主函数

#include "Wire.h"

#include "I2Cdev.h"

#include "MPU6050.h"

#include "U8glib.h"

int16\_t ax, ay, az,acc; //加速度检测参数

int16\_t gx, gy, gz;

int Button;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*加速度检测的变量\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int16\_t x,y,z;

int16\_t i\_1,j;

MPU6050 accelgyro;

#define LED\_PIN 13

int16\_t flag,p,q;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OLED代码的变量\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int16\_t OLED\_flag; //OLED启动关闭标志

int menu\_current = 1;

unsigned char OledFlag = 0;

U8GLIB\_SSD1306\_128X64 u8g(U8G\_I2C\_OPT\_NONE);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*心率检测代码的变量\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int data,num; //Heat:time data;

long timey,timer,time1,time2,time3; //定义变量为浮点数

int16\_t X;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*GPS的变量\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

boolean STA;

float f\_latitude,f\_longitude; //经纬度

float temp;

int GPS\_flag;

int GSM\_flag; //GSM启动标志

void setup() {

Wire.begin();

Serial.begin(38400);

Serial1.begin(38400);

Serial.println("Initializing I2C devices...");

accelgyro.initialize();

Serial.println("Testing device connections...");

Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");

// configure Arduino LED for

pinMode(LED\_PIN, OUTPUT);

digitalWrite(LED\_PIN,LOW);

q=0;

p=0; //超重次数标志

flag=0; //flag 摔倒标志

menu\_current = 1;

OledFlag = 5;

xinlv\_begin(); //启动心率模块

switch\_receive\_begin(); //接收yeelink开关初始化

}

void loop()

{

Button = analogRead(A7);

if(Button>55&&Button<100) //jinji qiujiu anniu

{

SendChineseMessage2();

}

MPU6050\_start();

if(OLED\_flag==1)

{ checkgsm(); //检测是否插入GSM卡

OLED\_start();

X=1;

}

else if(!OLED\_flag)

{

if(X)

OLED\_close();

X=0;

if(GSM\_flag) //有GSM卡插入时才能连接yeelink

switch\_receive\_start(2); //1 wei xinlv ,2 wei GPS

}

}

GPRS模块

#include <String.h>

#define PhoneNunber "13246528267"

#define InforMation "Fall off"

static char buf[20];

void checkgsm() //shezhi GSM shifou zhengchang biaozhi

{ String showimis="";

// Serial1.println("AT");

Serial1.println("AT+CIMI");

delay(500);

while(Serial1.available()>0)

{

char a=Serial1.read();

showimis=showimis+a;

}

Serial.println(showimis);

if(showimis[2]=='4'&&showimis[3]=='6'&&showimis[4]=='0'&&showimis[5]=='0')

GSM\_flag=1;

else

GSM\_flag=0;

Serial.print(GSM\_flag);

}

void SendChineseMessage2() //jinji qiujiu anniu

{

Serial1.println("AT");

delay(500);

Serial1.println("AT+CMGF=0");

delay(500);

Serial1.println("AT+CSCS=\"UCS2\"");

delay(500);

Serial1.println("AT+CMGS=33");

delay(500);

Serial1.println("0011000D91683136215297F00008AA1280014EBA904752307D27602595EE98983002"); //lao ren yudao jinji wenti.

delay(500);

Serial1.println((char)26);

delay(1000);

}

void SendChineseMessage() //laoren diedao

{

Serial1.println("AT");

delay(500);

Serial1.println("AT+CMGF=0");

delay(500);

Serial1.println("AT+CSCS=\"UCS2\"");

delay(500);

Serial1.println("AT+CMGS=25");

delay(500);

Serial1.println("0011000D91683136215297F00008AA0A80014EBA8DCC5012FF01"); //lao ren die dao!

delay(500);

Serial1.println((char)26);

delay(1000);

}

void SendChineseMessage1() //xinlv buzai zhengchang fanwei nei

{

Serial1.println("AT");

delay(500);

Serial1.println("AT+CMGF=0");

delay(500);

Serial1.println("AT+CSCS=\"UCS2\"");

delay(500);

Serial1.println("AT+CMGS=33");

delay(500);

Serial1.println("0011000D91683136215297F00008AA125FC373874E0D57286B635E38830356F4FF01"); //xinlv buzai zhengchang fanwei!

delay(500);

Serial1.println((char)26);

delay(1000);

}

void SendTextMessage() //

{

GSMConnect(); // the GPRS baud rate

Serial1.print("AT+CMGF=1\r");

delay(100);

ShowSerialData();

sprintf(buf, "AT+CMGS=\"%s\"",PhoneNunber);

