BRAC University

Department of Computer Science and Engineering

CSE 341: Microprocessor

Section: 01

Project Title: Creating a system which controls and monitors a device remotely using Dragino Yun and Temboo

Team Members' Info:

Name	Student ID	Signature
Sadman Araf	**17101354**	
Sajid Hossain	**17101352**	
Shaima Khandoker	**17101003**	
Sayed Us Sadat	**17101364**	

TABLE OF CONTENT

Abstract: One thing people usually want to do in home automation is follow the energy consumption of their electrical devices and turn them on and off remotely, for example, using their smartphones or tablets. So, we are creating a system which focuses on energy management by creating a project to switch a device on and off (like a lamp), measuring its energy consumption and storing this data to the Web. We are going to interface a current sensor to measure the energy consumption of the device that is connected to the project. This project will also be able to switch the device on and off remotely, and we are going to build an interface for you to control this switch from your computer and mobile device using the feature of the Arduino Yun to connect to the Web using the web service Temboo.

Keywords: Home Automation, IOT, Energy Monitoring and Control, Yun Shield, Temboo**.**

1. Introduction

Power crisis has been a long-term issue for Bangladesh and it is proliferating every day. There have been many attempts to solve this issue but the results were not very effective. One of the many reasons of power crisis is people's lack of consciousness towards saving electricity. There have been many promotional campaigns and advertisements to encourage people about saving electricity by turning off their lights and fans when they leave their room, turning off devices when they are not being used. But this also did not help the cause as people are slave to their habits. Observing all these, motivated us to develop a system where the users will be notified about their power consumption and also will give them the ability to remotely access their devices. This will make the users conscious about their power usage and they will also be able to turn off their device from anywhere if they forgot about turning off their devices in the first place. We will be implementing the demo using a bulb but it can be used to control other low powered devices. So, we will be using Home Automation along with IOT concepts to achieve efficient energy consumption. [1]

In section 2, we have provided the Literature Review about the technologies that we will be implementing in our project. Then in section 3, we have compared some related works with ours. Afterwards, in section 4 we have provided the hardware specifications followed by a brief detail about our competitors in section 5. In section 6 we have discussed about our proposed system and our implementation. Finally, we have ended with conclusion in section 7.

Our Contributions:

- Monitor energy consumption of their electrical devices.
- Remotely turn on/off their devices from anywhere by connecting to the internet.
- Creating a sophisticated web site to control device
- Storing data in Google Docs spreadsheet
- For research purpose, our power consumption data may be used to identify trends in the use of domestic appliances.
- Many power monitoring devices uses GSM or Bluetooth to remotely access their device, we will be using REST API to command our device.
- Data generated can be used for data analytics

2. Literature Review

Home Automation is to facilitate us perform a function without our intervention by interconnecting different electrical devices in a household. This Home Automation may be a variety of things, for example, it may be able to control light remotely, it may be able to set your room temperature and lights according to your preference without your intervention. [2] Home Automation has been rapidly growing because people want ease and comfort. [3] This Home Automation System connects the digital technology and computers to the appliances at home and manages it, thus making it user friendly webpage interface that can be accessed using a Personal Computer or a Smartphone. [4] It enables the users to realize the access control of the in-home appliances by the mobile devices with the help of cloud computing. [5] The results obtained from the consumption of energy usage in a home will be used to monitor the energy saving and all electric bills from annual electricity consumption used by all the appliances in all homes provided by the power plants and grid stations to estimate the energy consumption. [6] In residential sectors, which are a major consumer of energy, Home Automation System is a way to identify the problem and improve energy efficiency which has been a main global challenge. [7]

We will be using Representational State Transfer (REST) to remotely access our device from a server. REST [8] is used as a Web-based interaction for controlling household appliances using Web techniques such as HTTP caching and push messaging. Also, we will be designing a web-based graphical user interface to manage our device where users will be able to control various home appliances. Couple of scientists have additionally introduced utilization of Web administrations, Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) as an interoperable application layer to remotely get to home robotization frameworks. [9]

We are also going to implement The Internet of Things (IoTs) which can be defined as the connection of everyday devices such as mobile phones, smart TVs, smart watch, smart fridges to the Internet where the devices are able to communicate intelligently forming an interactive communication system between people and things. [10] The demand for IoTs have increased in the last few years because of its new dimension being introduced to the world in terms of communication technologies. It is expected, according to [11], that the number of devices using Internet will increase from 100.4 million in 2011 to 2.1 billion by the year 2021, growing at a high rate of 36% per year. It is expected the ratio of machine to machine (M2M) connections over mobile networks will increase from 80% in 2011 to 93% by 2021 since the cost of machine to machine over mobile networks are comparatively cheaper than fixed networks. Developing the Internet of Things will revolutionize multiple sectors from energy, transportation, healthcare to nanotechnology. A lot of home appliances such as air condition, lightings, entertainment systems are now connected to the Internet so that they can remotely controlled using mobile phones or Tablets. The technology is not only limited to controlling devices but it can also be used to monitor amount of energy consumption and also maintain certain temperature. [12] Hence, this will be able to reduce cost and save energy which is one of the major concerns for the world today.

3. Related Work

3.1 Remote Energy Monitoring, Profiling and Control Through GSM Network

This is an improvement of a GSM-based energy monitoring, profiling, and control framework which coordinates digital energy meters introduced at shopper unit with an electric supply organization's vitality checking framework. Single stage or three stages advanced electric meter can be utilized with indigenously created extra transmission module, which takes the meter perusing and uses the GSM system to transmit the vitality utilization perusing utilizing Short Message Service (SMS) back to the energy supplier. At the supplier end, a vitality observing framework is utilized to deal with all gotten meter readings, figure the charging cost, update the database and keep up a vitality utilization profile for every client. Different alarms and control can likewise be created by the provider.

Home energy system which detects the current values on a real-time basis calculates the instantaneous power and transfers the values to the server utilizing the Wi-Fi module. This Energy Monitoring System will screen control utilization of devices associated in the system and update this data on a server. Depending upon energy consumption and need of keeping specific gadget ON/OFF client can switch ON/OFF any device in the system. In this framework, each device or application is getting power supply through current sensor. Current sensor ACS712 will detect the present coursing through it and will send this information to Arduino Uno. Arduino Uno will update this data on a separate page where a client can see control utilization of different gadgets in the wake of reviving the page. Additionally, relying upon the need of keeping gadget ON/OFF and control utilization client can switch ON/OFF any gadget. The power utilization information on site page will be refreshed consequently by invigorating the website page.

Instead of an SMS sent to the supplier, we have designed a project where the consumers will be able to connect to the web using a web service called Temboo and they will be able to monitor and switch the devices on/off apart from reading the energy data. We have used the same Current sensor to sense the current flowing through the devices This sensor delivers an analog signal as an output, which can then can be easily converted to the corresponding current on the Arduino Yun board and value of power will be given by the Yun Board. All these data will be sent to Google Docs where energy will be calculated.

