IoT Based Dual Axis Solar Tracker

Prakhar Sachan
20BEE0217
School of Electrical Engineering, VIT
prakhar.sachan2020@vitstudent.ac.in

Koyna Chakravorty
20BEE0227
School of Electrical Engineering, VIT
koyna.chakravorty2020@vitstudent.ac.in

Devansh Jain
20BEE0237
School of Electrical Engineering, VIT
devansh.jain2020@vitstudent.ac.in

***Abstract*—** Nowadays solar energy is rapidly gaining importance as an alternative energy source as fossil fuels are limited and very expensive, and the sun is the largest source of free energy. Our project is about using energy as efficiently as possible. As farmers and other ordinary people in our country are not able to calculate the power consumed and the amount of backup received depending on the load connected to the battery. Our proposed system, therefore, offers an easy solution to both problems by using solar energy appropriately and efficiently to solve the fossil fuel reduction crisis, as solar energy is abundantly available. The main goal of this project is to design a precise dual-axis solar tracker and share the information through the IoT platform. In this work, the system tracks the position of the sun regardless of the weather condition. The inclination of solar collectors depends on solar irradiance. Using a panel that can be rotated along its axis for the sun's position will increase the conversion efficiency by at least 30-40%.

Keywords*— *My Devices Cayenne, Wiznet W5100, DHT22, Solar Tracker, RJ45 Cable, Arduino Mega

I. Introduction

The solar tracking system plays a major role in which it is used to capture the maximum power from the sunlight.

Solar power generation is one of the pollution-free and zero-emission processes rather than a non-renewable energy source.

Photovoltaic panels are used for harnessing solar energy from the sun and converting them into electrical energy. But they can't fully harness solar energy during the day if they are kept static at a particular angle. In order to obtain maximum power output, photovoltaic panels should be moved along with the sun.

The latest non-static trackers allow photovoltaic systems to decrease the angle of incidence between the incoming sunlight and the sensor in the solar tracking device, decreasing the amount of electricity. All focused solar systems have trackers in an effort to produce energy from direct sunlight into solar panels. These solar trackers have been shown to have dual-axis tracking technology designed to combine solar modules and reduce the cost of devices. These kinds of solar tracking systems are also ideal in rural areas where electricity production is inadequate. The solar panel, which transforms solar energy directly into electricity, is one of the key components.

Dual-axis solar tracker is more efficient than a single-axis solar tracker. Dual-axis tracking allows for the most accurate orientation of the solar device and is said to provide 40% more output through energy absorption. However, these solar trackers are more complex and expensive. Dual-axis trackers continually face the Sun as they can move in two different directions. There are many methods for tracking the sun in dual-axis solar system. A simple method is by using LDR (Light-dependent resistor) for finding the position of the sun. LDR is a photoresistor that changes its resistance based on the intensity of light falling on it. By placing four LDRs at each side of the photovoltaic panel, which rotates the photovoltaic panel in a particular direction when that particular side LDR output is low with this method the sun can be tracked with ± 4% degree precision

II. Ease of Use

The whole project is a low cost easy-to-use IoT-based system. The interfacing is user-friendly, one can easily customize and change the appearance of the whole dashboard on the MCayenne Platform. The threshold value for different parameters can be set by users which can be used to set triggers and notifications. The alerts sending service is also free of cost and demand no charges. Overall it's a user-friendly, customizable, one-time investment, low-maintenance IoT-based solution for improving efficiency and improving data clarity for Solar Trackers.

III. Modelling

Physical Layer

The perception Layer consists of the hardware components which are interfacing with the ATMega2560 microcontroller and the IoT sensors such as DHT22 for capturing and continuous monitoring of real-time temperature and humidity, LDR (light dependent resistor) for capturing the luminosity in ratio with the other LDRs. For providing motion for 2 degrees of freedom i.e. horizontal axis and vertical axis, we have used SG90 micro Servo motors. These are tiny motors that provide a great amount of torque to body weight ratio, they can easily handle small solar panels and structural weight. Arduino Mega 2560 is the main microcontroller that acts as the main processing unit and a bridge between IoT sensors and the Network Layer of the project. It receives data from different sensors and accordingly gives commands to the IoT sensors working as output i.e. SG90 Servo Motor.

