

深海无人遥控潜水器水面监控软件的设计与实现

李志强 吴超

(上海交通大学船舶海洋与建筑工程学院 上海 200240)

摘要 针对深海无人遥控潜水器(ROV)实时远程监控的需求以及传统ROV水面监控软件人机交互性差的问题,详细研究软件的实时性策略和虚拟仪表开发等关键技术,基于海龙系列ROV平台设计开发一套水面监控软件。软件采用三层架构的体系结构,实现串口通信、数据处理、状态监视、设备控制、警报显示等主要功能。经测试,该水面监控软件可在Windows系统下稳定运行,具有良好的实时性以及友好的人机交互性。

关键词 无人遥控潜水器 水面监控软件 三层架构 实时性策略 虚拟仪表

中图分类号 TP311.5 **文献标志码** A **DOI**:10.3969/j.issn.1000-386x.2024.11.006

DESIGN AND IMPLEMENTATION OF DEEP-SEA REMOTELY OPERATED VEHICLE SURFACE MONITORING SOFTWARE

Li Zhiqiang Wu Chao

(School of Naval Architecture, Ocean & Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract In response to the demand for real-time monitoring of deep-sea remotely operated vehicle (ROV) and the problem of traditional ROV surface monitoring software visualization, the key technologies of software real-time strategy and development of virtual dashboard were studied in detail. A set of surface monitoring software was designed and developed based on the Sea Dragon ROV platform. The software adopted a 3-tier architecture, which achieved the main functions such as serial communication, data processing, status monitoring, equipment control and alarm display. Tested, the surface monitoring software can run stably under Windows system, with good real-time and friendly human-computer interaction performance.

Keywords Remotely operated vehicle Surface monitoring software 3-tier architecture Real-time strategy Virtual dashboard

0 引言

海洋资源开发^[1]和深海科学考察^[2]的发展增加了对深海无人遥控潜水器(ROV)的需求。ROV采用脐带缆作业,由操作员在母船上对ROV进行远程控制。操作员在母船上可通过安装在ROV本体上的光学、声学探测设备实时观察海底状况,并通过脐带缆发送控制指令对ROV、机械手、配套的作业工具包进行操作。因此,ROV具备较强的深海作业能力和人工现场实时干预能力,是海洋水下工程的核心装备,广泛支

持各类典型深海水下应用,主要包括:深海矿产资源勘探开发、海洋油气勘探开发、深海调查研究和生物利用、深海打捞和应急处置^[3]。面对如此广泛的应用前景,ROV装备研发已经成为深海探测技术和海洋装备技术的重要课题^[4]。与此同时,由于ROV系统是横跨水面和水下的空间大系统,而水面监控软件是连接ROV水面和水下系统的中枢控制系统,完成对ROV运行状态的在线监视和设备控制等重要工作。

在目前ROV研究领域,美国、加拿大、英国、法国、德国和日本等国家处于领先地位,这些国家已经拥有了各种作业能力的ROV先进装备,并且配套形成了一

系列功能完善和人机交互性能好的 ROV 水面监控软件^[5-6]。国内 ROV 研究起步晚,且由于深海装备研发投入大、风险高、周期长,直到近十年国内的 ROV 技术才有了快速发展,但与国外先进水平仍存在较大的差距^[7]。其中,ROV 水面监控软件相较国外同类软件,在人机交互性等方面存在较大的改进空间^[8]。

ROV 在深海作业时,受水下复杂环境等影响,为了保证整个系统的安全性和可靠性,需要针对性地开发水面监控软件,以实现 ROV 的在线监控。为此,ROV 水面监控软件应当具备实时通信、数据处理、状态监视、设备控制、警报显示等功能。由于 ROV 水面监控软件面临的是实时作业环境且存在大量的人机交互场景,因此如何满足软件的实时性要求和提高软件的人机交互性是当前 ROV 水面监控软件设计开发的主要问题。

针对上述软件的功能需求和主要问题,本文基于海龙系列 ROV 平台^[9],设计实现一套水面监控软件。并从数据帧格式定义和数据帧同步算法两方面,详细研究软件的实时性策略,同时基于 Qt 自定义控件开发技术,设计实现虚拟罗盘、姿态仪表、位置仪表、推进器仪表等 ROV 虚拟仪表,有效地提高了软件的人机交互性和可视化程度。

