# 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]为了解决水下机器人功能扩展不灵活、硬件外设与算法参数调试不便、数据反馈显示不直观等问题。基于 Qt(Qt Company 开发的跨平台 C++图形用户界面应用程序开发框架。)设计的水下机器人上位机软件。通过 Socket 嵌套字配置 TCP/IP 协议，作为上位机软件与水下机器人下位机间的通信协议。使用 JSON（JavaScript Object Notation, JS 对象简谱）作为上位机与水下机器人的功能协议。上位机通过微软 Xbox 协议控件，将手柄数据映射为功能指令发送至下位机，以控制水下机器人。设计调试界面实时调节水下机器人推进的死区、动力、方向等外设参数与 PID 算法等算法参数，并将反馈数据每秒 10 组数据直观的显示在图表上。上位机实时显示水下机器人通过 Gstreamer 控件分发的视频推流,保证视频 1080P 分辨率，30 帧帧率，并且优化视频延时在 150ms 左右。该上位机集成了水下机器人的控制、参数调试、数据显示和实时视频播放等功能。

[关键词] 图形开发框架 嵌套字 功能协议 上位机 视频流组件

Research on Upper Computer of Underwater Vehicle

Li Xinkai

NO: 201821112051, Communication Engineering Major, 2018

Information Engineering College of Jimei University

Abstract: In order to solve the problems of inflexible function expansion, inconvenient debugging of hardware peripherals and algorithm parameters, non intuitive data feedback display and so on. Based on QT (cross platform C + + graphical user interface application development framework developed by QT company.) The upper computer software of underwater robot is designed. TCP / IP protocol is configured through socket nested word as the communication protocol between upper computer software and lower computer of underwater vehicle. JSON (JavaScript object notation) is used as the functional protocol between the upper computer and the underwater robot. The upper computer maps the handle data into functional instructions and sends them to the lower computer through Microsoft Xbox protocol control to control the underwater robot. The design and debugging interface can adjust the dead zone, power, direction and other peripheral parameters of underwater vehicle propulsion and PID algorithm parameters in real time, and visually display 10 groups of feedback data per second on the chart. The upper computer displays the video streaming distributed by the underwater robot through the GStreamer control in real time, ensures the video resolution of 1080p, 30 frame rate, and optimizes the video delay at about 150ms. The upper computer integrates the functions of underwater robot control, parameter debugging, data display and real-time video playback

Key Words: Qt Socket JSON Master computer Gstreamer

# 目录

# 引言

随着人类对陆地的了解愈加完善，世界各国将探索的目光转向了知之甚少的海洋。海洋占据的地球表面 70%的面积，复杂的环境纵深，低至几十米的台湾海峡，深至一万一千米的马里亚纳海沟。近年来，对海洋的展开了大量的生物与能源的勘探，有大量的新发现了，例如可燃冰等新能源。由于海洋环境过去复杂，大量的区域人类无法探测，但可以预测的是海洋中蕴含的资源远远超过了陆地。使各国意识到海洋装备技术是保证国家海洋安全、资源安全、甚至是国家安全的重要保障。因此，世界各国已经海洋的开发作为战略目标，投入大量的资金与科研力量。

另一方面，随着国家经济的不断发展，以及国家海洋牧场政策的支持。海洋渔业规模不断扩大，海洋捕捞问题日渐显著。大量的海洋渔场活动需要人工潜水进行捕捞。人工捕捞不仅效率低下，长时间潜水对人体造成不可逆的严重损伤。效能不足与不断扩大的需求形成严重的矛盾。

因此，对先进的水下装备的需求越发迫切。水下机器人作为一种可以代替人工前往危险水域的同时，具有更高的机动性、更好的观察效果、更优越的功能等的特点的水下装备，已经成为海洋勘测和作业的重要装备。因此，水下机器人研发的必要性与重要性不言而喻。

水下机器人定义：一种通过遥控或自主控制操作，对环境状况有感知探测机能，具有图像识别与传输功能，并且可在水中用于移动和控制其他设备的能力，辅助或代替人工进行水下工作的机器设备。由于电磁波在水中的衰减十分严重，有效的传播距离不足支撑长距离控制，所以水下机器人一般由水下机器人、负责数据传输各供电的脐带缆、上位机三部分组成。

水下机器人本体是一个独立的控制系统，并且往往远距离操作水下机器人进行水下工作，无法直接通过机器人本体获取信息，所以需要上位机作为控制和观测的媒介。本课题研究的水下机器人上位机分为调试、控制、监控三个主要功能。水下器机器根据需求的不同，设计时会采用不同的产商不同型号的零部件，尽管是相同的零件，也会存在无法忽视的误差。调试功能意在简化个设备参数的调节过程，提高机器人的生产效率。控制功能兼顾扩展性、灵活性与简易性。监控功能在保证视频流质量清晰流畅的同时，可拓展额外的图像算法。数据信息规避多个单字节数据解析为可单个读数据的繁琐过程。

