/\*

Reverse Geocache

This program unlocks a box that has reached a certain location.

by Stephen O'Gara December 2014

\*/

// installed in Arduino software Library folder

#include <math.h>

#include <LiquidCrystal.h>

#include <Adafruit\_GPS.h>

#include <SoftwareSerial.h>

SoftwareSerial mySerial(3,2); // ultimate gps from adafruit rx on pin 2, and tx on pin 3

Adafruit\_GPS GPS(&mySerial); // defines gps unit version 3 from adafruit

// DEBUG GPS switch

#define GPSECHO false //make true to debug GPS

//#define GPSECHO true

boolean usingInterrupt = false;

void useInterrupt(boolean);

//Servo

#include <PWMServo.h>

PWMServo servoLatch;

//Declarations

const float deg2rad = 0.01745329251994;

const float rEarth = 3958.75; //can replace with 3958.75 mi, 6370.0 km, or 3440.06 NM

float range = 3000; // distance from HERE to THERE

String here; // read from GPS

// PIN ASSIGNMENTS

/\* lcd pins to arduino pins (lcd pin# - arduino pin#)

1-gnd, 2-5v, 3-x, 4-12, 5-gnd, 6-11, 7-x, 8-x, 9-, 10-x, 11-8, 12-7, 13-6, 14-5, 15-5v, 16-gnd

\*/

LiquidCrystal lcd(12, 11, 8, 7, 6, 5); // LCD (rs,e,D4,D5,D6,D7} tie RW to gnd

int gpsWasFixed = HIGH; // did the GPS have a fix?

int ledFix = 13; // pin for fix LED

int ledpin = 4; // button power ring

int servoPin = 9; // pin for servo

int servoLock = 40; // angle (deg) of "locked" servo was 45 35 worked ok

int servoUnlock = 3; // angle (deg) of "unlocked" servo. zero causes binding.

/\*

===================================================================

example of desired location syntax for below

String there = "N34 08.902, W118 44.966";

Make sure you use the same syntax and number of characters

==================================================================

\*/

String there = "N34 08.902, W118 44.966"; // ENTER DESTINATION HERE

/\*

==================================================================

-some of my test coords-

far away N34 48.902, W118 04.966

near N34 08.902 W118 44.966

michaels N34°09.280 W118°47.645 2.57 miles or 4152 meters

======================================================================

\*/

int a;

int b;

void setup()

{

// set pin modes

pinMode(ledpin, OUTPUT);

// set starting pin positions

digitalWrite(ledpin, HIGH); // turn on button ring led

// set servo into locked position

servoLatch.attach(SERVO\_PIN\_A);

servoLatch.write(servoLock);

delay(50);

lcd.begin(16, 2);

Serial.begin(115200);

Serial.println("Debug GPS Test:");

GPS.begin(9600);

GPS.sendCommand(PMTK\_SET\_NMEA\_OUTPUT\_RMCGGA); // RMC (recommended minimum) and GGA (fix data) including altitude

GPS.sendCommand(PMTK\_SET\_NMEA\_UPDATE\_1HZ); // 1 Hz update rate

useInterrupt(true); // reads the steaming data in a background

delay(1000);

}

void loop()

{

if (GPS.newNMEAreceived()) { // Parse GPS and recalculate RANGE

if (!GPS.parse(GPS.lastNMEA())) // 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 (GPS.fix)

{

gpsWasFixed = HIGH; // when a fix is attained, calculate variable - range

digitalWrite(ledFix, HIGH);

here = gps2string ((String) GPS.lat, GPS.latitude, (String) GPS.lon, GPS.longitude);

range = haversine(string2lat(here), string2lon(here), string2lat(there), string2lon(there));

Serial.print(" Here: "); //for GPS debug

Serial.print(here);

Serial.print(" There: ");

Serial.println(there);

Serial.print(" Range: ");

Serial.print(range);

Serial.println(" Miles ");

delay(50); // pause the processor to get a clean screen display, no glitching

lcd.clear(); //write range distance to go on LCD screen

lcd.setCursor(0,0);

lcd.print("Distance to go");

lcd.setCursor(0,1);

if (range < .2)

{

//lcd.print(range \* 5280);

a=(int)(range \* 5280);

b=round(a);

lcd.print(b);

Serial.print("a=");

Serial.println(a);

Serial.print("b=");

Serial.println(b);

Serial.println();

lcd.print(" Feet");

delay(500);

