案。

**2.6 流媒体传输协议**

**2.6.1 RTP 传输协议**

实时传输协议（RTP）是由Internet专家任务组IETF针对多媒体实时数据传输而提出的新型应用层协议。该协议基于单播或多播网络为用户提供端到端的流媒体数据传输服务[32]。RTP仅负责数据的实时阐述，通过提供时间戳信息来保持收发端的数据同步，而传输中的差错控制和网络QoS则依赖同族协议RTCP来完成。 RTP协议一般运行在UDP之上，一个RTP协议帧的结构主要由12个字节组成的固定报头和有效载荷两部分构成，如图2.3所示。其中报头又包括了以下10个部分：



（

**2.6.2 RTCP 传输协议**

实施传输控制协议RTCP作为RTP的控制部分，主要负责数据传输中的流量控制和拥塞控制。RTCP通过在收发两端周期性地发送包含发送、丢失统计信息的数据包，来反馈数据在网络中的传输情况。RTCP协议同样使用UDP来发送数据，包含五种分组类型：

（1）发送端报告SR SR用于描述发送端发送和接收的统计数据。SR报文主要包括SSRC标识符、NTP时间戳、 RTP时间戳、发送包数以及发送字节数等信息。

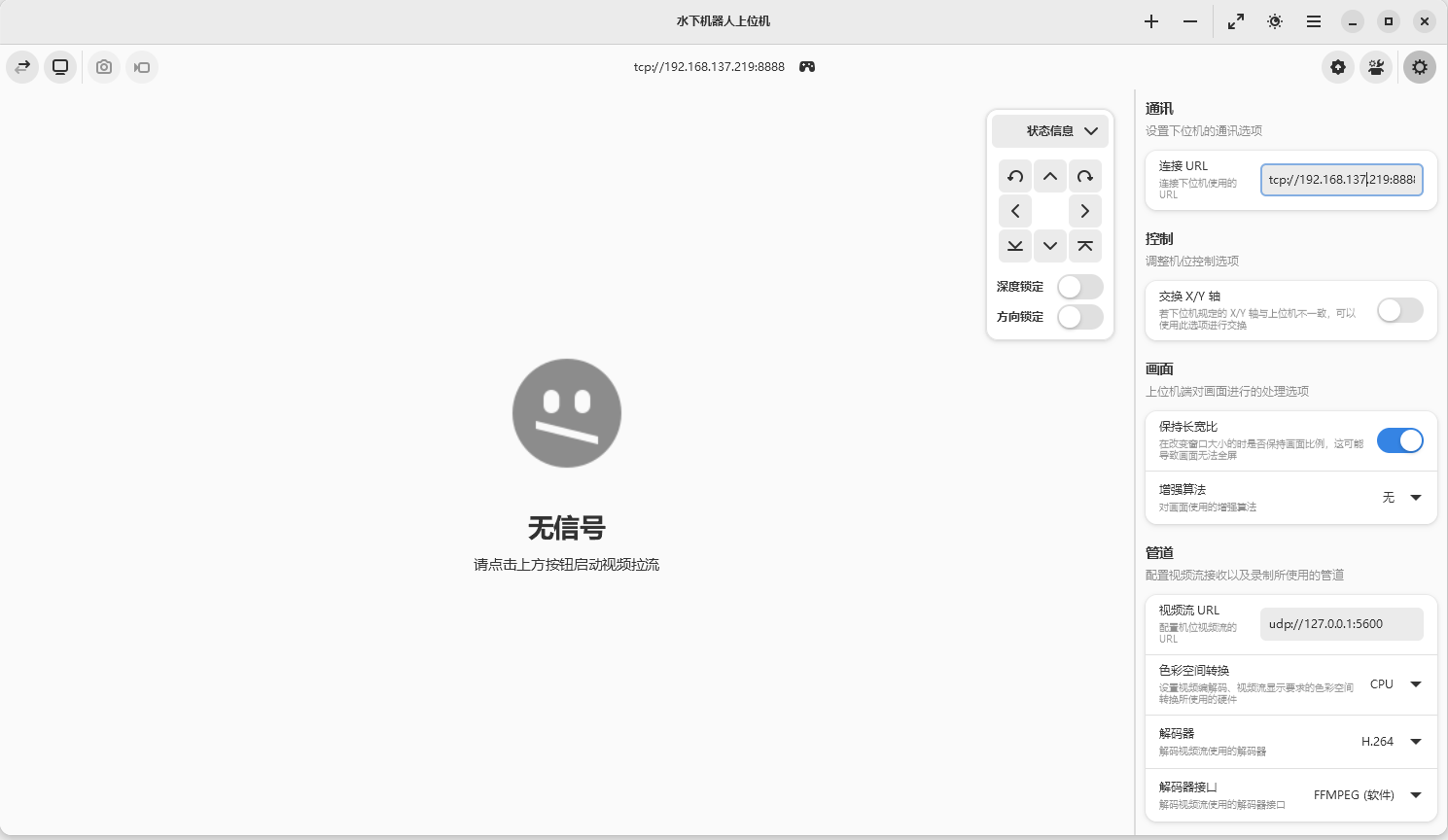
（2）接收端报告RR RR用于描述接收端的统计数据。RR报文有接收端发送，提供了累计丢包数、同步源标 识符、到达时间抖动、接收最后一个SR的时间、接收最后一个SR的延迟等信息。