Serial1.println(buf);

for(int i=0; i<20; i++)

buf[i]=NULL;

delay(100);

Serial1.println(InforMation);//the content of the message

delay(100);

Serial1.println((char)26);//the ASCII code of the ctrl+z is 26

delay(100);

Serial1.println();

// ShowSerialData();

delay(5000);

}

///DialVoiceCall

void CallPhone()

{ GSMConnect(); // the GPRS baud rate

sprintf(buf, "ATD%s;",PhoneNunber);

Serial1.println(buf);//send sms message, be careful need to add a country code before the cellphone number

for(int i=0; i<20; i++)

buf[i]=NULL;

delay(100);

Serial1.println();

delay(2000);

//ShowSerialData();

}

void ShowSerialData()

{

while(Serial1.available()!=0)

{

Serial.write(Serial1.read());

}

}

void GSMConnect()

{

delay(500);

Serial1.println("AT"); //Because we want to send the SMS in text mode

delay(200);

Serial1.println("AT"); //Because we want to send the SMS in text mode

delay(200);

Serial1.println("AT+CMGF=1");//TEXT

delay(500);

ShowSerialData();

Serial1.println("AT+CMGD=1");//Delete message return 32 length

delay(500);

ShowSerialData();

Serial1.println("AT+CNMI=2,1");

delay(500);

ShowSerialData();

Serial1.println("AT+CSCS=\"GSM\"");

delay(500);

ShowSerialData();

Serial1.println("AT");

ShowSerialData();

delay(10000);

}

void gettime() //huoqu shijian

{

String GSM\_time="";

GSMConnect();

// Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Serial1.println("AT+CIPSTART=\"TCP\",\"time.nist.gov\",13");

delay(500);

while(Serial1.available()>0)

{

char a=Serial1.read();

GSM\_time=GSM\_time+a;

}

Serial.println(GSM\_time);

}

void getsite() //huoqu weizhi

{

String GSM\_site="";

Serial1.println("AT+SAPBR=2,1");ShowSerialData();

Serial1.println("ATE0");ShowSerialData();

Serial1.println("AT+SAPBR=3,1,\"Contype\",\"GPRS\"");ShowSerialData();

Serial1.println("AT+SAPBR=3,1,\"APN\",\"CMNET\"");ShowSerialData();

Serial1.println("AT+SAPBR=1,1");ShowSerialData();

// Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Serial1.println("AT+CIPGSMLOC=1,1");delay(500);

while(Serial1.available()!=0)

{

char a=Serial1.read();

GSM\_site=GSM\_site+a;

}

Serial.println(GSM\_site);

Serial1.println("AT+SAPBR=0,1");delay(500);ShowSerialData();

}

GPS模块

#include <Adafruit\_GPS.h>

#include <SoftwareSerial.h>

#include<EEPROM.h>

#include <U8glib.h>

SoftwareSerial a(8,9);

Adafruit\_GPS GPS(&Serial);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//\*\*\*\*\*\*\*两个共用结构体存储浮点数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

union LAT

{ float lat;

byte a[4];

}m;

union LON

{ float lon;

byte b[4];

}n;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//-------字体设置，大、中、小

#define setFont\_L u8g.setFont(u8g\_font\_courB14)

#define setFont\_M u8g.setFont(u8g\_font\_fixed\_v0r)

#define setFont\_S u8g.setFont(u8g\_font\_chikitar)

#define u8g\_logo\_width 128

#define u8g\_logo\_height 18

#define GPSECHO false

//==========================

//GPS状态

int i\_lat,i\_lon;

char c\_lat,c\_lon; //经纬极向

int itime[3]; //时间

int idate[3]; //日期

float f\_Speed; //速度

int i\_Speed[2]; //速度格式化

float f\_Height; //海拔

int i\_satellites; //卫星数

float f\_fixquality; //信号质量

#define init\_updata 1000 //gps数据刷新时间

#define init\_oled 500 //OLED刷新时间

unsigned long time\_oled = millis();

unsigned long time\_gps = millis();

void GPS\_begin()

{

GPS.begin(38400);

}

void GPS\_start()

{

float temp;

char c = GPS.read();

// if you want to debug, this is a good time to do it!

// if a sentence is received, we can check the checksum, parse it...

if (GPS.newNMEAreceived()) {

// a tricky thing here is if we print the NMEA sentence, or data

// we end up not listening and catching other sentences!

