ESP8266

AT+CWMODE=2

AT+CIFSR

AT+RST

AT+CIPMUX=1

AT+CIPSERVER=1,8080

AT+CIPSTART=0,”TCP”,”192.168.4.2”,8080

AT+CIPSEND=0,2

AT+CJWAP=”lwz”,”123456789”

QT

Mainwindow.h文件

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QSerialPort>

#include <QSerialPortInfo>

#include <QSqlDatabase>

#include <QTimer>

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~*MainWindow*();

private slots:

void on\_btnOpen\_clicked();

void ReadData();

void ReadData\_N();

void timeUpdate();

void on\_startButton\_clicked();

void on\_stopButton\_clicked();

void on\_btnOpen\_n\_clicked();

private:

Ui::MainWindow \*ui;

QSerialPort \*serial;

QSerialPort \*esp\_serial;

QSqlDatabase db;

QTimer \*timer;

QByteArray Combuf\_n;

QString wen,shi,hw;

int iwen,ishi;

};

#endif // MAINWINDOW\_H

Main.cpp

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

Mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <Qserialport>

#include <QSerialPortInfo>

#include <QSqlError>

#include <QMessageBox>

#include <QSqlDatabase>

#include <QSqlQuery>

#include <QDebug>

#include <QTimer>

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

db = QSqlDatabase::addDatabase("QMYSQL");

db.setHostName("localhost");

db.setDatabaseName("zigbee");

db.setUserName("root");

db.setPassword("");

if(!db.open())

{

QMessageBox::critical(0,QObject::tr("无法打开数据库"),

"无法创建数据库连接！",QMessageBox::Cancel);

return;

}

else {

qDebug()<<"suceess Connect!";

}

timer = new QTimer();

foreach (const QSerialPortInfo &info, QSerialPortInfo::availablePorts()) {

qDebug()<<"Name:"<<info.portName();

qDebug()<<"Description:"<<info.description();

qDebug()<<"Manufacturer:"<<info.manufacturer();

QSerialPort serial;

serial.setPort(info);

if(serial.*open*(QIODevice::ReadWrite))

{

ui->cmbPortName->addItem(info.portName());

ui->cmbPortName\_n->addItem(info.portName());

serial.*close*();

}

}

ui->cmbBaudRate->setCurrentIndex(0);

ui->cmbBaudRate\_n->setCurrentIndex(0);

}

MainWindow::~*MainWindow*()

{

delete ui;

}

void MainWindow::on\_btnOpen\_clicked()

{

if(ui->btnOpen->text() == tr("打开数据串口"))

{

serial = new QSerialPort;

//设置串口名

serial->setPortName(ui->cmbPortName->currentText());

//打卡串口

serial->*open*(QIODevice::ReadWrite);

//设置波特率

serial->setBaudRate(ui->cmbBaudRate->currentText().toInt());

//设置数据位数

switch (ui->cmbDataBits->currentIndex())

{

case 8:

serial->setDataBits(QSerialPort::Data8);

break;

default:

break;

}

//设置校验位

switch (ui->cmbParity->currentIndex())

{

case 0:

serial->setParity(QSerialPort::NoParity);

break;

default:

break;

}

//设置停止位

switch (ui->cmbStopBits->currentIndex())

{

case 1:

serial->setStopBits(QSerialPort::OneStop);

break;

case 2:

serial->setStopBits(QSerialPort::TwoStop);

default:

break;

}

//设置流控制

serial->setFlowControl(QSerialPort::NoFlowControl);

//关闭设置菜单使能

ui->cmbPortName->setEnabled(false);

ui->cmbBaudRate->setEnabled(false);

ui->cmbDataBits->setEnabled(false);

ui->cmbParity->setEnabled(false);

ui->cmbStopBits->setEnabled(false);

ui->btnOpen->setText(tr("关闭数据串口"));

//连接信号槽

QObject::connect(serial,&QSerialPort::readyRead,this,&MainWindow::ReadData);

}

else

{

//关闭串口

serial->clear();

serial->*close*();

serial->deleteLater();

//恢复设置使能

ui->cmbPortName->setEnabled(true);

ui->cmbBaudRate->setEnabled(true);

ui->cmbDataBits->setEnabled(true);

ui->cmbParity->setEnabled(true);

ui->cmbStopBits->setEnabled(true);

ui->btnOpen->setText(tr("打开数据串口"));

}

}

void MainWindow::on\_btnOpen\_n\_clicked()