（3）源描述报文SDES SDES通过直观的文本信息来描述会话的参加者，包括终端标识、名称、电子邮箱、电话 号码、用户地理位置等源描述项，这些为接收方获取发送方的有关信息提供了方便。

（4）通知离开报文BYE BYE用于通知成员的离开。当某个会话成员结束会话时会发出包含离开原因的BYE类型

**第三章 上位机软件的实现**

**3.1上位软件的功能与实现方法**

本软件基于QT作为主窗口类设计的上位机。主窗口包括主菜单栏、工具栏和状态栏，操作方便、可扩展性好。在主窗口中加入QWidget类，QWidget是所有用户界面对象的基类，可以在其中加入各种界面组件，如 QLabel、QpushButton、QlineEdit等，这些界面组 件的具体功能都是通过QWidget类得以实现。同时使用了QTabWidget组件进行分页处理，实不同功能界面的切换，上位机软件的主界面如图所示。

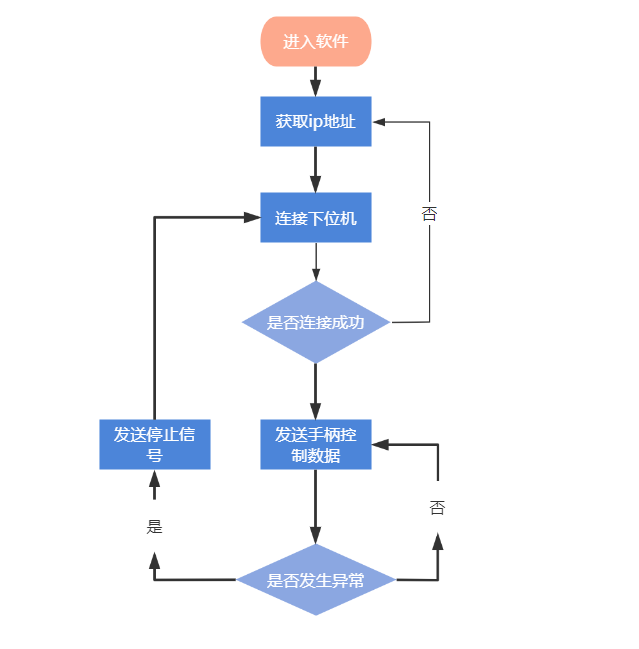
该上位机软件主要由主界面与调试界面组成：主从机的登录通信模块以及机器人当前话题列表和运动信息显示界面、可视化界面以及机器人启动设置和控制模块显示界面。可根据需要修改配置，具有较强的灵活性与通用性。上位机软件所实现的主要功能与实现方法如下：