// so be very wary if using OUTPUT\_ALLDATA and trytng to print out data

if (!GPS.parse(GPS.lastNMEA())) // this also sets the newNMEAreceived() flag to false

return; // we can fail to parse a sentence in which case we should just wait for another

}

if (time\_gps > millis())

time\_gps = millis();

if (millis() - time\_gps > init\_updata)

{

time\_gps = millis(); // reset the time\_gps

if(GPS.hour>=0&&GPS.hour<=16)

itime[0]=GPS.hour+8;

else

itime[0]=GPS.hour;

itime[1]=GPS.minute;

itime[2]=GPS.seconds;

itime[2] += 20;

if(itime[2]>=60)

itime[2] -= 60;

idate[0]=GPS.year;

idate[1]=GPS.month;

idate[2]=GPS.day;

f\_fixquality=GPS.fixquality; //信号质量

STA=GPS.fix; //GPS定位状态

Serial.print("STA = ");

Serial.println(STA);

if (STA) //当GPS定位上

{

GPS\_flag++;

f\_latitude=GPS.latitude;

f\_longitude=GPS.longitude;

c\_lat=GPS.lat;

c\_lon=GPS.lon;

lat\_lon\_transform(); //经纬度转化

f\_Speed=1.852\*GPS.speed; //速度转化

i\_Speed[0]=int(f\_Speed\*10)%10; //速度格式化

i\_Speed[1]=int(f\_Speed); //速度格式化

f\_Height=GPS.altitude; //海拔

i\_satellites=GPS.satellites; //卫星数

temp = i\_Speed[1] + 0.1\*i\_Speed[0];

if(GPS\_flag == 20||GPS\_flag == 50)

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*EEPROM save\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

m.lat=f\_latitude;

n.lon=f\_longitude;

for(i\_lat=0;i\_lat<4;i\_lat++)

EEPROM.write(i\_lat,m.a[i\_lat]);

for(i\_lon=4;i\_lon<8;i\_lon++)

EEPROM.write(i\_lon,n.b[i\_lon-4]);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*yeelink post\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(GSM\_flag) //有GSM卡插入时才能连接yeelink

post\_data(1,0,f\_latitude,f\_longitude,temp);

GPS\_flag = 21;

}

}

vooled();//OLED显示

}

}

//lat\_lon\_transform================================

void lat\_lon\_transform()

{

f\_latitude=(int(f\_latitude)/100)+((int(f\_latitude)%100)/60.0)+((f\_latitude-int(f\_latitude))/60.0);

if(c\_lat=='S') //南纬

f\_latitude=-f\_latitude;

//---------------------------------

f\_longitude=(int(f\_longitude)/100)+((int(f\_longitude)%100)/60.0)+((f\_longitude-int(f\_longitude))/60.0);

if(c\_lon=='W') //西经

f\_longitude=-f\_longitude;

}

void draw\_gps(void)

{

for(int a=0;a<3;a++)

{

u8g.drawFrame(106+(6\*a), 4-(a\*2), 5, 4+(a\*2));

}

for(int a=0;a<f\_fixquality+1;a++)

{

u8g.drawBox(106+(6\*a), 4-(a\*2), 5, 4+(a\*2));

}

u8g.setPrintPos(0, 8);

u8g.print("Sat:");

u8g.print(i\_satellites);

u8g.setPrintPos(36, 8);

u8g.print("ELE:");

u8g.print(f\_Height);

u8g.print("m");

u8g.drawLine(0,11 , 128, 11);

setFont\_L;

u8g.setPrintPos(2, 27);

u8g.print("Speed:");

if(STA)

{

u8g.print(i\_Speed[1]);

setFont\_M;

u8g.print(".");

u8g.print(i\_Speed[0]);

}

else

{

u8g.print("N/A");

setFont\_M;

}

u8g.drawLine(0,29 , 128, 29);

/\*\*\*\*\*\*\*\*\*\*read EEPROM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for(i\_lat=0;i\_lat<4;i\_lat++)

m.a[i\_lat]=EEPROM.read(i\_lat);

for(i\_lon=4;i\_lon<8;i\_lon++)

n.b[i\_lon-4]=EEPROM.read(i\_lon);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

u8g.setPrintPos(2, 41);

u8g.print("Lat.: ");

u8g.print(m.lat,4);

u8g.print(" ");

u8g.print( c\_lat);

u8g.setPrintPos(2, 50);

u8g.print("Lon.: ");

u8g.print(n.lon,4);

u8g.print(" ");

u8g.print( c\_lon);

u8g.drawLine(0, 53, 128, 53);

u8g.setPrintPos(2, 64);

u8g.print("20");

u8g.print("14");

u8g.print("-");

u8g.print("9");

u8g.print("-");

u8g.print("10");

u8g.print(" ");

u8g.print(itime[0]);

u8g.print(":");

u8g.print(itime[1]);

u8g.print(":");

u8g.print(itime[2]);