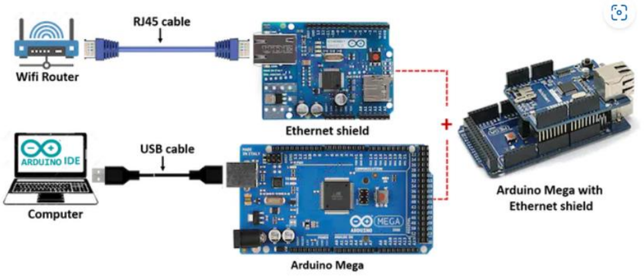
Communication Layer

The main component used for transmission is Ethernet Shield W5100, based on the Wiznet5100 chip that helps establish and provide internet connectivity to Arduino. It acts as a bridge between IoT Platform Mcayenne and Arduino. It helps in sending and receiving data packets over the internet which is only done through Ethernet Shield. We are also using RJ45 cable which is plugged into an Ethernet Shield and the other end of the cable to a router or PC.

Application Layer

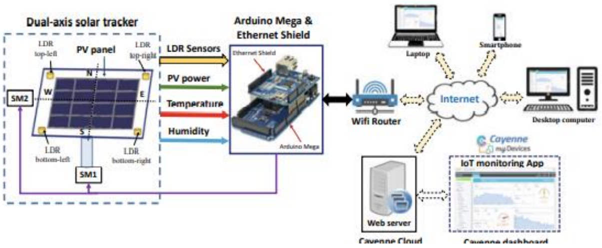
My Devices Cayenne platform is used for the abstraction of data for the project. Users will be able to visualize the monitoring and current status of all sensors as well as AT Mega with the help of a dashboard. In the dashboard with the help of various customizable widgets, we can control the motion manually as well as automatically through a button widget and so on. Power generated by Solar panels can be monitored from anywhere in the world. Triggers and Alerts are one of very useful features of Mayenne. Users can take control of all the aspects of Dual Axis Solar Tracker with a simple user-customized and free-of-cost IoT Platform.

IV. Research Methodology



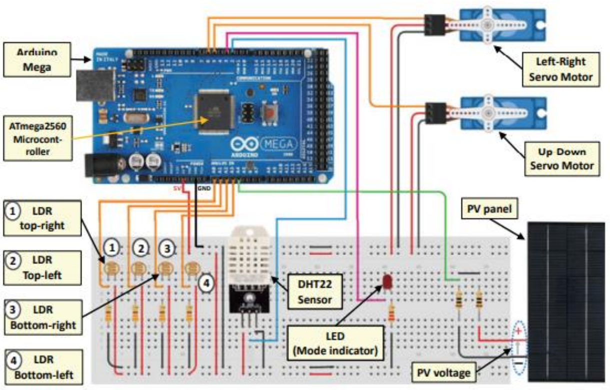
A. System Description

The proposed IoT-based solar tracker system. It is a dual-axis solar tracker that can rotate automatically to track the sun's position using LDR sensors, or manually by the user through the dashboard of an IoT application. The system starts with detecting the sun position (intensity of light) by LDR sensors and sends the data to the controller (Arduino Mega board). This latter then processes these data to command servomotors (SM1 and SM2) that hold the PV panel to rotate toward the sun. The values of the generated PV voltage and current, temperature, and humidity are also sent to the Arduino through associated sensors. Next, the Ethernet shield, which is mounted with Arduino and allows it to be connected to the Internet, will send the data that has been taken and processed by Arduino to the cloud (webserver). Lastly, the solar tracker data, including LDR sensors, PV power, temperature, and humidity, are displayed in real-time in the IoT monitoring application via pre-created Widgets. The IoT monitoring application is designed using the Cayenne myDevices platform. Once the user is connected to the internet from his computer or smartphone, he can visualize all solar tracker data in the IoT application's dashboard in their associated widgets. Therefore, the user has the necessary data linked to the environment and performance of the PV panel. In the manual mode, the servomotors will take angle directions from their associated widgets in the dashboard. Hence, the user can control his system to seek the best environmental conditions and extract the maximum energy from the PV panel. The IoT application is also programmed to send notification alerts (SMS or Email) when a sensor reaches a predefined threshold value.



