一個基於QT設計串口調試工具

嵌入式大雜燴 2022-07-24 22:07 發表於廣東

摘要：今天再次分享一個基於QT設計串口調試工具，源碼在Gitee，代碼簡單，可操作性強！

```
https://gitee.com/ErichMoonan/serial-master
```

1、概述

在開始軟件設計之前，我們來簡略地分析一下這樣一個小軟件其要包含的主要內容有哪些。我們認為軟件需要如下幾個方面的內容：

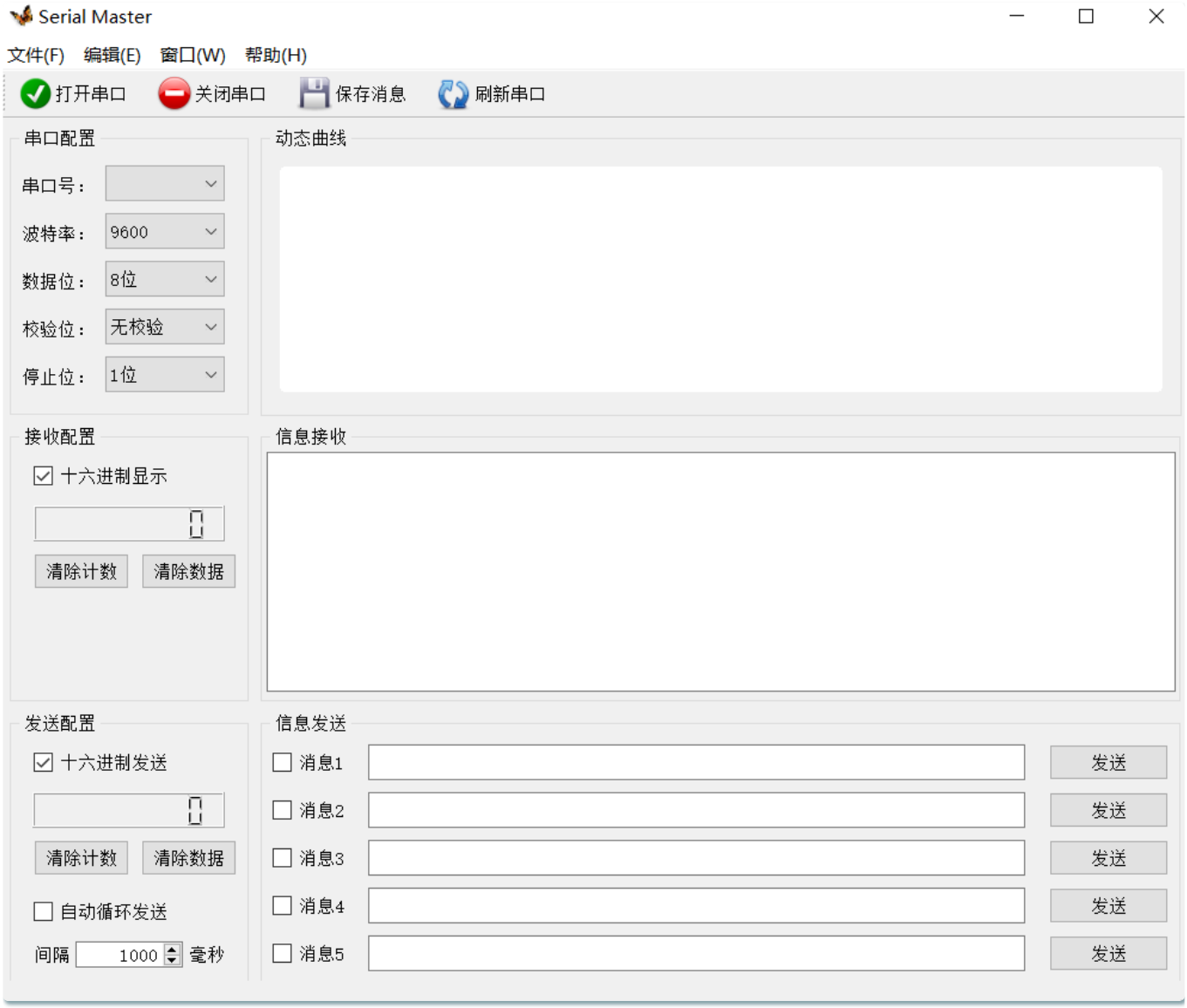
- 串口參數的配置，我們希望串口號能夠自動搜索，而相應的配置參數我們可以選擇。
- 發送數據的輸入，對於本軟件我們需要輸入相應的數據以實現命令及消息的發送，所以我們需要設計數據的輸入區域以及發送交互按鈕等。
- 接收信息的顯示，作為調試工具，我們肯定希望能夠一目了然地看到接收到目標設備發送過來的消息，所以我們需要一個顯示區域來對接收的區域進行顯示。
- 運行狀態的顯示， 我們希望對操作的狀態進行反饋以指示操作的動作是否執行，所以我們需要狀態欄來實現這一需求。
- 其它輔助功能， 還有如發送計數、接收計數、數據存儲等功能有時候也是需要的，所以我們一併考慮。