}

//OLED===================================================

void vooled()

{

if (time\_oled > millis())

time\_oled = millis();

if(millis()-time\_oled>init\_oled)

{

u8g.firstPage();

do

{

draw\_gps();

}

while( u8g.nextPage() );

}

}

OLED显示模块

#include "U8glib.h"

char \*menu\_strings[5] = { "\*\*\*Main Menu\*\*", "Heart Rate", "Ring up", "Location","Information" };

#define back 1

#define up 3

#define down 4

#define confirm 5

#define error -1

int key\_up;

int sroll;

//

static unsigned char u8g\_logo\_bits1[] U8G\_PROGMEM = {

0x00,0x00,0x00,0x00,0x00,0x00,0x78,0x3C,

0xFC,0x7E,0xFE,0xFF,0xFE,0xFF,0xFE,0xFF,

0xFE,0xFF,0xFC,0x7F,0xF8,0x3F,0xF0,0x1F,

0xE0,0x0F,0xC0,0x07,0x80,0x03,0x00,0x01//xin xing

};

static unsigned char u8g\_logo\_bits2[] U8G\_PROGMEM = {

0xFC,0x3F,0x02,0x41,0x05,0xA1,0x01,0x80,

0x01,0x80,0x81,0x80,0x81,0x80,0x87,0xE0,

0x81,0x80,0x01,0x81,0x01,0x82,0x01,0x84,

0x01,0x88,0x05,0xA1,0x02,0x41,0xFC,0x3F,//zhongbiao

};

void draw\_Information() //xinxi jianjie

{

u8g.setDefaultForegroundColor();

u8g.setFont(u8g\_font\_unifont);

u8g.setPrintPos(sroll, 10);

u8g.print("\*\* Information \*\*");

u8g.drawStr(0,25, "name:ZhangSan");

u8g.drawStr(0,40, "Add:Zhuhai BNUZ");

u8g.drawStr(0,55, "Call:13631225790");

}

//

void draw\_close(void) {

// graphic commands to redraw the complete screen should be placed here

u8g.setFont(u8g\_font\_unifont);

//u8g.setFont(u8g\_font\_osb21);

u8g.drawStr( 0, 22, " ");

}

void draw\_Heart(void) {

u8g.drawXBMP( 90, 0, 16, 16, u8g\_logo\_bits1);

u8g.drawXBMP( 90, 20, 16, 16, u8g\_logo\_bits2);

u8g.setFont(u8g\_font\_fub20);

u8g.setPrintPos(32, 64);

u8g.print(num); //count

u8g.setFont(u8g\_font\_9x18B);

u8g.print("N/min");

u8g.setFont(u8g\_font\_7x14);

u8g.setPrintPos(0, 16);

u8g.print("Heat:");

u8g.setFont(u8g\_font\_9x18B);

u8g.print(data);

u8g.setFont(u8g\_font\_7x14);

u8g.print("N");

u8g.setPrintPos(0, 34);

u8g.print("Time:");

u8g.setFont(u8g\_font\_9x18B);

u8g.print(timey); //timey

u8g.setFont(u8g\_font\_7x14);

u8g.print("S");

}

void drawMenu() //hua caidan kuangkuang

{

uint8\_t i, h;

u8g\_uint\_t w, d;

u8g.setFont(u8g\_font\_7x14);

u8g.setFontRefHeightText();

u8g.setFontPosTop();

h = u8g.getFontAscent()-u8g.getFontDescent();

w = u8g.getWidth();

for( i = 0; i < 5; i++ )

{

d = (w-u8g.getStrWidth(menu\_strings[i]))/2;

u8g.setDefaultForegroundColor();

if ( i == menu\_current ) // i == menu\_current

{

u8g.drawBox(0, i\*h+1, w, h);

u8g.setDefaultBackgroundColor();

}

u8g.drawStr(d, i\*h, menu\_strings[i]);

}

if(GSM\_flag)

u8g.drawStr(113,0, "ok");

else

u8g.drawStr(113,0, "!!");

}

void draw(unsigned char choose)

{

int button;

u8g.firstPage();

do

{

switch(choose)

{

case 1: //xinlv jiance

// xinlv\_begin(); //buneng fang zheli

while(1)

{

xinlv\_start();

button = analogRead(A7);

if(button>20&&button<50)//Main menu

break;

}

break;

case 2: // boda dianhua

CallPhone();

while(1)

{

u8g.firstPage();

do

{

u8g.setFont(u8g\_font\_unifont);

