#include <Stepper.h>

#include <MsTimer2.h>

#include <SPI.h> // include libraries

#include <LoRa.h>

#include <dht11.h>

#include <DHT.h> //分别引用步进电机、定时器中断、DHT传感器的头文件，方便下一步定义

#define DHTPIN 7 //定义温湿度针脚号为7号引脚

#define DHTTYPE DHT11

#define AD5 A5 //定义模拟口A5

#define STEPS 250

Stepper stepper(STEPS, 3,5,4,6); //步进电机进行初始设置

DHT dht(DHTPIN, DHTTYPE);

const int csPin = 10; // LoRa radio chip select

const int resetPin = 9; // LoRa radio reset

const int irqPin = 2; // change for your board; must be a hardware interrupt pin

int pinRelay = 8; //继电器引脚

String a, b, m, i, j;

double tem, hum;

String outgoing; // outgoing message

String c,w;

double Intensity = 0; //光照度数值

String incoming = "";

long lastSendTime = 0; // last send time

int interval = 2000; // interval between sends

byte msgCount = 0; // count of outgoing messages

byte localAddress = 0xfe; // address of this device

byte destination = 0xbb; // destination to send to

int counter = 0;

int flagmotor = 0;

int flagwater = 0;

int flagdir = 0;

int flagjidian = 0;

int flaglock = 0;

String sk = "101", sg = "100", cq = "111", cl = "110"; //水泵和步进电机的状态

String inttostring(int n)//int转string函数

{

if (n == 0) {

return "00"; //n为0

}

char result[25];

itoa(n, result, 10);

return result;

}

void setup() {

Serial.begin(9600); // initialize serial

//LoRa.setSyncWord(0x12);

pinMode(DHTPIN, OUTPUT); //定义输出口

pinMode(pinRelay, OUTPUT); //设置继电器

while (!Serial);

Serial.println("LoRa Duplex");

// override the default CS, reset, and IRQ pins (optional)

LoRa.setPins(csPin, resetPin, irqPin);// set CS, reset, IRQ pin

Serial.flush();

if (!LoRa.begin(4335E5)) { // initialize ratio at 915 MHz

Serial.println("LoRa init failed. Check your connections.");

while (true); // if failed, do nothing

}

//LoRa.onReceive(onReceive);

//LoRa.receive();

Serial.println("LoRa init succeeded.");

/\*while (Serial.available() > 0)

{

Serial.read();

delay(2);

}\*/

stepper.setSpeed(90);

}

void loop() {

Serial.flush();

if (millis() - lastSendTime > interval) {

if (LoRa.available()) {

return;

}

Intensity = analogRead(AD5); //读取模拟口AD5的值，存入Intensity变量

tem = dht.readTemperature(); //将温度值赋值给tem

hum = dht.readHumidity(); //将湿度值赋给hum

//Intensity = int((Intensity / 1021) \* 100);

i = inttostring(Intensity);

a = inttostring(tem);

b = inttostring(hum);

m = '0' + a + b + i;

Serial.print("Sending packet: ");

Serial.println(counter);

sendMessage(m);

Serial.println("Temp Hum Int:" + m);

lastSendTime = millis(); // timestamp the message

interval = random(5000) + 10000; // 2-3 seconds

counter++;

}

onReceive(LoRa.parsePacket());

//Serial.println(c);

// parse for a packet, and call onReceive with the result:

//200水泵关

//201水泵开

//210窗帘起

//211窗帘落

if (c[0] == '2')

{

flaglock = 1; //远程模式锁存

}

if (c[0] == '3')

{

flaglock = 0; //本地控制

}

if (flaglock == 1) //执行远程控制

{

if (c[1] == '1' && flagjidian == 0)

{

//浇水开

digitalWrite(pinRelay, HIGH);

sendMessage(sk);

Serial.println(sk);

Serial.println("jiaoshui yuan kai");

flagjidian = 1;

}

if (c[1] == '0' && flagjidian == 1)

{

//浇水关

digitalWrite(pinRelay, LOW);

sendMessage(sg);

Serial.println(sg);

Serial.println("jiaoshui yuan guan");

flagjidian = 0;

