

第7章 使用 TCP 协议进行网络通信

7.1 TCP 基础知识

书中 7.1.5~7.1.7 节介绍的内容已经无法使用，读者可以跳过。但是 7.1.1 节~7.1.4 节仍有实践价值，读者务必阅读。

7.2 实践案例：简易气象站程序 V2.0 的实现

这一节使用 TCP 网关替换了 OneNET 平台。建议读者直接阅读本节内容，并与书上的 7.2 节加以对比。

在这一部分中，将会为 V1.0 版的简易气象站程序增加 TCP 通信的功能，将测量结果通过 TCP 协议上传到 TCP 服务器上（参见示例代码\ch7New\ch7-3SimpleWeatherStationV2.0\和示例代码\ch7New\ch7-3TCP 网关程序\TCPGateway.exe）。

1. 建立 TCP 连接并登录服务器

首先为主窗口类增加 QTcpSocket 类型的成员变量指针：

```
private:
    QTcpSocket *m_TCPSocket;
```

然后在主窗口类的构造函数中为该指针分配内存：

```
m_TCPSocket = new QTcpSocket();
```

由于 TCP 支持长连接，因此可以在开启 TCP 通信时建立连接并发送登录报文，在关闭 TCP 通信功能时关闭连接。仿照前文的方法，为 TCP 通信开关控件 imageSwitchTCP 注册事件过滤器：

```
ui->imageSwitchTCP->installEventFilter(this);
```

然后为 TCP 通信开关控件 imageSwitchTCP 编写鼠标单击事件处理程序：

```
if (obj == ui->imageSwitchTCP)
{
    if (event->type() == QEvent::MouseButtonPress)
    {
        QMouseEvent *event2 = static_cast<QMouseEvent *>(event);
        if (event2->button() == Qt::LeftButton)
        {
            if (!ui->imageSwitchTCP->getChecked())
            {
                m_TCPSocket->connectToHost("127.0.0.1", 1811);
                connect(this, SIGNAL(signal_newDataArrived()), this,
                    SLOT(slot_TCPSendToOneNET()));
            }
        }
    }
}
```

```

        QString AuthInfo =
QString("%1#%2#%3*").arg(ui->lineEditProductID->text()).arg(ui->lineEditAuthCode->text()).arg(ui->lineEditScriptName->text()); //组装登录报文
        m_TCPSocket->write(AuthInfo.toUtf8());

        QEventLoop eventLoop;
        connect(m_TCPSocket, SIGNAL(readyRead()), &eventLoop,
SLOT(quit()));

        QTimer::singleShot(2000, &eventLoop, SLOT(quit()));
        eventLoop.exec();
        disconnect(m_TCPSocket, SIGNAL(readyRead()), &eventLoop,
SLOT(quit()));

        QByteArray buffer = m_TCPSocket->readAll();
        printLog("服务器返回", buffer);
    }
    else
    {
        m_TCPSocket->close();
        disconnect(this, SIGNAL(signal_newDataArrived()), this,
SLOT(slot_TCPSendToOneNET()));
    }
}
}
}
}

```

代码中的 127.0.0.1 是服务器 IP，1811 是服务器的端口号。在本章中，气象站程序与 TCP 网关程序进行通信。假设气象站程序与 TCP 网关程序都运行在一台电脑上，服务器 IP 可以使用 127.0.0.1。如果二者运行在不同电脑上，服务器 IP 需要填写网关程序所在电脑的 IP。

2. HEX 数据的生成

使用 TCP 协议向服务器发送数据时，要以十六进制（HEX）的形式传输，就像第 2 章中 GY-39 模块发送的十六进制数据一样（见 2.2.2 节）。要使用这种格式发送数据，需要先定义数据的格式。在本例中，按照图 7-11 的格式将气象站测量结果转换为 HEX 数据。如果数据位数不足，则在数据左侧补零。

编号	1	2	3	4	5	6	7	8	9
含义	湿度 如 23%RH 即发送 23			温度 如 -13.61℃ 即发送 -13.61					
编号	10	11	12	13	14	15	16	17	18
含义	海拔 如 -160m 即发送 -160			气压 如 101.325kPa 即发送 101.325					
编号	19	20	21	22	23	24	25	26	27
含义	气压		照度 如 123400Lux 即发送 123400						风速
编号	28	29	30	31	32	33	-	-	-
含义	风速			风向			-	-	-