### 3.1.1通信功能

建立上位机与下位机之间的通信：将启动上位机的电脑与下位机的嵌入式系统设置为静态IP地址（上位机与下位机静态IP地址需要在同一网段下，测试之中下位机设置为192.1468.137.219，上位机设置为192.168.137.50），在上位机通信模块中的连接URL输入下位机的静态IP地址及端口号，再通过TCP协议让上位机机登录下位机控制并获取下位机的状态数据及视频推流，上位机持续发送手柄控制数据或调试界面的调试数据，从而完成登录通信过程。

**实现方法：**调用socket库函数搭建客户端，使用QT的输入文本框API在主界面中设置连接目标的输入框。获取目标的IP地址及端口号传入connect（）函数请求连接下位机。

异常处理：除开TCP自带的异常处理，上位机与下位机发生连接异常时，上位机会先做出反应，向下位机发送异常警告与停止信号。下位机收到异常警告与停止信号时或者连接断开信号时，会停止水下器机器人的一切运动（包括推进器、机械臂、云台），以防止不受控制而暴走。



### 3.1.2 Xbox手柄控制

上位机使用市面使用率最高的Xbox手柄作为控制器，使用微软提供的Xbox手柄通信协议库获取手柄参数，按照需要映射成下位机控制数据发送至下位机，也可由上位机控制界面直接控制。手柄可控制下位机进已下动作。

1. 运动\t停止
2. 前后\左右\上浮下潜\左旋右旋
3. 机械臂抓紧\放开
4. 探照灯亮\灭
5. 摄像头变焦、对焦
6. 摄像头云台的上升\下降



**实现方法：**

1. 获取Xbox官方库

git clone git@github.com:WanderROS/Xbox360GamePad\_Qt.git

1. 对协议上图拟定的按键功能对xbox按键进行功能映射

//连接设备

QGamepad \*m\_gamepad = **new** QGamepad(0, **this**);

...

//显示手柄按下的按键的名称

connect(m\_gamepad, &QGamepad::buttonAChanged, **this**, [=](**bool** pressed){

    showCurrentBtnName(pressed, "buttonA");

});

connect(m\_gamepad, &QGamepad::buttonBChanged, **this**, [=](**bool** pressed){

    showCurrentBtnName(pressed, "buttonB");

});

...    //中间省略映射功能

//显示摇杆的值

connect(m\_gamepad, &QGamepad::buttonL2Changed, **this**, [=](**double** value){

    showBtnVaule(1, value);

});

connect(m\_gamepad, &QGamepad::buttonR2Changed, **this**, [=](**double** value){

    showBtnVaule(2, value);

});

1. 发送下位机

### 视频显示

水下机器人前段装有一颗H.264编码1080p的摄像头通过插件Gstreamer进行视频推流，上位机接收并显示在主界面中

**实现方法：**

1. 安装好Gstreamer的相关插件

yum install gstreamer-devel gstreamer-plugins-base-devel gstreamer-plugins-base-tools gstreamer-plugins-espeak gstreamer-plugins-base

gstreamer-python gstreamer-rtsp gstreamer-plugins-bad-free-devel gstreamer-tools gstreamer-plugins-bad-free gstreamer-plugins-good gstreamer-plugins-ugly

1. 下位机视频推流

进入下位机系统/etc/rc.local自启文件加入下列指令。其中根据插入的摄像头的设备号自行选择设备号输出。以H.264编码协议30帧帧率1080p格式发送至ip为192.168.137.50端口号为8888的设备上。

gst-launch-1.0 -v v4l2src device=/dev/video1 ! 'video/x-h264,width=1920,height=1080,framerate=30/1' ! rtph264pay ! udpsink clients="192.168.137.50:8888"