//u8g.setFont(u8g\_font\_osb21);

u8g.drawStr( 10,35, "Calling......");

} while( u8g.nextPage() );

button = analogRead(A7);

if(button>20&&button<50)//Main menu

break;

}

break;

case 3: //GPS dingwei

GPS\_begin();

while(1)

{

GPS\_start();

button = analogRead(A7);

if(button>20&&button<50)//Main menu

break;

}

break; //xianshi dangqian weizhi,tongguo GPS

case 4: draw\_Information(); break;

case 5: drawMenu(); break;

}

} while( u8g.nextPage() );

if(millis()-timer>2)

{

timer=millis();

sroll-=4;

}

}

void OLED\_start() {

int button = analogRead(A7); //5 ge xiao jianpan

//Serial.println(button);

if(button>0&&button<50)//Main menu fanhui

{

key\_up=back;

OledFlag = 5;

}

else if(button>100&&button<200) //shang

{

key\_up=up;

if(menu\_current==1)

{

menu\_current = 4;

}

else

menu\_current--;

}

else if(button>200&&button<300) //xia

{

key\_up=down;

if(menu\_current==4)

{

menu\_current = 1;

}

else

menu\_current++;

}

else if(button>300&&button<400) //queding

{

key\_up=confirm;

Serial.println(menu\_current);

OledFlag = menu\_current;

}

else

key\_up=error;

draw(OledFlag);

}

void OLED\_close()

{

u8g.firstPage();

do {

draw\_close();

} while( u8g.nextPage() );

// rebuild the picture after some delay

delay(500);

}

心率检测模块

#include "U8glib.h"

int pulsePin = A0; // Pulse Sensor purple wire connected to analog pin 0

int blinkPin = 5; // pin to blink led at each beat

int fadePin = 6; // pin to do fancy classy fading blink at each beat

int fadeRate = 0; // used to fade LED on with PWM on fadePi

// these variables are volatile because they are used during the interrupt service routine!

volatile int BPM; // used to hold the pulse rate

volatile int Signal; // holds the incoming raw data

volatile int IBI = 600; // holds the time between beats, must be seeded!

volatile boolean Pulse = false; // true when pulse wave is high, false when it's low

volatile boolean QS = false; // becomes true when Arduoino finds a beat.

void xinlv\_begin(){

//u8g.setRot180();

timer=millis();

timey=millis();

time1=millis();

time2=millis();

time3=millis();

pinMode(blinkPin,OUTPUT); // pin that will blink to your heartbeat!

pinMode(fadePin,OUTPUT); // pin that will fade to your heartbeat!

pinMode(pulsePin, INPUT\_PULLUP);

// Serial.begin(115200); // we agree to talk fast!

interruptSetup(); // sets up to read Pulse Sensor signal every 2mS

}

void xinlv\_start(){

if (QS == true){ // Quantified Self flag is true when arduino finds a heartbeat

fadeRate = 255; // Set 'fadeRate' Variable to 255 to fade LED with pulse

QS = false; // reset the Quantified Self flag for next time

}

ledFadeToBeat();

if(fadeRate>230)

data++;

if(data>0)

{

// Serial.print("+++++++++:............................");

// Serial.println(millis()-timer);

if(millis()-timer>1000)

{

timer=millis();

timey++;

}

if(timey>10)

num=60\*(data)/timey;

}

if(fadeRate==0)

{

if(millis()-time2>1000)

{

time2=millis();

time3++;

}

}

else

time3=0;

if(time3>1||timey>30)

{

if(timey>30)

{ if(num<50||num>80)

SendChineseMessage1();

delay(3000);

if(GSM\_flag) //有GSM卡插入时才能连接yeelink

post\_data(0,num,0,0,0); //0 ,xinlv = 90,0,0,0

}

data=0;

num=0;

timey=0;

time3=0;

}

/\*

delay(20); // take a break

Serial.print("xintiao:");

Serial.print(data); //Y

Serial.print(" ");

// Serial.println(fadeRate);

Serial.print(" timey: ");

Serial.print(timey);

// Serial.print(" time3: ");

//Serial.println(time3);

Serial.print(" xintiao: ");

Serial.print(num);

Serial.println("num/min");

\*/

u8g.firstPage();

do {

draw\_Heart();

}

while( u8g.nextPage() );

}

void ledFadeToBeat(){

fadeRate -= 15; // set LED fade value

fadeRate = constrain(fadeRate,0,255); // keep LED fade value from going into negative numbers!