	如 12.3m/s 即发送 12.3	如 190° 即发送 190			
--	--------------------	----------------	--	--	--

图 7-1 气象站程序在进行 TCP 通信时的 HEX 数据格式

由于 GY-39 模块和 PR-3000 模块均有自己的类,所以在这两个类中分别生成各自的 HEX 数据,最后将两段 HEX 数据拼接在一起。在本例中,用于生成 HEX 数据的函数统一命名为 dataToHex()。ClassGY39 类 dataToHex()函数的代码为:

```
QByteArray ClassGY39::dataToHex()
{
    QByteArray qbaHexData;
    qbaHexData.append(QString("%1").arg(getHumidity(), 3));
    qbaHexData.append(QString("%1").arg(getTemperature(), 6));
    qbaHexData.append(QString("%1").arg(getAltitude(), 4));
    qbaHexData.append(QString("%1").arg(getPressure(), 7));
    qbaHexData.append(QString("%1").arg(getIllumination(), 6));
    return qbaHexData;
}
```

ClassPR3000 类 dataToHex()函数的代码为:

```
QByteArray ClassPR3000::dataToHex()
{
    QByteArray qbaHexData;
    qbaHexData.append(QString("%1").arg(getWindSpeed(), 4));
    qbaHexData.append(QString("%1").arg(getWindDirection(), 3));
    return qbaHexData;
}
```

```
QByteArray qbaDataToSend;
qbaDataToSend = m_GY39Device->dataToHex();
qbaDataToSend.append(m_PR3000Device->dataToHex());
```

3. 使用 TCP 发送数据

新建一个槽函数 slot_TCPSendToOneNET(), 负责发送 TCP 数据并接收服务器响应。虽然 GY-39 模块和 PR-3000 模块的 HEX 数据分别由不同的函数生成,但是在进行 TCP 通信时只要将两个函数生成的 HEX 数据拼合在一起即可。最终槽函数的代码为:

```
void MainWindow::slot_TCPSendToOneNET()
{
    QByteArray qbaDataToSend;
    qbaDataToSend = m_GY39Device->dataToHex();
    qbaDataToSend.append(m_PR3000Device->dataToHex()); //拼合 HEX 数据
    m_TCPSocket->write(qbaDataToSend);

    QEventLoop eventLoop;
    connect(m_TCPSocket, SIGNAL(readyRead()), &eventLoop, SLOT(quit()));
    QTimer::singleShot(1000, &eventLoop, SLOT(quit()));
    eventLoop.exec();
    disconnect(m_TCPSocket, SIGNAL(readyRead()), &eventLoop,
SLOT(quit()));

    QByteArray qbaResponse = m_TCPSocket->readAll();
}
```

```
printLog("服务器返回", qbaResponse);  
}
```

7.3 程序运行结果

本章主要完成了程序的 TCP 通信功能。这部分内容和程序界面的关联较少，因而 V2.0 版程序的界面和 V1.0 版程序的界面几乎完全相同。

1、运行 TCP 服务器（见示例代码\ch7New\ch7TCP 网关程序\TCPGateway.exe）。运行后根据实际情况选择监听地址。本例中，TCP 服务器和气象站程序运行在同一电脑上，因而 IP 地址选择 127.0.0.1。若运行在不同电脑上，则需要根据实际情况进行选择。然后点击“开始监听”按钮，如图 7-12 所示。