…

1. 在上位机中嵌入Gstreamer的拉流指令，其中以H.264方式解码，1080p播放视频流。

gst-launch-1.0 rtspsrc location=rtsp://username:passwd@ipaddr:port  latency=0 ! rtph264depay  !  capsfilter caps="video/x-h264"  ! h264parse  ! vaapidecode  !  vaapipostproc  width=1920 height=1080  !  vaapisink sync=false

### JSON通信协议

上位机与下位机之间使用 [JSON](https://wikipedia.org/wiki/JSON) 数据包通过 [TCP](https://wikipedia.org/wiki/Transmission_Control_Protocol) 协议进行通信，一个数据包可以包含任意数量的有效命令，基本框架如下：

{

    "command1" : null,

    "command2" : "arg",

    "command3" : ["arg1", "arg2"],

    "command4" : { "key1": "value1", "key2": "value2" }

    // ...

}

当前有效的发送命令如下：

|  |  |
| --- | --- |
| 命令 | 描述 |
| x | 控制机器人前进或后退 |
| y | 控制机器人左右平移 |
| z | 控制机器人的上浮或下沉 |
| set\_propeller\_parameters | 设置推进器参数 |
| set\_control\_loop\_parameters | 设置控制环参数 |
| save\_parameters | 保存参数 |
| load\_parameters | 请求读取参数 |
| firmware\_update | 进行固件更新 |
| feedbacks | 反馈数据 |
| set\_propeller\_parameters | 设置推进器参数 |

自定义协议包根据功能分为控制指令数据包与调试指令数据包

1. 控制指令数据包

{

    "x"                : 0.0,   // 左右平移

    "y"                : -0.5,  // 前进后退

    "z"                : 0.8,   // 上浮下沉

    "rot"              : 0.1,   // 左右旋转

    "depth\_locked"     : **false**, // 深度锁定

    "direction\_locked" : **true**   // 方向锁定

}

1. 调试指令数据包

{

    "set\_propeller\_parameters": {    // 推进器参数

        "back\_right": {              // 推进器名称

            "deadzone\_upper" : 8,    // 死区上限

            "deadzone\_lower" : -12,  // 死区下限

            "power"          : 0.75, // 动力百分比

            "enabled"        : **true**  // 启用/禁用推进器

        }

        // ...

    },

    "set\_control\_loop\_parameters": { // 控制环参数

        "depth\_lock": {              // 控制环名称

            "p" : 1.0,               // 比例调节

            "i" : 2.0,               // 积分调节

            "d" : 0.5                // 微分调节

        }

        // ...

    }

}

实现方法：

按照上位机与下位机定好的协议，将key值与value对应打包。

# 第四章上位软件的功能的整体测试与分析

本文所使用的实验设备为第二章中所提到的自己设计的控制板。其核心控制芯片为全志科技的Nanopi H3。该控制板上搭载着MS5837深度传感器、DCM250B三轴传感器、PCA9685 PWM扩展芯片。具体实验步骤如下：

