NORTH SOUTH UNIVERSITY

Department of Electrical & Computer Engineering

EEE 299: Junior Design Project

Project Name: Sun Light Tracking Solar Panel with Real Time Power Logger

Group:01

Name	ID
Md.Rofiqul Islam	1521070643
Nuren Shams	1530784043
Abrar Fahim	1610522043

INTRODUCTION

- In the modern world of science and technology renewable and green energy is a buzz word that seems to be uttered constantly. One of the key source of renewable energy comes from harnessing the solar energy from the Sun light. For harnessing solar energy we use solar panels. Most of these solar panels that we used are mounted at a fixed angle. For this type of non -moveable solar panels the user can harness coral energy form the sun at a fixed period of day time . So, rest of the solar energy gets wasted. Thus, keeping this problem in mind we have designed a solution to this energy wastage problem. The solution of this problem is a rotating solar panel module that follows the sun light through the day using LDRs (Light Dependent Resistor) and servo motor as the feedback system, which rotates the module in a vertical and a horizontal direction. In this project we have impelled that solution and also added a measurement device with it which measures the voltage current and also the power delivered by the solar panel to a particular object.

Picture2 Source:https://www.directenergy.com/blog/solar-panels-help-hurt-resale-value-home/
Picture1 Source:https://www.slashgear.com/sharp-solar-cells-can-reach-60-percent-efficiency-20334561/

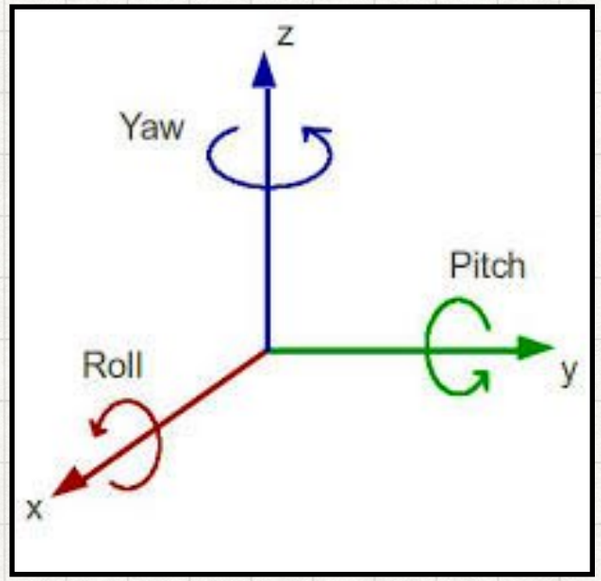
PROBLEM STATEMENT AND CHALLENGES TOWARDS ACCOMPLISHMENT

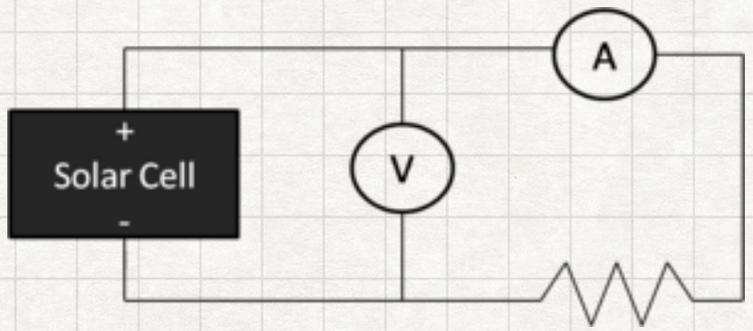
- The problem that we want to deal is energy wastage. Currently the solar panels that we are using is of very low efficiency which is about 15% to 17%. The max efficiency of solar panels that are being used is about 22.8% (which is SunPowers SPR X21-345 according to solarreviews.com).Thus thinking about this using a fixed angle non-moveable solar panel which waste a lot of solar energy can never be an option.

Now for the challenges that we have faced solving this problem are as follows:

1. Rotation of the solar panel: Which was done by using the LDRs and servos with the help of an Arduino Uno development board.

2. The second problem that we have faced building the project is measuring the current that is being supplied to the load by the solar panel. This problem was solved using an INA219 Bi-directional current sensing IC.





Picture 1 Source: https://www.researchgate.net/figure/Average-roll-pitch-and-yaw-angles_fig2_262055313
Picture 2 Source: http://photonicswiki.org/index.php?title=Photovoltaics-_Outreach_Kit

IMPLEMENTATION

THE ROTATING SOLAR PANEL MODULE

- **The project has basically two parts:**

1. **The Rotating Solar Panel module part/SOLAR TRACKER.**

2. **The Power Logger module part.**

Rotating Solar Panel Module/SOLAR TRACKER: For this part we have used an Arduino Uno as the brain of the module which actually processes all the data sent by the LDR panel mounted just above the Solar panel. Form this data the Arduino processes the information using a set of codes previously written for the development board. Then the processed information goes to the two servos responsible for the yaw and pitch of the solar panel. For generating the Arduino signal we have utilised the some of the in built registers and comparators of Arduino micro-controller ATmega 328/P. The name of the used registers and comparators are given below:

TC=Timer/Counter
TCCR1A = TC1 Control Register A (page 170 in the data sheet)
TCCR1B = TC1 Control Register B (page 173 in the data sheet)

