IoT based Smart Energy Management System

Group ID-B21PB02

Guide -Dr. Paul Braineard

T.Charan Reddy-S20180020254

S.Sai Jyothi Kumar-S20180020250

Final Report

Dec 16, 2021

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

SRICITY

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled "**IoT based Smart Energy Management System**" in the partial fulfillment of the requirements for the award of the degree of B. Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of my own work carried out during the time period from August 2021 to December 2021 under the supervision of Prof. Paul Braineard, Indian Institute of Information Technology SriCity, India.
The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

Signature of the student with date

Thathykalva Charan reddy

(S20180020254)

Date: 16-12-21

– –

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of BTP Supervisor with date

Dr. Paul Braineard

Date: 16-12-21

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

SRICITY

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled "**IoT based Smart Energy Management System**" in the partial fulfillment of the requirements for the award of the degree of B. Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of my own work carried out during the time period from August 2021 to December 2021 under the supervision of Prof. Paul Braineard, Indian Institute of Information Technology SriCity, India.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

Signature of the student with date

SAI JYOTHI KUMAR

(S20180020250)

Date: 16-12-21

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of BTP Supervisor with date

Dr. Paul Braineard

Date: 16-12-21

ABSTRACT

The project's main focus is on analyzing the power usage and to give user power usage statistics in real time and manage power consumption using IoT. It also focuses on controlling the electrical appliances via Internet using a website. We created a website to see power consumption trends of electrical devices, for producing output graphs and for turning devices on and off and controlling glow and speed, of lights and fans and automating on and off of water pump based on water in tank.

We are using thingspeak cloud platform for transferring data between Raspberry pi and website and hosting website using ngrok. We send data from arduino to raspberry and to the thingspeak. From thingspeak we send data to website. Text and mail will be sent to user.

Contents

List of Figures

IoT based Smart Energy Management System

1 Introduction

Electricity has become an integral part of our day to day life and it is not an indefinite resource. So managing the energy efficiently and conserving it is very much important.

To achieve this, there has been a lot of research in using power efficiently and generating it using renewable resources. Increasing economic growth and consumption patterns are leading to ever growing demand for energy. Since most of the energy supply is from fossil fuels, the resources are depleting and are causing lots of damage to environment. Hence it is imperative that house holds, industries and commercial enterprises take steps to use electricity only when needed and become energy efficient, reduce energy wastage which in turn reduce expenditure.

Internet has become an part of our lives, many devices that we use in our daily lives like TV, air conditioner, refrigerator, washing machine, can be monitored and controlled remotely on internet via Internet of Things (IoT) technology. Smart Home applications as one of the elements of smart cities. In this project we are aiming to make a prototype of an IoT based Smart Energy Management System which will help us get smart and also gets us a step closer towards our sustainable development goals.

1.1 Problem Statement

To develop a working prototype and website for smart energy management for homes.

1.2 Objectives

- To take out distance barrier between human and device.
- Optimizing electricity usage by using energy only in the required time and level.
- Analyse Energy consumption by appliances..
- To Reduce electricity bill.
- Connect device and user via internet and make system autonomous.

1.3 Applications

- We can monitor our energy consumption at house hold and device level.
- Devices can be operated over internet.
- Make systems more autonomous like water tank turning off automatically when tank is full and auto adjust fan speed based on temperature.
- Helps us optimise our power usage.

1.4 Why Website?

Website is an important component in an IoT system. User interface(UI) and user experience(UX) are one of the most important parts of IoT system and website comes in this part of an IoT system. It provides user an interface where they can easily interact with IoT device with minimal effort and even with zero programming knowledge. Using our website we can provide data with analysed information and control our devices.

Using the website we are providing an option to switch on and off devices, set temperature and other parameters of interest via the internet which greatly simplifies certain tasks.

Most of the papers online which had implemented projects similar to ours had used website as a medium to communicate data to users and to increase ease of usage.

2 Literature Survey

In this Section, we would be discussing papers from which we draw our inspiration and idea for the project. Reference links to the papers are also included.

IoT makes managing the electrical appliances easy from anywhere. The following works created a number of systems for different usages like IoT for remote management, Raspberry pi system for smart management. The papers discussed using technologies like PV(Photovoltaic) energy, AMI(Advanced Metering Infrastructure) which was made in a Lab.

2.1 Internet of Things based Smart Energy Management for Smart Home

link

In this paper Smart Energy Management (SEM) system is made using NodeMCU, arduino mini and Android. The SEM system helps operating electrical appliances for long hours as pre-programmed. The real time monitoring of Devices(active,passive,fault etc.,) and parameters(current,voltage,power etc.,) using an android based system has been proposed.

2.2 IoT Based Smart Energy Management System

link

The above paper focuses on controlling the electrical appliances based on environmental conditions using M2M communication. The prototype consists of different sensors(temperature, humidity, light intensity etc.) and the data from those are fed to the Arduino micro controller. Raspberry pi is used in order to upload the data to a cloud server using a wifi module.

2.3 IoT based smart home energy management system

link

This paper focuses on automation of electrical devices that we use in our daily life. The prototype's major processing unit is the Arduino micro controller. A connection is established to relay information from each sensor to an alarm based system which buzzes off every time the devices reach its limit. For this project, they are using LDR, Buzzer, LM35 and other sensors.

2.4 Real Time Power Consumption Monitoring Using Arduino

link

This paper describes a smart system for calculation of power. Using an Arduino platform as a micro controller to read the voltage and current from sensors and caluculate power using average and then wirelessly send the measured data to monitor the results using an Android application.

3 Methodology

Figure 1: Project methodology

Complete project methodology:

1. We use an Arduino board as our micro controller .
2. Arduino uno controls the sensors and acquires data from them.
3. We measure parameters such as current, temperature etc, using respective sensors and send it to raspberry pi.
4. ACS712 is used to calculate the current and voltage divider circuit to measure voltage.
5. GSM modules can be used to send text messages to the user's mobile .
6. Arduino reads the data from raspberry pi via serial communication and controls the electronic devices via light dimmer module.

7. We use Raspberry pi to send and receive data to our website via Thingspeak.
8. Thingspeak acts as a bridge between our device and website.
9. Website help us track our energy consumption.
10. We can see detailed graphs and control our devices.
11. Email will be sent to us from raspberry pi via Yagmail.
12. User can also read and write reviews for an item he posses in website.