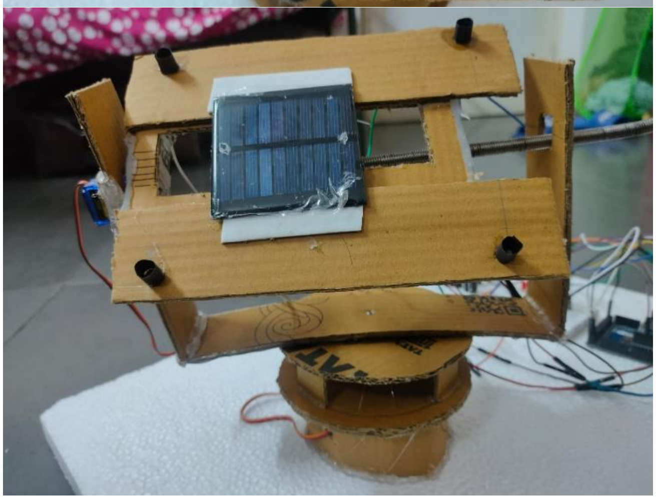
B. Hardware Design

As shown in Figure, the IoT solar tracker system consists of a PV panel, two servomotors, four LDR sensors, a voltage divider circuit, a temperature and humidity sensor, a Led, and the Arduino Mega board. The used PV panel is 115 by 85 mm in size with a 1.6 W output and can generate a voltage of up to 6 V. Two 180° servomotors are operated to motorize the solar tracker and they are controlled by the Arduino board through PWM pins 5 and 6. The left-right (L-R) servomotor (MG996R) rotates the solar tracker on the vertical axis (East/West), while the Up-down (U-D) servomotor (SG90) rotates the solar tracker on the horizontal axis (South/North). Four LDRs (Cds GL5528) are used to sense the sun's position and have been fixed in the four corners of the PV panel. The LDR sensors are connected to the Arduino through analog pins from A0 to A3. The LDR is a resistor whose value decreases with increasing light intensity incident on its surface. The LDR sensor is designed as a voltage divider circuit. The output of the voltage divider is connected to an analog input (A0 for instance) of the Arduino. Then, the microcontroller's Analog Digital Converter (ADC) converts the analog value read by A0 into a digital value between 0 and 1023 because the ADC is coded in 10 bits. The value of the series resistor in the LDR sensor circuit is 330 ohms. The temperature and humidity are measured through the DHT22 sensor, which is an ultra-low-cost sensor that is widely used in embedded projects. DHT22 has a thermistor and a capacitive humidity sensor embedded in it to measure temperature and relative humidity. Its temperature range is from −40 to 80 °C with < ± 0.5 °C of accuracy, and its humidity range is from 0 to 100% with ± 2% (Max ± 5%) of accuracy. This sensor uses one signal wire to transmit data to Arduino (digital pin 2), and two wires for the power supply. The PV voltage and current are measured through a voltage divider circuit that also acts as a load and which consists of two series resistors of 10 Ohms. The divider circuit output is connected to the Arduino's analog pin A4. Furthermore, a LED, which is connected to digital pin 3, reflects in the system circuit the mode state of the solar tracker (manual or automatic). The Arduino Mega with ATmega2560 microcontroller is used as the embedded controller that interacts with the Arduino Ethernet shield along with the monitoring platform. The Ethernet shield, which is mounted above the Arduino board, must be connected to a Wi-Fi router (or PC) through an RJ45 cable. The Ethernet Shield is based on the Wiznet W5100 Ethernet chip that provides a network (IP) stack for TCP and UDP protocols.