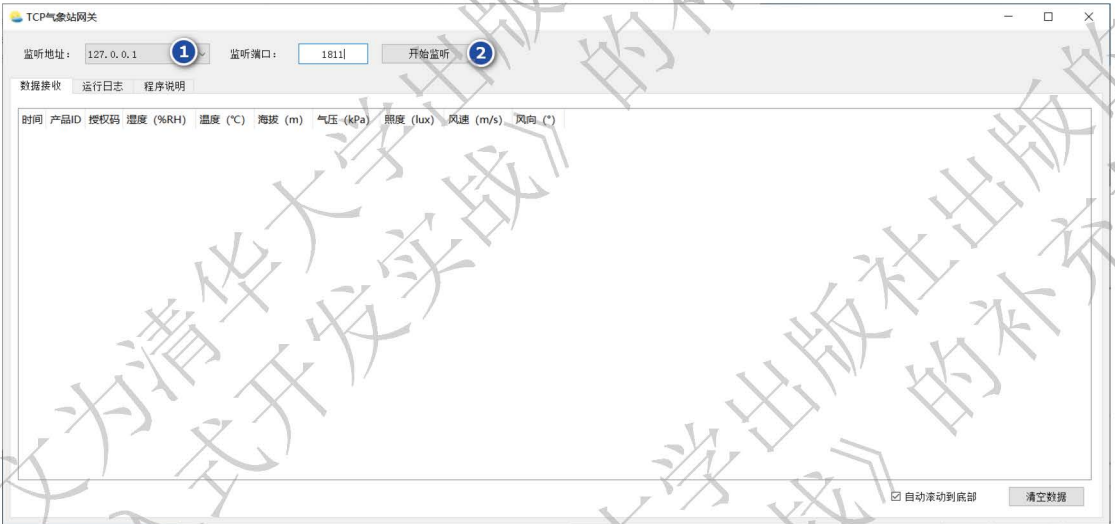


图 7-2 TCP 网关程序界面

2、在程序的 TCP 通信区域输入产品 ID、鉴权码、脚本名称（该信息原本用于 OneNet 平台鉴权使用，用 TCP 服务器替代后，此处可随意填写，亦可留空）。然后在程序中打开 TCP 通信开关。



图 7-3 程序中打开 TCP 通信开关

2、打开 TCP 通信开关，在程序中读取一组硬件测量数据，程序会自动将结果上传到 TCP 网关，如图 7-14 和图 7-15 所示。

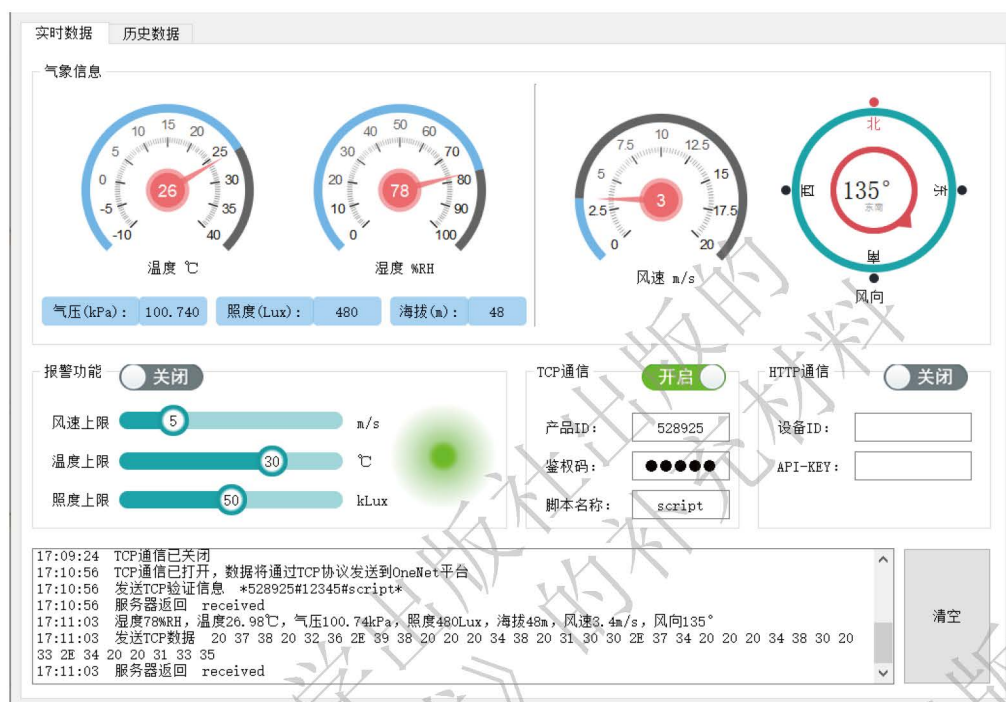


图 7-4 程序读取的气象数据



图 7-5 TCP 服务器接收并解析的气象数据