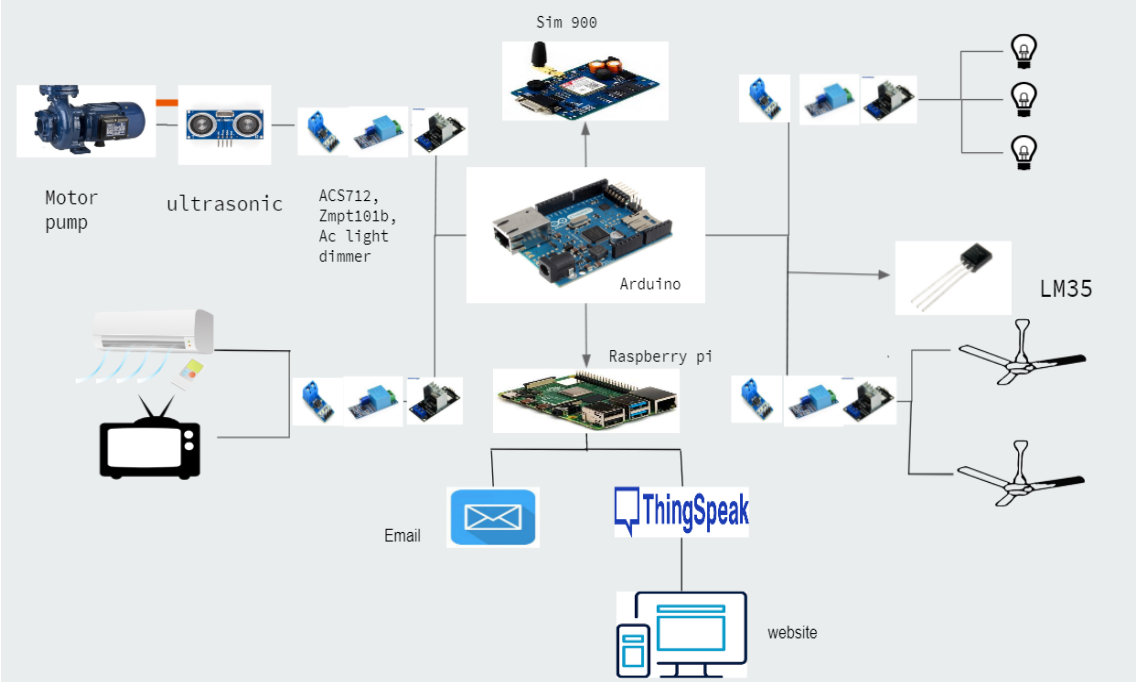

Figure 2: Model

3.1 Work Done

- We made a prototype of smart energy management system.
- We used Arduino and Raspberry pi as micro controller and LM35 (Temperature Sensor),HC-SR04 (Ultra Sonic sensor), ACS712 (Current Sensor), Voltage Divider circuit (Voltage measurement), Light Dimmer Module, bulb and Sim 900.
- Arduino and Raspberry Pi communicates through serial communication.

- All sensors are connected to arduino. It collects the data and sends it to Raspberry Pi and Raspberry Pi reads from Thingspeak and sends it to arduino.
- Calculation of electricity bill is done by Raspberry Pi based on slabs of power billing.
- Electricity bill is intimated to user via mail and text message.
- Raspberry pi reads data related to switches from ThingSpeak and uploads data related to electricity consumption.
- Django reads data from Thingspeak and data is shown on website.
- When we use controls such as switches and intensity in website then that control data is transferred into Thingspeak by Django.
- In Website power consumed and and consumption trends will be shown.
- User can also search for feedback of a device which they wanted buy and post a review for a device he posses.

3.2 Website Flowchart

Index page → *Signup* → *login* → *links* → *devicenames* → *homepage* → *dashboardpage*

3.3 Data Flow diagram:

3.3.1 Arduino to Website:

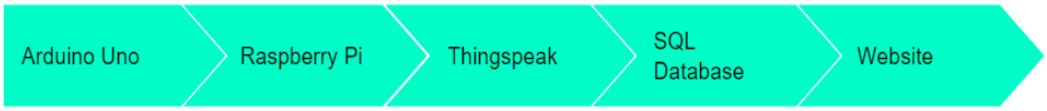

Figure 3: Data flow from Arduino to Website

1.Data transfers in this direction when power consumption data is sent from arduino to Website.

2. ACS712 and Voltage divider circuit measures current and voltage. Arduino calculates instantaneous power and sends it to raspberry pi.

3. Raspberry pi multiplies power with time difference between each successive message received. Now power used is sent to cloud.

4. Django framework reads data from thingspeak and writes it to Sqlite db which is then displayed on website.

3.3.2 Website to Arduino:

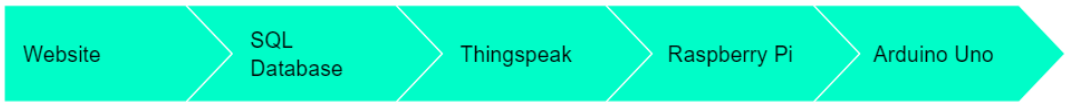

Figure 4: Data flow from Website to Arduino

1. Data is sent in this direction when we use switches and intensity control.

2. When a switch is used then django reads data and is written in Sqlite db and thingspeak cloud platform.

3. Raspberry pi reads data from thingspeak and sends it to arduino.

4. When arduino receives data it checks if device control switch is on or off.

5. Arduino then uses Light dimmer module to turn on or off device with specified intensity.

3.4 Tools Implemented

3.4.1 Thingspeak:

ThingSpeak is an open-source Internet of Things application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network (source: wikipedia). It is an IoT analytics platform service that allows you to visualize and analyze live data streams in the cloud.

In the free version we can create 4 channels and each channel has 8 fields for data entry. To write and read data we have to use "write a channel feed" and "read a channel feed" links which we are getting from our users via links form in the website.

3.4.2 Django Framework :

We are using Django as the back end to our website. Django is a python based free and open source web framework that follows the model-template-views architectural pattern (source: wikipedia).

Model: model is where we create a table and fix the datatype of each row of table.

Template: templates is where we store our HTML pages.