1 总体设计

1.1 软件的体系架构

本文设计的 ROV 水面监控软件是基于 Qt 开发平台的 PC 端上位机软件,其体系结构采用三层架构,包括用户界面层、业务逻辑层、数据访问层,如图 1 所示。

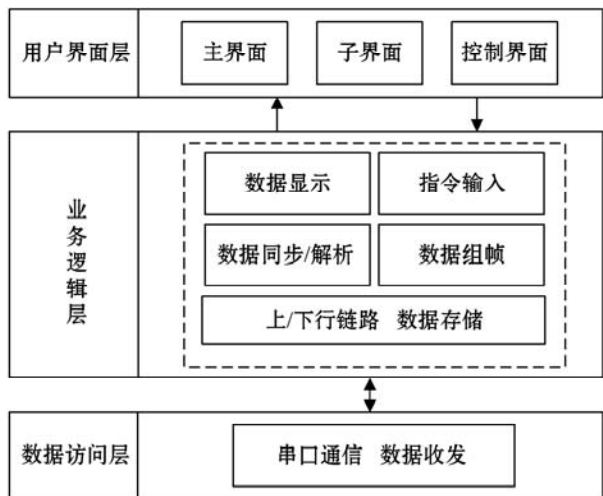


图 1 软件分层架构

1) 用户界面层是 ROV 水面监控软件的人机交互层,操作人员可以通过软件的用户界面对水下 ROV 执行

状态监视和指令下发等操作。本文设计的 ROV 水面监控软件用户界面采用主/子界面架构,包括主界面、控制界面 2 个主要界面以及串行数据界面和数据诊断界面等 6 个子界面。软件每个界面都承担不同的功能,如表 1 所示。

表 1 软件界面架构

序号	界面名称	界面主要功能
1	主界面	ROV 状态信息显示
2	控制界面	ROV 控制指令下发
3	串行数据界面	显示最新通信数据
4	数据诊断界面	显示板卡数据信息
5	变量设置界面	用户变量设置
6	软件说明界面	软件使用说明
7	保存设置界面	数据保存设置
8	串口信息界面	串口信息查询

2) 业务逻辑层位于用户界面层和数据访问层之间,用以完成传输数据的处理和转换。图 2 给出了一种简单情形的 ROV 水面监控软件交互时序,在实际作业时,水下 ROV 按照 30 ms/帧的通信频率向水面监控软件发送上行数据帧,水面监控软件也按照同样的通信频率向水下 ROV 发送下行数据帧。图 2 反映了沿着不同的数据流方向,业务逻辑层的模块调用时序和各个模块所完成的主要功能。值得注意的是,数据同步是指 ROV 水面监控软件将异步到达的原始上行数据,按照通信双方制定的上行数据帧格式,进行有效数据帧的识别和抓取。数据解析是指将获取的有效上行数据,根据数据帧格式,进行字节逻辑转换,将数据帧中的原始字节转换为相应的 ROV 状态信息。而数据组帧是指将操作员输入的控制指令,按照通信双方制定的下行数据帧格式,进行字节逻辑转换,组装成规定格式的下行数据帧。



图 2 软件交互时序

3) 数据访问层完成上位机和下位机之间原始数据的传输。本文设计开发的水面监控软件采用串口通信的方式与 ROV 进行实时的信息交互。通信双方需要约定串口通信的波特率、起始位、数据位、停止位和校验方式等,以保证通信双方能够正确地收发数据。

1.2 软件的模块划分

本文对 ROV 水面监控软件进行模块化设计,根据软件的体系架构,将软件划分 9 个模块,如图 3 所示。

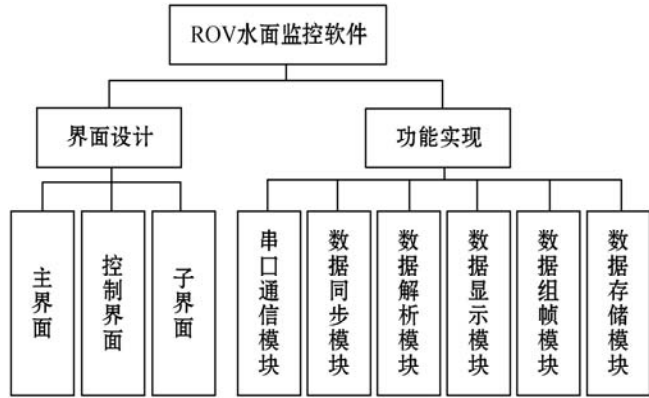


图 3 软件模块划分

基于模块化和层次化的软件架构设计可以有效地提高 ROV 水面监控软件的可维护性和可扩展性^[10]。软件模块之间保持相对独立,不同层次之间逐层调用,通过模块间的接口,进行数据的访问和传输,使得软件趋于高内聚低耦合,同时也使得软件更加易于开发和维护。

