FINAL PROJECT REPORT Dual Axis Solar Tracker

EMT 2461 Electromechanical Systems Software Interface Section D484, Instructor: Dr. Xiaohai Li

Eric Chan and Jorge Martinez

Submission Date: May 28th, 2017

Contents

[Detailed Description 2](#_Toc483694328)

[Parts 2](#_Toc483694329)

[Wiring Schematic 3](#_Toc483694330)

[Component Design 3](#_Toc483694331)

[System Design 4](#_Toc483694332)

[Potential Modifications 5](#_Toc483694333)

[Testing Procedure and Analysis 5](#_Toc483694334)

[Conclusion 5](#_Toc483694335)

[Appendix 6](#_Toc483694336)

[References 9](#_Toc483694337)

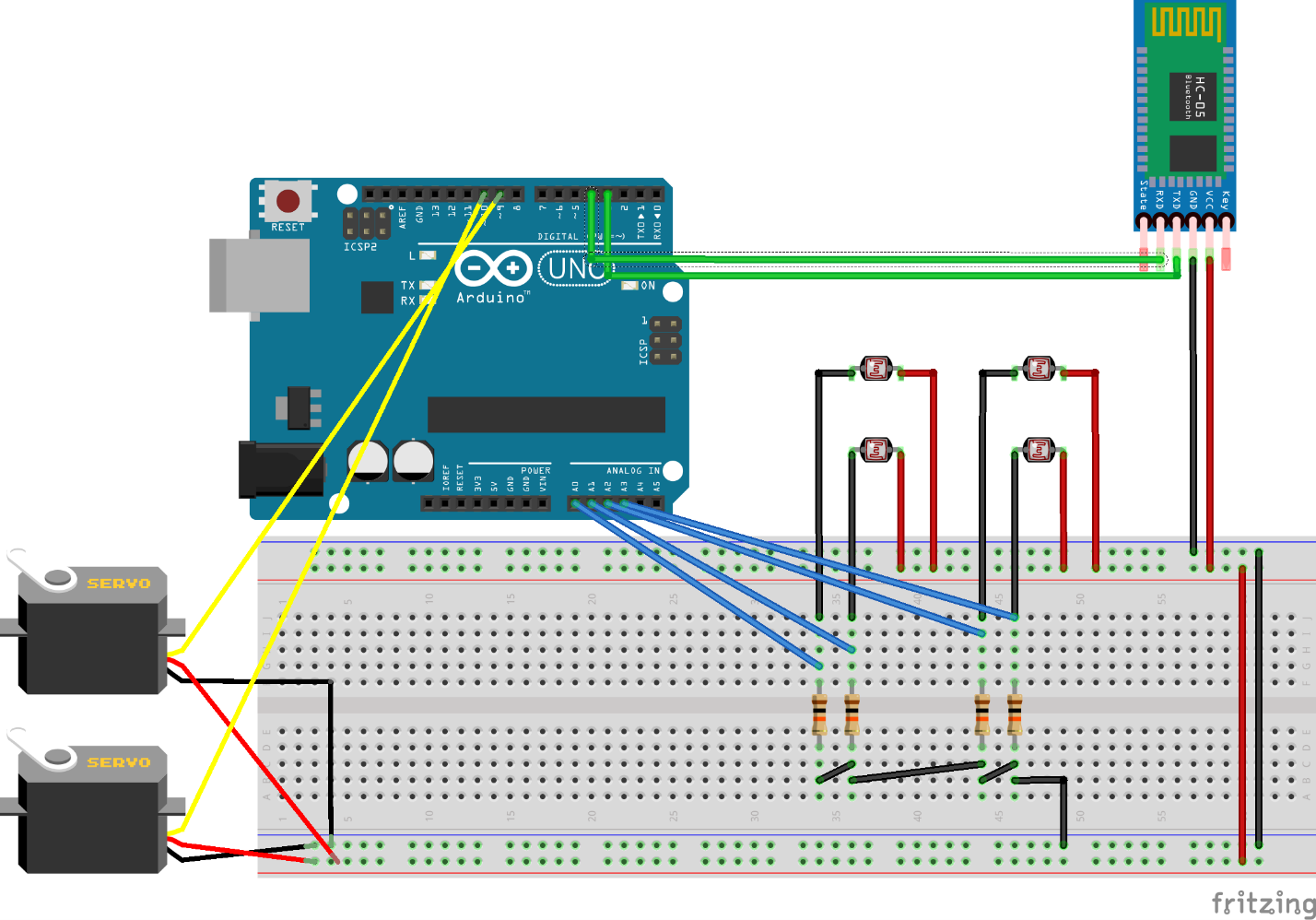
# Detailed Description

We will be emulating a dual axis solar tracker by utilizing two servo motors to serve as the dual axis of rotations and installing four light-dependent resistors or LDRs in a square array to detect changes of intensity of any light source in each corner. We chose to use servo motors because of their precise angular control, their PWM control, and their range of motion of 0 degrees to 180 degrees. When light is detected by one or more of the LDRs, it will reorient itself based on the intensity of the light source. If the light source is stronger up top or down bottom, then the north – south servo motor will start and begin positioning moving north or south until all the LDRs share the approximate same intensity of light. This will be the same for the east-west servo motor, detecting any stronger light in either direction and moving the panel thusly until all LDRs are satisfied with the approximate same intensity of light. If the light happens to be stronger diagonally of the LDRs, it will execute a combination of positioning, utilizing both north-south and east-west servo motors until the condition is satisfied.

# Parts

* Arduino Uno
* USB A to USB B cable
* A computer with a USB port
* Four light dependent resistors
* Two 6 V 200 mA Solar panels
* Two servo motors
* Bluetooth HC-06 module
* Four 330 Ohm resistors
* Voltage sensor
* Wires
* Breadboard

# Wiring Schematic



Here, all the required motors, sensors, and some of the peripherals for the dual axis solar tracker are connected. The servo motors are connected to the PWM pins with a potentiometer each connected to their own motor. The Bluetooth module is connected to connected to the transmit and receive pins respectively. The light dependent resistors are with their own resistor.

# Component Design

One of the major components in our project were the servo motors, we had to make sure both are supplied with 5 volts of electricity, as if we connect them directly to the Uno we might run the risk of drawing too much current. All the electricity was provided through a laptop’s 5 volt USB port for the Uno and the rest of the components and a separate breadboard power supply just for the servo motors since they draw the most power out of the entire system. In programming, we had to create functions that separated the delays for the reading of the voltage from the voltage sensor and the movement of the solar tracker, as if we increase the delay, the solar tracker will move too slow or not at all since it needs a constant refresh from the short delay and if we decrease the delay, the serial monitor and Bluetooth will bombard and spam it with messages of its readings.