## 4.1 测试环境配置

1. 下位机环境
2. 使用以下指令打开/etc/network/interface.conf文件

sudo nano /etc/network/interfaces

1. 向文件中添加以下指令，将IP地址修改为静态IP：192:168:137:219

allow-hotplug eth0

iface eth0 inet **static**

address 192.168.137.219

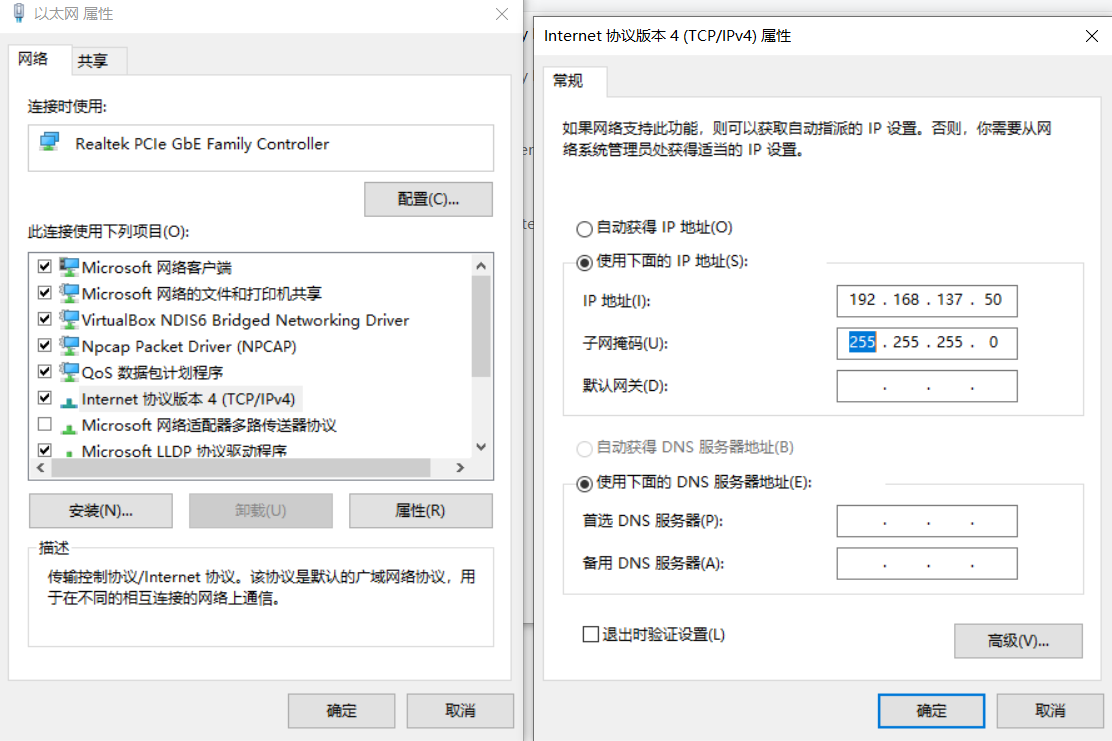
netmask 255.255.255.0

gateway 192.168.137.1

dns-nameservers 192.168.137.1

1. 上位机环境

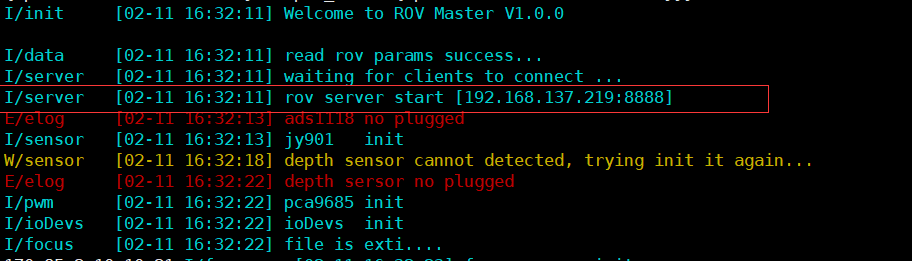
打开电脑网络适配器，将IPV4网络IP地址配置为192:168:137:50



## 4.2通信功能测试

### 4.2.1测试流程

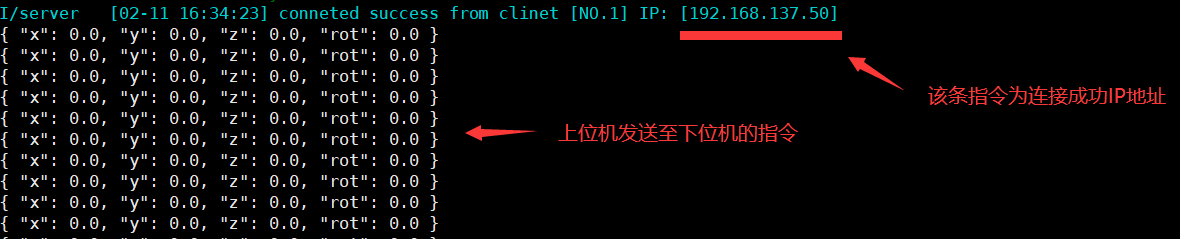
1.启动下位机程序，下位机显示当前IP为192.168.137.219:8888