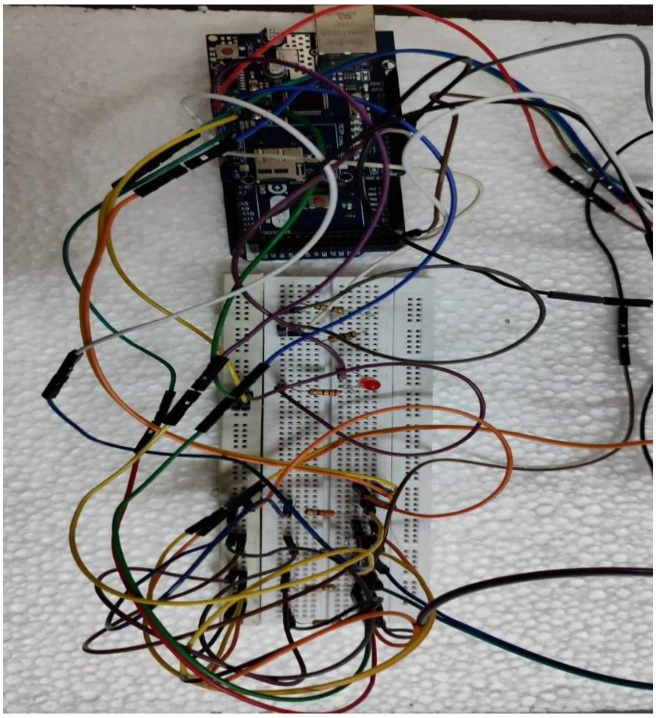


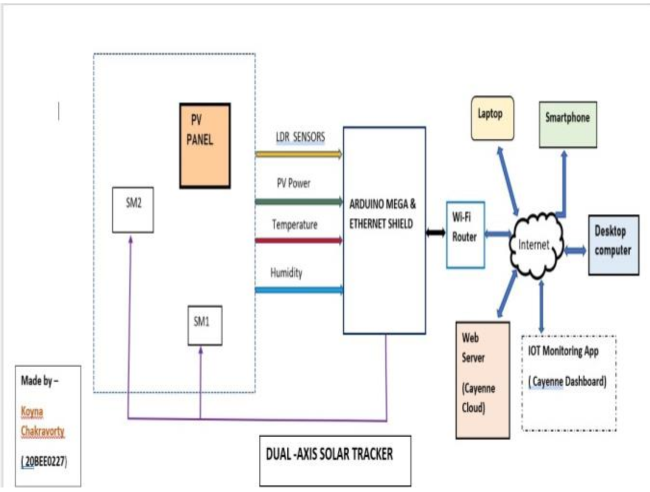
METHODOLOGY

C. Prototype



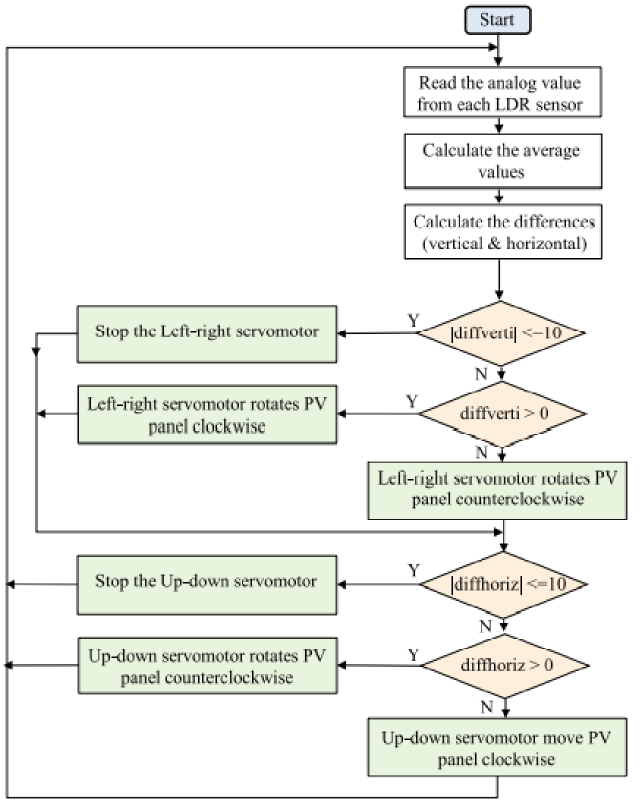
The above Figure presents the solar tracker prototype in its detached and assembled state. It consists of the PV panel, the L-R, and U-D servomotors, and the LDR sensors. The panel is attached to the U-D servomotor on one side and with a bearing on the other side to ensure better flexibility when the solar tracker rotates around the horizontal axis. The assembly is attached to the L-R servomotor. The LDR sensors are fixed in the four corners of the panel inside hollow cylinders. If the panel is not perpendicular to the sun, at least one LDR will be covered by shadow caused by the surrounding cylinder. Hence, there will be a difference in light intensity. The best orientation is when the light intensities are equal in all LDR sensors. Figure 5 shows the entire prototype of the IoT-based solar tracker system, and it is clear that all reported components in the hardware part have been used to build it.





