

HG3000 软件平台的驱动开发手册

编写人：张爱华

时间：2015 年 4 月 21 日

1 适用范围

本手册适用于 HG2002 监控平台和 HG3000 监控平台的通信驱动开发。

2 通信驱动的作用

整个监控系统由上位机监控平台、通信链路和现场终端设备三部分组成。上位机监控平台负责采集分析系统数据，现场终端设备负责借助传感器收集装置范围内的现场数据，但是现场终端设备是各个独立无交互的设备，数据分散，需要通过通信链路将数据汇集到监控平台上，进行处理后，直观地展示给用户。

因为现场复杂，各个设备来自于不同的厂家，每个厂家采用的通信协议标准可能都不相同，比如 modbus、cdt、DNP3.0 等。为了将这些现场终端的设备采集到监控平台，需要按照约定的通信协议，编写对应的通信驱动，实现数据的交互。

3 开发环境的说明

现有的开发环境为 VC6.0，开发者需要进行安装，安装包见附件。

4 常用的通信驱动包的介绍

一个标准的通信驱动包由以下几个部分组成：

(1) 驱动名称.exe: exe 文件为驱动的主执行文件，这一部分处理用户界面，显示通信过程中的收发数据，并且调用目标协议的 dll 动态链接库，来实现通信过程；

(2) 驱动名称.dll: dll 文件为协议动态链接库，这一部分是整个通信驱动的核心数据处理部分，这一部分将在后文中详细介绍；

(3) 驱动名称.ini: ini 文件为驱动的配置文​​件，在这一部分，现场运行人员通过这一部分的自定义来实现现场通信的不同需求，这一部分包括常用的通信链路的设置、采集巡回方式的设置等。

(4) readme.txt: 这个文档为开发者提供给现场调试人员的帮助文档，通过这一文档能够认识如何去配置和调试驱动。

5 常用的通信物理链路介绍

在现场的实际应用中，我们通常用到的是两种物理链路：232/422/485 总线、以太网。

232/422/485 总线：采用我们常说的串口通信方式，通信的双方需要按照约定波特率、数据位、停止位和奇偶校验位来进行数据传输和解析；

以太网：以太网要求通信双方都有网口支持，能够进行局域网组网。一般采用标准的 TCP/IP 协议来进行通信。

6 驱动的编写步骤

6.1 资料的收集

首先需要跟要通信的终端装置的厂家协商，弄清楚以下内容：

(1) 通信的链路是什么？

a 232/422/485

b 以太网

(2) 如果是 232/422/485 方式，确定到底是其中的哪一种方式？

(3) 如果是 232/422/485 方式，确定支持的串口通信参数是什么？

a 波特率 b 数据位 c 停止位 d 奇偶校验位

(4) 如果是以太网方式，确定使用是 TCP 还是 UDP 方式？

如果是 TCP，确定终端装置是 TCP Server 端，还是 TCP Client 端，或者都支持？

确定终端装置使用的端口号是多少？

(5) 跟终端装置的厂家沟通，要求对方提供完整版的通信协议说明，如果没有，请对方提供测试程序或者协议 dll 亦可。

6.2 协议的理解

在拿到终端厂家提供的通信协议之后，需要仔细阅读，了解其中的协议内容，主要内容是通信物理链路，通信参数，读写收发规则，报文格式等。

我公司的 HG3000 保护装置的通信规约和常用的 MODBUS 通信规约可作为开发者选读理解材料。详见附件中的《modbus 通讯协议 中文版.pdf》和《hg3000 保护通信规约.doc》

6.3 实时数据库与驱动层的接口定义理解

在通过驱动采集到数据以后，经过处理以后，需要将这些数据按照约定存放到实时数据库理解的接口中去，这样才能完成与 HG3000 的交互，对于实时数据库与驱动层的接口的详细说明，详见附件中的《实时数据库与驱动程序开发接口-解析.doc》

6.4 协议 dll 的结构说明

exe 文件使用公用进程，只需要更换图标即可，这里就不做说明。

协议 dll 的编写是整个驱动程序开发的关键步骤，其中的逻辑流程可参考附件中的《驱动开发的培训.ppt》。

其中关键类的说明：

QRamDrive 类：该类定义了共享内存的接口，包括上行结构、下行结构的读取和设置，驱动程序是通过这一接口与实时数据库进行数据交互；

CSerial 类：该类定义了串口这一通信结构，包括串口的设置、读取和发送，通过这一接口类，驱动完成与终端的串口数据通信；

原创力文档

max.book118.com

预览与源文档一致 下载高清无水印

CTCP 类：该类定义了使用 TCP/IP 通信时，包括 TCP/IP 客户端的 IP 地址、端口号等参数的设置、读取和发送，这是一个使用 TCP/IP 的以太网接口类；