2 关键技术

2.1 实时性策略

由于 ROV 水面监控软件面临的是实时作业场景,一方面需要不断地接收来自下位机的数据,通过数据同步和解析等处理,实时地显示 ROV 的状态信息;另一方面需要迅速响应水面操作员的人工干预,实时地下发控制指令。与此同时,ROV 部分运动控制功能有向水面迁移的趋势,比如 ROV 动力定位和循线控制等,这对通信频率的要求越来越高,达到 30 ms/帧,实时性也逐渐成为水面监控软件的关键要求。

本文设计的 ROV 水面监控软件采用串口通信的方式,完成上位机和下位机的数据传输,其上/下行链路的通信频率为 30 ms/帧。图 4 给出了 ROV 水面监控软件的实时性描述。数据帧的传输耗时约为 5 ms,控制功能以动力定位为例,其核心控制算法可以控制在 10 ms^[11]。为保证软件的实时性,水面监控软件则必须在 20 ms 内完成对数据帧的规定处理。实际上为

了给控制算法预留更多的时间以保障控制的精确性和可靠性,数据处理耗时应尽可能短。对此,本文从上行数据帧格式定义和数据帧同步算法两方面研究了 ROV 水面监控软件的实时性策略。

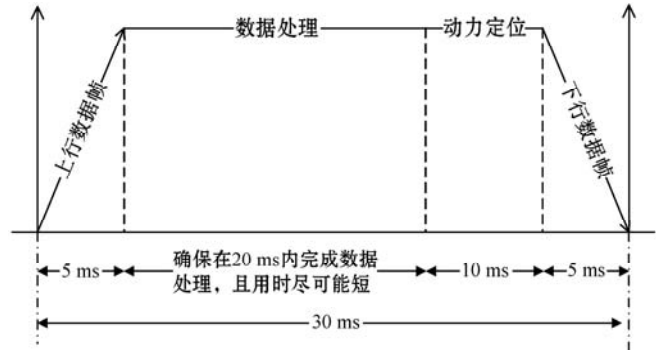


图 4 软件的实时性描述

1) 上行数据帧格式定义。由于 ROV 水面监控软件数据处理耗时,主要用于上行数据帧的同步和解析等操作。而数据帧作为串口通信数据传输的基本单元,它的长度直接影响了后续数据处理的效率和速度。

本文的 ROV 平台,其上行数据帧需要定义的信息主要包括 ROV 方向、位置、姿态、传感器数据、警报信息、推进器状态、ROV 板卡状态等信息。由于 ROV 不同的状态信息对更新率的要求不同,比如 ROV 的板卡信息,它显示了水下 ROV 内部板卡的状态信息,其不一定需要在每帧数据中进行更新。而本文的 ROV 平台搭载了包括功率继电器板卡、信号继电器板卡、传感器板卡、灯光系统板卡、摄像系统板卡等在内的 11 块板卡,每块板卡的信息在定义数据域时被分配了 16 个字节。如果在每帧数据中对所有板卡的状态信息全部进行更新,则上行数据帧的长度至少会增加 11×16 个字节。因此,针对 ROV 状态信息更新率不同这一特点,本文在定义上行数据帧时,按照表 2 进行设计定义。

表 2 上行数据帧格式定义

区域	字节偏移量	内容
帧头	0 ~ 2	0x10 0x82 0x80
数据域	3	板卡地址
	4 ~ 19	板卡数据 0 ~ 15 字节
	20 ~ 77	其余 ROV 状态信息
帧尾	78 ~ 79	CRC 校验码

如表 2 所示,本文设计定义的上行数据帧格式共 80 个字节,在设计数据域时,通过设置板卡地址标识字节,将具有不同更新率要求的 ROV 状态信息区别开。ROV 本体搭载的 11 块板卡对应不同的板卡地

址,在实际通信时,按照一定的顺序轮流发送。相当于板卡信息的更新率为 330 ms/帧,其余 ROV 状态信息则为 30 ms/帧。比起直接在每帧数据帧中将所有板卡信息全部定义,采用这种方式定义的数据帧的长度被大幅缩短,而由于数据帧同步和解析等操作是以数据帧为单位进行处理,因此上行数据帧长度的缩短必然会降低后续的数据处理耗时。