本意在拟定一套用于开发水下机器人的标准。提高水下机器人的研制效率，降低学习门槛。

# 第一章 系统整体设计

## 1.1 系统设计方案

本文研究一种用于水下机器人通信、控制、监控、调试的上位机软件。系统分为五个主要功能模块：分别是图形模块、通信模块、控制模块、监控模块、调试模块。其中图形模块用于上位机界面的显示与引导，作为其他功能模块的载体与接口；通信模块采用 socket 配置的 TCP/IP 协议用于建立稳定的连接，保证数据交互中稳定性与准确性；控制模块采用微软 Xbox 控件用于解析 Xbox 手柄协议，映射为水下机器人动作指令并发送至下位机；监控模块中包括通过各传感器模块实现水下机器人状态于环境数据监控，如深度、温度、航向角、偏航角、距离等。另外采用 Gstreamer 视频推流组件用于实时视频监控；调试模块依赖于 JSON 协议的扩展性，实时调试机器人外设参数与算法参数，并将效果数据通过图标的形式呈现。

上位机实现具体功能如下：

1. 网络连接与数据交互
2. 实时视频监控
3. 水下环境与机器人状态监控
4. 手柄控制机水下机器人作业
5. 外设与算法参数实时调试
6. 数据图标反馈

水下机器人上位机软甲整体设计框架图如下图 1-1：

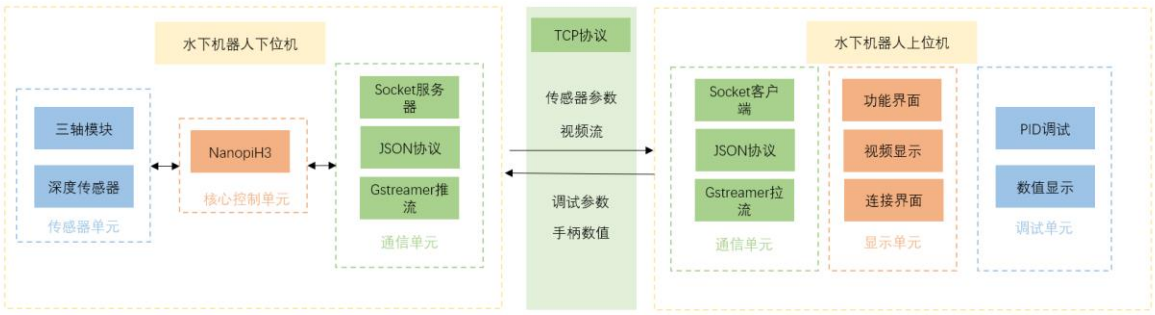



图 1-1 整体设计框架图

## 1.2 UI 设计

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。

## 1.3 Qt 控件使用

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[1]。

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。

### 1.3.1 窗体 Form 控件：

表 1-1 介绍了窗体 Form 控件中常用的一些属性和方法：

表 1-1 Form 类基本功能和方法

| 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 |
| 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] |

### 1.3.2.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 1-2 所示。

表 1-2 Button 类基本功能和方法

| 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。

### 1.3.3.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 1-3 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

### 1.3.4Label 控件

表 1-4 Label 类基本功能和方法

| 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.

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

## 1.4 socket 技术

Socket 用于描述 IP 地址和端口，是一个通信链的句柄。socket 套接字技术用于建立终端与终端通信链路，或多个终端设备间的稳定连接。它是一种基于 TCP/IP 协议，处于应用层与传输层之间的的通信技术。由于该数据传输技术采用了“点对点”传输模式，相比串口、IIC 等直接传输数据的协议，具有很高的可靠性。目前全球的数据网络都由 IPv4 协议作为网络的识别号，世界上有着数以万记，因此有着许多 IP 地址，每个地址可以挂载数万个端口作为 API 接口。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。主机一般运行了多个服务软件，同时提供几种服务。每种服务都打开一个 Socket，并绑定到一个端口上，不同的端口对应于不同的服务（应用程序），因此，在网络协议中使用端口号识别主机上不同的进程。该数据交互方式通过套接字描述符区分不同的端口与应用线程,一张网卡就可以分出多个虚拟仲端,实现多 TCP 连接的数据并发服务,节省了计算机资源并降低了硬件成本。

一个 Socket 嵌套字连接服务，由服务端和客户端两个端口组成，基于 QT 的网络控件，实现 Socket 配置的示意图。

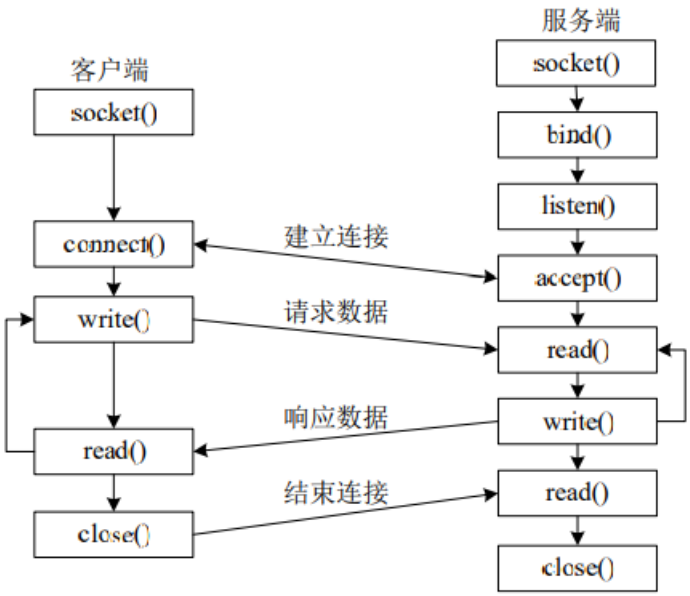



图 1-2 Socket 配置流程图

服务器被动接受连接请求，不会主动发送，而是作为监听器，监听特定 IP+端口，等待客户端发送的 socket 连接请求并确认。 利用 socket 控件创建服务端的过程为[2]：