{

if(ui->btnOpen\_n->text() == tr("打开网络串口"))

{

esp\_serial = new QSerialPort;

//设置串口名

esp\_serial->setPortName(ui->cmbPortName\_n->currentText());

//打卡串口

esp\_serial->*open*(QIODevice::ReadWrite);

//设置波特率

esp\_serial->setBaudRate(ui->cmbBaudRate\_n->currentText().toInt());

//设置数据位数

switch (ui->cmbDataBits\_n->currentIndex())

{

case 8:

esp\_serial->setDataBits(QSerialPort::Data8);

break;

default:

break;

}

//设置校验位

switch (ui->cmbParity\_n->currentIndex())

{

case 0:

esp\_serial->setParity(QSerialPort::NoParity);

break;

default:

break;

}

//设置停止位

switch (ui->cmbStopBits\_n->currentIndex())

{

case 1:

esp\_serial->setStopBits(QSerialPort::OneStop);

break;

case 2:

esp\_serial->setStopBits(QSerialPort::TwoStop);

default:

break;

}

//设置流控制

esp\_serial->setFlowControl(QSerialPort::NoFlowControl);

//关闭设置菜单使能

ui->cmbPortName\_n->setEnabled(false);

ui->cmbBaudRate\_n->setEnabled(false);

ui->cmbDataBits\_n->setEnabled(false);

ui->cmbParity\_n->setEnabled(false);

ui->cmbStopBits\_n->setEnabled(false);

ui->btnOpen\_n->setText(tr("关闭网络串口"));

//连接信号槽

QObject::connect(esp\_serial,&QSerialPort::readyRead,this,&MainWindow::ReadData\_N);

QObject::connect(timer,&QTimer::timeout,this,&MainWindow::timeUpdate);

}

else

{

//关闭串口

esp\_serial->clear();

esp\_serial->*close*();

esp\_serial->deleteLater();

//恢复设置使能

ui->cmbPortName\_n->setEnabled(true);

ui->cmbBaudRate\_n->setEnabled(true);

ui->cmbDataBits\_n->setEnabled(true);

ui->cmbParity\_n->setEnabled(true);

ui->cmbStopBits\_n->setEnabled(true);

ui->btnOpen\_n->setText(tr("打开网络串口"));

}

}

//数据串口读取算法

void MainWindow::ReadData()

{

QByteArray Combuf;

if(serial->*bytesAvailable*()>7)

{

Combuf = serial->readAll();

qDebug()<<Combuf;

}//因为QT接收串口数据会断层，故用bytesAvailable函数，作用是当接收数据大于5时才调用readall接收数据

if(!Combuf.isEmpty())

{

QByteArray allArray,jingArray,qianArray,ywArray;

allArray = Combuf;

if(allArray.contains("#") && allArray.contains("$"))

{

QString checkData;

QString checkData2,checkData3;

checkData = allArray.data();

checkData2 = checkData.mid(0,1);

checkData3 = checkData.mid(5,1);

if(checkData2 == "$" && checkData3 == "#")

//yw&hw

{

QString data;

bool hok;

data = allArray.data();

hw = data.mid(7,1);

int ihw = hw.toInt(&hok,10);

qDebug()<<ihw;

switch(ihw)

{

case 1:ui->hwLabel->setText("有人,已报警!");break;

case 0:ui->hwLabel->setText("无人");break;

}

//wenshi

jingArray = allArray.split('#').at(0);

qDebug()<<jingArray;

for(int i=1;i<jingArray.length();i++)

{

qianArray[i-1]=jingArray[i];

}

qDebug()<<qianArray;

//QByteArray转int

QString str\_data;

str\_data=qianArray.data();

wen = str\_data.mid(0,2);//温度

shi = str\_data.mid(2,2);//湿度

bool wenok,shiok;

iwen = wen.toInt(&wenok,10);

ishi = shi.toInt(&shiok,10);

ui->wdlabel->setText(wen);

ui->sdlabel->setText(shi);

qDebug()<<wen;

qDebug()<<shi;

QByteArray w\_buf,s\_buf;

w\_buf = wen.toLatin1();

s\_buf = shi.toLatin1();

if(ui->btnOpen\_n->text() == tr("关闭网络串口"))

{

esp\_serial->write("AT+CIPSTART=0,\"TCP\",\"192.168.4.2\",8080\r\n");

}

//数据库插入部分；

QSqlQuery query(db);