# System Design

C:\Users\Eric Chan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\EMT 2461 Project Proposal Block Diagram.png

From this system diagram, there are eight possible processes that happen from the start, which is when the solar tracker detects no differentiable light source. Four processes are for when either two adjacent LDRs detects similar amounts of strong light and four processes are when one of each LDRs detects a strong amount of light. Each process is denoted with differently colored arrows.

# Potential Modifications

* Add more photo resistors for more accuracy
* Change resistor values to increase or decrease sensitivity
* Have the photoresistors more spread-out
* Control speed and direction of the tracker using Bluetooth app
* Increase solar panel voltage/current

# Testing Procedure and Analysis

In testing the project, we encountered many trials and errors, one huge part was the servo motors. We had to replace servo motors when we noticed that their transition wasn’t smooth or it made clicking and/or ratcheting sounds when it moved. Another problem we encountered was the placement of the pins for our voltage sensor and other components, sometimes we would place the pins in the wrong position, for example putting a wire to a digital # pin instead of an analog # pin, and it would produce weird results that wouldn’t match theoretically during our testing. Bluetooth was especially a complete mystery as well, using the default 0 and 1 digital pins on the Uno produced results but with numbers accompanying the messages sent, so we had to use the Serial Software Bluetooth alternative for to clear the numbers.

# Conclusion

The advantages of a solar tracker far exceed the disadvantages of one, as the energy needed to move the solar panel is surpassed by the efficiency of the tracker being able to seize direct sunlight from sunrise till sunset. Solar trackers are already implemented all around the world where there’s flat, sunny areas like China, India, and Spain.

# Appendix

#include <Servo.h> // include Servo library

// 180 horizontal MAX

Servo horizontal; // horizontal servo

int servoh = 180; // 90; // stand horizontal servo

int servohLimitHigh = 180;

int servohLimitLow = 65;

// 65 degrees MAX

Servo vertical; // vertical servo

int servov = 45; // 90; // stand vertical servo

int servovLimitHigh = 80;

int servovLimitLow = 15;

// LDR pin connections

// name = analogpin;

int ldrlt = A0; //LDR top left - BOTTOM LEFT

int ldrrt = A1; //LDR top rigt - BOTTOM RIGHT

int ldrld = A2; //LDR down left - TOP LEFT

int ldrrd = A3; //ldr down rigt - TOP RIGHT

// VOLTAGE SENSOR DECLARATIONS

int analogInput = A5;

float vout = 0.0;

float vin = 0.0;

float R1 = 30000.0; //

float R2 = 7500.0; //

int value = 0;

// Bluetooth Declarations

#include <SoftwareSerial.h> //To use the SoftwareSerial libary

SoftwareSerial myBluetooth(4, 5); //Define a Bluetooth software serial that use

// Arduino’s port4 as rxPin and port5 as txPin, i.e., connects

// Bluetooth’s TX pin to Arduino port4, RX pin to Arduino port5.