D. Software Design

Arduino is an open-source electronics prototyping platform with easy-to-use hardware and software. The Arduino platform provides an integrated development environment (IDE), which includes support for C and C++ programming languages. The Arduino board used in this work is programmed by the IDE that serves as a code editor and from which the program code can be uploaded to the microcontroller through a USB cable. The Arduino Mega board is utilized to implement all software requirements of the IoT-based solar tracker.



MyDevices is a company that offers IoT solutions. It offers an end-to-end platform for the IoT. In our project, we will focus on Cayenne, one of the solutions from myDevices. This tool allows developers, designers, and engineers to build prototypes of the IoT. Cayenne uses the Message Queuing Telemetry Transport (MQTT) protocol to connect any device to the Cayenne cloud. Once connected, the user can send and receive data from the device to the Cayenne dashboard via the Widgets created. MQTT is a publish-subscribe messaging protocol based on the TCP/IP protocol. The publish-subscribe methodology uses a message agent responsible for delivering messages to the client. The MQTT is the API for sending information to the Cayenne cloud, or devices controlled by Cayenne. The messaging agent in this connection is the cloud, it manages the different clients (sensors and actuators) that send and receive the data. To use MQTT with Cayenne, we need to use the Cayenne libraries. For Arduino, the CayenneMQTT library can be installed from the IDE's Library Manager. To program our Cayenne IoT platform-based IoT application, we will take advantage of the predefined functions. For example, to establish the connection between Cayenne cloud and Arduino Mega equipped with the Ethernet module, we call the CayenneMQTT Ethernet library where we declare our authentication information (the username, password, and the ClientID) which should be obtained from the Cayenne Dashboard. Then, in the setup part of the program, we call Cayenne. begin () function to establish the connection with the Cayenne dashboard. For each actuator, we create a function with an integer parameter between 0 and 31, called CAYENNE IN (VIRTUAL CHANNEL). For each sensor, we create a function with an integer parameter between 0 and 31, called CAYENNE_OUT (VIRTUAL_CHANNEL). In the loop part of the program, we call the predefined function Cayenne. loop (), this function itself calls the functions CAYENNE_OUT and CAYENNE_IN. The virtual channel as its name suggests is a channel that does not physically exist, it characterizes visualization or command widgets. It allows them to be linked with the corresponding sensor or actuator The embedded software is the piece that will be embedded in the Arduino Mega to interact between the Ethernet module and Cayenne cloud. It is designed as follows: (i) The IoT-based solar tracker has two function modes: manual and automatic. A button created in the Cayenne dashboard has a role to switch between the two modes. When it is inactive, the manual mode is selected, otherwise automatic mode. Besides, a function is established in the Arduino code that allows recovering the state of the button. The LED in the system circuit reflects the state of this switch. Therefore, for the controller to know the selected operating mode, we just need to test the state of the pin to which the LED is connected. For example, if the LED state is low, the controller will call the manual mode function to execute, otherwise, it will call the automatic function. (ii) If the manual mode is selected, the user can directly control the positions of the servomotors to orient the PV panel from east to west by the L-R servomotor or from south to north by the U-D servomotor.