2) 数据帧同步算法。数据帧同步用于对有效上行数据帧的快速抓取,是后续数据解析和存储等操作的前提和基础,其抓取的有效性和效率是影响数据处理耗时的重要因素。

基于有限状态机理论,上行数据帧同步的过程可以分为帧长判断、帧头识别、数据域判断、帧尾校验四个状态^[12],数据帧的同步算法实际上也就是在以上四个状态间循环切换,以达到实时抓取有效数据帧的目的。据此本文设计实现基于有限状态机的数据帧同步算法。

当进行数据帧同步时,基于有限状态机的数据帧同步算法流程如图 5 所示,分为以下几步。

步骤 1 将串口通信模块接收到的数据写入识别缓冲区,进入帧长判断状态,判断缓冲区字节数是否大于等于一帧数据长度。若小于,说明该缓冲区内未包含有效数据帧,继续将接收数据写入缓冲区尾部。

步骤 2 如果缓冲区字节数满足条件,则进入帧头判断状态,搜寻缓冲区内是否存在帧头。如果不存在,说明此时缓冲区内无有效数据帧,清空缓冲区,重新进入帧长判断状态。

步骤 3 如果缓冲区内存在帧头,则帧头判断通过,记录此时的帧头位置,进入数据域判断状态,判断缓冲区大小与帧头位置之差是否大于等于一帧数据长度。如果小于,说明此时缓冲区内暂无有效数据帧,清除帧头之前的数据,重新进入帧长判断状态。

步骤 4 如果数据域判断通过,则进入校验状态,本文采用循环冗余校验算法^[13],判断该帧数据是否有效。同理,如果校验未通过,说明此时缓冲区内该段数据为无效数据帧。同时判断缓冲区是否存在第二个帧头,如果存在,则清除缓冲区内第二个帧头之前的数据,再次进入数据域判断状态。否则说明缓冲区内无有效数据帧,清空缓冲区内数据,重新进入帧长判断状态。

步骤 5 如果校验通过,则抓取该数据帧,同时清除缓冲区内已抓取字节,判断此时数据是否已全部同步完毕。若未结束,则再次进入帧长判断状态。如此循环进行,直到所有串行数据帧同步抓取完毕。

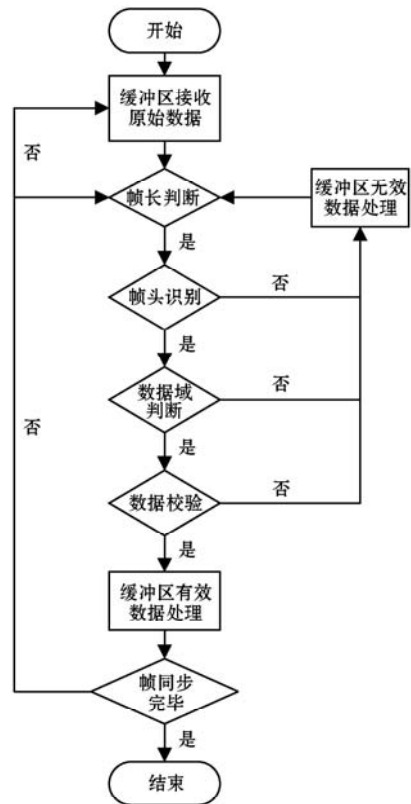


图 5 基于有限状态机的数据帧同步算法流程

1) 算法的有效性:根据上述处理流程,可以对数据帧格式进行严格审查,过滤无效数据帧,实现上行数据包的有效抓取。根据本文定义的上行数据帧格式编写测试用例,模拟可能会出现的数据帧同步情形,以验证本文算法的有效性。实验记录如表 3 所示,为方便表述,表 3 中 A 代表帧头, B 代表数据域, C 代表帧尾(校验码)。字母带有下划线,说明数据帧的该部分存在问题,属于无效数据帧,需要排除。

表 3 数据帧同步算法有效性

同步情形	测试用例	步骤流程	结果分析
无错	A B C	1 2 3 4 5	成功提取
无帧头	<u>A</u> B C	1 2 1	成功排除
多帧头	<u>A</u> B C	1 2 3 4 1(或 3)	成功排除
数据域错误	A <u>B</u> C	1 2 3 1	成功排除
校验错误	A B <u>C</u>	1 2 3 4 1	成功排除

