#include <TinyGPS.h>

#include <Wire.h>

int button1 = 10;

int button2 = 11;

#define led 13

void received\_sms(void);

int k=0;

float lat2,lon2;

void distance(void);

int button = 12;

int HMC6352Address = 0x42;

int slaveAddress;

byte headingData[2];

int i, headingValue;

// tiny gps library code

float flat, flon;

float heading=0;

int headinggps;

TinyGPS gps;

int x4=0;

void gpsdump(TinyGPS &gps);

bool feedgps();

void setup()

{

Serial2.begin(9600); // connect serial

Serial1.begin(4800);

Serial.begin(9600);

pinMode (button,INPUT);

pinMode (button1,OUTPUT);

pinMode (button2,OUTPUT);

}

void loop()

{

bool newdata = false;

if(k=0){

received\_sms();

}

unsigned long start = millis();

while (millis() - start < 250) // get new data every 1/4 of a second

{

if (feedgps())

newdata = true;

}

if (newdata)

{

gpsdump(gps);

}

}

void gpsdump(TinyGPS &gps)

{

float lat, lon;

unsigned long age, date, time, chars;

unsigned short sentences, failed;

feedgps(); // If we don't feed the gps during this long routine, we may drop characters and get checksum errors

gps.f\_get\_position(&lat, &lon, &age);

feedgps();

flat=lat;

flon=lon;

feedgps();

headinggps= gps.course();

feedgps();

gps.stats(&chars, &sentences, &failed);

if(digitalRead(button) == LOW)

{

Serial2.println("AT");

delay(1000);

Serial2.println("AT+CMGS=\"+923224541334\""); //sending sms number

delay(1000);

Serial2.print(lat,14);Serial2.print(",");// print latitude

Serial2.println(lon,14); // print longitude

delay(1000);

Serial2.write(byte(26));

delay(3000);

}

distance();

}

bool feedgps()

{

while (Serial1.available())

{

if (gps.encode(Serial1.read()))

return true;

}

return false;

}

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

void received\_sms(void)

{

int h,n,f;

char inByte;

char arr[100]={0};

char lat3[50]={0};

char lon3[50]={0};

k=1;

Serial.print( " m in function ");

/\* do{

while ( !Serial2.available() );

inByte = Serial2.read();

}while(inByte != 'K');

while ( !Serial2.available() );

inByte = Serial2.read();

while ( !Serial2.available() );

inByte = Serial2.read();\*/

for(int z=0; z<4; z++)

{

do

{

while ( !Serial2.available() );

} while ( ',' != Serial2.read() );

Serial.print("m in do while");

}

do

{

while ( !Serial2.available() );

} while ( '\r' != Serial2.read() );

Serial.print(" I have detected quotes ");

/\*while ( !Serial2.available() );

inByte = Serial2.read();

while ( !Serial2.available() );

inByte = Serial2.read();\*/

int y=0;

do

{

while ( !Serial2.available() );

inByte = Serial2.read();

arr[y]=inByte;

Serial.write( inByte );

y++;

}while(inByte != '\r');

int c=0;

Serial.print("lat=");

for(int a=1;a<18;a++)

{

lat3[c]= arr[a];

Serial.write( lat3[c] );

c++;

}

int b=1;

Serial.print("lon=");

for(int a=19;a<36;a++)

{

lon3[b]= arr[a];

Serial.write( lon3[b] );

b++;

}

lat2 = atof(lat3);

lon2= atof(lon3);

}

void distance(){

float x2lat = lat2; // setting x2lat and x2lon equal to our first waypoint

float x2lon = lon2;

float flat1=flat;

float flon1=flon;

float dist\_calc=0;

float dist\_calc2=0;

float diflat=0;

float diflon=0;

//------------------------------------------ distance formula below. Calculates distance from current location to waypoint

diflat=radians(x2lat-flat1); //notice it must be done in radians

flat1=radians(flat1);

x2lat=radians(x2lat);

diflon=radians((x2lon)-(flon1));

dist\_calc = (sin(diflat/2.0)\*sin(diflat/2.0));

dist\_calc2= cos(flat1);

dist\_calc2\*=cos(x2lat);

dist\_calc2\*=sin(diflon/2.0);

dist\_calc2\*=sin(diflon/2.0);

dist\_calc +=dist\_calc2;

dist\_calc=(2\*atan2(sqrt(dist\_calc),sqrt(1.0-dist\_calc)));

dist\_calc\*=6371000.0; //Converting to meters

//Serial.println(dist\_calc);

if(dist\_calc<1){ // are we within a meter of the waypoint?

done();

}

//-----------------------------------------heading formula below. Calculates heading to the waypoint from the current locaiton

flon1 = radians(flon1); //also must be done in radians

x2lon = radians(x2lon);

float heading=0;

heading = atan2(sin(x2lon-flon1)\*cos(x2lat),cos(flat1)\*sin(x2lat)-sin(flat1)\*cos(x2lat)\*cos(x2lon-flon1)),2\*3.1415926535;

heading = heading\*180/3.1415926535; // convert from radians to degrees

int head =heading;

if(head<0){

heading+=360; //if the heading is negative then add 360 to make it positive

}

//Serial.println("heading:");