對於串口工具其實網上就有不少，我們之所以要自己實現這麼一個串口調試工具，主要的原因有兩點。一是，網上找到的相應工具某一個單獨的工具有時候不能完全滿足我們的需求，所以我們根據自己的需求設計這個工具能更好的滿足我們串口調試的需要。二是，通過這樣一個工具的實現，我們能夠加深對串口通訊相關知識的理解。

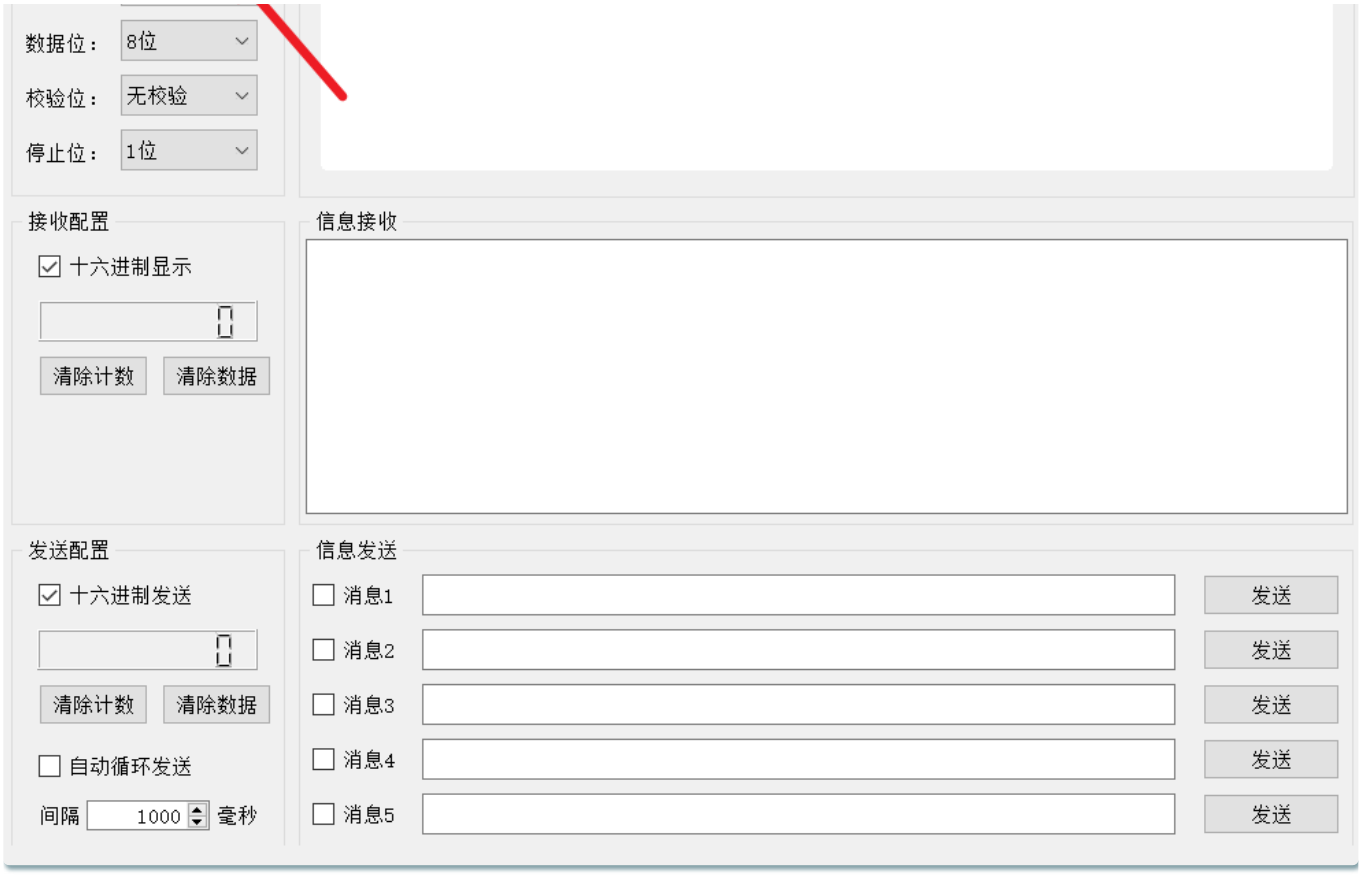
2、界面設計

根據上一節中分析的需求，我們先來設計軟件的界面。我們在QT中基於QMainWindow類生成一個操作界面，包括菜單欄、工具欄和狀態欄以滿足需求中對狀態顯示及操作命令的要求。

而在中間顯示區域，我們將其劃分為3行2列。在左邊的一列從上到下設置：串口配置操作區域、接收配置區域以及發送配置區域。在右側的一列從上到下設置：動態曲線顯示區域、信息接收顯示區域以及信息發送輸入區域。具體的界面設置如下圖所示：



完成如上圖的佈局後，我們可以選擇在屬性中配置空間的參數，也可以在代碼中添加相關的參數，本人習慣於在代碼中完成。完成整個佈局後我們先試著運行程序，正常運行則出現如下的界面：



上圖就是完成佈局後的運行界面，不過我們還沒有實現相應的編碼，所以目前還不能實現我們第一節中所提出來的功能。

3、編碼實現

接下來這一小節，我們將來編碼實現相應的功能。我們主要將功能分為參數設置與操作功能、數據的輸入與發送功能以及數據的接收與顯示功能三個部分來實現。

3.1、參數設置與操作功能

對於參數的配置除了串口號以外都可以直接使用ComboBox控件的相應函數添加。串口號這塊，我們希望搜索電腦安裝的串口並添加到控件中。具體的實現方式如下：

```
//搜索可用的串口，并添加到串口组合框
void MainWindow::SearchSerialPorts()
{
    ui->comboBoxPort->clear();

    foreach(const QSerialPortInfo &info,QSerialPortInfo::availablePorts())
    {
        ui->comboBoxPort->addItem(info.portName());
    }
}
```

配置好串口參數後，我們可以打開串口以建立連接。需要說明的是我們打開串口間離連接時，我們需要將該串口的數據接收與我們的數據接收和處理函數建立信號槽連接。具體實現如下：

```
//打开串口
void MainWindow::on_actionConnect_triggered()
{
    serialPort->setPortName(ui->comboBoxPort->currentText());

    if(serialPort->open(QIODevice::ReadWrite))              //打开串口成功