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 网络连接请求的发起者，通过目标主机 IP 地址与非指定的空闲端口发送连接请求[3]。利用 socket 控件创建客户端连接请求的过程为：

1）socket()创建套接字；

2）connect() 发起连接；

3）close()关闭连接。

在 Socket 连接中， 数据请求是发送数据包来进行交互， 数据按信道配置顺序尽心传输，就像水流过管道一样。它的优点是不会重复和没有错误。基于 TCP 协议的流量控制，为了防止过多的数据造成缓存拥塞，Socket 限制数据包的大小，并使用 ACK 应答的校验方式来保证数据传输。此外，Socket 是面向连接的数据交互方式，依照 TCP 协议的三次握手，优先确保连接的可靠性，在服务端接受客户端请求之后将持续保持与客户端连接。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。

## 1.5 JSON 协议

JSON 协议是一种结构简单、利用率高、轻量级的文本格式的数据结构，是一种独立所有变成语言之外的数据结构，其具有诸多优点：

1）有效信息描述十分简洁，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 协议的数据结构分为对象类和数组类两种结构类型，JSON 对象类数据结构表示方法 Key/Value 对来表示并保存。而 JSON 协议的数组类则是将 Key/Value 对作为一个 Value 对象。数组类结构是一种嵌套结构以 key 值为数组名，value 为新的对象结构，JSON 对象类和数组类的数据结构的语法结构分别如图

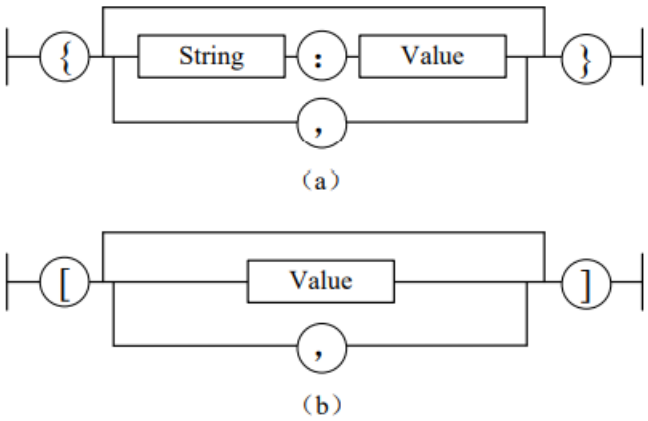



图 1-3 JSON 协议示意图

“{”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“Key-Value”元素或数组中的“Value”元素。例如，JSON 对象：{ “bool”：“true” ， “int”：22 ， “string”：lixinkai}；JSON 数组：{ “true” ， 22 ， “李昕凯”}。

（2）JSON 序列化

在使用 JSON 的过程中，没申明一个 key/value,实际上申请了以链表结构。封装 JSON 数据的过程，其实就是创建链表和向链表中添加节点的过程，其中各个 Key:value 对的地址并不连续。JSON 序列化则是将不连续的链表合并与转换为字符串的过程，客户端发送和存储的数据必须进行 JSON 链表的序列化。将信息按照 JSON 语法编写的顺序与逻辑合并为固定格式的字符串。在 QT 之中，JSON 可以直接调用 CJSON 库函数直接序列化，这种用法的优点是能够及时直观的验证语法鱼内容的正误。JSON 序列化功能函数中常用的有 JavaScriptSerializer，下位中则使用 cJSON_Print（）函数。

（3）JSON 反序列化

JSON 反序列化则是将具有特定格式的字符串转换为 JSON 链表结构的过程。客户端 JSON.parse()函数将接收的 JSON 格式字符串转化为 key-value 对的链表嵌套结构，对传输数据进行解析，然后使用 cJSON_GetObjectItem（）根据需要 key 值提取对应 value 值。JSON 虽然只是一种文本格式，但与串口协议、IIC 协议、SPI 协议有本质区别，通过 TCP 协议与 socket 嵌套字程序的支持，可以实现数据交互过程中的稳定传输。使用 JSON 协议进行通信交互，在不影响传输速率的基础上简化了自定义协议中功能位定义流程与数据流程。开发人员不需要理解 JSON 通讯协议或自定义协议具体每个字节的作用，只需要按照规定提取 JSON 对象元素中 key-value 对就能解析或修改传输的数据内容。基于 JSON 数据格式的特点以及优势，本系统将在控制、调试等多个模块的数据交互中应用。

## 1.6 视频编码标准

### 1.6.1 主流视频编码标准介绍

视频原理利用人眼反应的余辉效应，造成连续的错觉。所以视频是有一张张的连贯的图片组成的。而这一张张连贯的图片，往往每一张图片中有许多相同的的数据。这对数据的存储与解析是不要的冗余数据。例如一组连贯的图片的数据可能基本完全一致，使图片的数据存储产生空间冗余，视频播放产生了不要的时间冗余和编解码资源浪费。为了避免不必要的资源浪费，提高效率，就需要消除这组图像的空间冗余。主流的做法则是对视频的图像帧进行图像预测，消除多余的信息，减少冗余。还可以通过相邻像素预测来消除图像中亮度和色度相近的图像像素信息。最后本文会利用已有的技术探索实时视频的效果。在 ITU-T 和 ISO 等优秀编码技术的推广下，已经对视频编解码技术进行数个版本的迭代。各技术被大量应用于实时视频，监控录制，图像数据存储与传输等应用中。其中最为主流的视频编解码标准 有 MPEG-4、H.265 以及 H.264。