Disadvantages

- They have used GSM to collect the data.
- GSM is subject to Electromagnetic and Other Interference.
- These Interference can obstruct or degrade the performance of microwave signals.

3.2 Bluetooth Based Home Automation System Using Cell Phone

In this project, they have designed a minimal effort yet adaptable and secure cell phone-based home automation framework. The structure depends on an independent Arduino BT board and the home machines are associated with the information/yield ports of this board by means of relays. The communication between the mobile phone and the Arduino BT board is remote. This framework is intended to be easy and versatile enabling assortment of gadgets to be controlled with the least changes to its center. The communication between the cell phone and the Arduino BT board is remote. Extra devices can be associated with the framework with little alterations. Since the cell phone script is written in Python, it is portable and can keep running on any Symbian Operating System platform. Secret key assurance is being utilized to just permit approved clients from getting to the machines at home.

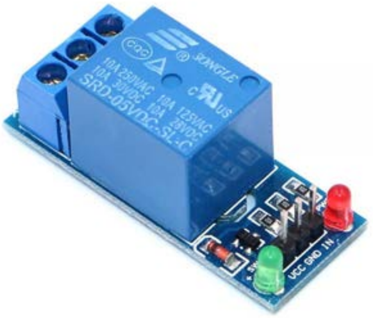
Our project will be a similar type as the users will be able to control their devices remotely and there will also be a secure login system which will enable only the owners to get access to the devices.

Disadvantages

- They have used Bluetooth.
- Bluetooth has a short range of 10-15 m (which is usually shorter than Wi-Fi).
- Bluetooth consumes a high amount of power.

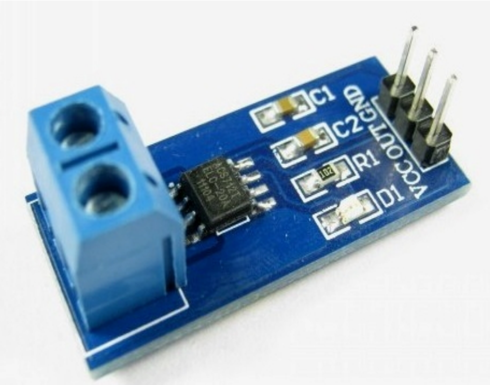
4. Required Components

4.1 Single Channel 5v Relay Module



It is basically an electromagnetic switch that is needed to switch a large voltage (110V or 230V) using a small voltage as the command signal from Arduino Board. This can switch up to 10A which is more than enough for many home appliances like lamp that we are using here uses only 130 mA. The module itself is simply a relay mounted on a printed circuit board along with the required components that are necessary to operate the relay and some large headers and traces to carry up to 10A if necessary. It uses an Omron G5LE-14-DC5 relay.

4.2 Analog Current Sensor

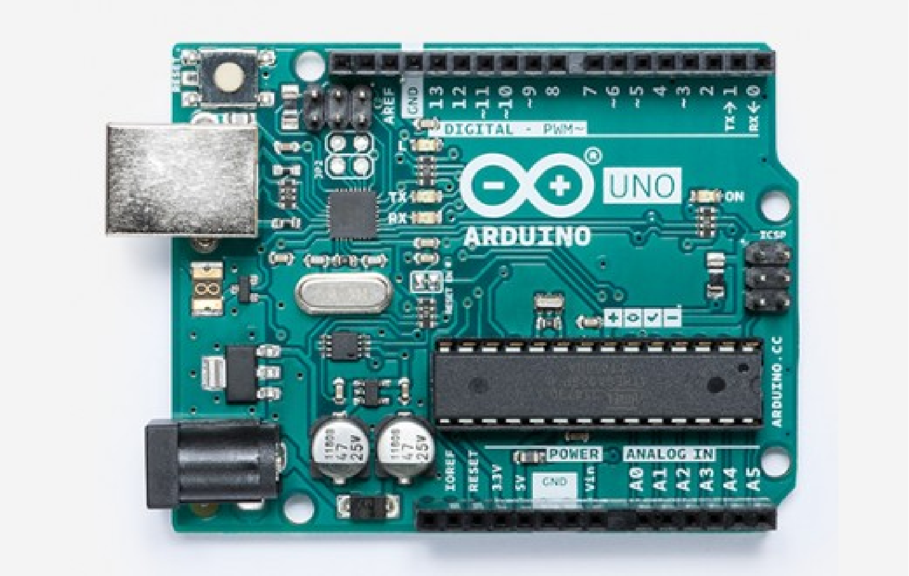


We are using a module from ITead Studio, which is basically a breakout board for the ACS712 sensor. A breakout board is simply a board that is composed of a printed circuit, the chip itself, and all the components required to make the chip work, such as resistors and capacitors. This sensor delivers an analogue signal as an output, which is proportional to the measured current. This signal then can be easily converted to the corresponding current on the Arduino Yun board. We will acquire this analogue signal using one of the integrated analog-digital converters of the Yun Board. The sensitivity of the sensor is 185 mV/A.

4.3 Power Cables

Power cables are used to connect our project to the wall socket on one side and one is used to connect the lamp to the project on the other side.

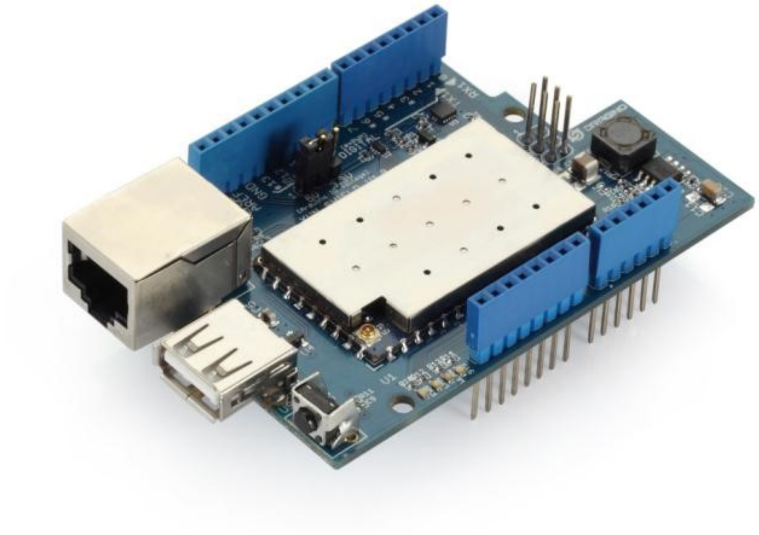
4.4 Arduino Uno



Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16

MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller.

4.5 Dragino Yun Shield



Yun Shield is one of the most powerful shields for Arduino Board. Yun Shield is designed to solve the Internet connectivity and storage issue for Arduino Board.