The control is made from the associated widgets of servomotors in the dashboard of the IoT application. In this mode, the controller calls the Cayenne.loop () function which itself calls all the functions CAYENNE_IN, including those related to servomotors, to execute. The Cayenne.loop () function will also call all the functions CAYENNE_OUT, linked to the sensors, to execute. Where the data related to LDR sensors, PV current, voltage and power, temperature, and humidity would be sent to the server so that they can be visualized in their associated widgets in the IoT application. (iii) If the automatic mode is selected, the algorithm starts by reading the analog values returned by LDR sensors. Then, it processes these data to command servomotors that move the PV panel toward the sun position. Considering the vertical axis-based solar tracker movement, the average values of the two LDRs on the left and the two LDRs on the right are compared and if the lefts receive more light, the PV panel will move in that direction (clockwise) through the L-R servomotor. The latter will stop when the difference result is between −10 and 10. This range is used to stabilize the controller and to reduce the power consumption of servomotors. Otherwise, if the right set of LDRs receives more light, the PV panel will move in that direction (Counterclockwise) through the L-R servomotor and will continue to rotate until the difference result is in the range [−10, 10]. The same approach is used for the horizontal axis-based solar tracker movement where the average values of the two LDRs on the top and the two LDRs on the bottom are compared.

Arduino Code

File Edit Sketch Tools Help

sketch1 §

```
/*****************************************************************
 PROJECT: IoT based solar tracker system / the embedded software
 Prakhar Sachan 20BEE0217
 Koyna Chakravorty 20BEE0227
 ******************************************************************/

#define CAYENNE_PRINT Serial
#include <CayenneMQTTEthernet.h>    //CayenneMQTT library
#include <Servo.h>                  //Servo motor library
#include <DHT.h>                    //DHT library
#define DHTTYPE DHT22
#define DHTPIN 2
DHT dht(DHTPIN,DHTTYPE);

//MQTT credentials
char username[]="5179f440-230c-11ed-bf0a-bb4ba43bd3f6";
char password[]="1c9acb501694ed38a8b565f3b802b2176594ab64";
char clientID[]="7349f5c0-230c-11ed-baf6-35fab7fd0ac8";

Servo servo_x;                    //up-down servomotor
int servoh = 0;
int servohLimitHigh = 170;
int servohLimitLow = 10;

Servo servo_z;                    //left-right servomotor
int servov = 0;
int servovLimitHigh = 170;
int servovLimitLow = 10;

int topl,topr,botl,botr;
int threshold_value=10;
float Vout;

void setup()
{ Serial.begin(9600);
  Cayenne.begin(username, password, clientID);
  servo_x.attach(5);
  servo_z.attach(6);
  dht.begin();
  pinMode(3,OUTPUT);
  digitalWrite(3,LOW);
}

void loop()
{ topr= analogRead(A2);
```