CUDP 类：该类定义了使用 TCP/IP 通信时，包括 UDP 客户端的 IP 地址、端口号等参数的设置、读取和发送，这是一个使用 UDP 的以太网接口类；

CQueryItem 类：该类实现对循环召唤命令的管理，包括循环召唤命令的获取和通过串口读取的循环回复数据的处理；

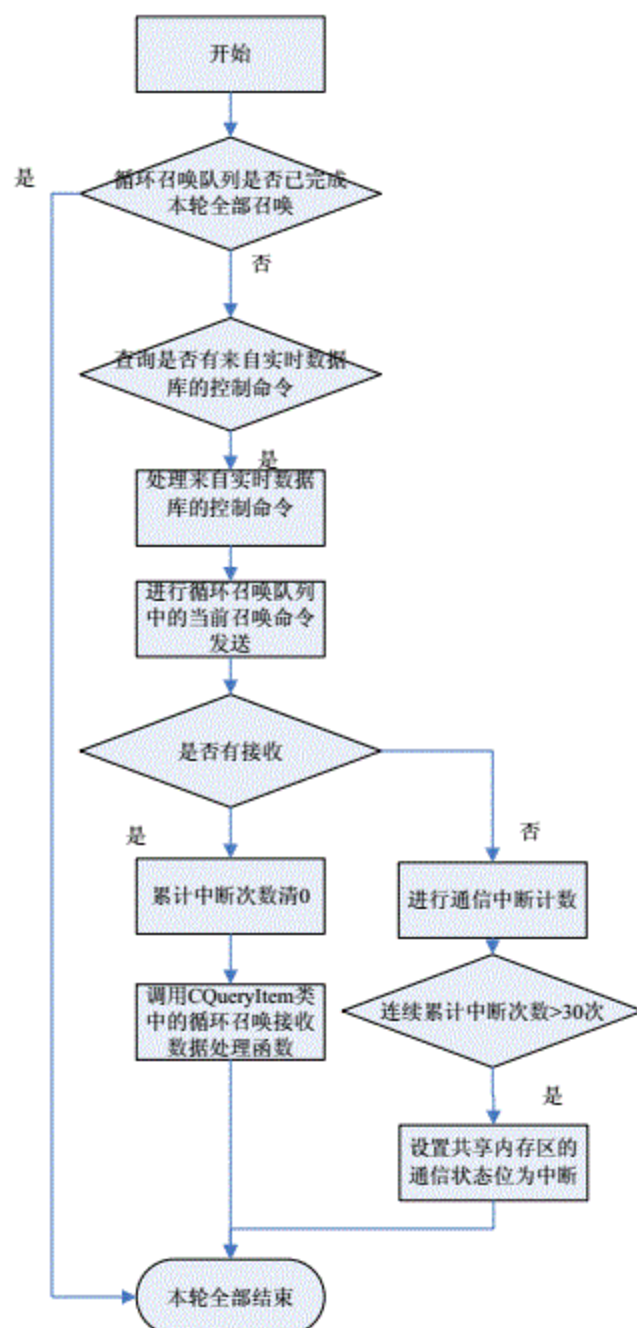
CCom 类：协议驱动的框架类，实现对 QRamDrive 对象和通信接口类的调用，定义通信驱动的实现框架；

CModbus 类：协议的实现类，继承于 CCom 类，CModbus 是指 Modbus 协议类，当然如果使用的 103 通信，可以命名为 C103 都可以。

在这里主要介绍协议类的实现和 CQueryItem 类中串口读取到的回复数据的处理的编码：

协议类一般为外界提供了一个反应通信流程的方法可供调用，比如 Run 或者其他，在这个函数中定义了整个通信流程如何有序地执行，以 Modbus 为例：

在 void CModbus::Run()这个方法中定义通信流程如下：



6.5 协议 dll 的程序编码

以 Modbus 协议为例, 通过这个流程图, 我们可以看到其中有两个数据发送的部分, 一是控制命令的下发, 二是循环召唤命令的下发; 同时有两个数据接收的部分, 一是控制命令回复数据的处理, 二是循环召唤命令回复数据的处理。那么在协议类的整体框架不变的情况下, 可以重点了解这四个部分的处理, 因为每个终端设备的驱动协议不同, 导致数据的发送和接收处理都会有不同之处, 但是整体的框架却是基本一致。

那么我们主要需要处理的函数就是 CModbus::

ExecCmd、CModbus::Run、CQueryItem::SetValue 和 CQueryItem::SetValue 这四个主要的函数。