}

if (flagdir == 0 && c[2] == '1')

{

sendMessage(cq);

Serial.println(cq);

//窗帘升起

for (int j = 1; j < 2; j++)

{

Serial.println("chuang sheng yuan");

stepper.step(12288); //4步模式下旋转一周用2048 步。

if (j == 1)

{ flagdir = 1;

}

delay(500);

}

}

if (flagdir == 1 && c[2] == '0')

{

sendMessage(cl);

Serial.println(cl);

//窗帘反转

for (int j = 1; j < 2; j++)

{

Serial.println("chuang jiang yuan");

stepper.step(-12288); //4步模式下旋转一周用2048 步。

if (j == 1)

{

flagdir = 0;

}

delay(500);

}

}

c = "";

//return;

}

else if(flaglock == 0) //if (c[0] == "") //执行本地控制

{

if (Intensity < 900 && Intensity > 450 && flagdir == 0)

{

sendMessage(cq);

Serial.println(cq); //窗帘升起

for (int j = 1; j < 2; j++)

{

Serial.println("stepper up"); //窗帘升起

stepper.step(12288); //4步模式下旋转一周用2048 步。

delay(500);

//flagmotor = 1;

if (j == 1)

{

flagdir = 1;

}

}

}

if (Intensity < 450 || Intensity > 900 && flagdir == 1)

{

sendMessage(cl);

Serial.println(cl);

for(int j=1;j<2;j++)

{

Serial.println("steper down"); //窗帘降落

stepper.step(-12288); //4步模式下旋转一周用2048 步。

delay(500);

// flagmotor = 1;

if (j == 1)

{

flagdir = 0;

}

}

}

if (hum < 60 && flagjidian == 0) {

sendMessage(sk);

Serial.println(sk);

for (int j = 1; j < 2; j++)

{

digitalWrite(pinRelay, HIGH); //输出HIGH电平，继电器模块闭合

Serial.println("jidian jin kai");

if (j == 1)

{

flagjidian = 1;

}

}

}

if (hum > 60 && flagjidian == 1) {

sendMessage(sg);

Serial.println(sg);

for (int j = 1; j < 2; j++)

{

digitalWrite(pinRelay, LOW); //输出HIGH电平，继电器模块闭合

Serial.println("jidian jin guan");

}

if (j == 1)

{

flagjidian = 0;

}

}

return;

}

}

void sendMessage(String outgoing) {

LoRa.beginPacket(); // start packet

LoRa.write(destination); // add destination address

LoRa.write(localAddress); // add sender address

LoRa.write(msgCount); // add message ID

LoRa.write(outgoing.length()); // add payload length

LoRa.print(outgoing); // add payload

LoRa.endPacket(); // finish packet and send it

msgCount++; // increment message ID

}

void onReceive(int packetSize)

{

if (packetSize == 0)

return; // if there's no packet, return

// read packet header bytes:

int recipient = LoRa.read(); // recipient address

byte sender = LoRa.read(); // sender address

byte incomingMsgId = LoRa.read(); // incoming msg ID

byte incomingLength = LoRa.read(); // incoming msg length

c= "";

while (LoRa.available()) {

c+= (char)LoRa.read();

// Serial.println((char)LoRa.read());

// Serial.println ("c");

// Serial.println (c);

}

/\* for(int j=0;j<3;j++)

{

c[j]=w[j];

}

/\*if (incomingLength != incoming.length()) { // check length for error

Serial.println("error: message length does not match length");

// return; // skip rest of function

}\*/

//if the recipient isn't this device or broadcast,

if (recipient != localAddress && recipient != 0xff) {

Serial.println("This message is not for me.");

return; // skip rest of function

}

// if message is for this device, or broadcast, print details:

Serial.println("Received from: 0x" + String(sender, HEX));

Serial.println("Sent to: 0x" + String(recipient, HEX));

Serial.println("Message ID: " + String(incomingMsgId));

Serial.println("Message length: " + String(incomingLength));

Serial.println("Message: " + c);

Serial.println("RSSI: " + String(LoRa.packetRssi()));

Serial.println("Snr: " + String(LoRa.packetSnr()));

Serial.println();

delay(1000); //延时一段时间

}