2) 算法的效率:通过算法测试,可以发现基于有限状态机的数据帧同步算法的运行效率优于传统逐次比较的帧同步算法^[14],如表 4 所示。

表 4 数据帧同步算法效率对比

算法	每 100 帧数据平均耗时/ms
逐次比较的帧同步算法	2.18
本文算法	1.96

综上所述,本文基于数据帧格式定义和数据帧同

步算法研究了 ROV 水面监控软件的实时性策略。经测试,在单个通信周期 30 ms 内,上行/下行数据帧的传输耗时各为 5 ms,数据处理时间耗时 4.6 ms,三者总耗时大约为 15 ms,如表 5 所示。这意味着本文的实时性策略可以满足 ROV 水面软件的实时性要求。

表 5 单帧数据处理过程耗时

单帧数据处理过程	平均处理时间/ms
上行数据帧传输	5
数据帧处理	4.6
下行数据帧传输	5
运动控制(预留)	剩余时间 > 10

2.2 虚拟仪表开发

虚拟仪表的设计开发是提高 ROV 水面监控软件人机交互性的重要手段。类似飞行器虚拟仪表^[15],ROV 水面监控软件的虚拟仪表主要包括虚拟罗盘、姿态仪表、位置仪表、推进器仪表四大类。

本文采用 Qt 自定义控件开发技术,完成 ROV 虚拟仪表的设计和开发。开发过程主要分为三个主要步骤:1) 虚拟仪表类的接口设计;2) 虚拟仪表的绘制;3) 虚拟仪表类的封装使用。

1) 虚拟仪表类的接口设计:在设计开发仪表类时,需要根据每个仪表的具体功能需求,设置相应的数值输入接口和状态切换接口,以完成 ROV 主要状态数据的可视化显示,包括 ROV 方向、姿态、位置、推进器状态数据,以提高软件的人机交互性。ROV 水面监控软件虚拟仪表接口设计如表 6 所示。

表 6 虚拟仪表接口设计

类型	接口	作用
虚拟罗盘	数值输入接口	输入 ROV 实时的方向角
姿态仪表	数值输入接口	输入 ROV 实时的姿态角
位置仪表	数值输入接口	输入 ROV 实时的位置数据
	模式切换接口	深度/高度模式切换
推进器仪表	数值输入接口	输入 ROV 实时的推进器状态数据

2) 虚拟仪表的绘制:利用 Qt 绘图系统对不同的虚拟仪表进行绘制,Qt 整个绘图系统基于 QPainter、QPaintDevice 和 QPaintEngine 三个类。三者的关系和主要作用如图 6 所示。



图 6 Qt 绘图系统

由于 QPainter 在 Qt 内部实现,一般情况下不需要关注。因此在实际绘制仪表时,只需关注如何创建绘图空间以及如何进行具体的仪表绘制。绘图空间可以通过实例化 QPaintDevice 的子类 QWidget 进行创建。而仪表的具体绘制则通过 QPainter 类进行操作。QPainter 类提供了高度优化的图形绘制接口,以支持各种复杂界面图形的绘制。

由于 QWidget 中的 paintEvent 事件处理器可以在子类中被重写来接收绘图事件,因此可以通过 QPainter 在绘图事件中重绘 QWidget 的类对象。此时,Qt 事件循环会自动处理绘图事件,并根据重写的内容做出响应,从而在指定区域完成 ROV 虚拟仪表的绘制。

3) 虚拟仪表类的封装使用:通过 Qt 自定义控件开发技术,将开发好的 ROV 仪表类进行编译,将生成的链接库封装进 Qt Creator 和 Qt Designer 的控件库中。这样就可以在 ROV 水面监控软件开发的 Qt 项目中完成相关文件的配置和调用,利用 Qt 界面编程特性,对 ROV 虚拟仪表进行拖拽式用户界面程序开发,提高虚拟仪表的可复用性和易用性。

图 7 显示了 ROV 虚拟罗盘,用于显示 ROV 当前的方向和目标角度。在默认情况下,目标角度显示为问号。只有当 ROV 自动定向功能打开,才会显示具体的角度数值,同时在虚拟罗盘的外圈显示出目标角度的指示小箭头。