经过多个版本的迭代 H.264 视频编解码技术已经成熟，历代的升级极大提升了 H.264 的编码效率。提出了先进的 4*4DCT 变换算法，极大降低了算法复杂度。

优秀的帧内预测算法与多模型帧间补偿与预测相结合，提高了编解码的压缩比，在理想情况下最高能减少 70%的压缩数据。改进的码流结构模型提高了信道的利用率。相比上一代的 H.263，相同环境与条件下能够减少 50%左右的带宽，大大提高了视频的实时性与图像质量的同时，减小的对带宽的要求，从而减少硬件成本。应用的领域也更加广泛。

### 1.6.2 H.264 的技术特点和选择说明

因为先进的编码方案和优秀的编码算法 H.264 不仅具备优秀的编码效率，也兼具出色的网络传输能力。因此 H.264 是开发者最常使用的视频编码方式。H.264 领先于其他编码算法在于先进的算法思想和优越的系统设计方案。H.264 的系统设计中，按功能分为了两块。一块专门负责视频编码，对视频中的冗余信息进行预测并剔除。其中适用多种先进算法，包括运动多模型补偿算法、图像预测算法，定义了新的编码特性。在此基础上，提出了一个通用的视频编码标准框架，并给出了其详细实现步骤。因为视频传输所需要的带宽比较大，通常会占满整个带宽，所以不会设置校验位，对数据的可靠性的要求往往比较低。因此视频编码通常使用 UDP 这种无需建立的网络协议。采用新的理论进行设计的编解码在编码特性上有着卓越的编码效果，H.264 通过帧间预测算法将位置相近，属性相似的像素点进行整合算法优化。配合运动补偿算法减少图像数据的重复性，消除视频流编码的时间冗余。使用融合了变换编码与运动补偿算法特点的混合编码方案来消除空间冗余[4]。H.264 的主要编码特性包括：

（1）帧内预测编码：H.264 通过使用当前图像块左边和上侧的图像对当前图像块作出估计并对现实值和预期值的差异编号，能够有效地减小空间冗余率。H.264 根据宏块尺寸提出了灵活多样的帧内预测编码模型，具有很强的应用适应性。

（2）帧间预测编码：通常在视频流信息中，每一帧图像包含的数据与数据与前后图像间存在一定的联系。这种物体的运动关系可以用较少的函数关系进行表达，也就成了时间上的冗余。而这可以预测的运动相关性，图像中携带多余的数据信息，也是影响视频实时性的关键。H.264 在在传统的 PBI 帧预测方案基础上，加入了 多参考帧间预测与融合算法，使压缩效率提高了 50% 。

（3）量化与整数 DCT 变异在完成转换编码时，H.264 采用了整数 DCT 转换来逼近离散余弦变换特征，以整数为基本变异算法不但大大降低了统计量和复杂性，而且反转换毫无误差，物质运动的分析也更为准确，控制了块效果和振铃效应。

（4）在编码过程中，H.264 编码算法根据所建立的分割方法将图像分割为多个宏块，然后宏块中的各个子块进行运动预测，数据变换和量化编码，而基于宏块的预测编码结构可能会导致图像中的块效应。 为了克服这个缺点，H.264 提出了一种基于局部二值模式（LBP）和小波变换的方法来处理图像中出现的块效应现象。因此，需要对块效应数据进行去块滤波因此需要对产生块效应的数据进

行去块滤波。

（5）熵编码编码是建立在统计学基础上一种编码方式，以统计概率的思想对冗余数据进行区别和编码。 它能有效地降低传输过程中因丢失信息而导致图像质量下降的问题。为了进一步提高压缩比和减少计算量，熵编码被引入到视频编码器当中 H.264 中集合了 CAVLC 模型与 CABAC 模型。其中 CAVLC 模型是一种根据前后帧数据而自适应长度的编码模型。CAVAC 则是根据数据内容调节模型参数的二进制算数编码模型。CAVLC 算法可以根据开发者的需要进行更改其中的参数和模式，选择调整码表中的内容，使其自适应需要处理编码的数据。因为底层思想中使用的是静态概率调整码表，使其编码效率难以提升。CABAC 模型可以根据数据内容实时动态更新概率以使用最合适的码表进行编解码，更好地适应高码率，改善了这个问题。在保证压缩比的同时，CAVAC 模型还针对网络传输做了一定的优化。在视频编码标准中新提出了一钟 HEVC 的编码标准。HEVC 有着及其出色的压缩比，但由于算法过于复杂，需要的编码时间过长。对处理单元的要求过高，不适用于实时视频。在压缩率、实时性对硬件的要求来考虑，该系统最适合使用的视频编码标准就是 H.264。并且目前市面上或软件接口中 H.264 的解决方案更加齐全和完善。因此选择 H.264 编码标准作为上位记得视频编解码方案。