    {
        serialPort->setBaudRate(ui->comboBoxBaud->currentText().toInt());       //设置波特率

        switch(ui->comboBoxData->currentIndex())                    //设置数据位数
        {
            case 1:serialPort->setDataBits(QSerialPort::Data8);break;
            default: break;
        }

        switch(ui->comboBoxParity->currentIndex())                  //设置奇偶校验
        {
            case 0: serialPort->setParity(QSerialPort::NoParity);break;
            default: break;
        }

        switch(ui->comboBoxStop->currentIndex())                    //设置停止位
        {
            case 1: serialPort->setStopBits(QSerialPort::OneStop);break;
            case 2: serialPort->setStopBits(QSerialPort::TwoStop);break;
            default: break;
        }

        serialPort->setFlowControl(QSerialPort::NoFlowControl);     //设置流控制

        //连接槽函数
        QObject::connect(serialPort, &QSerialPort::readyRead, this, &MainWindow::ReadSerialDa

        // 设置控件可否使用
        ui->actionConnect->setEnabled(false);
        ui->actionClose->setEnabled(true);
        ui->actionRefresh->setEnabled(false);
    }
    else    //打开失败提示
    {
        QMessageBox::information(this,tr("错误"),tr("打开串口失败！"),QMessageBox::Ok);
```

```
    }
}
```