Views: views file contains functions for every work in our website. This file tells what to do when we click a button, give a data entry etc;

3.4.3 SQlite Database :

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private.(from: SQlite website)

Update button in the home page can be used to update data in the sqlite database with latest data from Thingspeak which will be displayed in the web page. In sqlite we create tables to store our data. Here are some tables where we stored our data:

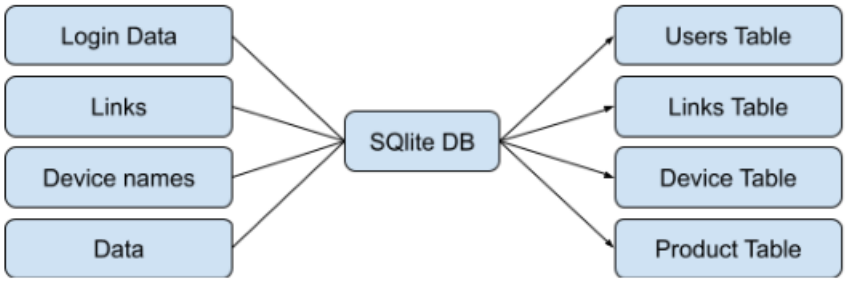

Figure 5: Tables in Database

3.4.4 Ngrok:

Ngrok is a cross-platform application that exposes local server ports to the Internet. It is used for web hosting. ngrok provides a real-time web UI where you can introspect all HTTP traffic running over your tunnels. Replay any request against your tunnel with one click.It is useful for run personal cloud services from your home, demo websites without deploying, test mobile apps connected to your locally running backend. (from: Ngrok website)

3.5 Hardware components:

3.5.1 Arduino Uno:

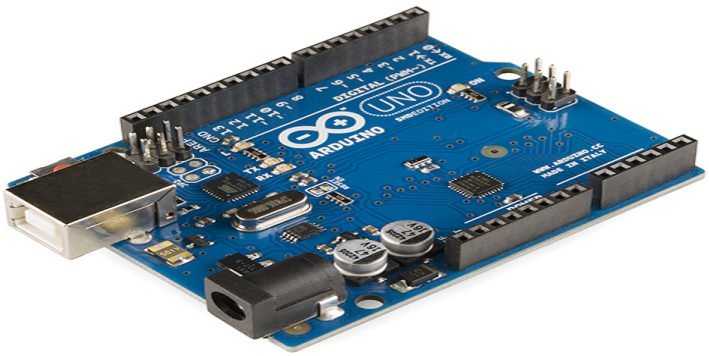

Figure 6: Arduino Uno

Arduino is an low-cost, flexible, and easy-to-use programmable open source micro controller board with sets of digital(14 I/O) pins and analog(6 I/O) pins that are used for various applications. It can be connected to computer using cable and programmed. We use arduino in our project to gather data from sensors, control appliances and send collected data to raspberry pi using usb cable. (Image: wikipedia)

Specifications:

- Operating Voltage : 5V
- DC current per I/O pin : 20 mA
- DC current for 3.3V pin: 50 mA
- Clock Speed : 16MHz
- Dimensions : 68.6 mm x 53.4 mm

3.5.2 Raspberry pi:

Figure 7: Raspberry pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. Raspbian OS is official Operating System it is free to use and efficiently optimized to use with Raspberry Pi. Raspbian

have GUI which includes tools for Browsing, Python programming etc. We should use SD card to store the OS. Raspberry Pi also provides on-chip SPI, I2C, I2S and UART modules. Primary function raspberry pi is to communicate data between arduino and cloud. It is also used to send mail with details regarding electricity bill.(Image: wikipedia)

Specifications:

- Supply : 5V, 2.5A
- Data storage : MicroSD card
- Network : Ethernet, WiFi and Bluetooth
- Peripherals : 17 x GPIO
- Processor : 64 bit quad core ARM cortex
- USB 2.0 ports : 4
- Dimensions : 85.60mm x 53.98mm x 17mm

3.5.3 LM 35:

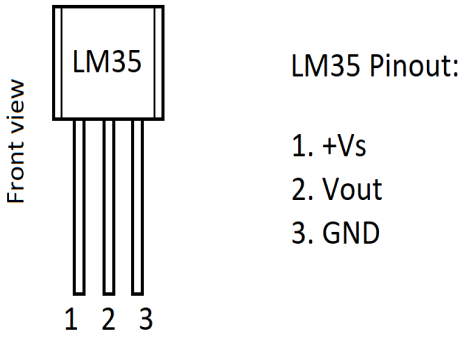

Figure 8: Temperature sensor

It is a temperature sensor that outputs an analog signal which is proportional to the instantaneous temperature. In LM35 sensor as the temperature changes, the voltage across a diode changes at a known rate and precisely amplifying the voltage change, it

is easy to generate an analog signal that is directly proportional to temperature. It does not require external calibration.(Image: link)

- Operating Voltage : 4 V to 30 V
- Linear + 10-mV/°C Scale Factor
- Accuracy: 0.5°C at 25°C and 1°C from -55°C to 150°C
- Low cost temperature sensor

3.5.4 HC-SR04:

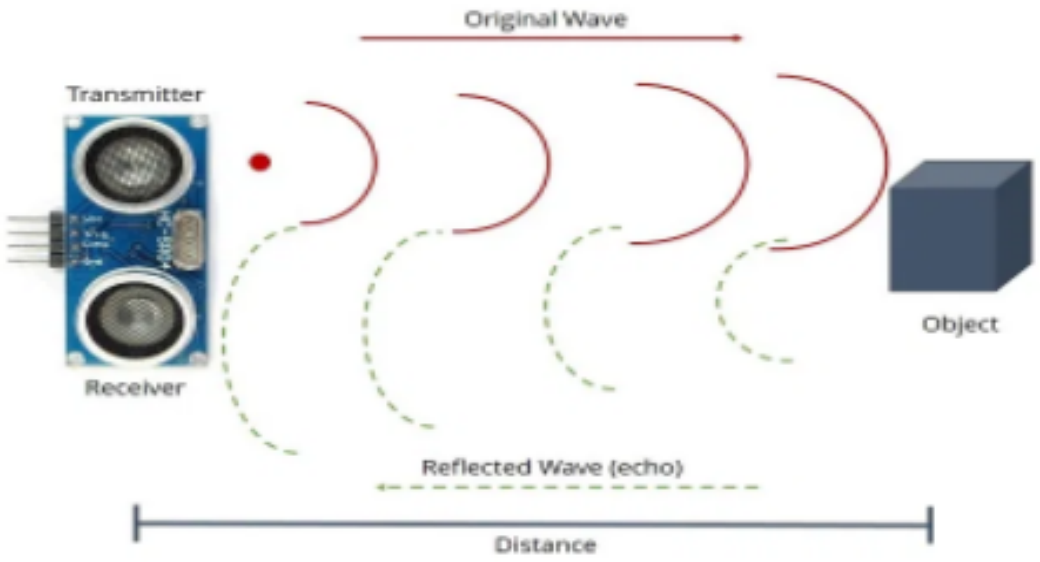

Figure 9: Temperature sensor

It is a sensor that can measure distances through ultrasound which travels through air. If ultrasound hits an object or obstacle on its path, it will then bounce back towards the sensor. We are using [HC-SR04] with pin configurations VCC, TRIG, ECHO and GND.(Image: link)