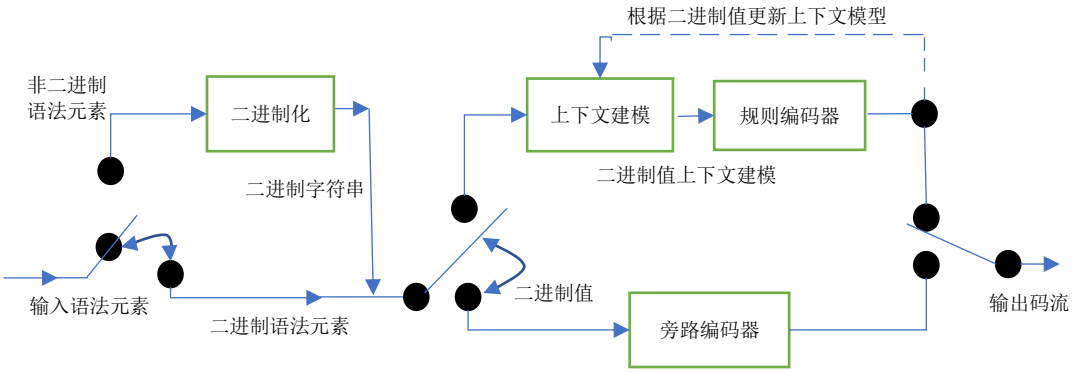



图 1-4 CABAC 编码器框架结构图

## 1.7 流媒体传输协议

### 1.7.1 RTP 传输协议

实 RTP(Real-time Transport Protocol)实时传输协议，是研究者针对语音传输或视频传输设计的应用层实时传输协议。RTP 协议提供服务端到客户端，单播或广播的语音、视频数据传输服务。RTP 只负责发送时间信息与语音、视频的同步标志位，并不负责保证传输的质量。流媒体传输协议的纠错控制则是通过 RTCP(RTP Control Protocol)负责。RTP 协议通常搭配 UDP 一起使用。一个 RTP 协议帧如下图。一个数据帧长度为 12 个字节，其中由包头和数据负载两个功能组成。


图 1-5 RTP 协议帧功能图

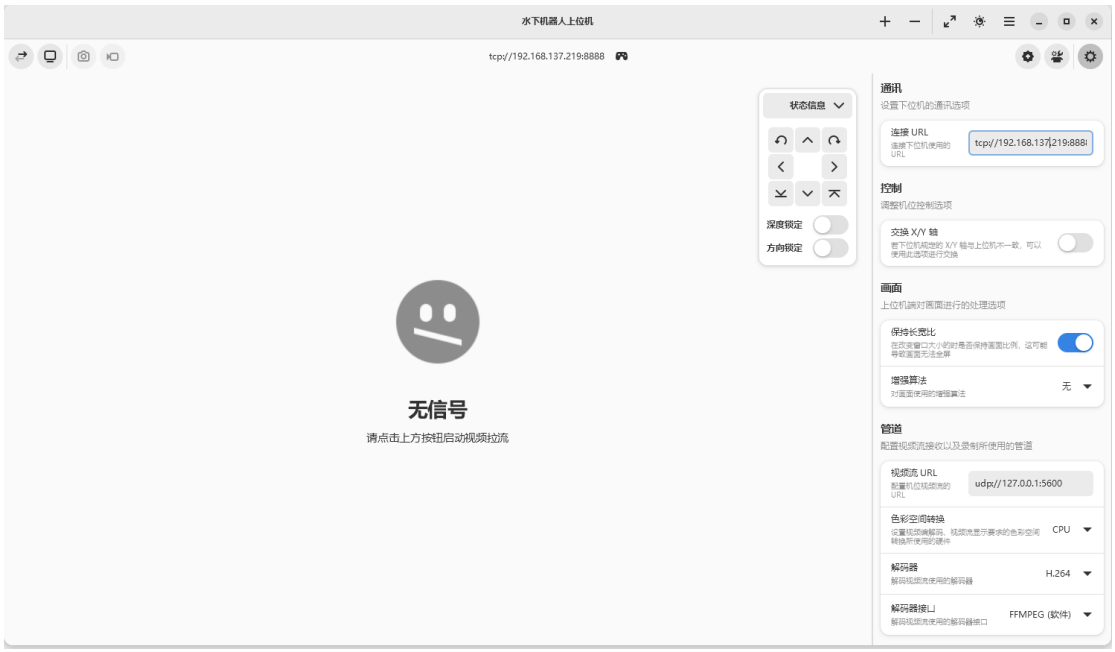
### 1.7.2 RTCP 传输协议

RTCP 传输协议与 TCP 的传输控制方式相同,有类似的流量控制和拥塞控制机制。 由于 RTCP 具有良好的扩展性，可以很方便地扩展到各种应用场景之中,RTCP 已经成为网络传输中一种重要的协议形式。由于 RTCP 具有良好的扩展性,可以很方便地扩展到各种应用场景之中,RTCP 已经成为网络传输中一种重要的协议形式。RTCP 协议会统计发送成功的数据量和丢失信息的地址，反应传输的成功率和丢包率。TCP 协议也基于面对传输效率的 UDP 协议发送数据包。

# 第二章 上位机软件的实现

## 2.1 上位软件的功能与实现方法

该上位机是基于 QT 作为主机主窗口类的设计。该系统主要分为三个界面：主界面、菜单显示界面以及数据存储与调试界面。主窗口包含主菜单栏、工具栏、状态栏，操作简单，可扩展性好。将 QWidget 类添加到主窗口中，主窗口是所有用户界面对象 QWidget 基类。您可以添加各种界面组件，如 QLabel、QpushButton、QlineEdit 等[5]。同时，使用 QTabWidget 组件进行分页处理、不同功能的接口切换，上位机软件的主界面如图 2-1 所示。