同樣的，我們除了要打開串口建立連接外，還需要關閉串口斷開連接，具體的代碼如下：

```
//关闭串口
void MainWindow::on_actionClose_triggered()
{
    serialPort->clear();
    serialPort->close();

    // 设置控件可否使用
    ui->actionConnect->setEnabled(true);
    ui->actionClose->setEnabled(false);
    ui->actionRefresh->setEnabled(true);
}
```

3.2、數據的輸入與發送功能

數據的輸入與發送，我們設計了5條命令，每條命令可以通過後面的按鈕手動發送，也可以自動循環發送。自動循環發送時，將對每條選中的命令以設定的時間間隔輪詢發送。

首先我們來看看定時周期發送的過程。我們定義了一個計時器，以我們設定的時間週期觸發發送命令，每次發送複選框被選中的命令一條，依次循環直到人為停止循環發送為止。具體的代碼如下：

```
//定时周期发送
void MainWindow::CycleSendData()
{
    QCheckBox* cbSend;

    while(true)
    {
        snIndex=snIndex>=6?1:snIndex;

        cbSend=ui->groupBoxMessage->findChild<QCheckBox*>(QString("checkBoxSendEnable%1").arg

        if(cbSend->isChecked())
        {
            WriteSerialData(snIndex);
            snIndex++;
            break;
        }
```

```
            snIndex++;
        }
    }
```

手動單次發送則判斷是哪一個按鈕觸發的動作則操作對應的數據輸入框，將其中的內容以指定的格式發送出去。具體的操作代碼如下：

```
//按钮触发发送
void MainWindow::SingleSendData()
{
    // 判断如果Sender是QPushButton就执行
    if (QPushButton* btn = dynamic_cast<QPushButton*>(sender()))
    {
        QString senderName;
        int sn=0;

        senderName = btn->objectName();
        sn = senderName.replace("pushButtonSend", "").toInt();

        if((0<sn) && (sn<6))
        {
            WriteSerialData(sn);
        }
    }
}
```

3.3、數據的接收與現實功能

在我們的設計中，數據的接收相對要簡單一些。當串口接收到數據後就會觸發我們的接收數據處理函數，並將以我們設定的格式顯示出來，具體的實現代碼如下：

```
//从串口接收数据
void MainWindow::ReadSerialData()
{
    QByteArray rxDatas;
    QString context;

    rxDatas=serialPort->readAll();

    if(!rxDatas.isNull())
    {
        if(ui->checkBoxRecieve->isChecked())    //十六进制显示
        {
             context = rxDatas.toHex(' ');
```