WGM = Waveform Generation Mode (page 171 in the data sheet)
CTC = Clear Timer on Compare Match (CTC) Mode

COM1 = Compare Output Mode for Channel (page 170 in the data sheet)

ICR1L =Input Capture Register 1 Low byte (page 178 in the data sheet)

ICR1H = Input Capture Register 1 High byte (page 179 in the data sheet)

OCR1A L = Output Compare Register 1 A Low byte (page 180 in the data sheet)

OCR1A H = Output Compare Register 1 A High byte (page 181 in the data sheet)

OCR1B L = Output Compare Register 1 B Low byte (page 182 in the data sheet)

OCR1B H = Output Compare Register 1 B High byte (page 183 in the data sheet)

Output Compare pins (OC1A and OC1B)

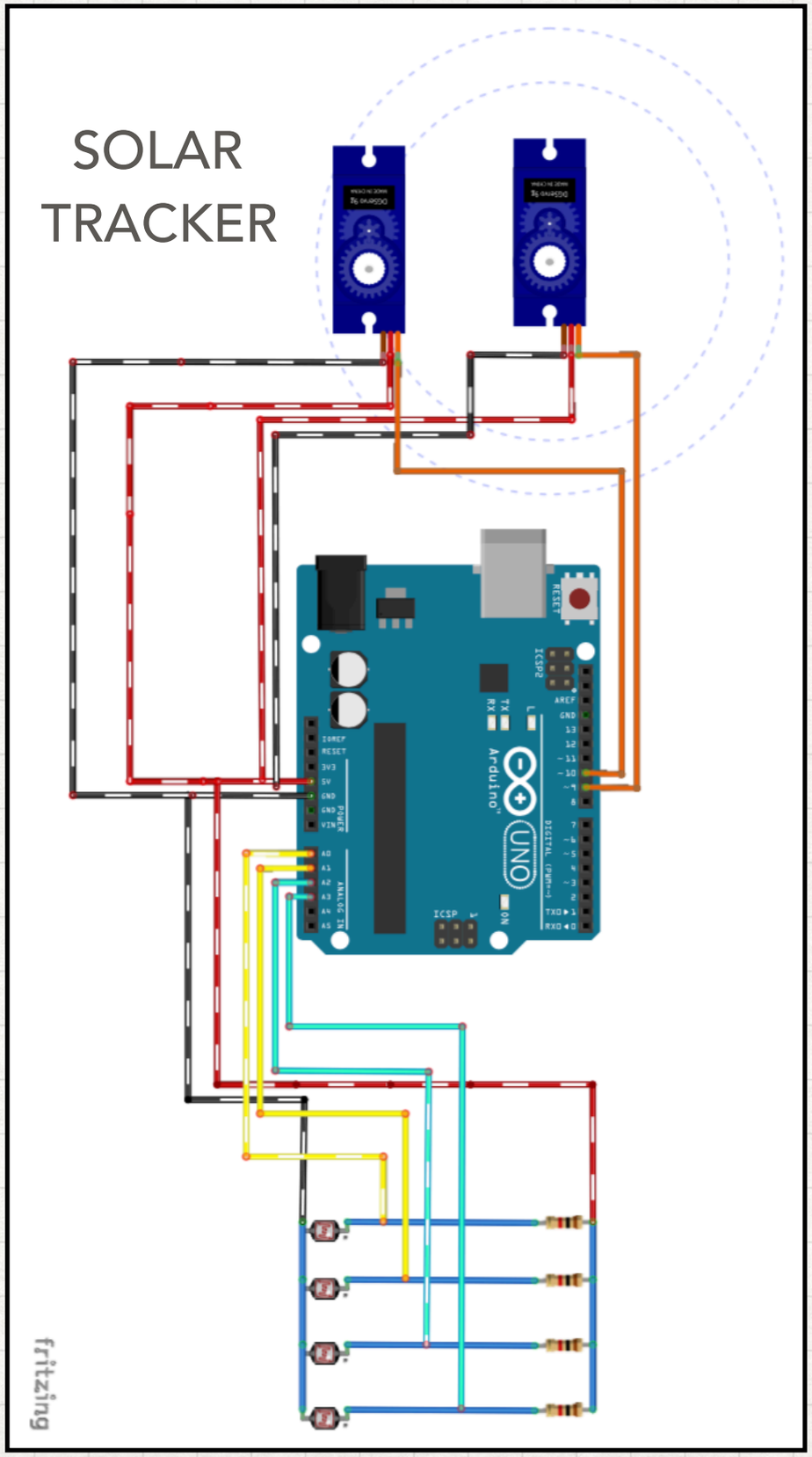
CS10, CS11, CS12: Clock Select

CODE BASE

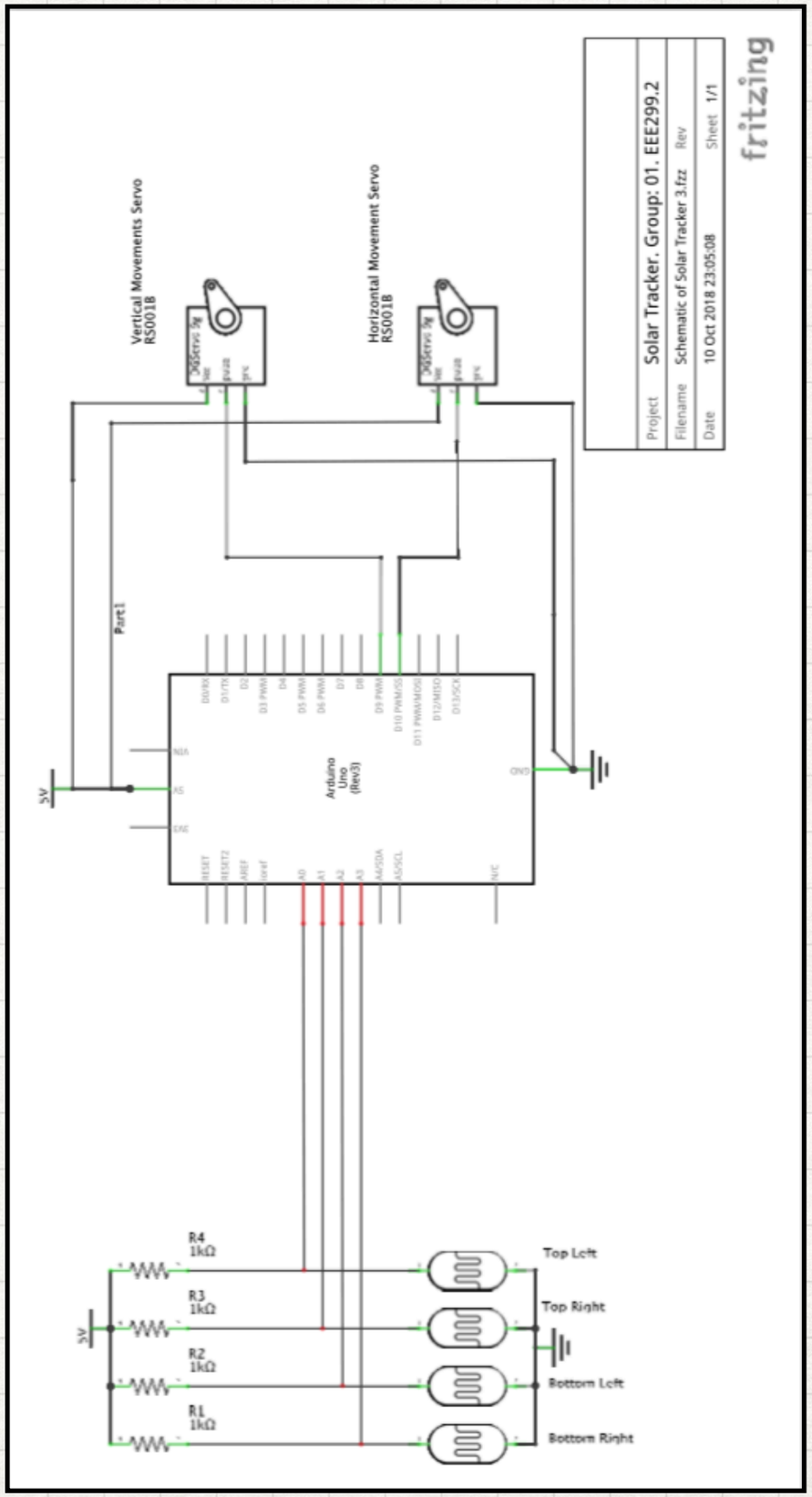
THE ROTATING SOLAR PANEL MODULE

```
int topleft;
int topright;
int downleft;
int downright;
int waittime = 1;

void setup() {
  pinMode(9, OUTPUT); //For Horivontal servo movement
  pinMode(10, OUTPUT); //For Vertical Servo movement
  TCCR1A = 0;
  TCCR1A = (1 << COM1A1) | (1 << COM1B1) | (1 << WGM11);
  TCCR1B = 0;
  TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11);