2-1 上位机主界面

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：

## 2.2 socket 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。

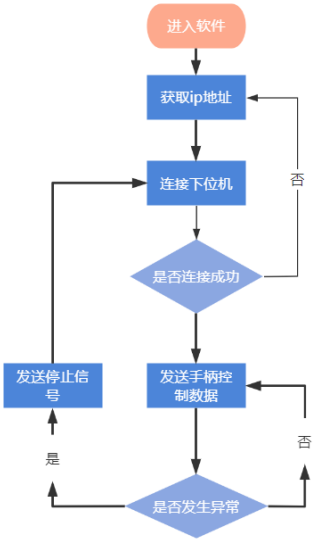



图 2-2 连接模块流程图

## 2.3 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。

a) 运动\t 停止
b) 前后\左右\上浮下潜\左旋右旋
c) 机械臂抓紧\放开
d) 探照灯亮\灭
e) 摄像头变焦、对焦
f) 摄像头云台的上升\下降


图 2-3 手柄功能定义图

**实现方法：**

从 github 上获取 Xbox 官方库。对协议上图拟定的按键功能对 xbox 按键进行功能映射，以 JSON 协议格式编码，发送下位机。主界面分为控制界面与调试界面。控制界面与调试界面发送数据包独立开来，所以在发送数据包是，程序先判断当前是否处于控制界面。当处于控制界面时，获取手柄数值判断是否有动作。确认手柄动作时，向下位机发送控制指令。如图 2-4。

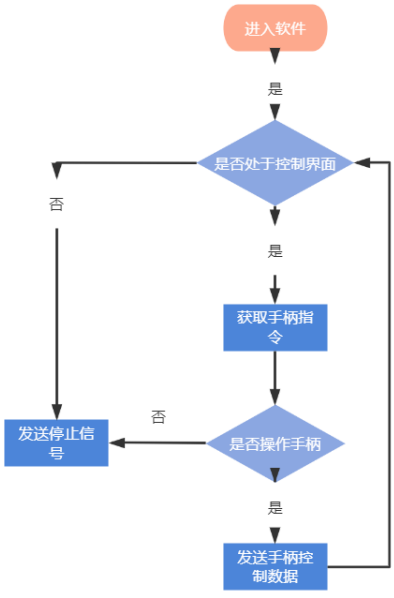



图 2-4 手柄模块

## 2.4 视频流分发

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。为了保证视频流的流畅度与画面质量，需要设定好视频流的分辨率与帧率。图像数据大小计算公式如式(1)。

$$带宽大小（M）= \frac{视频编码率(kpbs)}{8} * 帧率（N） \quad (1)$$

1080P 视频分辨率的码率一般为 3500kpbs，由上图公式计算需要带宽为 12Mbyte。核心板 Nanopi 可支持的带宽为 12.5Mbyte。

**实现方法：**

安装好 Gstreamer 的相关插件下位机视频推流。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。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。当拉流失败时，上位机确认是否连接成功，随后重新拉流。程序流程如图 2-5。

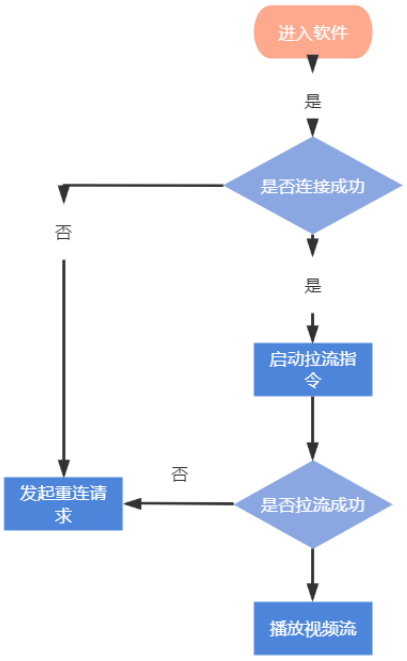



图 2-5 视频拉流流程图

## 2.5 功能协议

上位机与下位机之间使用 JSON 数据包通过 TCP 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表 2-1：