//使用占位符：使得能在sql语句用变量

query.prepare("insert into cgq(wd,sd,hw) values(:iwen,:ishi,:ihw)");

//数字表示变量在sql语句的位置，后面即对应变量，这个函数实现变量和占位符的绑定。

query.bindValue(0,iwen);

query.bindValue(1,ishi);

query.bindValue(2,ihw);

//这一句一定要写！

query.exec();

}

}

}

Combuf.clear();

}

void MainWindow::ReadData\_N()

{

timer->start(100);

Combuf\_n.append(esp\_serial->readAll());

}

//延时函数

void MainWindow::timeUpdate()

{

timer->stop();

qDebug()<<Combuf\_n;

if(Combuf\_n.length()!=0 && !Combuf\_n.contains("OK") && Combuf\_n.contains(":")){

QString code = Combuf\_n.data();

QStringList list = code.split(':');

QString str = list[1];

qDebug()<<list[1];

if(str.mid(0,5) == "start"){

on\_startButton\_clicked();

}

else if(str.mid(0,4) == "stop"){

on\_stopButton\_clicked();

}

}

if(Combuf\_n.contains("OK")){

QByteArray w\_buf,s\_buf;

w\_buf = wen.toLatin1();

s\_buf = shi.toLatin1();

esp\_serial->write("Tmperature:");

esp\_serial->write(w\_buf);

esp\_serial->write("Humidity:");

esp\_serial->write(s\_buf);

if(hw == "1"){

esp\_serial->write("有人");

}

else if(hw == "0"){

esp\_serial->write("无人");

}

}

if(Combuf\_n.contains("CONNECT")){

if(iwen > 10 && ishi > 10)

{

esp\_serial->write("AT+CIPSEND=0,30\r\n");

}

else if (iwen < 10 || ishi < 10)

{

esp\_serial->write("AT+CIPSEND=0,29\r\n");

}

else if (iwen < 10 && ishi< 10)

{

esp\_serial->write("AT+CIPSEND=0,28\r\n");

}

}

qDebug()<<Combuf\_n;

Combuf\_n.clear();

}

void MainWindow::on\_startButton\_clicked()

{

serial->write("start");

QMessageBox::information(this,"TIPS","开始检测",QMessageBox::Cancel);

}

void MainWindow::on\_stopButton\_clicked()

{

serial->write("stop");

QMessageBox::information(this,"TIPS","停止检测",QMessageBox::Cancel);

}

ZigBee

协调器消息处理函数

static void GenericApp\_MessageMSGCB( afIncomingMSGPacket\_t \*pkt )

{

//unsigned char bufferwenshi[4];

if(osal\_memcmp(uartbuf,"start",5))

{

//人体红外传感器检测

switch ( pkt->clusterId )

{

case GENERICAPP\_CLUSTERID:

HalUARTWrite(0,"$",1);

HalUARTWrite(0,&pkt->cmd.Data[0],2);

HalUARTWrite(0,&pkt->cmd.Data[2],2);

HalUARTWrite(0,"#",1);

#if defined( LCD\_SUPPORTED )

HalLcdWriteScreen( (char\*)pkt->cmd.Data, "rcvd" );

#elif defined( WIN32 )

WPRINTSTR( pkt->cmd.Data );

#endif

break;

case GENERICAPP\_CLUSTERID\_YW:

HalUARTWrite(0,pkt->cmd.Data,pkt->cmd.DataLength);

byte state;

if(DATA\_PIN\_HW ==1)

{

MicroWait(10000);

if(DATA\_PIN\_HW ==1)

{

state = 0x31; //有人进入

HalUARTWrite(0,"1",1);

HalLedBlink(HAL\_LED\_1,0,50,100);

HalLedBlink(HAL\_LED\_2,0,50,200);

}

}

else

{

state = 0x30;

HalUARTWrite(0,"0",1);

HalLedSet(HAL\_LED\_1,HAL\_LED\_MODE\_ON);

HalLedSet(HAL\_LED\_2,HAL\_LED\_MODE\_ON);

}

break;

}

}

}

协调器回调函数

static void rxCB(uint8 port, uint8 event)

{

unsigned char error[9]="ERROR CMD";

unsigned char stop[10]="STOP CHECK";

HalUARTRead(0,uartbuf,5);

if(osal\_memcmp(uartbuf,"start",5))

{

GenericApp\_MessageMSGCB;

}

else if(osal\_memcmp(uartbuf,"stop",4))