  ICR1 = 40000;
  OCR1A = 3000;
  OCR1B = 3600;
}

//LDR input pins
void loop() {
  topleft = analogRead(A0);
  topright = analogRead(A1);
  downleft = analogRead(A2);
  downright = analogRead(A3);
 if (topleft > topright) {     //For Horizontal Movement
Start
    OCR1A = OCR1A + 1;
    delay(waittime);
  }
  if (downleft > downright) {
    OCR1A = OCR1A + 1;
    delay(waittime);
  }
if (topleft < topright) {
  OCR1A = OCR1A - 1;
  delay(waittime);
}
if (downleft < downright) {
  OCR1A = OCR1A - 1;
  delay(waittime);
}                          //For Horizontal Movement End
if (OCR1A > 4000) {
  OCR1A = 4000;
}
if (OCR1A < 2000) {
  OCR1A = 2000;
}
if (OCR1A < 2000) {
   OCR1A = 2000;
  }
  if (topleft > downleft) {    //For Vertical Movement start
    OCR1B = OCR1B - 1;
    delay(waittime);
  }
  if (topright > downright) {
    OCR1B = OCR1B - 1;
    delay(waittime);
  }
  if (topleft < downleft) {
    OCR1B = OCR1B + 1;
    delay(waittime);
  }
  if (topright < downright) {
    OCR1B = OCR1B + 1;
    delay(waittime);
  }
  if (OCR1B > 4200) {
    OCR1B = 4200;
  }
  if (OCR1B < 3000) {
    OCR1B = 3000;
  }                          //For vertical Movement end
}
```