File Edit Sketch Tools Help

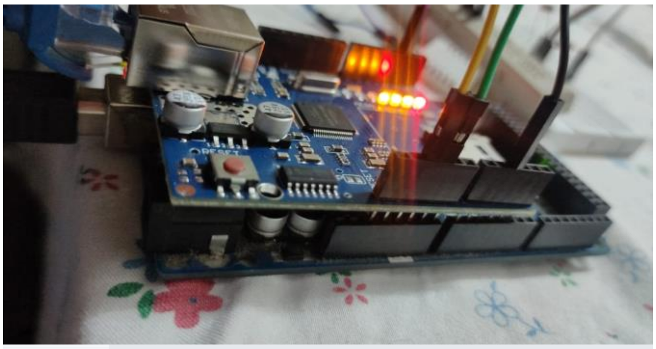
sketch1 §

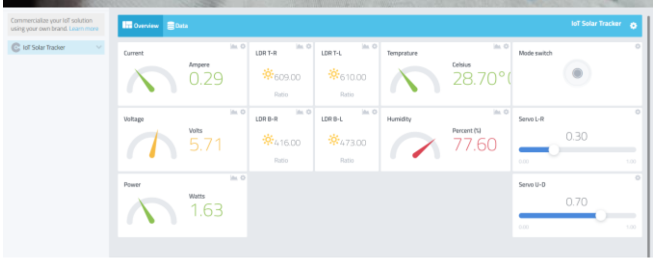
```
void loop()
{ topr= analogRead(A2);
  topl= analogRead(A3);
  botl= analogRead(A4);
  botr= analogRead(A5);
  Vout=(analogRead(A1) * 5.0) / 1023;
  Serial.println(" Manual-mode");
  Cayenne.loop();

  if (digitalRead(3)==HIGH){
    Serial.println(" Automatic-mode");
    servoh = servo_x.read();
    servov = servo_z.read();
    int avgtop = (topr + topl) / 2;
    int avgbot = (botr + botl) / 2;
    int avgright = (topr + botr) / 2;
    int avgleft = (topl + botl) / 2;
    int diffhori= avgtop - avgbot;
    int diffverti= avgleft - avgright;

    /*tracking according to horizontal axis*/
    if (abs(diffhori) <= threshold_value)
    {
     servo_x.write(servoh);            //stop the servo up-down
    }else {
       if (diffhori > threshold_value)
          { Serial.println(" x - 2 ");
          servo_x.write(servoh -2);    //Clockwise rotation CW
          if (servoh > servohLimitHigh)
          {
           servoh = servohLimitHigh;
          }
          delay(10);
          }else {
           servo_x.write(servoh +2);   //CCW
           if (servoh < servohLimitLow)
           {
           servoh = servohLimitLow;
           }
           delay(10);
           }
      }
    /*tracking according to vertical axis*/
    if (abs(diffverti) <= threshold_value)
```

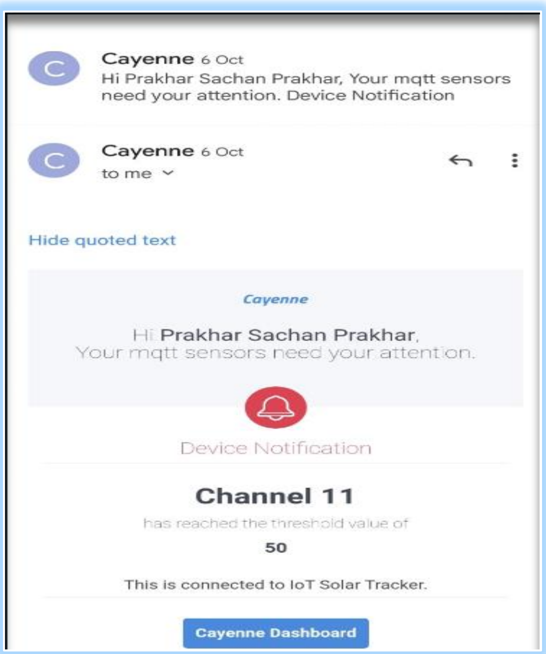
V. Results and Discussion



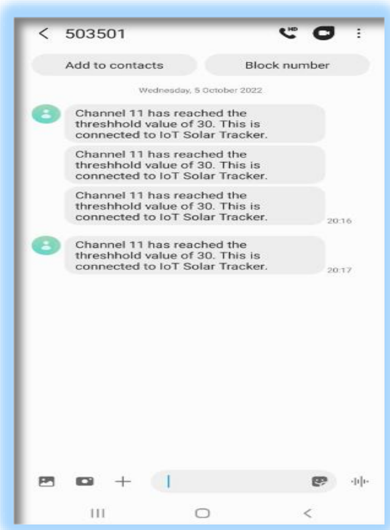


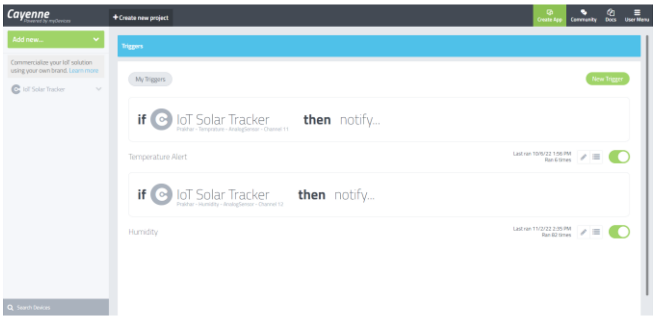
Different tests have been carried out to examine the developed IoT-based solar tracker prototype. The Arduino board is powered with the computer through a USB cable, which is also used to display, in the Serial Monitor of

Arduino IDE, the measured parameters and data received from the IoT application. This will allow us to verify whether the captured data from Arduino are correctly and in real-time sent to the IoT application or not. Whereas, Arduino can be powered with an external DC power supply. The Ethernet shield connects the Arduino board to the internet via RJ45 cable. Once the connection with the IoT application is established, the data of the solar tracker system are sent to the monitoring application, where we can view these data live and send commands to the controller.