{

HalUARTWrite(0,stop,10);

}

else

{

HalUARTWrite(0,error,9);

}

}

终端 发送消息函数

static void GenericApp\_SendTheMessage( void )

{

DHT11();

uint8 wenshi[4];

byte state;

wenshi[0]=wendu\_shi+0x30;

wenshi[1]=wendu\_ge+0x30;

wenshi[2]=shidu\_shi+0x30;

wenshi[3]=shidu\_ge+0x30;

if(DATA\_PIN\_YW == 1)

{

state = 0x31;

}

else

{

MicroWait(10000);

if(DATA\_PIN\_YW == 0)

{

state = 0x30;

}

}

afAddrType\_t my\_DstAddr;

my\_DstAddr.addrMode = (afAddrMode\_t)Addr16Bit; //addr16bit表示单播

my\_DstAddr.endPoint = GENERICAPP\_ENDPOINT;

my\_DstAddr.addr.shortAddr = 0x0000; //表示发送对象为协调器

if ( AF\_DataRequest( &my\_DstAddr, &GenericApp\_epDesc,

GENERICAPP\_CLUSTERID,

osal\_strlen("1234")+1,

wenshi,

&GenericApp\_TransID,

AF\_DISCV\_ROUTE, AF\_DEFAULT\_RADIUS ) == afStatus\_SUCCESS )

{

HalLedBlink(HAL\_LED\_1,0,50,1300);// Successfully requested to be sent.

}

if(AF\_DataRequest(&my\_DstAddr, &GenericApp\_epDesc,

GENERICAPP\_CLUSTERID\_YW,

1,

&state,

&GenericApp\_TransID,

AF\_DISCV\_ROUTE, AF\_DEFAULT\_RADIUS ) == afStatus\_SUCCESS

)

{

HalLedBlink(HAL\_LED\_2,0,50,1300);

}

}

红外热释电传感器代码

if(DATA\_PIN\_HW ==1)

{

MicroWait(10000);

if(DATA\_PIN\_HW ==1)

{

state = 0x31; //有人进入

HalUARTWrite(0,"1",1);

HalLedBlink(HAL\_LED\_1,0,50,100);

HalLedBlink(HAL\_LED\_2,0,50,200);

}

}

else

{

state = 0x30;

HalUARTWrite(0,"0",1);

HalLedSet(HAL\_LED\_1,HAL\_LED\_MODE\_ON);

HalLedSet(HAL\_LED\_2,HAL\_LED\_MODE\_ON);

}

安卓APP

mainactivity.java

package com.example.meow.testapp;

import android.app.Activity;

import android.os.Bundle;

import android.os.Message;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.os.Handler;

import android.widget.Toast;

public class MainActivity extends Activity implements View.OnClickListener {

private TextView tv\_content,tv\_send\_text;

private Button bt\_send,stop\_btn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

InitView();

PhoneServer phoneServer = new PhoneServer();

phoneServer.setHandler(handler);

new Thread(phoneServer).start();

}

private void InitView(){

tv\_content = (TextView) findViewById(R.id.tv\_content);

tv\_send\_text = (TextView) findViewById(R.id.tv\_send\_text);

bt\_send = (Button) findViewById(R.id.bt\_send);

stop\_btn = (Button) findViewById(R.id.stop\_btn);

stop\_btn.setOnClickListener(this);

bt\_send.setOnClickListener(this);

}

@Override

public void onClick(View v){

switch (v.getId()){

case R.id.bt\_send:

String str = "start";

new S\_AsyncTask().execute(str);

tv\_send\_text.setText(str);

break;

case R.id.stop\_btn:

String str\_s = "stop";

new S\_AsyncTask().execute(str\_s);

tv\_send\_text.setText(str\_s);

}

}

Handler handler = new Handler(){

@Override

public void handleMessage(Message msg){

switch (msg.what){

case 1:

tv\_content.setText(""+msg.obj);

//Toast.makeText(MainActivity.this,"接收到的消息",Toast.LENGTH\_LONG).show();

}

}

};

}

PhoneServer.java

package com.example.meow.testapp;

import android.os.Message;

import android.os.Handler;

import java.io.DataInputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

public class PhoneServer implements Runnable {

private ServerSocket server;

private DataInputStream in;

private byte[] receice;

private Handler handler = new Handler();

public PhoneServer(){

}

public void setHandler(Handler handler){

this.handler = handler;