Breadboard Design



Schematics



IMPLEMENTATION

THE POWER LOGGER MODULE

- **Power Logger Module:** For this part we have used an Arduino mega as the brain of the module and INA 219 IC as the heart. Here the INA 219 IC's Vin+ pin is connected with the source's positive terminal. The load's one portion is connected with Vin- pin and other portion is connected with the negative terminal of the source. From the SCL and SDA pin connected with the Arduino's SCL(pin 21) and SDA(pin 20) pin the INA 219 IC shares it's captured data about the voltage and the current measured by it. Then the Arduino processes those data and shows the data on the OLED screen connected with it using the same SCL(pin 21) and SDA(pin 20) pin. Here we have also connected a Micro SD Storage Board TF Card Shield Module for Arduino to store this data in to a Micro SD card. For tis we have connected the modules MOSI to pin 51, MISO to pin 50, SCK to pin 52 and CS to pin 53 of Arduino. From this captured data we can plot Voltage ,Current and Power graph.

Elaboration of the used pins:

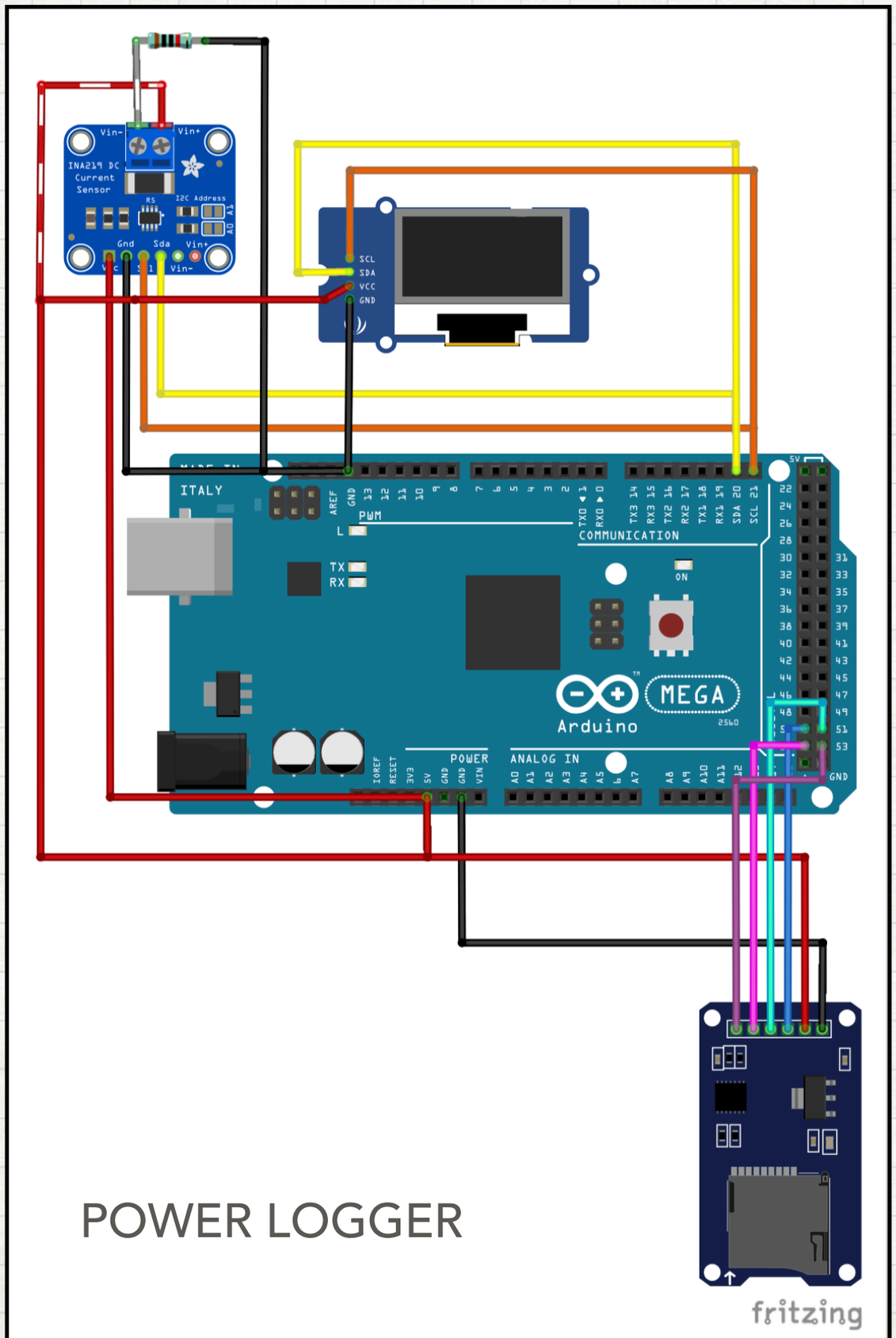
- MISO (Master In Slave Out) - The Slave line for sending data to the master,
- MOSI (Master Out Slave In) - The Master line for sending data to the peripherals,
- SCK (Serial Clock) - The clock pulses which synchronize data transmission generated by the master and one line specific for every device:
- SS (Slave Select) - the pin on each device that the master can use to enable and disable specific devices.
- CS(Chip Select)
- SDA(Data Line)
- SCL(Clock Line)