//Timers

int dtime1 = 1000;

int dtime2 = 50;

void setup()

{

Serial.begin(9600);

// \*\*\*BLUETOOTH\*\*\*\*\*\*

Serial.println("Hello, Serial Monitor is activated!");

pinMode(4, INPUT); //Set Arduino’s port4 as INPUT

pinMode(5, OUTPUT); //Set Arduino’s port5 as OUTPUT

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*analogRead SETUPS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pinMode(A0, INPUT);

pinMode(A1, INPUT);

pinMode(A2, INPUT);

pinMode(A3, INPUT);

pinMode(A4, OUTPUT);

myBluetooth.begin(9600); // Set Bluetooth baud rate. Check the datasheet!!

myBluetooth.println("Hello, Bluetooth is activated!");

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

// servo connections

// name.attacht(pin);

horizontal.attach(9);

vertical.attach(10);

horizontal.write(180);

vertical.write(45);

}

void loop()

{

timers();

}

void loop1()

{

// read the value at analog input

value = analogRead(analogInput);

vout = (value \* 5.0) / 1024.0; // see text

vin = (vout / (R2 / (R1 + R2))) / 2;

Serial.print("The precise voltage from the light is: ");

Serial.println(vin, 2);

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BLUETOORTH\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

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

if (myBluetooth.available()) //Detect any data being received by Bluetooth

//Use “read()” to read in data received by Bluetooth RX pin, then use

//Use Serial.println() to display the data on the Serial Monitor

Serial.write(myBluetooth.read());

if (Serial.available()) //Ready to send data from Serial Monitor

//Read the data typed in Serial Monitor, then send to Bluetooth TX pin.

myBluetooth.write(Serial.read());

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

}

void loop2()

{

int lt = analogRead(ldrlt); // top left

int rt = analogRead(ldrrt); // top right

int ld = analogRead(ldrld); // down left

int rd = analogRead(ldrrd); // down rigt

// int dtime = analogRead(4)/20; // read potentiometers

// int tol = analogRead(5)/4;

int tol = 50;

int avt = (lt + rt) / 2; // average value top

int avd = (ld + rd) / 2; // average value down

int avl = (lt + ld) / 2; // average value left

int avr = (rt + rd) / 2; // average value right

int dvert = avt - avd; // check the diffirence of up and down

int dhoriz = avl - avr;// check the diffirence og left and rigt

if (-1 \* tol > dvert || dvert > tol) // check if the diffirence is in the tolerance else change vertical angle

{

if (avt > avd)

{

servov = ++servov;

if (servov > servovLimitHigh)

{

servov = servovLimitHigh;

}

}

else if (avt < avd)

{

servov = --servov;

if (servov < servovLimitLow)

{

servov = servovLimitLow;

}

}

vertical.write(servov);

}

if (-1 \* tol > dhoriz || dhoriz > tol) // check if the diffirence is in the tolerance else change horizontal angle

{

if (avl > avr)

{

servoh = --servoh;

if (servoh < servohLimitLow)

{

servoh = servohLimitLow;

}

}

else if (avl < avr)

{

servoh = ++servoh;

if (servoh > servohLimitHigh)

{

servoh = servohLimitHigh;

}

}

else if (avl = avr)

{

// nothing

}

horizontal.write(servoh);

}

}

void timers()

{

static long prev = 0; // Creates static variable that remains in memory.

long now = millis();

short diff = now - prev; // Finds the difference between the last time it ran and the now time.

static short lp1t = 0;

lp1t += diff;

if (lp1t > dtime1)

{

loop1();

lp1t -= dtime1; // Loop timer is reset back to its original value.

}

static short lp2t = 0;

lp2t += diff;

if (lp2t > dtime2)

{

loop2();

lp2t -= dtime2; // Loop timer is reset back to its original value.

}

prev = now;

}

# References

<http://www.elegoo.com/>

<https://www.allearthrenewables.com/product/dual-axis-solar-tracker-advantages>

<http://www.solarpowerworldonline.com/2016/05/advantages-disadvantages-solar-tracker-system/>

<http://fritzing.org/home/>

<https://en.wikipedia.org/wiki/Solar_tracker>