To check the reliability of the monitoring application to notify the user when an event occurs, it has been programmed to send an alert. For example, when the monitored temperature is higher than 40 °C. Figure shows the alert notification received in our mailbox at the same time when the temperature exceeds 40° . Other alerts can be added to the application, such as a malfunction of one of the sensors and/or actuators and a rapid decrease in PV power.





VI. Future Scope

The goals of this project were purposely kept within what was believed to be attainable within
the allotted timeline. As such, many improvements can be made upon this initial design.

- The sensitivity and accuracy of tracking can be enhanced by using a different light sensor. A photodiode with an amplification circuit would provide improved resolution and better tracking accuracy/precision.

- Real Time clocks (RTCs) can be interfaced instead of the sensors used for seasonal monitoring here. With the use of more powerful motors, the trackability and lifespan could be increased, but it could be a bit costly.

- Instead of a microcontroller board, using Arduino is much cheaper, although it is complicated. Solar tracker, whether single or dual axis, will help to achieve optimal solar output levels. Today it looks very regarding the future of the solar tracking industry.

VII. Conclusion

We successfully developed an Iot based two-axis smart tracking system that could capture the sunlight incident on the panel and move it in the direction of maximum sunlight incidence. We also looked into the pros and cons. Some of the main challenges identified when processing the data real-time collected from the solar tracker are 1) Connectivity issues 2) data imbalance 3) Sensor selection and sensor disposition The main conclusions of this study are:
i.The proposed system is inexpensive and compact compared to other tracking systems used for the same application. ii. Extremely easy to program and modify as it is based on Arduino and does not require an external programmer. iii. The designed system is easy to use and increases panel efficiency. iv. The developed system captures real-time data on an Android device.

Acknowledgment

Our team would like to express special thanks of gratitude to our IoT Fundamentals Faculty, Prof.Sitharthan R who gave us immense guidance and helped us in completing this project. We would also like to thank our GUE Faculty, Prof. Gokulkrishnan who generously helped us by providing information and compiling this report.

References

[1] Kang B, Kim D, Choo H (2017) Internet of everything: a large-scale autonomic iot gateway. IEEE Trans Multi Scale Comput Syst 3(3):206–214

[2] M, Komeily A, Wang Y, Srinivasan RS (2019) Adopting internet of things for the development of smart buildings: a review of enabling technologies and applications. Autom Constr 101:111–126

[3] El Hammoumi A, Motahhir S, El Ghzizal A, Chalh A, Derouich A (2018) A simple and low-cost active dual-axis solar tracker. Energy Sci Eng 6(5):607–620

[4] AL-Rousan N, Isa NAM, Desa MKM (2018) Advances in solar photovoltaic tracking systems: a review. Renew Sustain Energy Rev 82:2548–2569. (Elsevier Ltd)

[5] Singh KJ, Kapoor DS (2017) Create your own internet of things: a survey of IoT platforms. IEEE Cons Electron Mag Inst Electr Electron Eng Inc 6(2):57–68

[6] El Hammoumi A, Motahhir S, Chalh A, El Ghzizal A, Derouich A (2018) Low-cost virtual instrumentation of PV panel characteristics using Excel and Arduino in comparison with traditional instrumentation. Renew Wind Water Sol 5(1):1–16

[7] . Nur M, Mohd A, Jumaat A, Rimong C, Jawa A, Jumaat SA (2020) Dual axis solar tracker with IoT monitoring system using arduino. Int J Power Electron Drive Syst 11(1):451–458.