Specifications:

- Power supply : 5V DC supply

- Working Current : 15mA
- Working Frequency : 40Hz
- Ranging Distance : 2cm - 400cm [depends on sensitivity adjustment]
- Resolution : 0.3cm
- Trigger Input Pulse width : 10us
- Dimension : 45mm x 20mm x 15mm

3.5.5 ACS712:

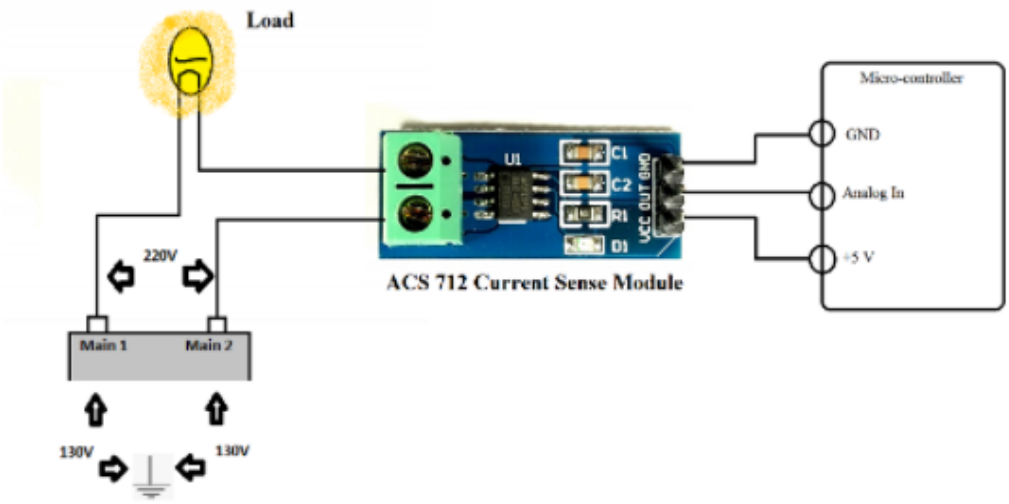

Figure 10: Current sensor

The ACS712 module uses the Hall Effect Principle to measure current. It has Pin configurations VCC, Output, GND, Wire In(T1), Wire Out(T2). Current flows through the onboard hall sensor circuit in its IC The hall effect sensor detects the incoming current through its magnetic field generation. Once detected, the hall effect sensor generates a voltage proportional to its magnetic field that's then used to measure the amount of

current(Image: link)

Specifications:

- Able to measure both AC and DC current and can be isolated from Load.
- Sensor: ACS712T ELC-05B
- Offers 185mV/Amp
- current range: +5A to -5A
- Total output error of 1.5% at TA = 25°C
- Power: Pin 5V power supply, on-board power status LED
- Dimension : 3.1 cm x 1.2 cm x 1.1 cm

3.5.6 Light Dimmer module:

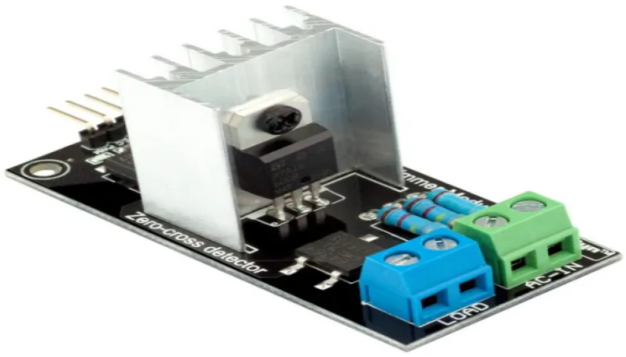

Figure 11: Light Dimmer module

The AC light dimmer module allows us to vary the power of an alternating current. It can be used for operations like varying the brightness of a lamp supplied with 220V or to vary the speed of a fan. The AC dimmer consists of a triac (equivalent to a DC transistor) and a phase zero crossing detector to synchronize the voltage variation and the phase of the AC current. The module is connected to the circuit via the AC-IN terminal

block and the bulb is connected to the LOAD terminal block.(Image: link)

Specifications:

- AC Voltage: 220V/110V; 50/60Hz
- Logic: 3.3V/5V
- Peak Current: 16A
- Environment: For indoor and outdoor use
- Operating Temperatures: -20°C to +80°C
- Dimension : (L x W x H) mm: 55 x 22 x 32

3.5.7 Sim 900:

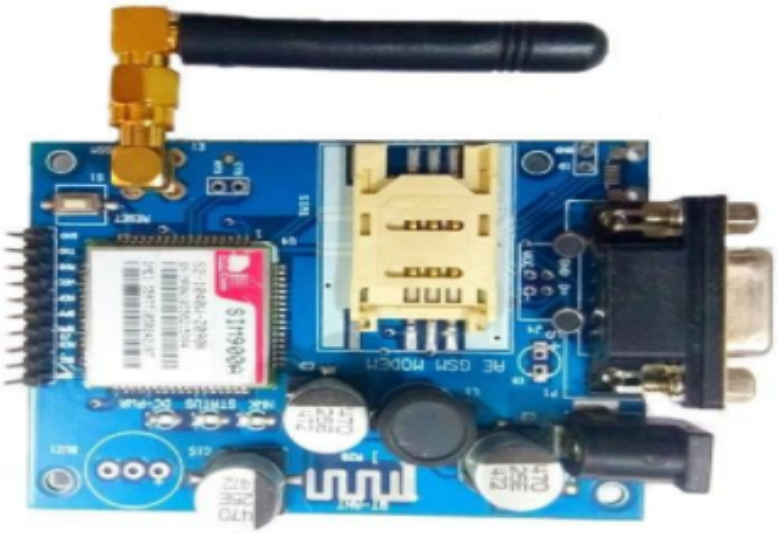

Figure 12: Sim 900

The SIM900 is a complete Quad-band GSM/GPRS solution in a SMT module which can be embedded in the customer applications. Featuring an industry-standard interface it cab be used for voice, SMS, Data, and Fax in a small form factor and with low power

consumption. The SIM900 uses UART protocol to communicate with an Arduino. The chip supports baud rate from 1200bps to 115200bps with Auto-Baud detection. Using (RX,TX) pins we can communicate with arduino.(Image: link)