}

else

{

Serial.println("range is over .2 miles away ");

lcd.print(range);

lcd.print(" Miles");

delay(500); // allows lcd to be clear and not blinky

}

}

else { //No GPS fix- take box outside

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Hello...");

lcd.setCursor(0,1);

lcd.print("Take me outside!");

delay(200);

}

//================= winning process ==================

if (range < .0189) // set winning distance in miles

{ // .0379 is 200 feet

servoLatch.write(servoUnlock); // open box latch

delay(50);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("You Win! ");

lcd.setCursor(0,1);

lcd.print("Box is unlocked"); // leave box unlocked

digitalWrite(ledpin, LOW); // blink button ring to indicate winning

delay(300); // 10 seconds of blinking

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

digitalWrite(ledpin, LOW);

delay(300);

digitalWrite(ledpin, HIGH);

delay(300);

}

}

// TO LOCK BOX - Move away from the winning coordinates and turn on box

// When the box is shut, you can power it up, closing the servo lock

// ==========GPS calculation stuff ===========================

SIGNAL(TIMER0\_COMPA\_vect) {

// Interrupt is called once a millisecond, looks for any new GPS data, and stores it

char c = GPS.read();

if (GPSECHO)

if (c) UDR0 = c;

}

void useInterrupt(boolean v) {

if (v) {

OCR0A = 0xAF;

TIMSK0 |= \_BV(OCIE0A);

usingInterrupt = true;

} else {

TIMSK0 &= ~\_BV(OCIE0A);

usingInterrupt = false;

}

}

String int2fw (int x, int n) {

// returns a string of length n (fixed-width)

String s = (String) x;

while (s.length() < n) {

s = "0" + s;

}

return s;

}

String gps2string (String lat, float latitude, String lon, float longitude) {

// returns "Ndd mm.mmm, Wddd mm.mmm";

int dd = (int) latitude/100;

int mm = (int) latitude % 100;

int mmm = (int) round(1000 \* (latitude - floor(latitude)));

String gps2lat = lat + int2fw(dd, 2) + " " + int2fw(mm, 2) + "." + int2fw(mmm, 3);

dd = (int) longitude/100;

mm = (int) longitude % 100;

mmm = (int) round(1000 \* (longitude - floor(longitude)));

String gps2lon = lon + int2fw(dd, 3) + " " + int2fw(mm, 2) + "." + int2fw(mmm, 3);

String myString = gps2lat + ", " + gps2lon;

return myString;

};

/\*

float string2radius (String myString) {

// returns a floating-point number: e.g. String myString = "Radius: 005.1 NM";

float r = ((myString.charAt(8) - '0') \* 100.0) + ((myString.charAt(9) - '0') \* 10.0) + ((myString.charAt(10) - '0') \* 1.0) + ((myString.charAt(12) - '0') \* 0.10);

return r;

};\*/

float string2lat (String myString) {

// returns radians: e.g. String myString = "N38 58.892, W076 29.177";

float lat = ((myString.charAt(1) - '0') \* 10.0) + (myString.charAt(2) - '0') \* 1.0 + ((myString.charAt(4) - '0') / 6.0) + ((myString.charAt(5) - '0') / 60.0) + ((myString.charAt(7) - '0') / 600.0) + ((myString.charAt(8) - '0') / 6000.0) + ((myString.charAt(9) - '0') / 60000.0);

//Serial.print("float lat: ");

//Serial.println(lat);

lat \*= deg2rad;

if (myString.charAt(0) == 'S')

lat \*= -1; // Correct for hemisphere

return lat;

};

float string2lon (String myString) {

// returns radians: e.g. String myString = "N38 58.892, W076 29.177";

float lon = ((myString.charAt(13) - '0') \* 100.0) + ((myString.charAt(14) - '0') \* 10.0) + (myString.charAt(15) - '0') \* 1.0 + ((myString.charAt(17) - '0') / 6.0) + ((myString.charAt(18) - '0') / 60.0) + ((myString.charAt(20) - '0') / 600.0) + ((myString.charAt(21) - '0') / 6000.0) + ((myString.charAt(22) - '0') / 60000.0);

//Serial.print("float lon: ");

//Serial.println(lon);

lon \*= deg2rad;

if (myString.charAt(12) == 'W')

lon \*= -1; // Correct for hemisphere

return lon;

};

float haversine (float lat1, float lon1, float lat2, float lon2) {

// returns the great-circle distance between two points (radians) on a sphere

float h = sq((sin((lat1 - lat2) / 2.0))) + (cos(lat1) \* cos(lat2) \* sq((sin((lon1 - lon2) / 2.0))));

float d = 2.0 \* rEarth \* asin (sqrt(h));

//Serial.println(d);

return d;

};