}

@Override

public void run(){

try {

server = new ServerSocket(8080);

while (true){

Socket client = server.accept();

in = new DataInputStream(client.getInputStream());

receice = new byte[50];

in.read(receice);

in.close();

Message message = new Message();

message.what = 1;

message.obj = new String(receice);

handler.sendMessage(message);

}

} catch (IOException e){

e.printStackTrace();

}

try {

server.close();

} catch (IOException e){

e.printStackTrace();

}

}

}

S\_AsyncTask

package com.example.meow.testapp;

import android.os.AsyncTask;

import java.io.IOException;

import java.io.PrintStream;

import java.net.Socket;

public class S\_AsyncTask extends AsyncTask<String,Void,Void> {

private static final String IP="192.168.4.1";

private static final int PORT = 8080;

private Socket client = null;

private PrintStream out = null;

@Override

protected Void doInBackground(String... params){

String str = params[0];

try {

client = new Socket(IP,PORT);

client.setSoTimeout(8080);

out = new PrintStream(client.getOutputStream());

out.print(str);

out.flush();

if(client == null){

return null;

} else {

out.close();;

client.close();

}

} catch (IOException e){

e.printStackTrace();

}

return null;

}

}

数据库

Index.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>数据管理页面</title>  
</head>  
<style>  
 .main{  
 text-align: center; /\*让div内部文字居中\*/  
 background-color: #fff;  
 border-radius: 20px;  
 width: 300px;  
 height: 350px;  
 margin: auto;  
 position: absolute;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 }  
 .big{  
 font-size: large;  
 }  
</style>  
<body style="background-color: indianred">  
  
 <div class="main">  
 <h1 >数据库数据</h1>  
 <form method="post" action="deal.php">  
 <input type="submit" name="database" value="查看数据库">  
 <br><br>  
 </form>  
 <form method="post" action="deal.php">  
 <br><br>  
 <input type="submit" name="data" value="当前温湿度">  
 </form>  
 </div>  
</body>  
</html>

Deal.php

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>数据管理页面</title>  
</head>  
<style>  
 .main{  
 text-align: center; /\*让div内部文字居中\*/  
 background-color: #fff;  
 border-radius: 20px;  
 width: 300px;  
 height: 350px;  
 margin: auto;  
 position: absolute;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 }  
 .big{  
 font-size: large;  
 }  
</style>  
<body>  
  
 <div class="main">  
 **<?php** //style="background-color: indianred"  
 $mysql\_link = mysqli\_connect('localhost','root',  
 '','zigbee');  
 **if**(**isset**($\_POST['database'])){  
 $result = mysqli\_query($mysql\_link,'select \* from cgq');  
 **if**(!$mysql\_link){  
 printf("can't connect to mysql server. Errorcode: %s",mysqli\_connect\_error());  
 **exit**;  
 }  
 **else if**($result){  
  
 **echo** ('ID 温度 湿度 红外')."<br/>";  
 **while**($row = mysqli\_fetch\_assoc($result)){  
 **echo** $row['id'],"&nbsp;&nbsp;&nbsp;",  
 $row['wd'], "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;",  
 $row['sd'],"&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;",  
 $row['hw'],"<br/>";  
 }  
 }  
 mysqli\_free\_result($result);  
 }  
 **if**(**isset**($\_POST['data'])){  
 $result2 = mysqli\_query($mysql\_link,'select \* from cgq order by id DESC limit 1');//mysql获取数据库中的最后一条数据  
 **if**(!$mysql\_link){  
 printf("can't connect to mysql server. Errorcode: %s",mysqli\_connect\_error());  
 **exit**;  
 }  
 **else if**($result2){  
 **while**($row1 = mysqli\_fetch\_assoc($result2)){  
 **echo** "<br/>";  
 **echo** "<br/>";  
 **echo** "<br/>";  
 **echo** "<br/>";  
 **echo** ('当前温度为：'),$row1['wd']."<br/>";  
 **echo** "<br/>";  
 **echo** ('当前湿度为：'),$row1['sd']."<br/>";  
 **echo** "<br/>";  
 **if**($row1['hw']==1)  
 **echo** ('安全状态：有人')."<br/>";  
 **else if**($row1['hw']==0)  
 **echo** ('安全状态：无人')."<br/>";  
 }  
 }  
 mysqli\_free\_result($result2);  
 }  
 mysqli\_close($mysql\_link);  
 **?>** </div>  
</body>  
</html>