Yun Shield runs Open Source OpenWrt system (Same system as runs in Arduino Yun) and it is fully compatible with Arduino IDE v1.5.4 or later. Yun Shield is the ideally choice for Arduino Projects which require various internet connections and more storage. Basically, Yun Shield + Leonardo equally to the official Arduino Yun, but Yun Shield is more flexible because it can work with other Arduino board such as Uno, Duemilanove, Mega etc. And Yun Shield uses external Wi-Fi antenna which provides stability and possibility for various environments.

Specifications

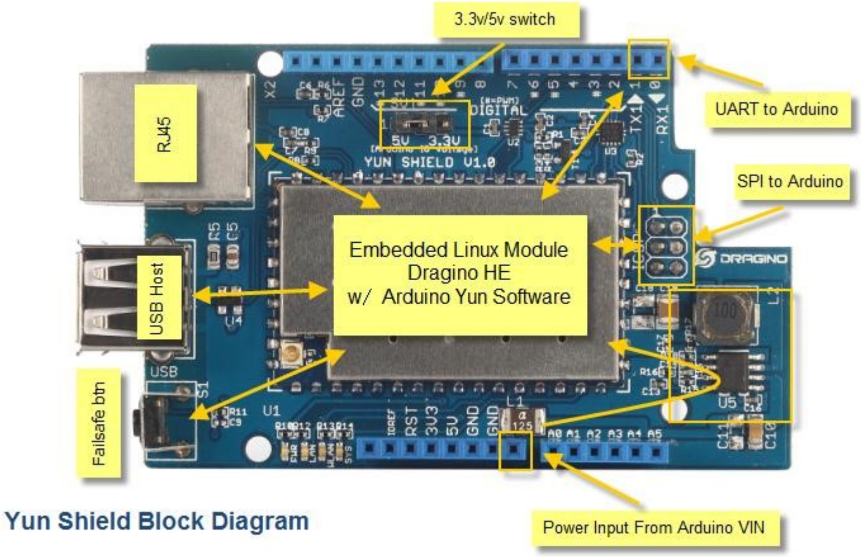
- Processor: 400MHz, 24K MIPS
- Flash: 16Mbytes
- RAM: 64Mbytes
- Power Input: 4.75v ~ 23v via Arduino VIN pin
- 1 x 10M/100M RJ45 connector
- 150M Wi-Fi 802.11 b/g/n
- External Antenna via I-Pex connector
- 1 x USB 2.0 host connector, used for USB storage or 3G connection
- 1 x Reset button
- Compatible with 3.3v or 5v I/O Arduino.

Features

- Open source Linux (OpenWrt) inside.
- Low power consumption.
- Compatible with Arduino IDE 1.5.4 or later, user can program, debug or upload sketch to Arduino board via Arduino IDE.
- Managed by Web GUI, SSH via LAN or Wi-Fi.
- Software upgradable via network.

- Built-in web server.
- Support internet connection via LAN port, Wi-Fi or 3G dongle.
- Support USB flash to provide storage for Arduino projects.
- Failsafe design provides robustly system.
- Compatible with Arduino Leonardo, Uno, Duemilanove, Diecimila, Mega

System Structure



Yun Shield Block Diagram

5. Competitors

5.1 Neurio Energy Monitor

The Neurio Home Energy Monitor introduces effectively inside the home's heap board and gives amazingly granular vitality information that is utilized to lessen energy utilization, screen sun-based execution and plan for future storage needs.

- Neurio Energy Monitors support Wi-Fi, ZigBee, XBee, and RS-485 communication protocols.
- Provides bill forecasting so you comprehend what your electricity bill will be before it arrives.
- Provides a continuous view on how your home's power is being utilized, shown on your telephone or web application.

5.2 CURB

Curb is the home energy monitoring system that helps you take responsibility for your home, condominium or loft and all the energy it consumes.

Curb connects straightforwardly to your breaker board, giving you real-time information on your energy utilization and creation, enabling you to:

- Settle on more intelligent choices about your energy use.
- Perceive unusual examples of energy use, indicating potential issues of different machines.

- Approximate your energy bill, enabling you to see the effect of your energy use on your CURB continuously.
- Works on iOS and Android devices, plus web for detailed desktop analysis.