CODE BASE

THE POWER LOGGER MODULE

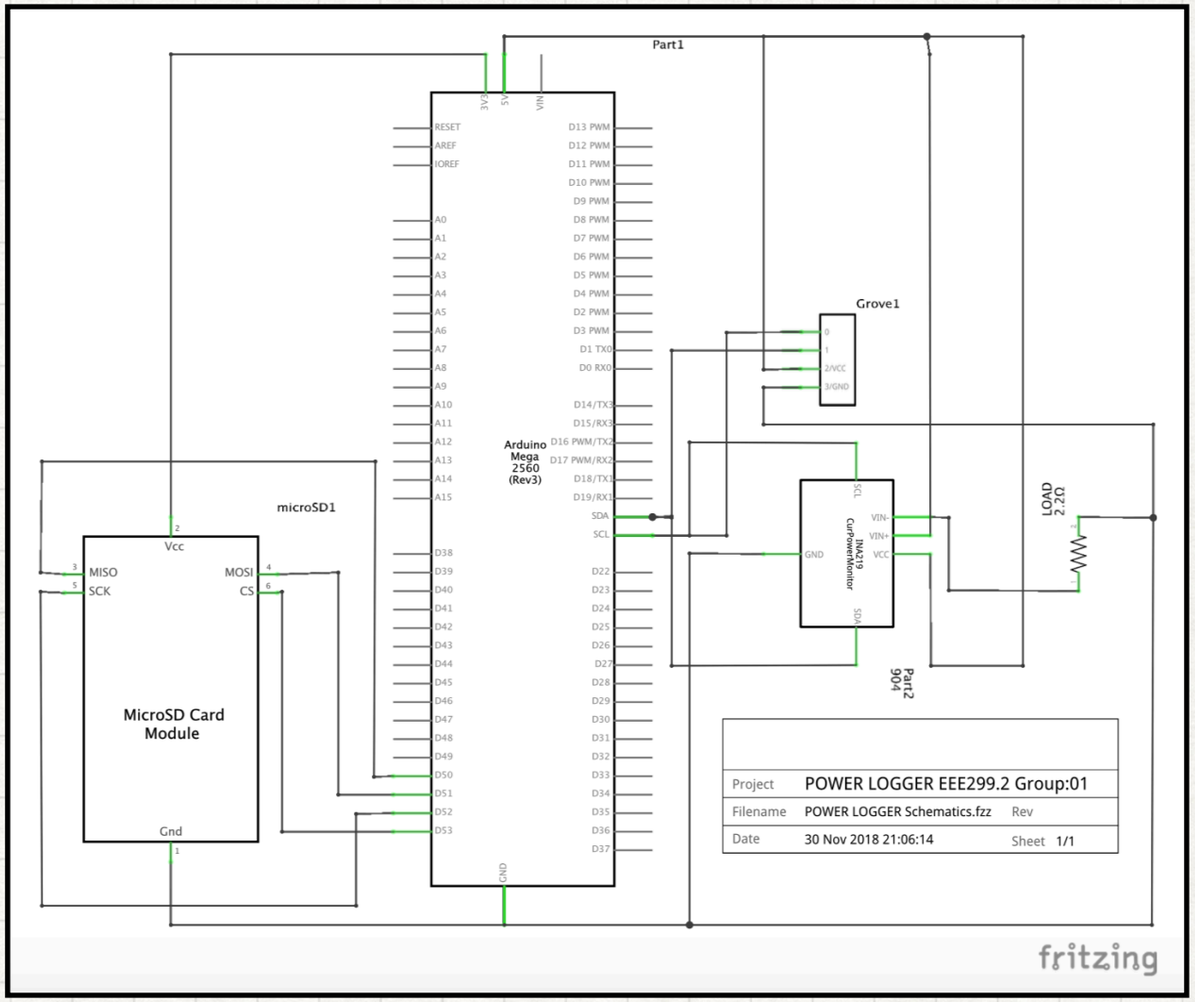
```
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <Adafruit_SSD1306.h>
#include <SPI.h>
#include "SdFat.h"   //Added library for SD card
SdFat SD;

#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);
Adafruit_INA219 ina219;


unsigned long previousMillis = 0;
unsigned long interval = 1000;   //Taking measurement after
1000 milliseconds or 1 second interval
const int chipSelect = 53;     //CS pin of arduino mega. It
differs with board type
float shuntvoltage = 0;
float busvoltage = 0;
float current_mA = 0;
float loadvoltage = 0;
float energy = 0;
File TimeFile;
File VoltFile;
File CurFile;

void setup() {
  SD.begin(chipSelect); //For initiating the data storing
process in the SD card
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  //For
initiating and displaying the specs on the
OLED :SSD1306_SWITCHCAPVCC
  ina219.begin(); //For initiating INA219 bi-directional current
sensing ic
}

void loop() {
  unsigned long currentMillis = millis(); //Taking the
millisecond time fro the arduino's clock
  if (currentMillis - previousMillis >= interval)
  {
    previousMillis = currentMillis;
    ina219values();  //Calling the function for reading the
values from INA219 bi-directional current sensing ic

    TimeFile = SD.open("TIME.txt", FILE_WRITE);
    if (TimeFile) {
      TimeFile.println(currentMillis);
      TimeFile.close();
    }

    VoltFile = SD.open("VOLT.txt", FILE_WRITE);
    if (VoltFile) {
      VoltFile.println(loadvoltage);
      VoltFile.close();
    }

    CurFile = SD.open("CUR.txt", FILE_WRITE);
    if (CurFile) {
      CurFile.println(current_mA);
      CurFile.close();
    }
displaydata();

 //For showing the data in the arduino's serial plotter.
    Serial.begin(9600);  //initiating the serial plotter
        Serial.print("Volt(V):");
        Serial.println(loadvoltage);
        Serial.print("Current(mA):");
        Serial.println(current_mA);
        Serial.print("Time(msec):");
        Serial.println(currentMillis);
        Serial.print("Power(mW):");
        Serial.println(loadvoltage*current_mA);
  }

//Function for displaying the data on the OLED
void displaydata() {
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(0, 0); //Setting the position
  display.println(loadvoltage);
  display.setCursor(35, 0);
  display.println("V");
  display.setCursor(50, 0);
  display.println(current_mA);
  display.setCursor(95, 0);
  display.println("mA");
  display.setCursor(0, 10);
  display.println(loadvoltage * current_mA);
  display.setCursor(35, 10);
  display.println("mW");
  display.setCursor(0, 20);
  display.println(energy);
  display.setCursor(35, 20);
  display.println("mWh ::Group01");
  display.setCursor(58, 10);
  display.println("::EEE299.2");
  display.display();
}

//Function For reading the values of INA219 bidiectional
current sensing ic
void ina219values() {
  shuntvoltage = ina219.getShuntVoltage_mV();
  busvoltage = ina219.getBusVoltage_V();
  current_mA = ina219.getCurrent_mA();
  loadvoltage = busvoltage + (shuntvoltage / 1000);
  energy = energy + loadvoltage * current_mA / 3600;
}
```