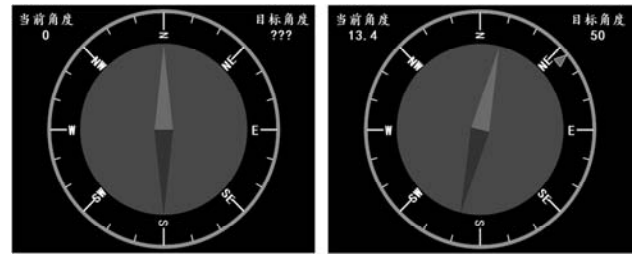


图 7 虚拟罗盘

图 8 显示了 ROV 姿态仪表,用于显示 ROV 当前的横倾角和俯仰角。

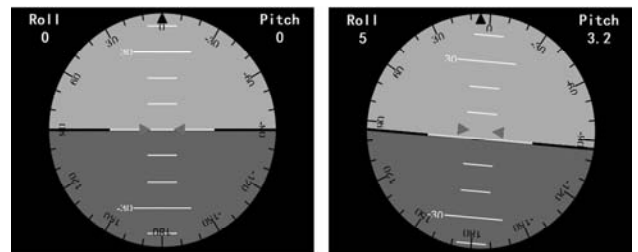


图 8 姿态仪表

图 9 显示了 ROV 位置仪表,用于显示 ROV 当前的水下位置,根据所采用的位置传感器的不同,其设有两种模式:深度模式(深度计)和高度模式(高度计)。

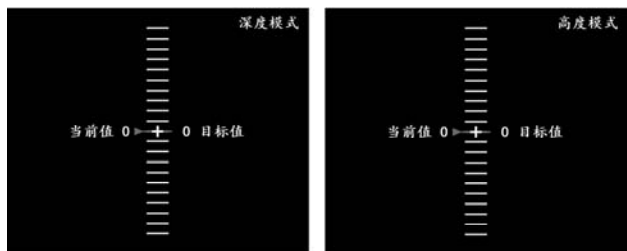


图 9 位置仪表

图 10 显示了 ROV 的推进状态,共包含四个仪表。从左往右、从上往下,依次为水平推进器仪表、垂直推进器仪表、水平合成推力仪表、垂直合成推力仪表。推进器仪表中的指示线反映了当前的推力或者合成推力指令数值,数值范围为 0% ~ 100%。当有数值输入时,对应的推进器仪表的数值将发生变化,同时推力指示线也会移动。

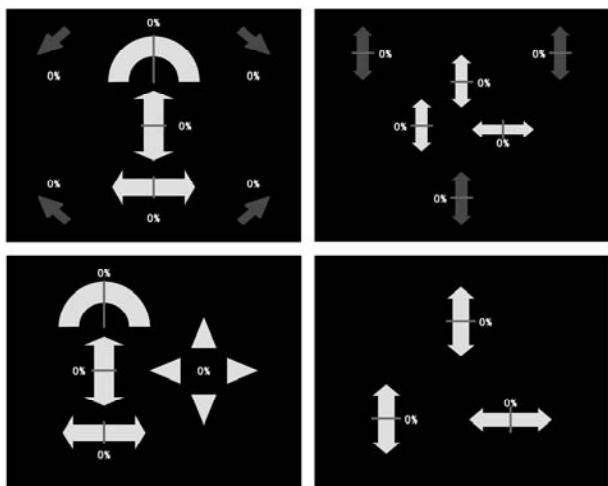


图 10 推进器仪表

3 软件实现

本文基于海龙系列 ROV 平台,采用 Qt 开发框架设计开发一套水面监控软件。以下将基于软件的主要界面,对软件的功能和性能进行说明和展示。

3.1 主界面

主界面体现了 ROV 水面监控软件的实时通信、数据处理、状态监视、警报显示等功能。在实际作业时,水面监控软件与水下 ROV 进行实时通信,将 ROV 内部板卡所采集到的原始数据进行处理和转换,得到所需要的数据,使得用户能够获取足够的 ROV 水下状态信息,包括 ROV 的方向、位置、姿态、设备状态、传感器数据、运动控制状态、警报等信息。这些信息分别显示在软件主界面的对应分区中,如图 11 所示。在主界面中央区域,绘制有 ROV 的方向、姿态、位置和推

进器仪表,这些虚拟仪表可以实时地将对应的 ROV 状态信息进行可视化显示,有效地提高了软件的人机交互性。



图 11 ROV 水面监控软件主界面

3.2 控制界面

ROV 水面监控软件提供了虚拟化的控制界面,实现了软件的设备控制功能。操作员通过控制界面输入操作指令,经由信息传输系统传给水下 ROV,以实现对 ROV 的相关控制,主要包括 ROV 摄像系统、灯光系统、阀门系统、云台控制系统、姿态控制系统、电源系统的控制。如图 12 所示。