Specifications:

- Sensor: SIM900 A
- Single supply voltage: 3.4V – 4.5V
- Power saving mode: Typical power consumption in SLEEP mode is 1.5mA
- Operating Temperature: -30^{o}C to +80^{o}C
- Features keypad interface, display interface, Real Time Clock and Communicates using AT commands
- Supports single SIM card
- Dimension : 24x24x3mm

3.6 How Electricity bill is calculated

In this section we will see how power consumed is calculated by using current and voltage measurement form arduino and time and how electricity bill is calculated

$$\text{Instantaneous Power (P)} = \text{I * V}$$

$$Units = (\frac{P_i + P_f}{2}) * (T_f - T_i)$$

$$\text{Bill} = \text{power Units consumed * charge}$$

Slab 1. Power used is less than 50 Unit charge per unit is 1Rs

Slab 2. Power used is more than 50 Unit but less than 100 Unit charge per unit is 2Rs

Slab 3. Power used is more than 100 Unit but less than 200 Unit charge per unit is 3Rs

Slab 4. Power used is more than 200 Unit charge per unit is 4Rs

3.7 Arduino code

Explanation of btp_arduino_code

1. First we define all the variables and the pins we use in code.

2. In void setup we define baud rate to communicate with raspberry pi and define if pins mentioned above are input or output pins.

3. void SendMessage() function is used to send text message of electric bill to user.

4. Code in void loop will be repeated until we terminate the code.

5. trigpin, echopin are used in ultrasonic sensor. We use ultrasonic sensor to measure water level in tank. Using if statement we had written code such that if waterlevel is less than 70% tank will turn on and if waterlevel is grater than 90% motor will turn off.

6. Serial.available() this helps us check if there is any serial message from raspberry pi if present it will set values to switches and intensity controllers.

7. Next we read current and voltage values from respective sensors 10 times and take average to calculate instantaneous power.

8. Serial.print is used to send data from arduino to raspberry pi we send switch details and power measurement to raspberry pi for further processing of data.

9. Now we act on data from raspberry pi such as switch, intensity and message details turning devices on and off with required intensity and send text message via gsm.

3.8 Raspberry pi code

Explanation of btp_raspberry_code

1. First we import all the modules we use in code such as requests, time, yagmail etc.

2. serial.Serial('/dev/ttyACM0', 9600, timeout=1) is used to establish usb port and start serial communication between arduino and raspberry pi. We initialize values of variables we use and start timer.

3. def mailing() function is used to send mail electric bill to user along with graph of power consumption.

4. Code in while(True) loop will be repeated until we terminate the code.

5. d.strftime("%m") will return current month and requests.get(..) is used to download data such as switch and intensity control from thingspeak.

6. Data downloaded will be printed as it is useful for debugging.

7. if d!=month this section of code is used to send mail when a new month arrives we store month value at 2 variables when month changes the 2 data in 2 variables will be different which will be used to find if a month changes and send message..

8. Value list is list of data that we want to send to raspberry pi and we join data in list using join function and name it send_string and send it to arduino using ser.write command and code it using utf-8 coding.

9. Receive_string is read using ser.readline(), decode it using utf-8 and split the string at ',' and make a list from data in receive_string.

10. We discard data if it is faulty else we read power data from list.

11. Now we see time to calculate time taken to complete one looping of 'while' which is 'gap' and we start timer again for next loop.

12. We take average of current instantaneous power(pn) and instantaneous power from previous while loop(po) and calculate total power units consumed and store pn as po for next while looping.

13. Electricity bill is calculated based on charges per slab and data is uploaded to thingspeak.

4 Results

1. We made a prototype of smart energy management system.

2. We calculated power consumption using readings of current and voltage and display it on website. System can can turn on and off water pump automatically and control intensity of device based on parameters set on website.

3. Mail and text message will be sent to user with electricity bill.

4. User can see detailed graphs of power consumption in website and control devices over internet.

5. Website has been hosted online using Ngrok.

4.1 Pages in website

4.1.1 Index page

Index page gives us basic information about our project idea .

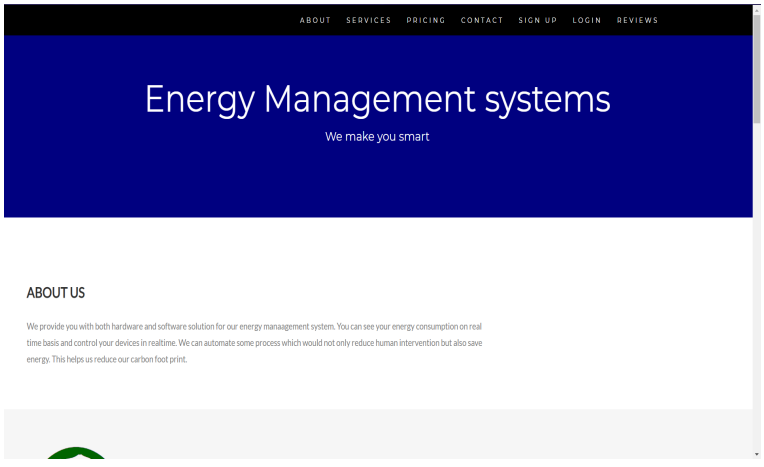

Figure 13: Index page

4.1.2 Sign up page

Here user can sign up with an unique username which he can further use to login.