6. Proposed System

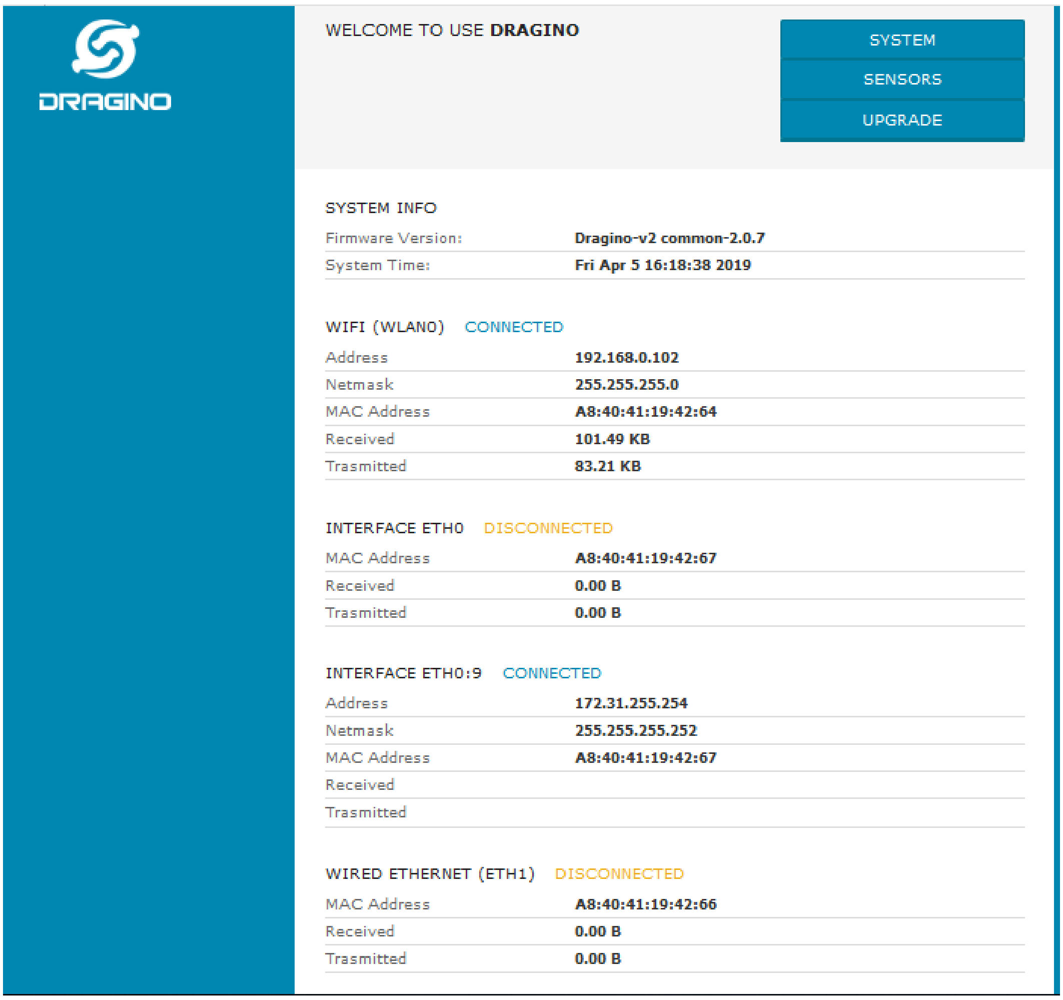
6.1 Procedure

- Connecting a relay to one of the Arduino Yun digital outputs and using the Yun REST API to command this relay from a web browser.
- Using an analogue current sensor to get a measurement of the instant current consumption rom the device that is connected to the relay, and calculate the instant power consumption from this measurement.
- Sending this data to Google Docs spreadsheet so that it can be accessed remotely from any web browser or from the Google Docs mobile app, and calculating the energy consumption and some other useful data such as total energy cost of the device connected to our project.
- Creating a simple web interface to control the lamp using a computer or any smartphone or tablet.

6.2 System Setup

a) Network Setup

The Yun Shield has a Wi-Fi interface and a LAN port. Each of them has an IP address and can be used for internet connection and device management. At the first boot of the Yun Shield, it will auto generate an unsecure Wi-Fi network called Dragino2-xxxxxx. The User can use their laptop to connect to this Wi-Fi network. The laptop will get an IP 192.168.240.xxx and the Yun Shield has the default IP 192.168.240.1. The Yun Shield runs an Open Source Linux system. If the user has a PC on the same network as the Yun Shield, the user can access its system via a Web Interface. Simply type the IP address into your browser and you will see the log in page of Yun Shield. After log in, the GUI will show the WIFI / ETH interface status. Click the Configure button and now the user can configure the device password and network parameters.



WELCOME TO USE **DRAGINO**

SYSTEM INFO

Firmware Version:	Dragino-v2 common-2.0.7
System Time:	Fri Apr 5 16:18:38 2019

WIFI (WLAN0) CONNECTED

Address	192.168.0.102
Netmask	255.255.255.0
MAC Address	A8:40:41:19:42:64
Received	101.49 KB
Trasmitted	83.21 KB

INTERFACE ETH0 DISCONNECTED

MAC Address	A8:40:41:19:42:67
Received	0.00 B
Trasmitted	0.00 B

INTERFACE ETH0:9 CONNECTED

Address	172.31.255.254
Netmask	255.255.255.252
MAC Address	A8:40:41:19:42:67
Received	
Trasmitted	

WIRED ETHERNET (ETH1) DISCONNECTED

MAC Address	A8:40:41:19:42:66
Received	0.00 B
Trasmitted	0.00 B

b) Temboo Setup

Set up an account on the web service Temboo, so the user can use the wide range of services provided by Temboo to upload data to Google Docs and to use their Google Spreadsheet API. The user must create an account first and then they can use the Temboo library called Choreo to upload the measurements of the Arduino Yun to the Web and log the data into a Google Docs spreadsheet. Below is a figure of the Temboo Dashboard we have created.

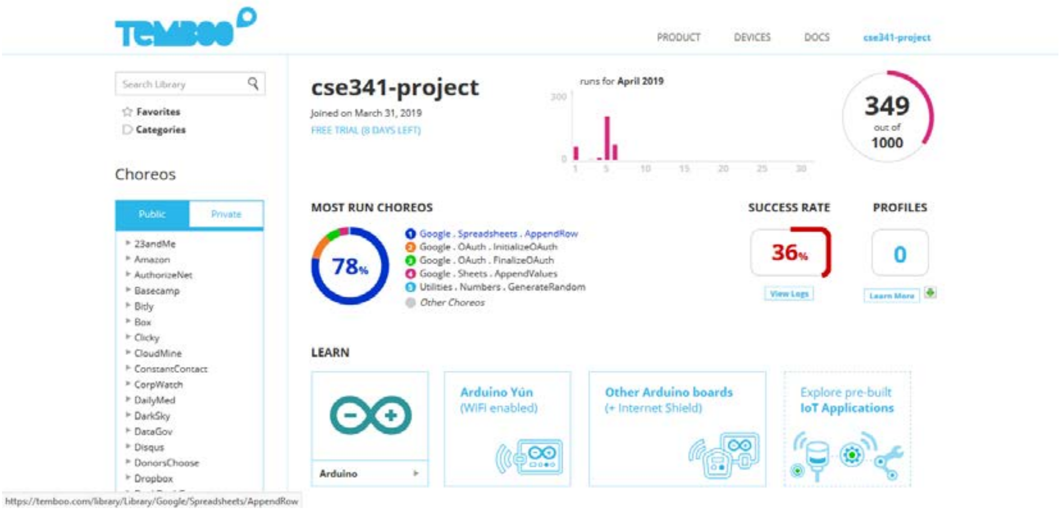
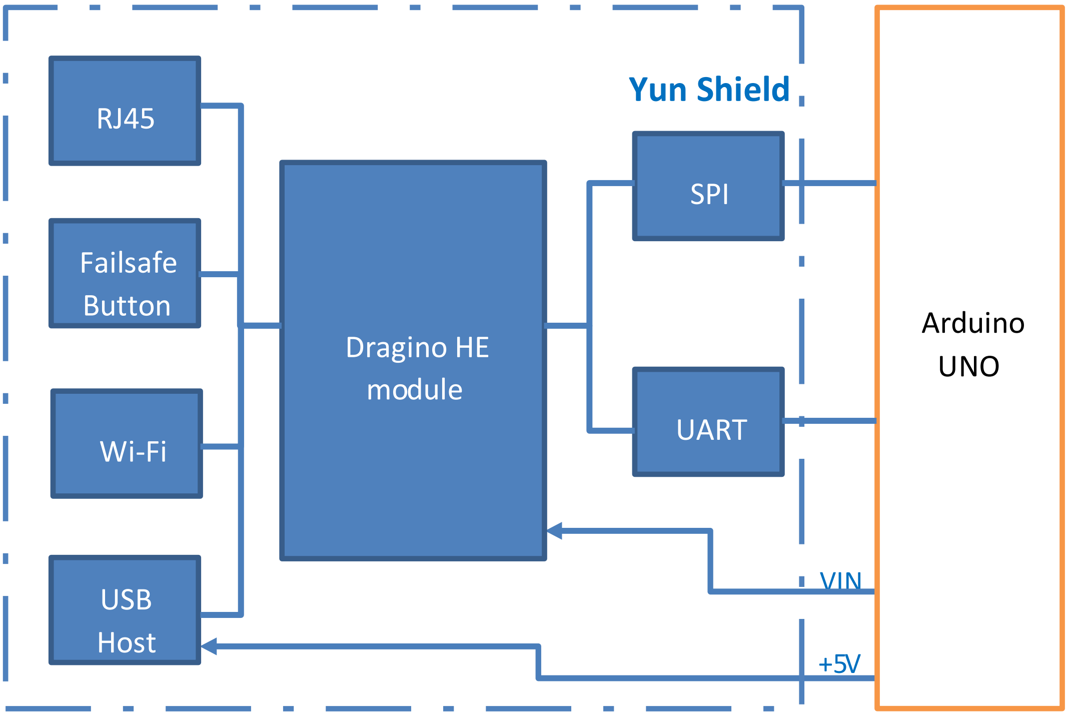