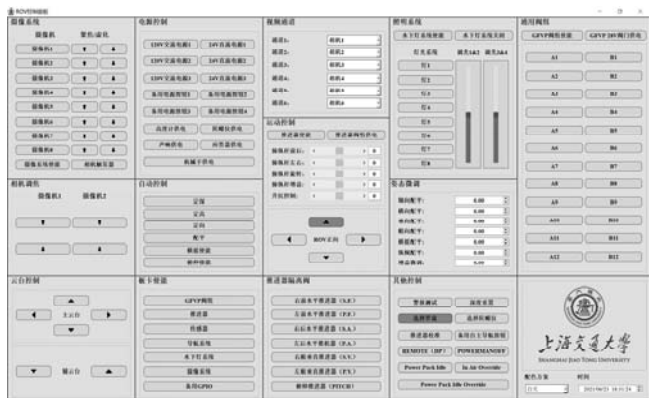


图 12 ROV 水面监控软件控制界面

3.3 子界面

软件的子界面可通过位于主界面顶部的界面控制按钮打开。子界面包括控制界面、串行数据界面、数据诊断界面、变量设置界面和软件说明界面。本文通过数据诊断界面展示软件子界面的设计开发情况。

数据诊断界面显示了 ROV 水下系统搭载的 11 块板卡所对应的状态信息。当软件出现警报提示时,操作员可以通过该界面提供的信息进行详细的警报分析。系统的每块板卡对应一个界面分页,同时该界面还包含所有板卡的接收数据和发送数据分页,因此共设有 13 个分页。图 13 显示了板卡接收数据分页,描述了所有板卡当前状态的原始数据。

板卡描述	PCB ID	SENSENET接收的数据
功率放大器 1	4433_00	00 00 01 00 00 00 00 00 00 00 00 00 00 00 77 00
功率放大器 2	4432_01	00 00 00 00 00 00 00 00 00 00 00 00 00 00 78 00
信号放大器 1	4423_08	02 09 7f 64 03 00 00 00 00 00 00 00 00 00 71 00
信号放大器 2	4422_09	00 00 00 00 00 00 00 00 00 00 00 00 00 00 78 00
电磁阀	4433_10	00 00 00 00 80 80 81 80 00 00 00 00 00 00 02 01
伺服电机 1A	4441_18	3c 2f 03 00 04 00 2d 09 0e 00 10 00 04 00 1a 00
伺服电机 1B	4441_19	00 00 00 00 00 00 00 00 00 00 00 00 00 00 0f 22 95 95
传感器 1A	4451_28	04 00 04 00 04 00 2f 2f 1a 00 08 00 1c 00 20 00
传感器 1B	4451_29	00 00 00 00 00 00 00 00 39 29 04 00 00 95 96 00 65
报警系统	4461_20	57 00 7c 48 03 00 5c 13 80 00 00 00 00 00 04 00
导航系统	4471_22	00 05 34 c2 29 2f 04 00 4e 4e 7f 4e 21 65 79 77
摄像系统	4481_24	00 00 20 20 02 00 02 04 00 00 00 22 73 79 00 00
灯光系统	4501_28	00 00 00 00 59 01 00 00 00 00 00 00 00 00 12 00

图 13 数据诊断界面

4 结 语

本文基于海龙系列 ROV 平台,利用 Qt 开发框架,设计实现一套水面监控软件。针对软件的实时性要求,详细研究了基于上行数据帧格式定义和数据帧同步算法的软件实时性策略。同时针对传统水面监控软件可视化程度低、人机交互性能差的问题,利用 Qt 自定义控件开发技术设计实现虚拟罗盘、姿态仪表、位置仪表和推进器仪表等 ROV 虚拟仪表。

经测试,该 ROV 水面监控软件可在 Windows 操作系统下稳定运行,各项功能测试正常,具备实时通信、数据处理、状态监视、设备控制、警报显示等功能,能够有效地对水下 ROV 进行在线监控,具备良好的实时性和友好的人机交互性能。值得注意的是,软件的设计开发过程不仅仅局限于潜水器的水面监控软件,对于其他领域相似的上位机,比如无人机地面监控站软件,也具备一定的参考价值。