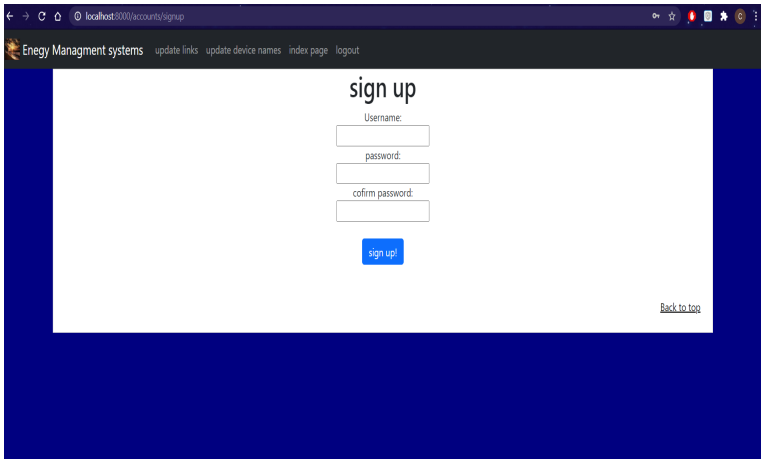

Figure 14: Sign up page

4.1.3 Login page

This page is used for logging in.

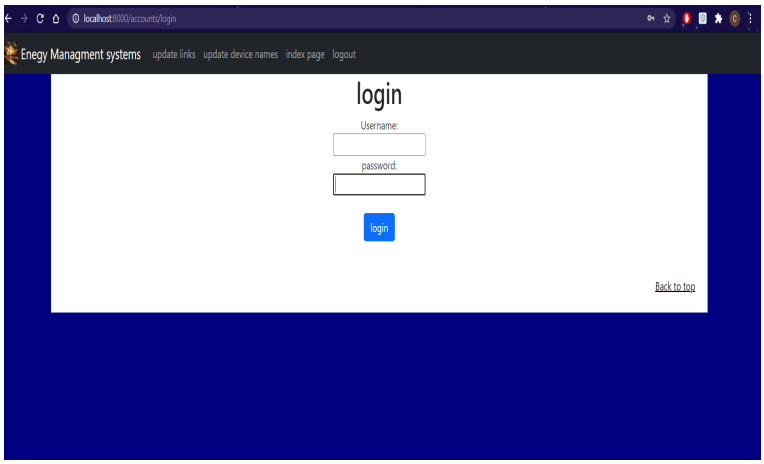

Figure 15: Login page

4.1.4 Add links page

In this page we have to provide links which connect databases to our cloud platform thingspeak and channel ID which will be used to display graphs.

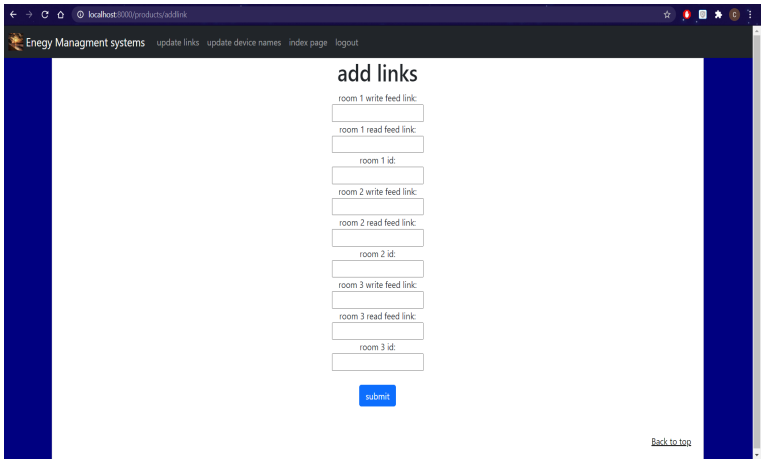

Figure 16: Add links page

4.1.5 Add device names page

Here we can provide names of our devices in that particular room which will be used to display current usage stats in website.

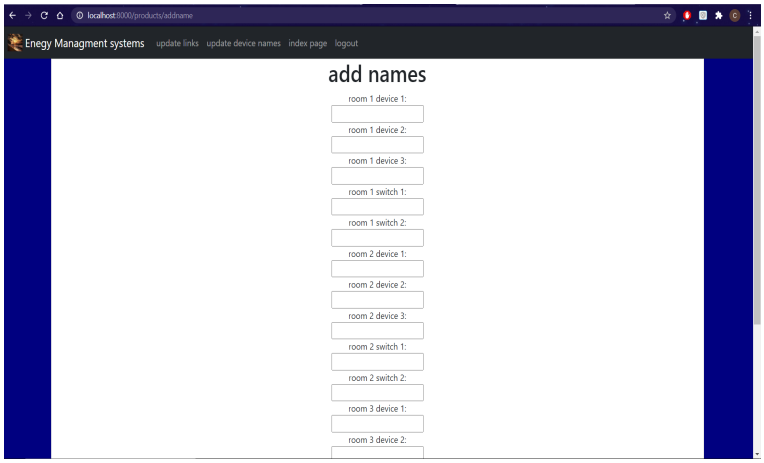

Figure 17: Add device names page

4.1.6 Home page

In this page we will show power consumed and electricity bill and the slab of billing. Basic information about power consumption in 3 rooms will be shown, 2 switches per room and option to control intensity of light and speed of fan are also present in this page.

Update button shown in figure below can be used to get latest data from cloud platform and switch for remote control option can be used to turn on and off remote control option.