//analogWrite(fadePin,fadeRate); // fade LED

if(fadeRate>200)

digitalWrite(fadePin,HIGH);

else

digitalWrite(fadePin,LOW);

}

MPU6050陀螺仪模块

#include "U8glib.h"

bool blinkState = false;

void MPU6050\_start()

{

// read raw accel/gyro measurements from device

OLED\_flag=0;

accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

if(abs(ax)<3000&&abs(ay)<3000&&abs(az)>17000)

OLED\_flag=1;

// these methods (and a few others) are also available

//accelgyro.getAcceleration(&ax, &ay, &az);

//accelgyro.getRotation(&gx, &gy, &gz);

acc=abs(ax)+abs(ay)+abs(az);

Serial.print("a/g:\t");

Serial.print(abs(ax)); Serial.print("\t");

Serial.print(abs(ay)); Serial.print("\t");

Serial.print(abs(az)); Serial.print("\t");

Serial.print(acc); Serial.print("\t");

Serial.print(p);Serial.print("\t");

Serial.print(flag);Serial.print("\t");

if(acc<3000&&acc>0) //判断是否有失速现象发生

{

for(i\_1=0;i\_1<80;i\_1++) //当发生失速现象后测出失速后大约80个加速度数值

{ accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

Serial.print("{");Serial.print(p);Serial.print("\t");

Serial.print(abs(ax)); Serial.print("\t");

Serial.print(abs(ay)); Serial.print("\t");

Serial.print(abs(az)); Serial.print("\t");

Serial.print("}");

if(ax>32700||ay>32700||az>32700||ax<-32700||ay<-32700||az<-32700) //判断失速后是否又发生超重现象

p=p+1;

if(p>=3)

{ p=0;

delay(2000); //延时2秒

for(j=0;j<30;j++) //取延时后的加速度数据

{ accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

Serial.print("|");Serial.print(flag);Serial.print("\t");

Serial.print(x-abs(ax)); Serial.print("\t");

Serial.print(y-abs(ay)); Serial.print("\t");

Serial.print(z-abs(az)); Serial.print("\t");

Serial.print(q); Serial.print("|");

if(x-abs(ax)>300&&y-abs(ay)>300&&z-abs(az)>300) //若发生超重一秒后,加速度在一个平稳的数值内.则可确定摔倒了

q=q+1;

if(q==2)

break;

x=abs(ax);

y=abs(ay);

z=abs(az);

if(j==29)

flag=1;

}

}

if(j==30)

break;

}

}

if(flag==1)

{ int button;

GPS\_flag = 0;

SendChineseMessage();

delay(3000);

//SendTextMessage();

//delay(3000);

post\_data(2,flag,0,0,0);

delay(7000);

for(;GPS\_flag<21; )

{

Serial.println("++++++++++++++++++++++++++++++++++++++++++++++");

GPS\_start();

}

while(1)

{

u8g.firstPage();

do

{draw\_Information();

} while( u8g.nextPage() );

button = analogRead(A7);

if(button>20&&button<50)//Main menu

break;

}

flag=0;

}

// blink LED to indicate activity

//blinkState = !blinkState;

//digitalWrite(LED\_PIN, blinkState);

}

Yeelink发送模块

#include <stdio.h>

#include <math.h>

int device[] = {12797};

int sense[] = {22589,21934,21121,22601}; // number,GPS

void yeelink\_begin()

{

Serial.begin(38400);

Serial1.begin(38400);

yeelink\_begins();

}

//post\_data(0,90,0,0,0); //0 ,xinlv = 90,0,0,0

// post\_data(1,0,20,150,speed); //1,0,lat,lng,speed

// delay(15000);

void post\_data(int choose,int number,float lat,float lng,float speeds)

{ //yeelink\_begin();

int length = 0;

int lats=lat;

int lngs=lng;

int speeda = speeds;

unsigned char count = 0;

if(choose == 0||choose == 2||choose==3)

{

if(number>=0&&number<10)

length = 1;

else if(number>=10&&number<100)

length = 2;

else

length = 3;

length = length + 11 - 1;

}

else if(choose == 1)

{

while(1)

{

lats /= 10;

if(lats != 0)

count++;

else

break;

}

while(1)

{

lngs /=10;

if(lngs != 0)

count++;

else

break;

}

while(1)

{

speeda /= 10;

if(speeda != 0)

count++;

else

break;

}

length =46 + count + 15;

}

connect();

Serial1.print("POST /v1.0/device/");

Serial1.print(device[0]);

Serial1.print("/sensor/");

Serial1.print(sense[choose]);

Serial1.println("/datapoints HTTP/1.1");

Serial1.println( "Host: api.yeelink.net");

Serial1.println("U-ApiKey:587407b0532721175514efd429f5dc98");