参 考 文 献

- [1] 黄明泉,徐景平,施林炜. ROV 在海洋油气田开发中的应用及展望[J]. 海洋地质前沿,2021,37(2):77-84.
- [2] 赵羿羽,曾晓光,金伟晨. 海洋科考装备体系构建及发展方向研究[J]. 舰船科学技术,2019,41(19):1-6.
- [3] 连琏,魏照宇,陶军,等. 无人遥控潜水器发展现状与展望[J]. 海洋工程装备与技术,2018,5(4):223-231.
- [4] 曹俊,胡震,刘涛,等. 深海潜水器装备体系现状及发展分析[J]. 中国造船,2020,61(1):204-218.
- [5] 钟宏伟. 国外无人水下航行器装备与技术现状及展望[J]. 水下无人系统学报,2017,25(4):215-225.
- [6] 杜志元,杨磊,陈云赛,等. 我国与美国潜水器的发展和对比[J]. 海洋开发与管理,2019,36(10):55-60.
- [7] 沈克,严允,晏红文. 我国深海作业级 ROV 技术现状及发展展望[J]. 控制与信息技术,2020,4(3):1-7.
- [8] 刘畅. 腹部作业型 ROV 定深控制及水面监控系统设计[D]. 武汉:华中科技大学,2017.
- [9] 任峰,张莹,张丽婷,等. “海龙 III”号 ROV 系统深海试验与应用研究[J]. 海洋技术学报,2019,38(2):30-35.

- [10] 温贝托·塞万提斯,里克·卡斯曼. 软件架构设计:实用方法及实践[M]. 刘旭斌,陈瑶,邵元英,等译. 北京:机械工业出版社,2017.
- [11] 王中华,葛彤,朱继懋. ROV 动力定位系统控制时序与逻辑设计[J]. 海洋工程,2006,4(2):61-66.
- [12] 陈孟春,冯建文. 基于有限状态机的高速串口通信收发器的 FPGA 设计[J]. 计算机应用与软件,2017,34(12):178-183.
- [13] 崔彦坤,马萌. CRC 校验算法的设计与实现[J]. 计算机与网络,2019,45(1):62-64.
- [14] 周奋,王婷. 嵌入式系统中串口通信帧的同步方法[J]. 单片机与嵌入式系统应用,2006(10):73-75.
- [15] 李帅,范项媛. 基于 Qt 的无人机地面站软件系统的设计[J]. 雷达科学与技术,2017,15(4):410-414,420.

(上接第 38 页)

- [10] Zhang Y B, Cui M, Zheng L J, et al. Research on electronic medical record access control based on blockchain[J]. International Journal of Distributed Sensor Networks, 2019, 15(11):450-456.
- [11] 田国华,胡云瀚,陈晓峰. 区块链系统攻击与防御技术研究进展[J]. 软件学报,2021,32(5):1495-1525.
- [12] 张乐君,刘智栋,谢国,等. 基于集成信用度评估智能合约的安全数据共享模型[J]. 自动化学报,2021,47(3):594-608.
- [13] Daraghmi E Y, Daraghmi Y A, Yuan S M. MedChain: A design of blockchain-based system for medical records access and permissions management[J]. IEEE Access, 2019, 7: 164595-164613.
- [14] Jakobsson M, Juels A. Proofs of work and bread pudding protocols[J]. Secure Information Networks, 2008, 10(5): 258-272.
- [15] Kiayias A, Russell A, David B, et al. Ouroboros: A provably secure proof-of-stake blockchain protocol[J]. Advances in Cryptology, 2017, 45(12):357-388.
- [16] 王同贺,华昊辰,曹军威. 共识边缘计算及其在能源互联网中的应用[J]. 电力建设,2021,42(2):116-125.
- [17] Baig M I, Shuib L, Yadegaridehkordi E. State of the art and research challenges[J]. Information Processing & Management, 2019, 56(6):76-81.
- [18] 芦效峰,付淞兵. 属性基加密和区块链结合的可信数据访问控制方案[J]. 信息安全,2021,21(3):7-14.
- [19] Ren W, Hu J, Zhu T Q, et al. A flexible method to defend against computationally resourceful miners in blockchain proof of work[J]. Information Sciences, 2020, 507:161-171.
- [20] Kato K, Takefusa A, Nakada H, et al. A study of a scalable distributed stream processing infrastructure using ray and Apache Kafka[C]//IEEE International Conference on Big Data, 2018:5351-5353.
- [21] 张志威,王国仁,徐建良,等. 区块链的数据管理技术综述[J]. 软件学报,2020,31(9):2903-2925.