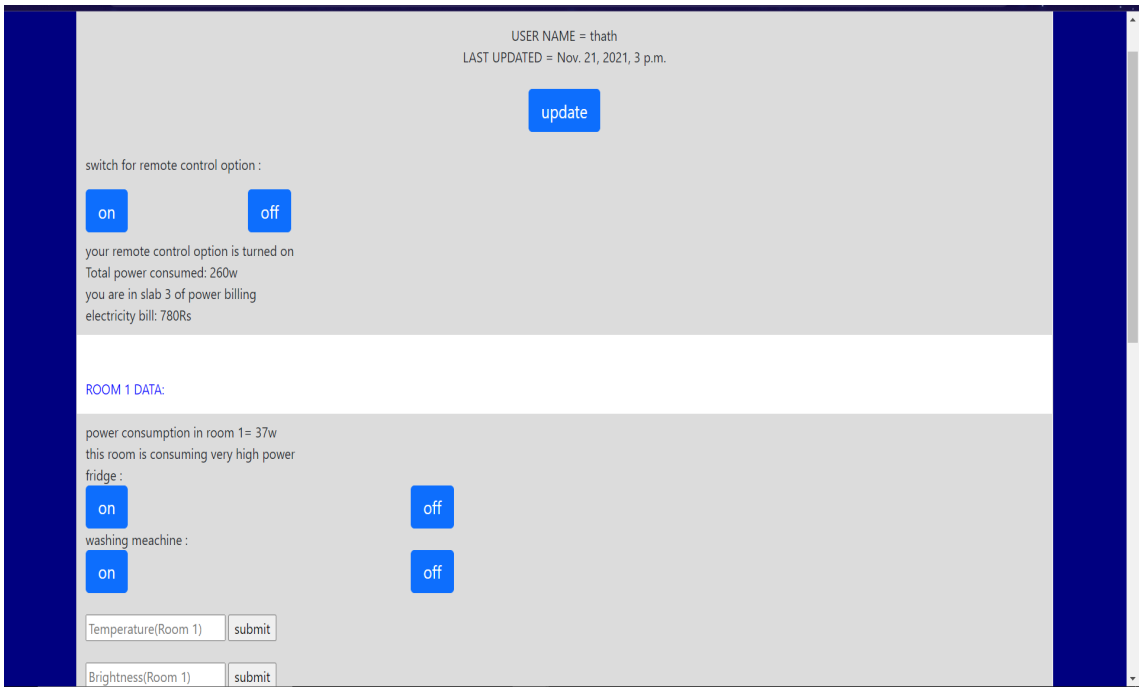

Figure 18: Home page

4.1.7 Data Dashboard page

Each room has a dashboard page with visualisations of our power consumption trends.

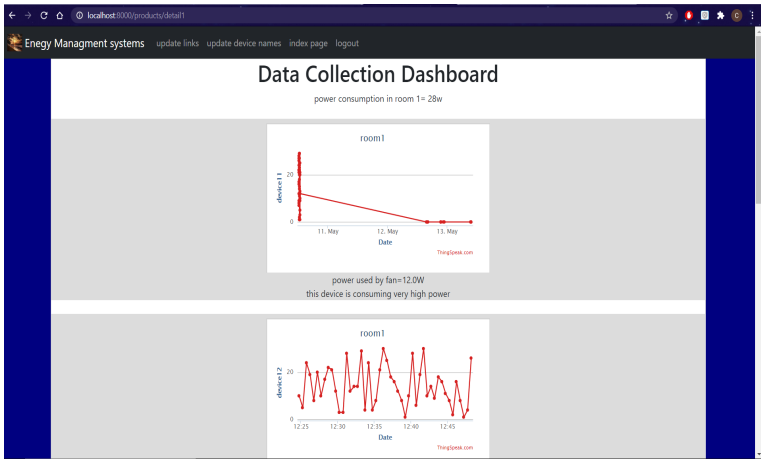

Figure 19: Data Dashboard page

4.1.8 Reviews page

In this page user can see reviews of electronic devices from users of the device. User can search for a device and also sort them. Users can upvote a review if they like it and are logged in.

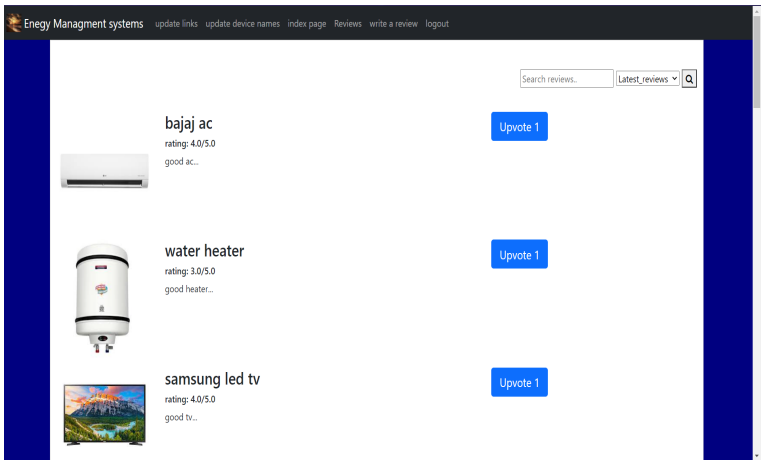

Figure 20: Reviews page

4.1.9 Write reviews page

This page is used to write review only logged in user can write a review.

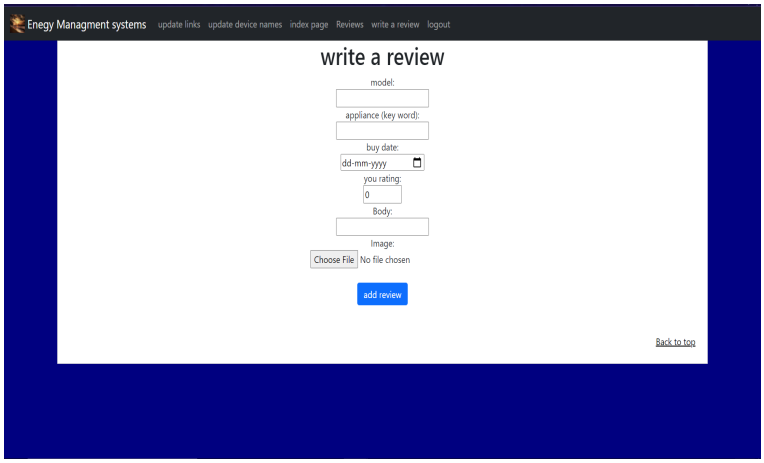

Figure 21: Write reviews page

4.1.10 Detailed review page

In this page user can see a detailed view of a review with information such as buy date, rating, votes, review etc.

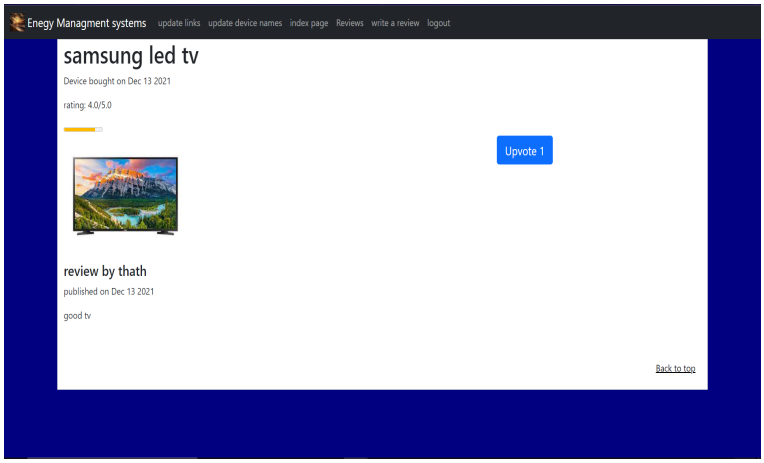

Figure 22: Detailed review

4.2 Hardware

- The Figure 23 shows prototype of smart energy management system.