表 2-1 功能协议参数与解释

| 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
}
```

2. 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：**

以 CJSON 函数，cjson_creat()创建头链，按照上位机与下位机定好的协议，将 key 值与 value 对应打包，控制协议以各个控制功能协议直接打包，调试功能由 parameters 嵌套型 cjsonadd（）内封装参数，其中需要根据数据类型分别使用添加函数。最后使用 unprintf（）转化为字符串发送，程序流程如图 2-6。

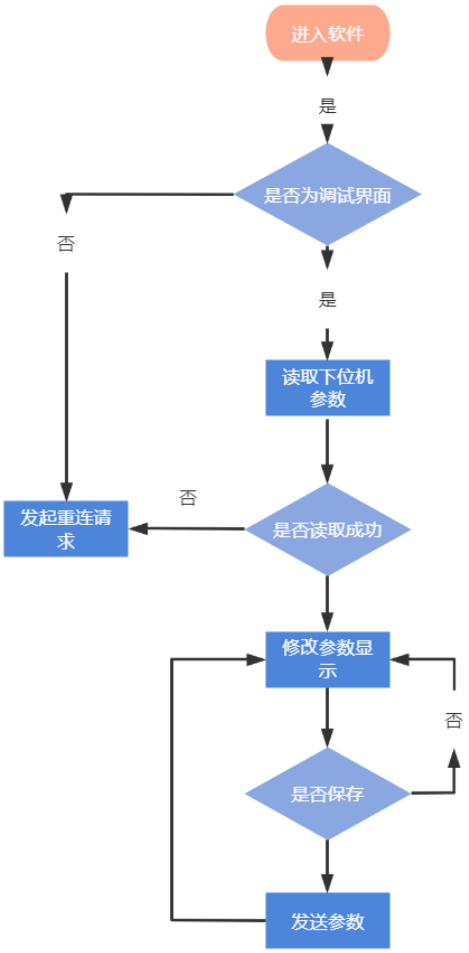



图 2-6 调试界面流程图

# 第三章 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：

## 3.1 测试环境配置

### 3.1.1.下位机环境配置

1) 使用以下指令打开/etc/network/interface.conf 文件

```
sudo nano /etc/network/interfaces
```

2) 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
```

3) 安装 gstreamer 组件

```
sudo sudo apt-get install libgstreamer*
```

### 3.1.2.上位机环境配置

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

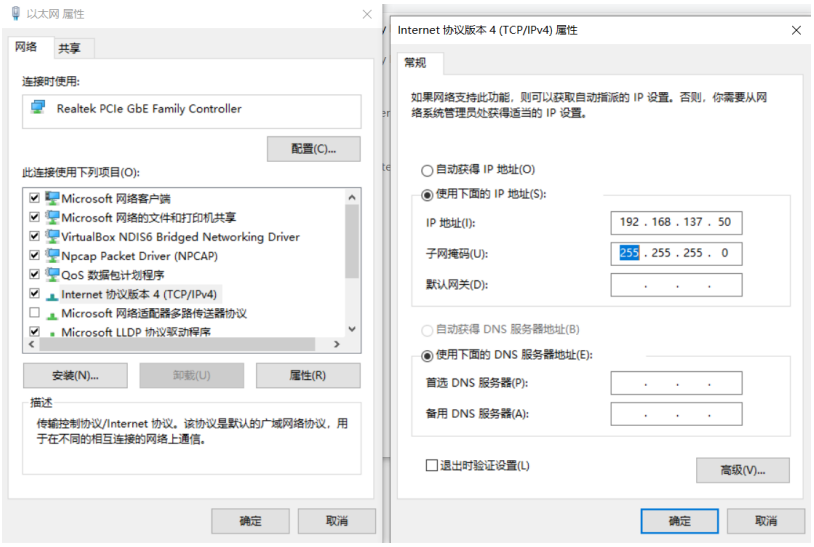



图 3-1 上位机网络环境配置

## 3.2 Socket 通信功能测试

**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 如图 3-2

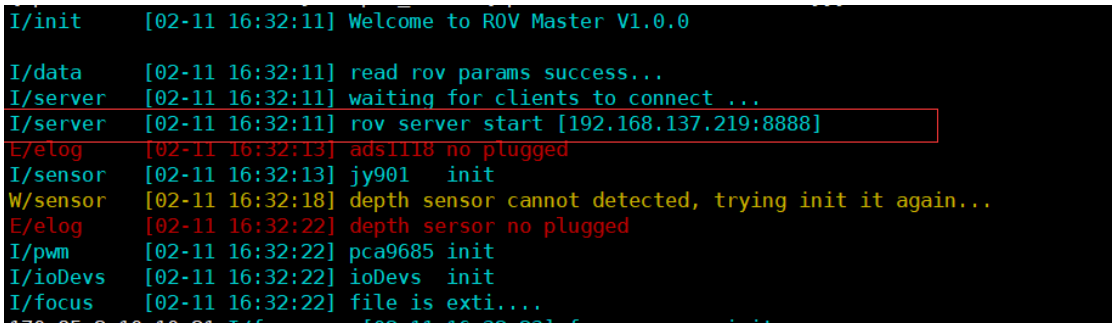



图 3-2 socket 服务器

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 如图 3-3。


图 3-3 上位机网络连接

3. 如图 3-4 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。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。

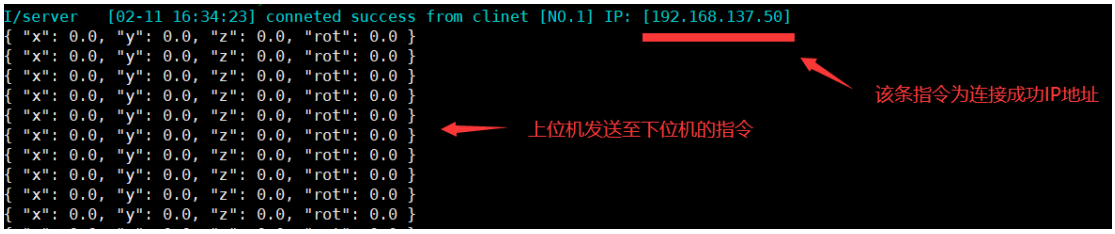



图 3-4 服务器连接请求

## 3.3 手柄控制测试

如图 3-5 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。如图 3-6 手柄控制功能正常，且数据未产生冲突。机器人按照指令进行运动。