Serial1.print("Content-Length: ");

Serial1.println(length); //length

Serial1.println("Content-Type: application/x-www-form-urlencoded");

Serial1.println("Connection: close");

Serial1.println();

if(choose == 0||choose == 2||choose==3)

{

Serial1.print("{\"value\":"); //zhi

Serial1.print(number);

Serial1.println("}");

}

else if(choose == 1)

{

Serial1.print("{\"value\":{\"lat\":");

Serial1.print(lat);

Serial1.print(",\"lng\":");

Serial1.print(lng);

Serial1.print(",\"speed\":");

Serial1.print(speeds);

Serial1.print(",\"offset\":\"yes\"");

Serial1.println("}}");

}

Serial1.println((char)26);

close();

}

void yeelink\_begins()

{

Serial1.print("ati\r\n");

Serial1.print("AT+CGCLASS=\"B\"\r\n");

Serial1.print("AT+CGDCONT=1,\"IP\",\"CMNET\"\r\n");

Serial1.print("AT+CGATT=1\r\n");

Serial1.print("AT+CIPCSGP=1,\"CMNET\"\r\n"); //uninet

}

void connect()//TCP连接

{

Serial1.print("AT+CLPORT=\"TCP\",\"2000\"\r\n");

delay(500);

Serial1.print("AT+CIPSTART=\"TCP\",\"42.96.164.52\",\"80\"\r\n"); //the ip addr to yeelink

delay(5000);

Serial1.print("AT+CIPSEND\r\n");

delay(500);

}

void close()//结束TCP连接

{

Serial1.print("AT+CIPCLOSE=1\r\n");

delay(500);

Serial1.print("AT+CIPSHUT\r\n");

delay(500);

}

Yeelink接收模块

#include "U8glib.h"

long previousMillis = 0; // will store last time LED was updated

long interval = 10000; // interval at which to get (milliseconds)

boolean ResponseBegin = false;

void switch\_receive\_begin()

{

// Serial.begin(9600);

// Serial1.begin(9600);

Serial1.println("AT+SAPBR=2,1");

Serial1.println("ATE0");

Serial1.println("AT+SAPBR=3,1,\"Contype\",\"GPRS\"");

Serial1.println("AT+SAPBR=3,1,\"APN\",\"CMNET\"");

Serial1.println("AT+SAPBR=1,1");

}

void switch\_receive\_start(unsigned char flag)

{

unsigned long currentMillis = millis();

if(currentMillis - previousMillis > interval)

{

previousMillis = currentMillis;

send\_data("AT+HTTPINIT",flag); //begin()

if(flag == 1) //xinlv switch

send\_data("AT+HTTPPARA=\"URL\",\"http://api.yeelink.net/v1.0/device/12797/sensor/22600/datapoints\"",flag);

else if(flag == 2) //GPS switch

send\_data("AT+HTTPPARA=\"URL\",\"http://api.yeelink.net/v1.0/device/12797/sensor/22601/datapoints\"",flag);

//set canshu

//delay(500);

send\_data("AT+HTTPPARA=\"CID\",1",flag);

//delay(500);

send\_data("AT+HTTPACTION=0",flag); //fangshi dongzuo

//delay(500);

send\_data("AT+HTTPREAD",flag);

//delay(500);

send\_data("AT+HTTPTERM",flag); //End

//delay(1000);

}

}

void send\_data(String msg,unsigned char flag1)

{

String str;

boolean ActionFlag = false;

MPU6050\_start();

if(msg.indexOf("HTTPACTION")>0)

ActionFlag = true;

Serial1.println(msg);

while(!Serial1.available())

delay(10);

while(Serial1.available())

{

char c = Serial1.read();

str += c;

delay(2);

}

if(ActionFlag)

{

while(!Serial1.available())

delay(10);

while(Serial1.available())

{

char c = Serial1.read();

str += c;

delay(2);

}

}

//{"timestamp":"2014-08-24T13:30:58","value":0} 53 60?

if(str.indexOf("value")>0) //jiance daozhi ,output it

{

// Serial.println("+++++++++++++++++++++");

// Serial.println(str.indexOf("\"value\":"));

// Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Serial.println(str[60]);

Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

if(str[60] == '1'&& flag1 == 1)

{

//draw(1); //xinlv

Serial.println("shibie dao xinlv kaiguan wei 1.");

}

else if(str[60] == '1'&& flag1 == 2) //GPS kaiguan

{

// draw(3);

GPS\_flag = 0;

delay(3000);

for(;GPS\_flag<21; )

{

Button = analogRead(A7);