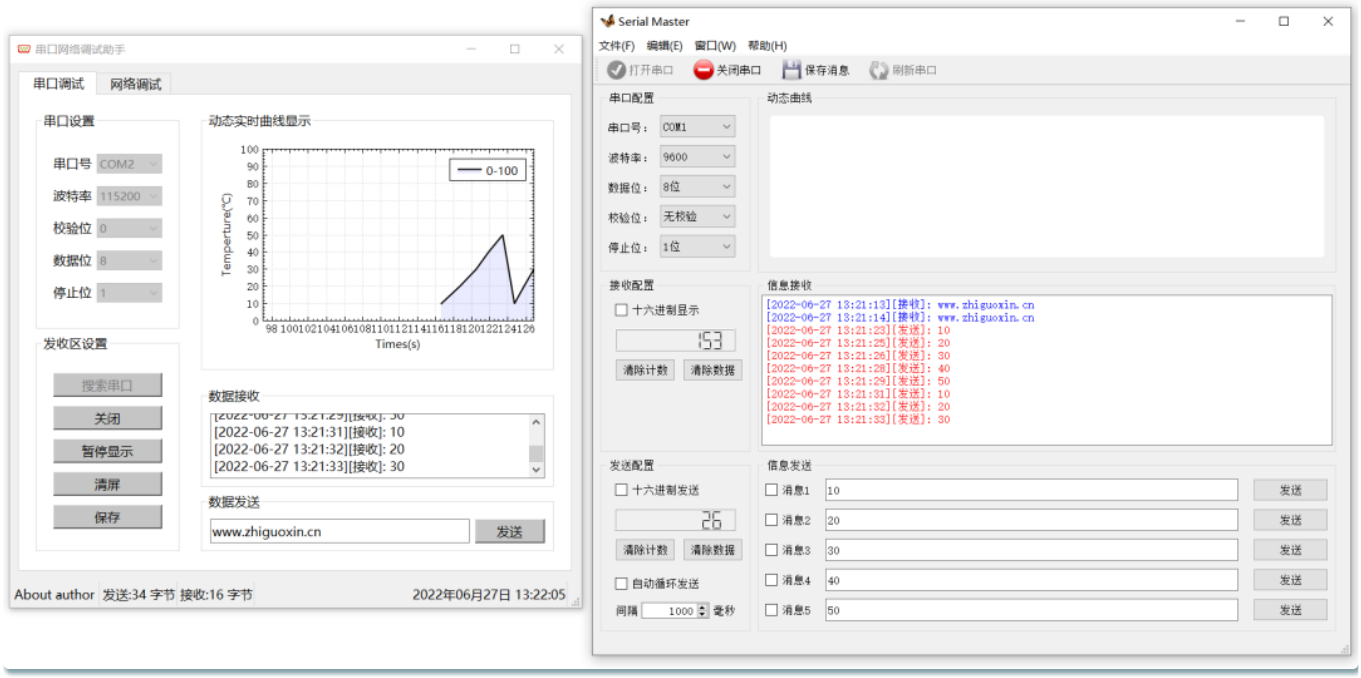
```
                context=context.toUpper();
            }
            else    //ASCII显示
            {
                context = rxDatas;
            }

            QString timeStrLine="["+QDateTime::currentDateTime().toString("yyyy-MM-dd hh:mm:ss")+
            context = timeStrLine+context+"\n\r";

            QString content = "<span style=\" color:blue;\">"+context+"</span>";
            ui->textBrowser->append(content);

            receivedBytes=receivedBytes+rxDatas.size();
            ui->lcdNumberRecieve->display(receivedBytes);

            ui->statusbar->showMessage(tr("成功读取%1字节数据").arg(rxDatas.size()));
        }

        rxDatas.clear();
    }
```

4、小結

完成了編碼調試後，我們來對開發的這一工具進行一些測試。首先我們安裝一個虛擬串口軟件用以虛擬我們用於測試的串口。如果有硬件接口最好，但是在我的電腦上沒有串口，所以我們使用虛擬串口來模擬一對串口。具體的配置如下圖所示：

我們使用另一個串口工具來實現與我們開發的這一工具實現通訊驗證。我們使用以前寫得一個串口工具來實現與這一工具的通訊。一個使用使用COM1，一個使用使用COM2。具體的配置如下圖所示：

注：使用虛擬串口波特率可以



來源：木南創智

往期推薦：

嵌入式並行多線程處理器，了解一下！

Linux大陸
Hello Linux
20篇原創內容

公眾號

嵌入式大雜燴
本公眾號專注於嵌入式技術，包括但不限於C/C++、嵌入式、物聯網、Linux等編程學...
289篇原創內容

公眾號

在公眾號聊天界面回复**1024**，可獲取嵌入式資源；回复 **m** ，可查看文章匯總。

喜歡此內容的人還喜歡

看完Vue源碼，總結了這個22個工具函數
大遷世界

C語言編寫的超好用的新一代SSH 終端- WindTerm
k8s技術圈

Vue2升級到Vue3到底是不是一個正確的選擇？(尤雨溪親自回复解讀)
前端瓶子君