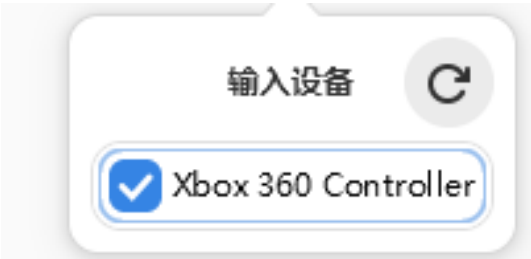



图 3-5 上位机手柄连接

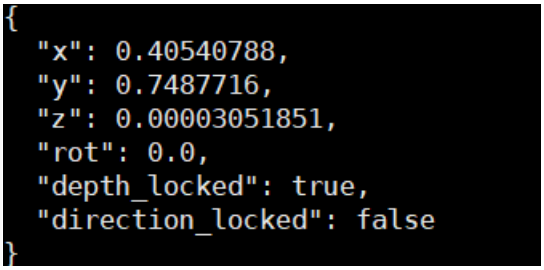



图 3-6 下位机接收手柄指令

## 3.4 视频显示

**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。

## 3.5 参数调试功能

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。

{"set_propeller_parameters":{"center_right":{"lower":0,"upper":0,"power":0.75,"enabled":true},"back_left":{"lower":-12,"upper":11,"power":0.75,"enabled":true
},"front_right":{"lower":-29,"upper":23,"power":0.75,"enabled":true},"front_left":{"lower":-14,"upper":17,"power":0.75,"enabled":true},"back_right":{"lower":
-20,"upper":23,"power":0.75,"enabled":true},"center_left":{"lower":0,"upper":0,"power":0.75,"enabled":true}},"set_control_loop_parameters":{"depth_lock":{"p"
:22.76,"i":7.59,"d":6.9},"direction_lock":{"p":1.0,"i":1.0,"d":1.0}}}
recv successful.....

图 3-6 读取参数文件

2. 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。

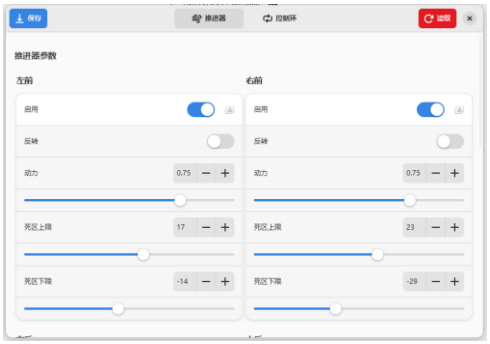



图 3-7 上位机调试界面

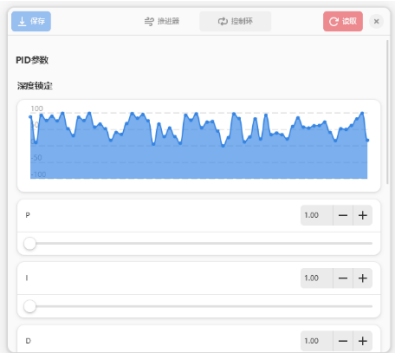



图 3-8 反馈数据图表

# 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，以 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。

同时我还要感谢陈彭老师，在我遇到难题的时候陈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陈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。非常感谢陈老师的悉心指导。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！

# [参考文献]

[1] 梁玉. 基于C#的数据采集上位机软件设计[D].西安电子科技大学,2014.
[2] 何军红,张迪,张力,薛文琦.基于 TCP/IP 协议的异构网络的数据采集[J].工业仪表与自动化装置,2019(03):77-80+110.
[3] 曾锡安. 基于C/S与B/S混合结构的生态放流远程监测系统研究与开发[D].兰州理工大学,2019.
[4] 宫健. 基于Gstreamer的嵌入式流媒体传输系统的研究与实现[D].南京邮电大学,2016.
[5] 崔奇,夏浩,滕游,刘安东.移动机器人自主导航系统及上位机软件设计与实现[J].计算机测量与控制,2022,30(01):141-146.DOI:10.16526/j.cnki.11-4762/tp.2022.01.022.
[6] 贾贝贝,康明才.基于 QT 的嵌入式系统文件传输上位机软件设计[J].电子设计工程,2022,30(03):122-125+130.DOI:10.14022/j.issn1674-6236.2022.03.027.
[7] 李辉. 基于RTP的H.264视频传输系统的设计与实现[D].吉林大学,2013.
[8] FAKIOTAKIS M，LANEDM，BRUCEJ，DAVIES C.Review of Fish Swimming Modes for Aquatic Locomotion[J]. IEEE Journal of Ocean Enginnering，1999: 237-238
[9] hang J,XuanD,KimAW,etal.Astudy of autonomous parking for a 4-wheel driver mobile robot [A].Proceedings of the 26 Chi nese Control Confetence [C].2007:179-183
[10] Jing Wu, Tongwen Chen. Design of networked Control Systems with Packet Dropouts[J]. IEEETransactions on Automatic Contr01. 2007, 52(7): 1314-1319.
[11] Mohammad Tabbara, Dragan, Stability of networked Control Systems with Stochastic Protocols[C]. Proceedings ofthe 2007 American Control Conference,new York, USA, July 11, 2007.
[12] (美)AerhonyJones, Jim Ohlund(著)，杨合庆（译），network Programing forMicrosoft windows，北京: 清华大学出版社，2002.