2.有上节环境变量配置，在上位机通信模块中输入tcp://192.168.137.219:8888如下图。



1. 下位机显示收到IP：192.168.137.50连接请求，并连接成功。并且成功收到上位机发送的停止运动指令。



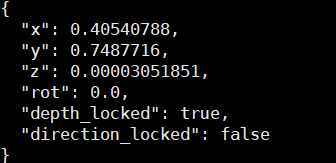
### 4.2.2 结论

上位机与下位机通信功能正常且上位机可以连接任意IP地址的主机。

## 4.3 手柄控制测试

4.3.1 测试流程

1.上位机成功识别到Xbox手柄的连接，并且下位机接收到手柄的映射数据包。



4.3.2 结论

手柄控制功能正常，且数据未产生冲突

## 4.4视频显示

4.4.1测试流程

1.下位机启动Gstreamer推流指令。

gst-launch-1.0 -v v4l2src device=/dev/video1 ! 'video/x-h264,width=1920,height=1080,framerate=30/1' ! rtph264pay ! udpsink clients="192.168.137.1:8888"

2.上位机点击拉流按键，上位机主界面能够显示水下机器人实时拍摄的画面。画面清晰流畅，没有模糊或失真。

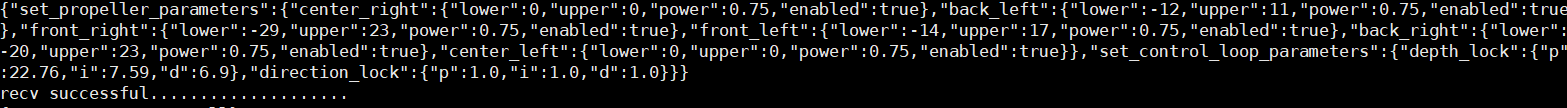
4.4.2结论

视频显示功能正常

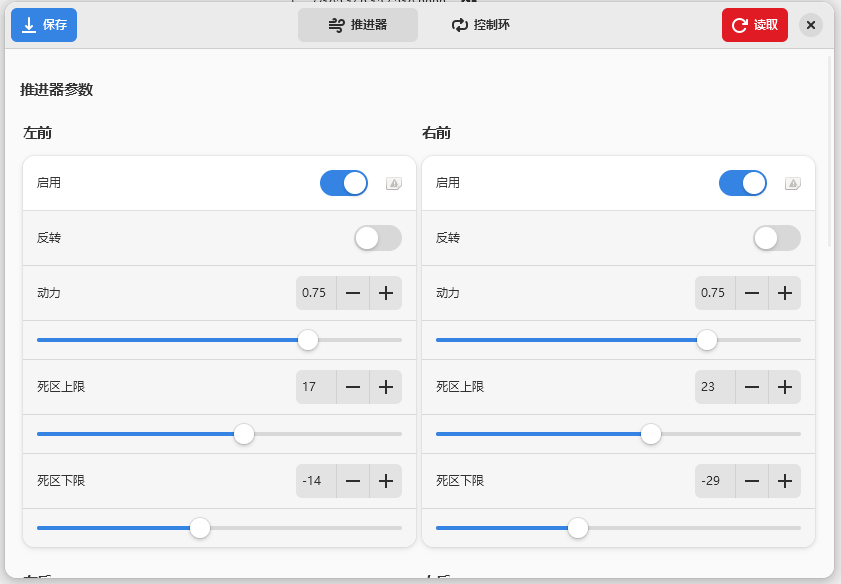
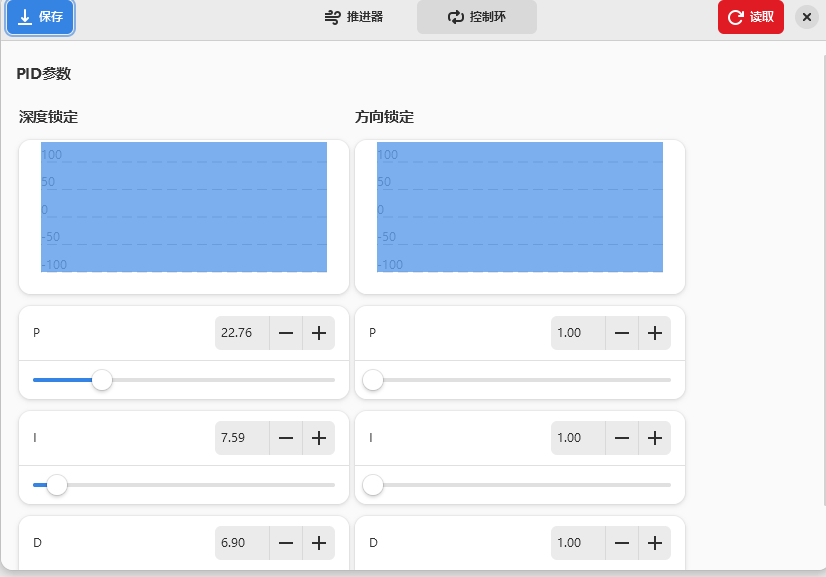
## 4.5 参数调试功能