Breadboard Design

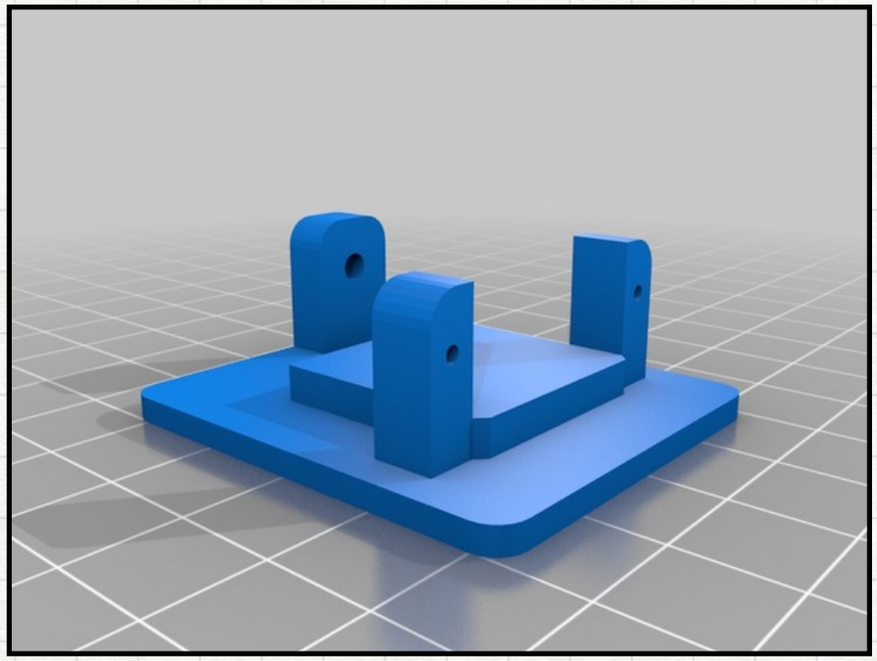


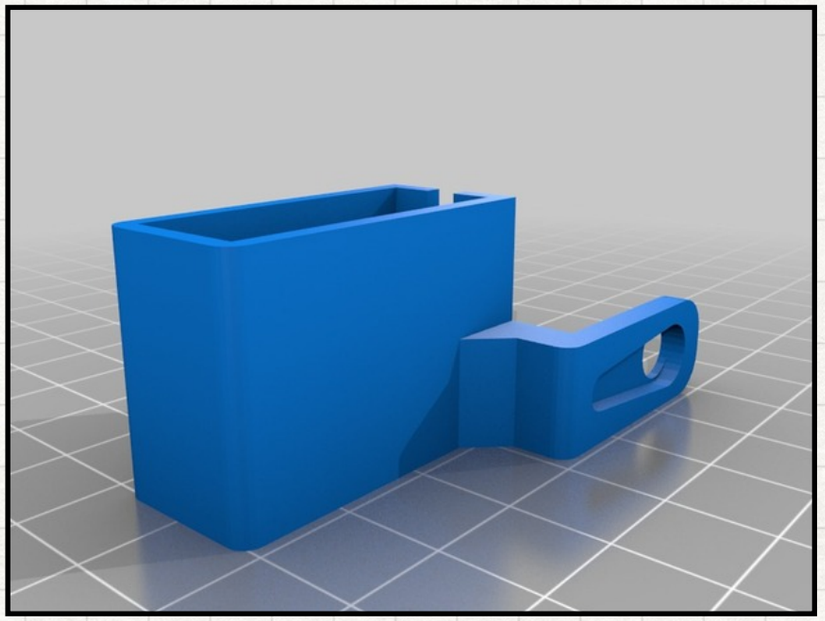
POWER LOGGER

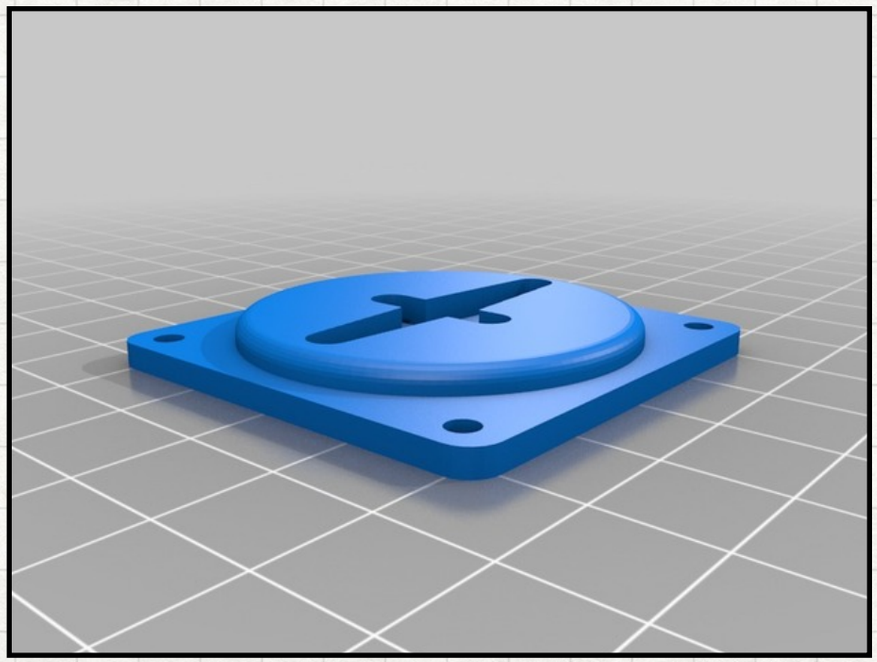
Schematics

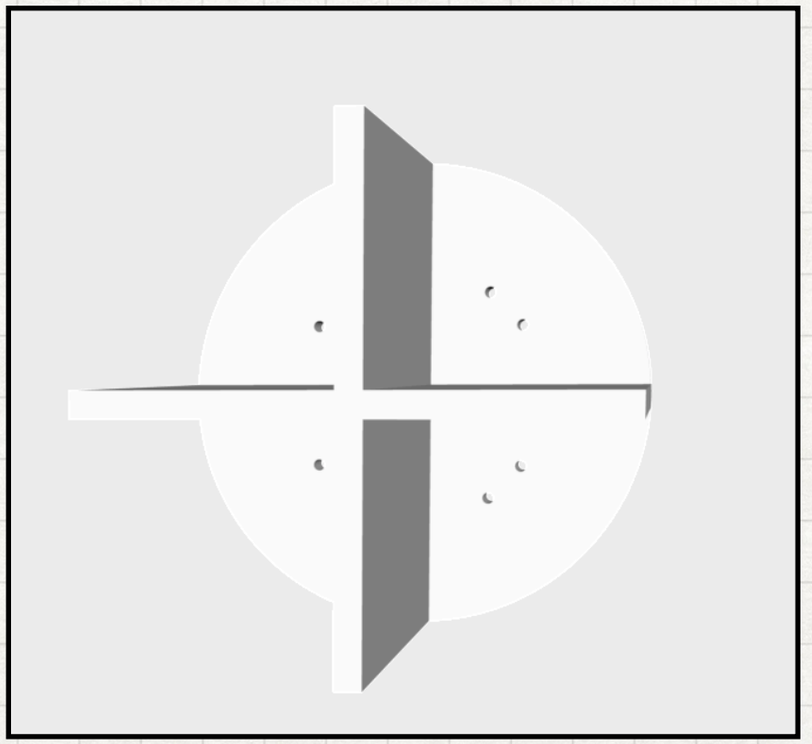


3D PRINTED STRUCTURES



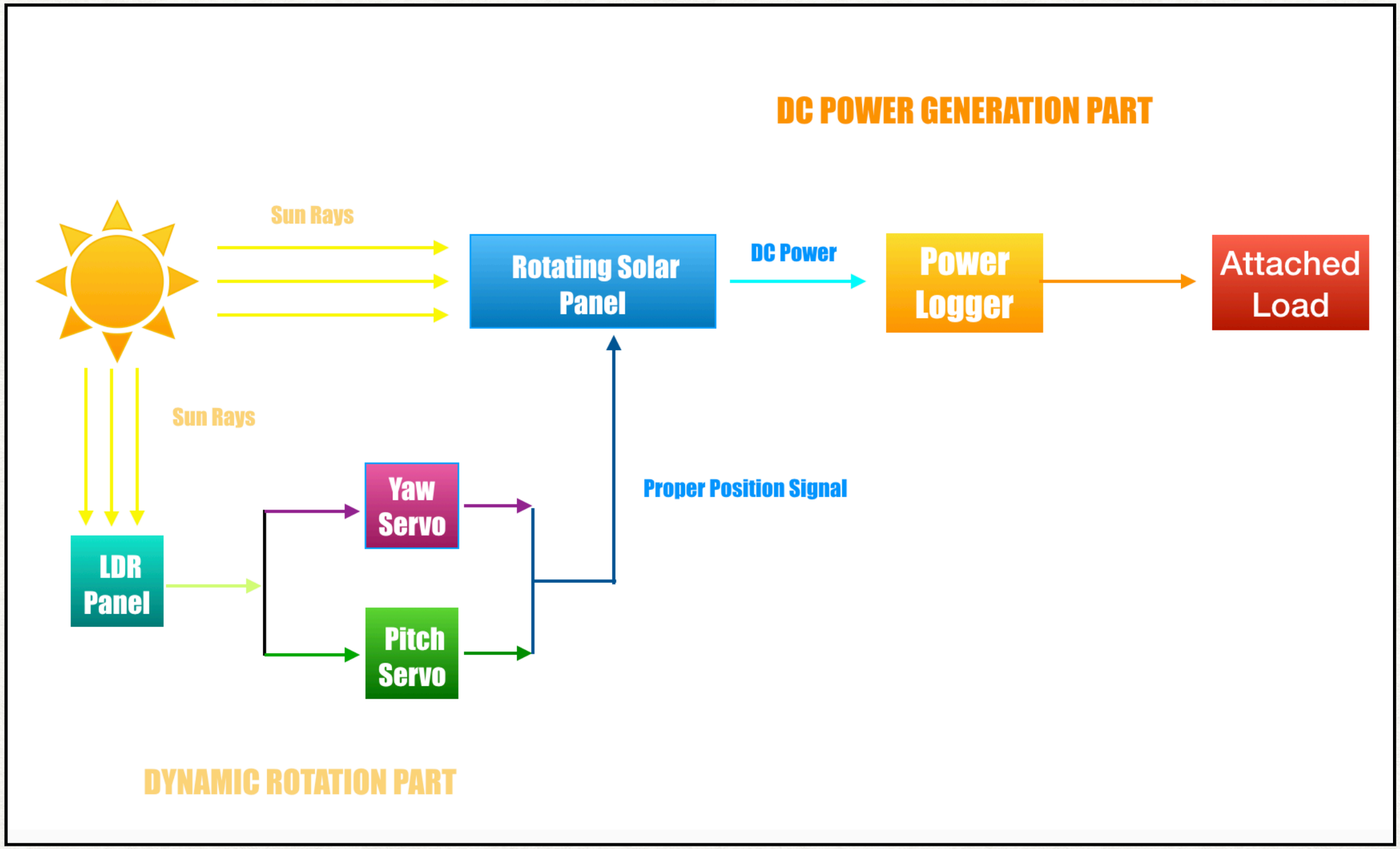






https://www.thingiverse.com/thing:708819

DIAGRAMMATIC OVERVIEW OF THE PROJECT



LIST OF MATERIALS AND THEIR COSTING

SERIAL NO	ITEM NAME	NUMBER OF UNITS	PER UNIT PRICE	COST (IN TAKA)
1	ARDUINO MEGA	1	850	850
2	ARDUINO UNO	1	450	450
3	OLED DISPLAY	1	500	500
4	INA 219 IC MODULE	1	1380	1380
5	SERVO 9g	2	130	260
