20160818 CompassGuidance

/\* Text moved to be bottom of the code in an attempt to drive people crazy.

\*

\*/

// This code works and gives degrees!!!

#include <Wire.h> // I2C Communications HMC5883 compass

#define Addr 0x1E // 7-bit address of HMC5883 compass

#include <Servo.h>

Servo myservo; // create servo object to control a servo

// twelve servo objects can be created on most boards

int pos = 0; // variable to store the servo position

float fHeading[9];

int readcount;

int displaycount;

float CurrentCourse = 180.0;

float error;

int Ierror;

int debug = 1;

int forwardReverse = 1; // 1 Forward 2 Reverse

float targetHeadingA = 170.0;

float targetHeadingB = 350.0;

float targetHeading;

void setup() // Built-in initialization block

{

Serial.begin(9600);

Wire.begin();

myservo.attach(9); // attaches the servo on pin 9 to the servo object

// checkinputs();

}

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

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

void loop() // Main loop auto-repeats

{

if (forwardReverse == 1)

{

targetHeading = targetHeadingA;

}

if (forwardReverse != 1)

{

targetHeading = targetHeadingB;

}

if (debug == 1)

{

// getRawSensor(10);

getcompass();

runGuidance();

}

}

void getcompass()

{

readcount = readcount + 1;

if (readcount > 2)

{

readcount = 0;

}

displaycount = displaycount + 1;

if (displaycount > 500)

{

displaycount = 0;

}

int x, y, z;

// Initiate communications with compass

Wire.beginTransmission(Addr);

Wire.write(byte(0x03)); // Send request to X MSB register

Wire.endTransmission();

Wire.requestFrom(Addr, 6); // Request 6 bytes; 2 bytes per axis

if(Wire.available() <=6) { // If 6 bytes available

x = Wire.read() << 8 | Wire.read();

z = Wire.read() << 8 | Wire.read();

y = Wire.read() << 8 | Wire.read();

}

// If compass module lies flat on the ground with no tilt,

// just x and y are needed for calculation

float heading=atan2(x, y)/0.0174532925;

if(heading < 0) heading+=360;

heading=360-heading; // N=0/360, E=90, S=180, W=270

fHeading[readcount] = heading;

heading = fHeading[0] + fHeading[2] + fHeading[3] + fHeading[4];

heading = heading + fHeading[5] + fHeading[6] + fHeading[7] + fHeading[8];

heading = heading + fHeading[5];

heading = heading/2 ;

if (displaycount <=0)

{

Serial.println(heading);

}

CurrentCourse = heading;

delay(1);

}

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

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

void runGuidance()

{

// digitalWrite(2, LOW); //

// digitalWrite(3, LOW); //

// digitalWrite(5, LOW); //

// digitalWrite(6, LOW); //

// digitalWrite(7, LOW); //

// digitalWrite(8, LOW); //

Serial.println(" ");

// Serial.print("------------------------------------------ ");

Serial.print(" TARGET>>> ");

Serial.print(targetHeading);

Serial.print(" <<< "); // 640------------------0----------------1360

Serial.print(" COURSE> ");

Serial.print(CurrentCourse);

Serial.print(" < "); // 640------------------0----------------1360

error = targetHeading - CurrentCourse;

// mapping to -179 .. 180

if (error <= -180.0) {error = 360.0 - error;}

if (error > 180.0) {error = -360.0 + error ; }

if (error <= 0.0) // (Turn LEFT)

{

left();

Serial.print(" Right Error: ");

Serial.print(error);

Serial.print(" "); //

}

if (error > 0.0) // RIGHT (TURN RIGHT)

{

right();

Serial.print(" Left Error: ");

Serial.print(error);

Serial.print(" "); // 640------------------0----------------1360

}

Serial.print(" Servo: ");

Serial.print(pos);

Serial.print(" "); //

myservo.write(pos); // tell servo to go to position in variable 'pos'

delay(150);

}

void right() //rrrrrrrr

{

pos = targetHeading - 20;

if (pos < 0)

{

pos = pos + 360;

}

}

void left() //lllllllll

{

pos = targetHeading + 20;

if (pos > 360)

{

pos = pos - 360;

}

}