Alieldin Ayman | Nahla Fawzy | Salma Yasser | Sama Foad

28/12/2016



Sun-Tracking Solar Cell

An electronics project

# **Motivation**

Energy sources have mostly become very expensive and not natural-friendly. That’s why we –a group of four students- have decided to invest in the development of an average-priced energy source that can generate enough power for simple electrical equipment, and can provide even higher values by charging batteries over extended periods of time.

# **Implementation**

So in order to implement those ideas into reality, we thought of using sunlight as our main power source, as it is considered the most “green” type of energy. We used an 18W Solar-cell which absorbs a rather large amount of sunlight, and added the sun-tracking functionality to make it even more efficient at the cost of a very low power to operate. The system is self-sustaining as the solar panel generates enough power to charge the circuit operating it.

# **Setup & Components Used**

* One **180°** Servo Motor
* One **360 °**Continuous Servo Motor
* Four Light-Emitting Diodes (LDR)
* Two Potentiometers (Variable Resistance)
* Arduino UNO Chip
* 18W Solar Panel
* 6V Battery (Two 3V Batteries in Series)
* Wooden-framed Body
* Epoxy (LOTS OF IT)

# **Circuit Diagram**

# 

# **\*Circuit drawn using Fritzing\***

# **Arduino Code**

#include<Servo.h>

Servo HORIZONTAL;

Servo VERTICAL1;

//Servo VERTICAL2;

int V1POS;

int DELAY = 250;

int TOL = 50;

//int V2POS = 60;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

HORIZONTAL.attach(9); //Horizontal 360 Servo

VERTICAL1.attach(10); // Vertical Servo

V1POS = 60;

VERTICAL1.write(V1POS);

delay(1000);

//VERTICAL2.attach(11); // Vertical Servo

}

void loop() {

// put your main code here, to run repeatedly:

int LDRTR = analogRead(0);

int LDRTL = analogRead(1);

int LDRBR = analogRead(2);

int LDRBL = analogRead(3);

int AVGT = (LDRTL + LDRTR) / 2;

int AVGB = (LDRBR + LDRBL) / 2;

int AVGR = (LDRTR + LDRBR) / 2;

int AVGL = (LDRBL + LDRTL) / 2;

int DIFFV = abs((AVGT - AVGB));

int DIFFH = abs((AVGR - AVGL));

/\*

\* Less than 92 will make it rotate clockwise

\* More than 92 will make it rotate anti-clockwise

\*/

if (DIFFH > TOL)

{

if (AVGR > AVGL)

{

HORIZONTAL.write(86);

}

if (AVGL > AVGR)

{

HORIZONTAL.write(98);

}

if (abs(AVGL - AVGR) <= 5) //<<<<-----------------------------VARIABLE

{

HORIZONTAL.write(92);

}

}

if (DIFFV > TOL)

{

if( V1POS >= 179)

{

V1POS=170;

}

else if(V1POS <= 1)

{

V1POS=10;

}

if (AVGT > AVGB)

{

V1POS++;

}

else if (AVGB > AVGT)

{

V1POS--;

}

VERTICAL1.write(V1POS);

}

Serial.print("Vertical Servo Position: ");

Serial.println(V1POS);

Serial.print("Top Right LDR Reading: ");

Serial.println(LDRTR);

Serial.print("Top Left LDR Reading: ");

Serial.println(LDRTL);

Serial.print("Bottom Right LDR Reading: ");

Serial.println(LDRBR);

Serial.print("Bottom Left LDR Reading: ");

Serial.println(LDRBL);

delay(DELAY);

}