Fig: Temboo Dashboard

c) Circuit Setup

The first step is to mount the Dragino Yun shield on top of the Arduino Uno as shown in the illustration below:



Then, we are going to connect the power supply of the relay module and the current sensor. The Arduino Yun Shield only has one 5V pin. This is why we connected the 5V pins of the two modules to a small piece of a breadboard first and then connected this breadboard to the Arduino 5V pin. After this, we have to connect the ground pins of the two modules to the ground pin on the Dragino Yun board. To finish with the connection of the two modules, we need to connect their respective signal pins to the Dragino board. The relay will be controlled via pin number 8 of the Dragino Yun board, so connect the signal pin of the relay module to pin number

8 of the Yun board. The current sensor has an analog output, so it has to be connected to one analog input on the Arduino board in order to acquire the signal using one of the Yun integrated analog-to-digital converters. This converter will acquire the analog signal that comes from the sensor and transform it into digital values that range from 0 to 1023 (which correspond to a 10-bit precision). Then we connected the output pin of the current sensor module to pin number A0 of the Dragino Yun board. That's basically all for the low-power part. Now, we will focus on connecting the project to the two power supply cables so that we can plug the project into the wall plug and plug the lamp to the project. We will start by connecting the cable that will go to the wall. Finally, we connected the female power plug that we will connect the lamp to, as shown in the following image. Finally, we powered up everything by plugging our Arduino board to a power adapter. Then, we plugged the lamp or the device that we want to control in to the female power plug of the project. Then we connected the male power plug to the power socket in the wall as shown in the figure below:

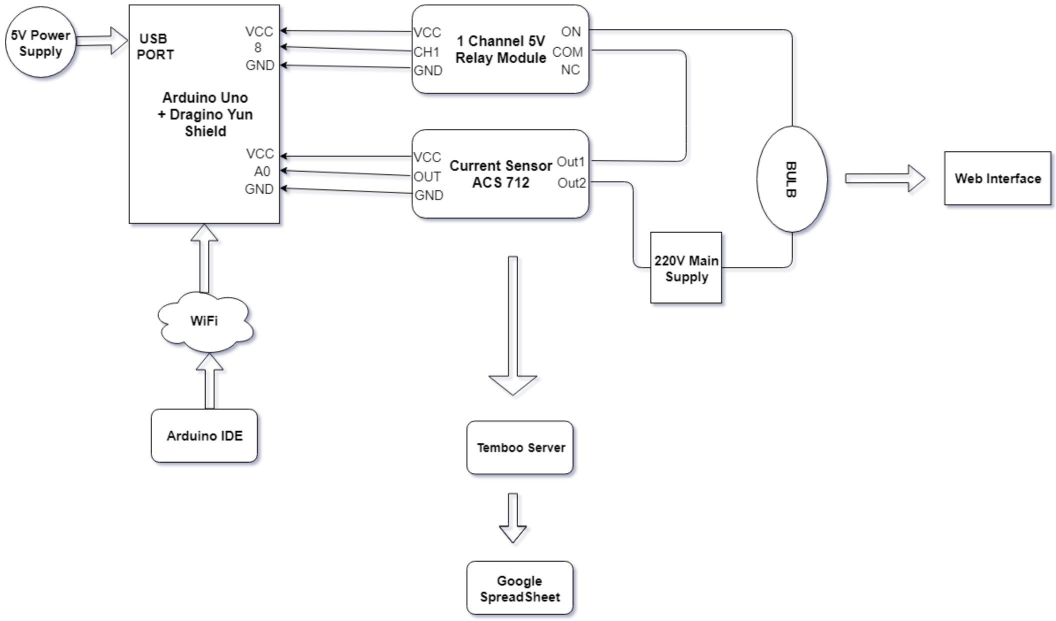
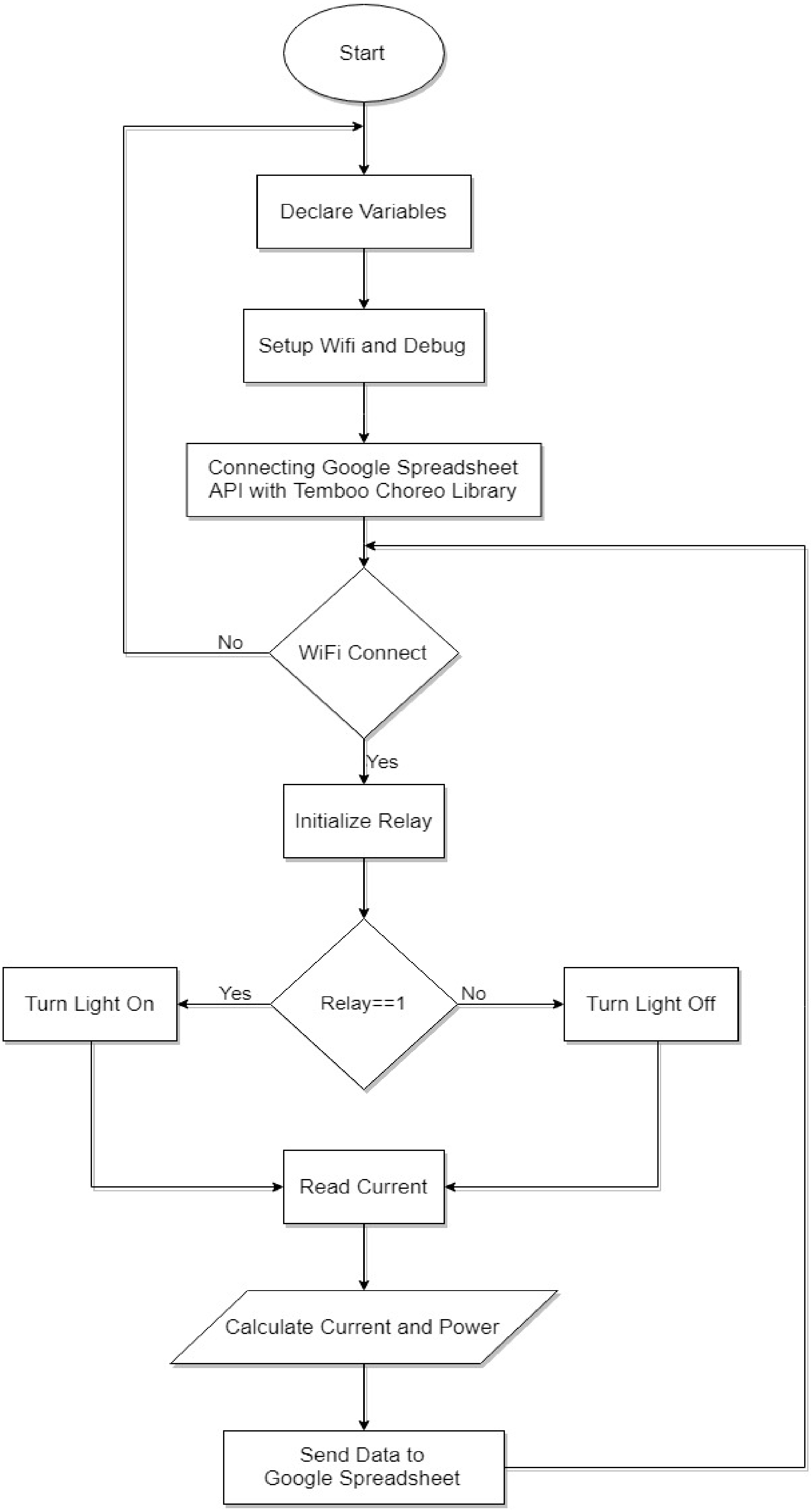
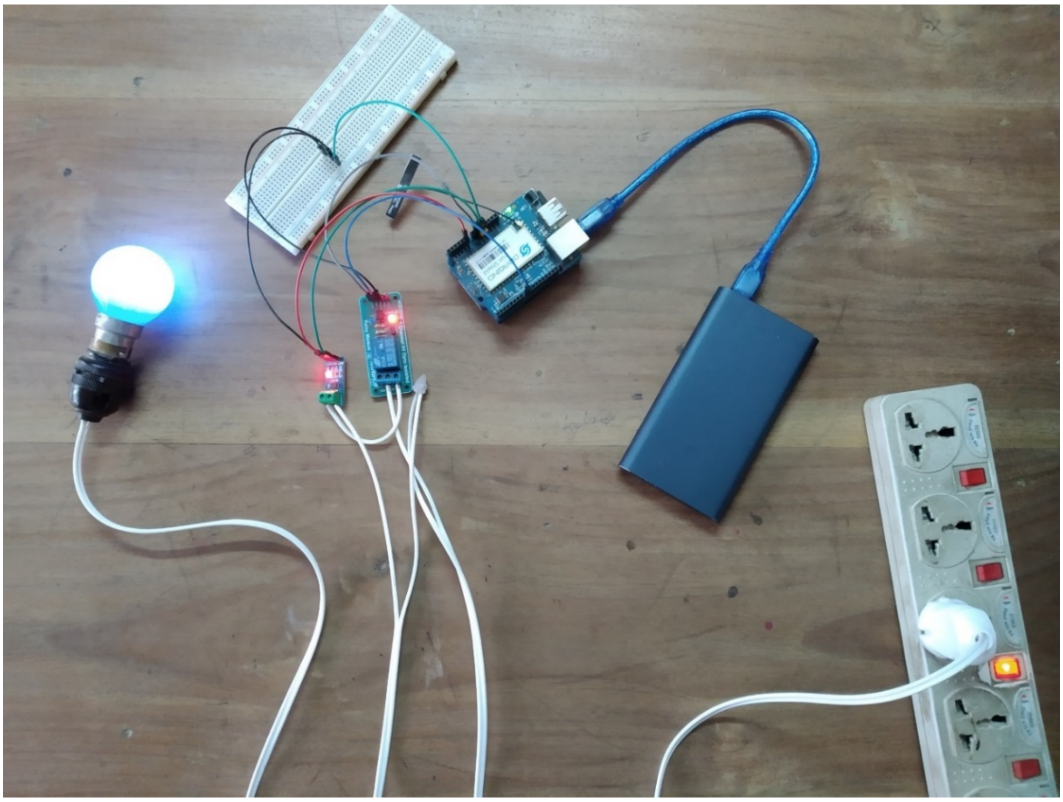