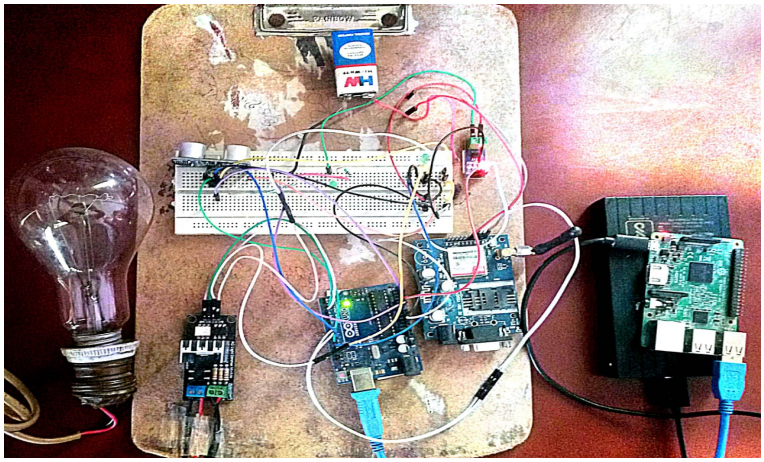

Figure 23: Prototype

- In Figure 24, we can see data communication between arduino and raspberry pi when power consumption is '0'. Where 0, 0, 60, 32, 1 is the data sent by raspberry pi where 0, 0, 1 s data related to switches and 60, 32 is related to intensity. The list data sent from arduino to raspberry. 10.122951 is time between every consecutive power reading and '0' below is power consumed.

```
data
0
0
60
32
1
['0', '0', '60', '32', '0', '0.00', '79.00', '1']
10.122951761999502
[0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
0.0
0.0

data
0
0
60
32
1
['0', '0', '60', '32', '0', '0.00', '92.00', '1']
10.036395568000444
[0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
0.0
0.0
```

Figure 24: Data communication

- In Figure 25, Power vs Time from 0 to 30 we can see graph is increasing and from 30 to 50 graph is stagnant as power is not consumed and again increased from 50 to 70.

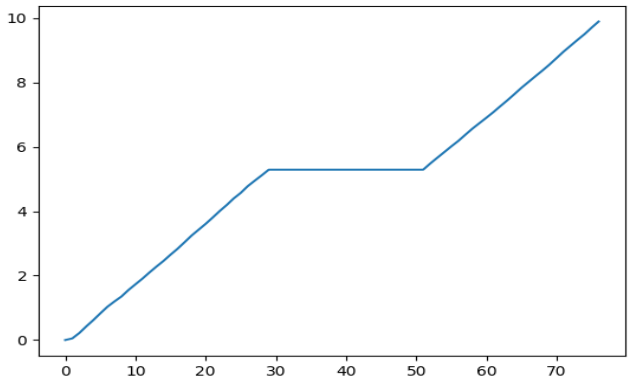

Figure 25: Power consumption graph

- In Figure 26, we can see mail sent by Raspberry pi while initiating where the power consumed is zero.

Figure 26: Mail

- In Figure 27, we see the code to control temperature. Here arduino changes rotation speed of fan based on current temperature of room and temperature set by user on website.

```
if (temp1>temp){
  if (fan_speed!=0){
    Serial.print("reduce speed");
    if (temp1-temp>5){
    fan_speed=fan_speed-1;
    }}
  }
if (temp1<temp){
  if (fan_speed!=5){
    Serial.println("increase speed");
    if (temp-temp1>5){
    fan_speed=fan_speed+1;
    }}
  }
rot=(fan_speed*50);
```

Figure 27: Temperature control

- In Figure 28, we can see the code of automatic water pump. Where we can see ultrasonic sensor gathers water depth and arduino acts accordingly.

```
cm = (duration/2) / 29.1;     // Divide by 29.1 or multiply by 0.0343
if(cm>100){
  cm=30;
  }
//inches = (duration/2) / 74;   // Divide by 74 or multiply by 0.0135
percent =((tank_l-cm)*100)/tank_l;
/*if(percent>100){
  percent=70;
  }*/
//Serial.print(inches);
//Serial.print("in, ");
//percent=40;
Serial.print(cm);
Serial.print("cm, ");
Serial.print(percent);
Serial.print(" % full");
Serial.println();
if (percent < 70){
digitalWrite(motor, HIGH);
}
else if(percent>90){
  digitalWrite(motor, LOW);
}
```

Figure 28: Mail

4.3 Prototype can

- Calculate power using readings from current and voltage sensor.
- Turn on and off devices on internet and control intensity.
- Send mail and text messages to user regarding electricity billing.

- Turn waterpump on and off based on water level and can control speed of fans based on temperature.
- Make interaction between system and user with help of website.
- Give user better analysed graph of power consumed.

5 Conclusion

We made a prototype of smart energy management system using senors, actuators, arduino, raspberry pi, Thingspeak, django, html, css, Ngrok. This project helps us keep us track of our daily energy consumption and helps us use energy when needed and in the level needed. This project helps us take out the distance barrier in operating devices and one can operate devices from any part of world. This IoT based project helps us get smart and encourage automation in houses. Future scope of our project is..

1. Integrating system with more electronics such as AC, Television etc;

2. Buying domain name and hosting it online using paid hosting service.

3. Adding more security features and new options in website .

4. Integrating smart home features into system which would make system more advanced.

5. Improving quality of sensors used which would help us reduce maintenance and avoid any damage to electronics.

6 List of Abbreviations

- IoT - Internet of Things.
- HTML - Hypertext Markup Language.

- LED - Light emitting diode.
- DB - Database.
- API - Application Programming Interface.
- HTTP - Hypertext Transfer Protocol.
- MQTT - Message Queuing Telemetry Transport.
- GSM - Global System for Mobile communication
- SQL - Structured Query Language.
- SEM - Smart Energy Meters.
- M2M - Machine to Machine.
- UI- User interface
- UX- user experience
- LDR - Light Dependent Resistor.
- ACS712 sensor: Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor.

7 Resources

- Template for index page: index page
- Course done for website: udemy django course
- Course done for Raspberry Pi: Udemy Raspberry Pi For Beginners

8 Codes:

Drive folder of codes: btp work

9 First Report

Drive link for report 1: btp 6 sem report