GPS\_start();

if(Button>20&&Button<50)//Main menu

break;

}

delay(1000);

post\_data(3,0,0,0,0);

}

// Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

Serial.println(str);

}

中断处理模块

volatile int rate[10]; // array to hold last ten IBI values

volatile unsigned long sampleCounter = 0; // used to determine pulse timing

volatile unsigned long lastBeatTime = 0; // used to find IBI

volatile int P =512; // used to find peak in pulse wave, seeded

volatile int T = 512; // used to find trough in pulse wave, seeded

volatile int thresh = 512; // used to find instant moment of heart beat, seeded

volatile int amp = 100; // used to hold amplitude of pulse waveform, seeded

volatile boolean firstBeat = true; // used to seed rate array so we startup with reasonable BPM

volatile boolean secondBeat = false; // used to seed rate array so we startup with reasonable BPM

void interruptSetup(){

// Initializes Timer2 to throw an interrupt every 2mS.

TCCR2A = 0x02; // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO CTC MODE

TCCR2B = 0x06; // DON'T FORCE COMPARE, 256 PRESCALER

OCR2A = 0X7C; // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE RATE

TIMSK2 = 0x02; // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND OCR2A

sei(); // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED

}

// THIS IS THE TIMER 2 INTERRUPT SERVICE ROUTINE.

// Timer 2 makes sure that we take a reading every 2 miliseconds

ISR(TIMER2\_COMPA\_vect){ // triggered when Timer2 counts to 124

cli(); // disable interrupts while we do this

Signal = analogRead(pulsePin); // read the Pulse Sensor

sampleCounter += 2; // keep track of the time in mS with this variable

int N = sampleCounter - lastBeatTime; // monitor the time since the last beat to avoid noise

// find the peak and trough of the pulse wave

if(Signal < thresh && N > (IBI/5)\*3){ // avoid dichrotic noise by waiting 3/5 of last IBI

if (Signal < T){ // T is the trough

T = Signal; // keep track of lowest point in pulse wave

}

}

if(Signal > thresh && Signal > P){ // thresh condition helps avoid noise

P = Signal; // P is the peak

} // keep track of highest point in pulse wave

// NOW IT'S TIME TO LOOK FOR THE HEART BEAT

// signal surges up in value every time there is a pulse

if (N > 250){ // avoid high frequency noise

if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)\*3) ){

Pulse = true; // set the Pulse flag when we think there is a pulse

digitalWrite(blinkPin,HIGH); // turn on pin 13 LED

IBI = sampleCounter - lastBeatTime; // measure time between beats in mS

lastBeatTime = sampleCounter; // keep track of time for next pulse

if(secondBeat){ // if this is the second beat, if secondBeat == TRUE

secondBeat = false; // clear secondBeat flag

for(int i=0; i<=9; i++){ // seed the running total to get a realisitic BPM at startup

rate[i] = IBI;

}

}

if(firstBeat){ // if it's the first time we found a beat, if firstBeat == TRUE

firstBeat = false; // clear firstBeat flag

secondBeat = true; // set the second beat flag

sei(); // enable interrupts again

return; // IBI value is unreliable so discard it

}

// keep a running total of the last 10 IBI values

word runningTotal = 0; // clear the runningTotal variable

for(int i=0; i<=8; i++){ // shift data in the rate array

rate[i] = rate[i+1]; // and drop the oldest IBI value

runningTotal += rate[i]; // add up the 9 oldest IBI values

}

rate[9] = IBI; // add the latest IBI to the rate array

runningTotal += rate[9]; // add the latest IBI to runningTotal

runningTotal /= 10; // average the last 10 IBI values

BPM = 60000/runningTotal; // how many beats can fit into a minute? that's BPM!

QS = true; // set Quantified Self flag

// QS FLAG IS NOT CLEARED INSIDE THIS ISR

}

}

if (Signal < thresh && Pulse == true){ // when the values are going down, the beat is over

digitalWrite(blinkPin,LOW); // turn off pin 13 LED

Pulse = false; // reset the Pulse flag so we can do it again

amp = P - T; // get amplitude of the pulse wave

thresh = amp/2 + T; // set thresh at 50% of the amplitude

P = thresh; // reset these for next time

T = thresh;

}

if (N > 2500){ // if 2.5 seconds go by without a beat

thresh = 512; // set thresh default

P = 512; // set P default

T = 512; // set T default

lastBeatTime = sampleCounter; // bring the lastBeatTime up to date

firstBeat = true; // set these to avoid noise

secondBeat = false; // when we get the heartbeat back

}

sei();

// enable interrupts when youre done!

}// end isr