Fig: The simplified circuit Diagram of the project showing the connections

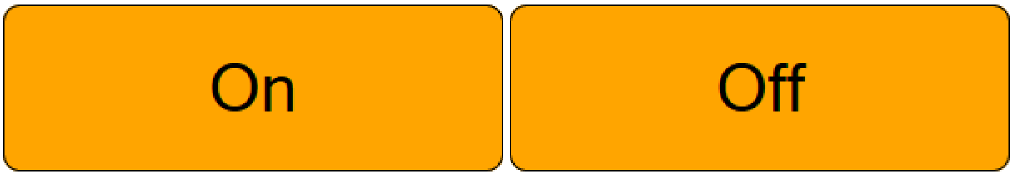
6.3 Work Flow


Start
Declare Variables
Setup Wifi and Debug
Connecting Google Spreadsheet API with Temboo Choreo Library
No
WiFi Connect
Yes
Initialize Relay
Turn Light On
Yes
Relay==1
No
Turn Light Off
Read Current
Calculate Current and Power
Send Data to Google Spreadsheet

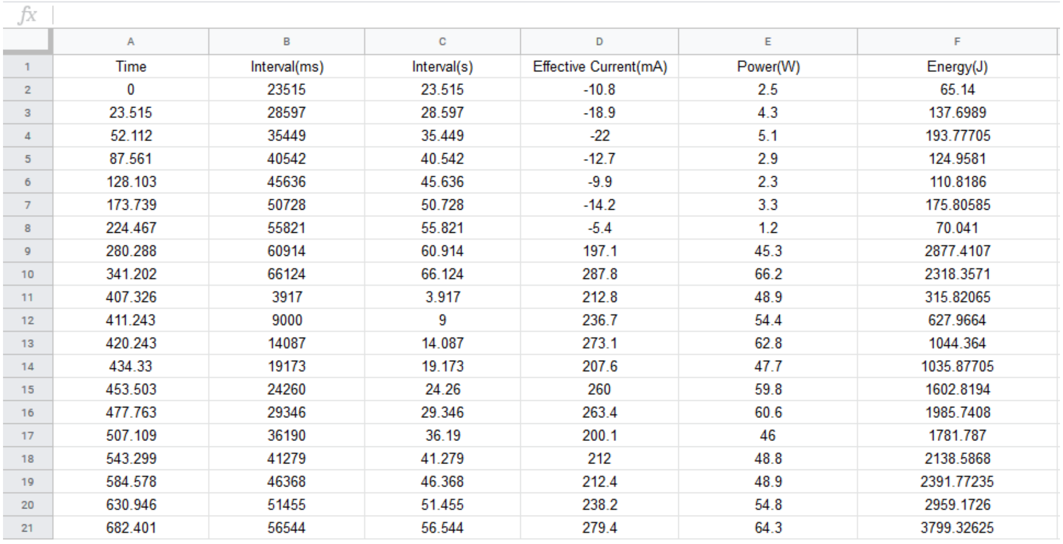
6.4 Experimental Setup and Result



The circuit has been setup as described above in section 6.2 a). The light bulb is switched on and off using simple commands from the web browser and typing the correct REST function with the name of the pin we want to change. However, that's not very convenient. So, a nice graphical interface with buttons is made that can be pressed to turn a light on or off. It would be even better if this interface could be accessed not only from a web browser on a computer but also from any smartphone or tablet in the home.