//Serial.println(heading);

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

int x4=0;

int turn=0;

headinggps\*=10E-3;

x4=headinggps-heading; //getting the difference of our current heading to our needed heading

// if(heading>headinggps+90){

// pd=2;

//}

//if(heading<headinggps-90){

// pd=2;

//}

//-------------------------------------- below tells us which way we need to turn

if(x4>=-180){

if(x4<=0){

turn=8; //set turn =8 which means "right"

}

}

if(x4<-180){

turn=5; //set turn = 5 which means "left"

}

if(x4>=0){

if(x4<180){

turn=5; //set turn = 5 which means "left"

}

}

if(x4>=180){ //set turn =8 which means "right"

turn=8;

}

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

float decheading=headinggps;

float hd = gps.course()\*10E-3;

if(hd>heading-2){

if(hd<heading+2){ // if we are close to the needed heading

turn=3; //then set turn = 3 meaning go "straight"

}

}

if(turn==3){

Serial.println("straight");

digitalWrite(button1, HIGH);

digitalWrite(button2, HIGH);

delay(150);

digitalWrite(button1, LOW);

digitalWrite(button2, LOW);

delay(700);

//enter straight code here

}

if(turn==8){

Serial.println("right");

rightturn();

}

if(turn==5){

Serial.println("left");

leftturn();

}

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

}

void done(){

done();

}

void rightturn()

{

unsigned long age;

if(headinggps+2>heading){

if(headinggps-2<heading){ // Go straight here

Serial.println("straight");

digitalWrite(button1, HIGH);

digitalWrite(button2, HIGH);

delay(150);

digitalWrite(button2, LOW);

digitalWrite(button1, LOW);

delay(700);

return;

}

}

digitalWrite(button, HIGH);

digitalWrite(button1, HIGH);

delay(200);

digitalWrite(button1, LOW);

delay(50);

digitalWrite(button, LOW);

delay(800);

//------------------------------------------------------Calculate heading and compare again

feedgps();

headinggps= gps.course();

feedgps(); // If we don't feed the gps during this long routine, we may drop characters and get checksum errors

gps.f\_get\_position(&flat, &flon, &age);

float x2lat = lat2; // setting x2lat and x2lon equal to our first waypoint

float x2lon = lon2;

float flat1=flat;

float flon1=flon;

flon1 = radians(flon1); //also must be done in radians

x2lon = radians(x2lon);

flat1=radians(flat1);

x2lat=radians(x2lat);

float heading=0;

heading = atan2(sin(x2lon-flon1)\*cos(x2lat),cos(flat1)\*sin(x2lat)-sin(flat1)\*cos(x2lat)\*cos(x2lon-flon1)),2\*3.1415926535;

heading = heading\*180/3.1415926535; // convert from radians to degrees

int head =heading;

if(head<0){

heading+=360; //if the heading is negative then add 360 to make it positive

int x4=0;

int turn=0;

headinggps\*=10E-3;

x4=headinggps-heading; //getting the difference of our current heading to our needed heading

if(x4<-180){

return; //set turn = 5 which means "left"

}

if(x4>=0){

if(x4<180){

return; //set turn = 5 which means "left"

}

}

rightturn();

}

}

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

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

//----------------------------------------------left turning

void leftturn(){

unsigned long age;

if(headinggps+2>heading){

if(headinggps-2<heading){ // Go straight here

Serial.println("straight");

digitalWrite(button1, HIGH);

digitalWrite(button2, HIGH);

delay(150);

digitalWrite(button2, LOW);

digitalWrite(button1, LOW);

delay(700);

return;

}

}

digitalWrite(button1, HIGH);

digitalWrite(button, HIGH);

delay(200);

digitalWrite(button, LOW);

delay(50);

digitalWrite(button1, LOW);

delay(800);

//------------------------------------------------------Calculate heading and compare again

feedgps();

headinggps= gps.course();

feedgps(); // If we don't feed the gps during this long routine, we may drop characters and get checksum errors

gps.f\_get\_position(&flat, &flon, &age);

float x2lat = lat2; // setting x2lat and x2lon equal to our first waypoint

float x2lon = lon2;

float flat1=flat;

float flon1=flon;

flon1 = radians(flon1); //also must be done in radians

x2lon = radians(x2lon);

flat1=radians(flat1);

x2lat=radians(x2lat);

float heading=0;

heading = atan2(sin(x2lon-flon1)\*cos(x2lat),cos(flat1)\*sin(x2lat)-sin(flat1)\*cos(x2lat)\*cos(x2lon-flon1)),2\*3.1415926535;

heading = heading\*180/3.1415926535; // convert from radians to degrees

int head =heading;

if(head<0){

heading+=360; //if the heading is negative then add 360 to make it positive

int x4=0;

int turn=0;

headinggps\*=10E-3;

x4=headinggps-heading; //getting the difference of our current heading to our needed heading

if(x4>=180){ //set turn =8 which means "right"

return;

}

if(x4>=-180){

if(x4<=0){

return; //set turn =8 which means "right"

}

}

leftturn();

}

}