### 4.5.1测试流程

1.点击调试界面的保存键时，下位机成功据保存至JSON文件中。



1. 点击调试界面的读取键时，上位机成功将数据显示至上位机的调试界面中

### 4.5.2结论

调试功能正常

### 结论

本文就水下机器人控制与调试不便捷这一问题，以跨平台Qt图形库框架为基础，用面向对象的C＋＋编程语言设计并实现了一个界面简洁、功能齐全、简单实用的以可靠的TCP协议作为通信协议的上位机软件。本文首先介绍了上位机软件的设计架构、TCP通信原理、JSON通信机制和Gstreamer的推流原理，然后基于原理详细的阐述了设计各功能模块的设计与实现的过程，并以最终的测试结果为实验用例来论证该上位机实现效果。并且实现下位机的状态的可视化，实验清晰的反映上位机的功能效果，验证了上位机所描述的所有功能的有效性。同时该上位机预留了控制、调试、图像增强算法等接口，增加其了扩展性与灵活性，使后续对机器人的开发更加简单。与此同时，可用用于其他嵌入式设备的控制与调试。

# 致谢语

但是在我的导师杜勇教授的帮助下，让我最终顺利完成了该课题的设计。在这期间，遇到很多难题，依靠自身难以解决，杜勇教授都在百忙之中抽出时间为我讲解，帮助我解决难题。在此，我非常感谢我的导师杜勇教授。胡教授毕生心血都花在科研事业上，为科学技术的发展奉献了很大的力量，虽然平常事情很多，但在我们遇到难题时还是抽出时间为我仔细，耐心的讲解，以严谨，认真的科学态度教导我。不仅是我的专业技术能力得以提高，还让我将技术与实际相结合，也让我对科学技术的态度更神往，更严谨。在此，我再一次感谢杜勇教授，感谢他的热心付出和辛苦栽培。

同时我还要感谢吴一亮老师，在我遇到难题的时候吴老师也给我很多帮助，并且教会了我很多的基础知识，在思路和方法不对是吴老师会给我适当的提醒，让我不至于弄错思路与方法，使我锻炼了自学能力和遇到问题的独立思考能力。非常感谢吴老师的悉心指导。我还要感谢在设计过程中同学们的帮助，让我节省了很大一部分时间，非常感谢写老师和同学们！

最后，我要感谢一直在被后默默付出不求回报的父母，在大学四年的路上给了我很大的鼓励与支持。