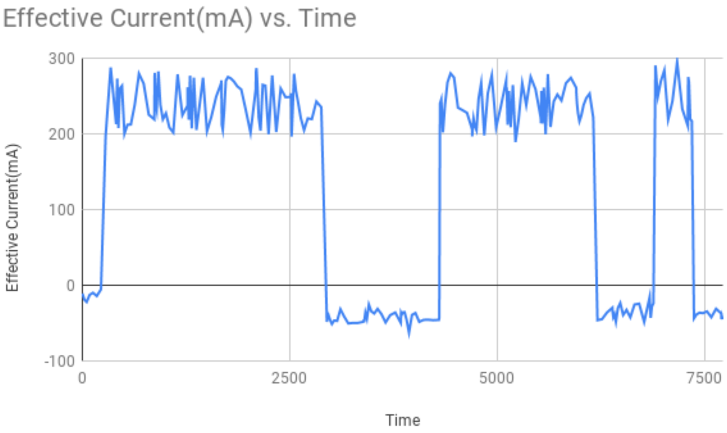


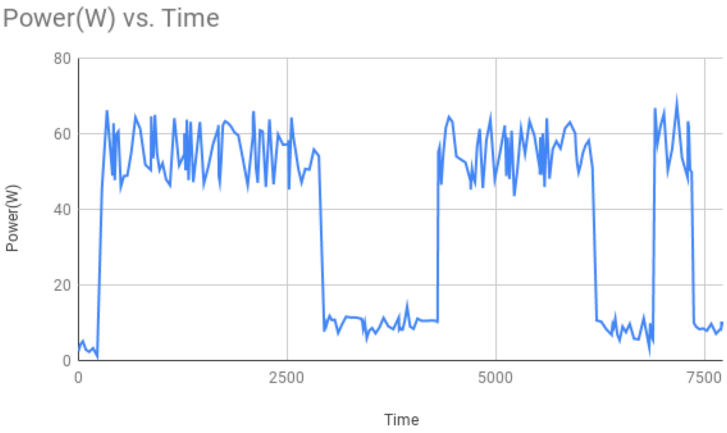
The following data were uploaded in the Google spreadsheet:

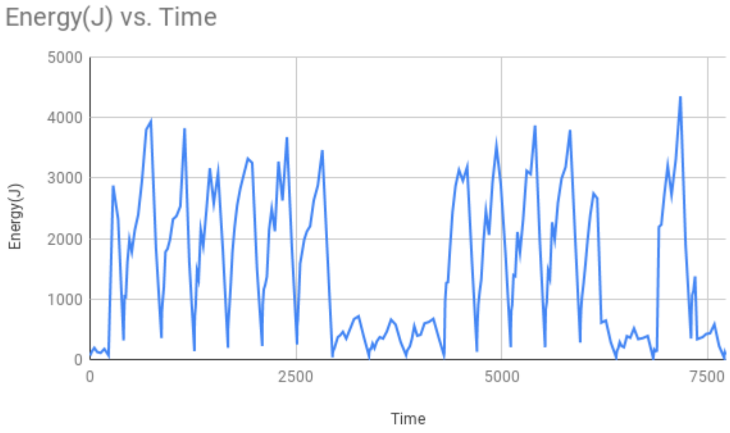


	A	B	C	D	E	F
1	Time	Interval(ms)	Interval(s)	Effective Current(mA)	Power(W)	Energy(J)
2	0	23515	23.515	-10.8	2.5	65.14
3	23.515	28597	28.597	-18.9	4.3	137.6989
4	52.112	35449	35.449	-22	5.1	193.77705
5	87.561	40542	40.542	-12.7	2.9	124.9581
6	128.103	45636	45.636	-9.9	2.3	110.8186
7	173.739	50728	50.728	-14.2	3.3	175.80585
8	224.467	55821	55.821	-5.4	1.2	70.041
9	280.288	60914	60.914	197.1	45.3	2877.4107
10	341.202	66124	66.124	287.8	66.2	2318.3571
11	407.326	3917	3.917	212.8	48.9	315.82065
12	411.243	9000	9	236.7	54.4	627.9664
13	420.243	14087	14.087	273.1	62.8	1044.364
14	434.33	19173	19.173	207.6	47.7	1035.87705
15	453.503	24260	24.26	260	59.8	1602.8194
16	477.763	29346	29.346	263.4	60.6	1985.7408
17	507.109	36190	36.19	200.1	46	1781.787
18	543.299	41279	41.279	212	48.8	2138.5868
19	584.578	46368	46.368	212.4	48.9	2391.77235
20	630.946	51455	51.455	238.2	54.8	2959.1726
21	682.401	56544	56.544	279.4	64.3	3799.32625

Using the data uploaded in the Google Spreadsheet some graphs have been generated which shows the graphs of Effective Current(mA) vs. Time, Power(W) vs. Time, and lastly Energy(J) vs. Time are shown below:







7. Conclusion

In this paper, we have proposed a system to remotely monitor energy and control Device using Rest API and a Web interface. We have optimized our project to monitor the power consumption of electronic devices as accurate as possible. The web interface will allow people to monitor who are connected to the same Wi-Fi connection making it seamless and efficient. This project can be further extended so that high powered devices can be controlled by using a current sensor which can deal with higher currents. And then consumers can monitor the power consumption of their devices from a longer distance. Advanced security system can also be added so that only authenticated users can monitor the device by building a login system on the web interface.. The above results and illustrations show that the proposed model is one of the intuitive methods to calculate and monitor the power consumption which almost any consumer can control.

REFERENCES

[1] S. R. Hernández & P. C. García, “**Definition of a Home Automation System for Energy Management and Efficiency**”, Dated September-October 2010, IEEE Computer Society.

[2] A. A. Adewale, I. A. Samuel, A. A. Awelewa & I. U. Dike, “**Design and Development of a Microcontroller Based Automatic Switch for Home Appliances**”, (2013).

[3] S. V. Shah, “**Home Control and Monitoring**”, 1995, Core, (2017).

[4] M. Park, J. Eune & S. Kim, “**Tangible user interface design for home automation energy management appliances**”, ACM Digital Library.

[5] J. Zhang, Z. Zhou, S. Li, L. Gan, X. Zhang, L. Qi, Xiaolong & W. Dou, “**Hybrid computation offloading for smart home automation in mobile cloud computing**”, ACM Digital Library.