6	SOLAR PANEL	1	250	250
7	LDR(5KΩ)	4	10	40
8	RESISTOR(1KΩ)	2	0.5	1
9	PLASTIC WOOD BOARD	1	180	180
10	JUMPER WIRES	2 (SET)	60	120
11	MICRO SD MODULE	1	250	250
12	SD CARD(4GB)	1	250	250
13	BREAD BOARD	1	100	100
14	3D PRINTING COST	1 (SET)	450	450
	TOTAL COST			4911

REFERENCES

1.https://www.arduino.cc
2.https://www.solarreviews.com/blog/what-are-the-most-efficient-solar-panels-for-2018
3.https://www.eia.gov/todayinenergy/detail.php?id=18871
4.https://www.thingiverse.com/thing:708819
5.https://www.youtube.com/watch?v=kUHmYKWwuWs&list=WL&index=43&t=0s
6.https://www.youtube.com/watch?v=BtLwoNJ6klE&list=WL&index=45
7.https://www.youtube.com/watch?v=Caw5PiIFk-w&list=WL&index=51
8.https://www.youtube.com/watch?v=8MvRRNYxy9c&index=52&list=WL
9.https://www.youtube.com/watch?v=5Dp-XatLySM&index=53&list=WL
10.https://www.youtube.com/watch?v=2kr5A350H7E&index=40&list=WL
11.https://www.youtube.com/watch?v=J61_PKyWjxU&index=41&list=WL
12.https://dronebotworkshop.com/lcd-displays-arduino/
13.https://www.youtube.com/watch?v=_e_0HJY0uIo&list=WL&index=36
14.https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ina219-current-sensor-breakout.pdf
15. ATmega 328P datasheet

“

THANK YOU

Any question

”