[6] R. K. Chauhan & K. Chauhan, “**Building automation system for grid-connected home to optimize energy consumption and electricity bill**”, ScienceDirect.

[7] G. P. R. Filho, L. A. Villas, V. P. Gonçalves, G. Pessin, A. A. F. Loureiro & J. Ueyamae, “**Energy-efficient smart home systems: Infrastructure and decision-making process**”, ScienceDirect.

[8] A. Kamilaris, V. Trifa, and A. Pitsillides, "**HomeWeb: An application framework for Web-based smart homes**", in Telecommunications (ICT), 2011 18th International Conference on, 2011, pp. 134-139.

[9] T. Perumal, M. N. Sulaiman, K. Y. Sharif, A. R. Ramli, and C. Y. Leong, "**Development of an Embedded Smart Home Management Scheme**", International Journal of Smart Home, vol. 7, pp. 15-26, 2013.

[10] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "**Smart objects as building blocks for the internet of things**", Internet Computing, IEEE, vol. 14, pp. 44-51, 2010.

[11] S. Hilton. (2012, 14 January). “**Progression from M2M to the Internet of Things: an introductory blog**”.

[12] K. Tweed, “**Is Energy Efficiency the Most Popular In-Home Automation?**”, Dated 1st October 2013, IEEE Spectrum.

APPENDIX (Program Code)

```
#include <Bridge.h>
#include <Temboo.h>
#include <Process.h>
#include "TembooAccount.h"
#include <YunServer.h>
#include <YunClient.h>
unsigned long time;
#define CURRENT_SENSOR A0
#define RELAY_PIN 8
float amplitude_current;
float effective_value;
float effective_voltage = 230;
float effective_power;
float zero_sensor;
int measurements_interval = 0;
int last_measurement = 0;
YunServer server;
int server_poll_time = 50;
int power_measurement_delay = 10000;
int power_measurement_cycles_max = power_measurement_delay/server_poll_time;
int power_measurement_cycles = 0;
Process date;
const String GOOGLE_USERNAME = "yourGoogleUsername";
const String GOOGLE_PASSWORD = "yourGooglePass";
const String SPREADSHEET_TITLE = "Power";
void setup() {
  Bridge.begin();
  Serial.begin(115200);
  time = millis();
  if (!date.running())  {
    date.begin("date");
    date.addParameter("+%D-%T");
    date.run();
  }
  pinMode(RELAY_PIN,OUTPUT);
  server.listenOnLocalhost();
  server.begin();
  zero_sensor = getSensorValue();
  last_measurement = millis();
}
void loop() {
  YunClient client = server.accept();
  if (client) {
    process(client);
    client.stop();
  }
    if (power_measurement_cycles > power_measurement_cycles_max) {
     Serial.println("\nCalling the /Library/Google/Spreadsheets/AppendRow Choreo...");
     float sensor_value = getSensorValue();
     measurements_interval = millis() - last_measurement;
     last_measurement = millis();
     amplitude_current=(float)(sensor_value-zero_sensor)/1024*5/185*1000000;
     effective_value=amplitude_current/1.414;
     float effective_power = abs(effective_value * effective_voltage/1000);
     runAppendRow(measurements_interval,effective_power);
     power_measurement_cycles = 0;
    }
  delay(server_poll_time);
  power_measurement_cycles++;
}
void runAppendRow(int measurements_interval, float effectiveValue) {
  TembooChoreo AppendRowChoreo;
  AppendRowChoreo.begin();
  AppendRowChoreo.setAccountName(TEMBOO_ACCOUNT);
  AppendRowChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);
  AppendRowChoreo.setAppKey(TEMBOO_APP_KEY);
  AppendRowChoreo.setChoreo("/Library/Google/Spreadsheets/AppendRow");
  AppendRowChoreo.addInput("Username", GOOGLE_USERNAME);
  AppendRowChoreo.addInput("Password", GOOGLE_PASSWORD);
  AppendRowChoreo.addInput("SpreadsheetTitle", SPREADSHEET_TITLE);
  if (!date.running())  {
    date.begin("date");
    date.addParameter("+%D-%T");
    date.run();
   }
  String timeString = date.readString();
  String data = "";
  data = data + timeString + "," + String(measurements_interval) + "," + String(effectiveValue);
  AppendRowChoreo.addInput("RowData", data);
  unsigned int returnCode = AppendRowChoreo.run();
  if (returnCode == 0) {
    Serial.println("Completed execution of the /Library/Google/Spreadsheets/AppendRowChoreo.\n");
  } else {
    while (AppendRowChoreo.available()) {
      char c = AppendRowChoreo.read();
      Serial.print(c);
    }
    Serial.println();
  }
  AppendRowChoreo.close();
}
void process(YunClient client) {
  String command = client.readStringUntil('/');
  if (command == "digital") {
    digitalCommand(client);
  }
  if (command == "analog") {
    analogCommand(client);
  }
  if (command == "mode") {
    modeCommand(client);
  }
}
void digitalCommand(YunClient client) {
  int pin, value;
  pin = client.parseInt();
  if (client.read() == '/') {
    value = client.parseInt();
    digitalWrite(pin, value);
  }
  else {
    value = digitalRead(pin);
  }
  client.print(F("Pin D"));
  client.print(pin);
  client.print(F(" set to "));
  client.println(value);
  String key = "D";
  key += pin;
  Bridge.put(key, String(value));
}
void analogCommand(YunClient client) {
  int pin, value;
  pin = client.parseInt();
  if (client.read() == '/') {
    value = client.parseInt();
    analogWrite(pin, value);
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" set to analog "));
    client.println(value);
    String key = "D";
    key += pin;
    Bridge.put(key, String(value));
  }
  else {
    value = analogRead(pin);
    client.print(F("Pin A"));
    client.print(pin);
    client.print(F(" reads analog "));
    client.println(value);
    String key = "A";
    key += pin;
    Bridge.put(key, String(value));
  }
}
void modeCommand(YunClient client) {
  int pin;
  pin = client.parseInt();
  if (client.read() != '/') {
    client.println(F("error"));
    return;
  }
  String mode = client.readStringUntil('\r');
  if (mode == "input") {
    pinMode(pin, INPUT);
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" configured as INPUT!"));
    return;
  }
  if (mode == "output") {
    pinMode(pin, OUTPUT);
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" configured as OUTPUT!"));
    return;
  }
  client.print(F("error: invalid mode "));
  client.print(mode);
}
float getSensorValue() {
        int sensorValue;
        float avgSensor = 0;
        int nb_measurements = 100;
        for (int i = 0; i < nb_measurements; i++) {
          sensorValue = analogRead(CURRENT_SENSOR);
          avgSensor = avgSensor + float(sensorValue);
        }
        avgSensor = avgSensor/float(nb_measurements);
            return avgSensor;
}
```