INTERNET OF THINGS (IoT) based ONLINE ENERGY MONITORING SYSTEM

Presentation

By

Prof.M.CHAKRAVARTHY

MOHAMMED JALAL TAQUI

MOHSIN AHMED

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

CONTENT

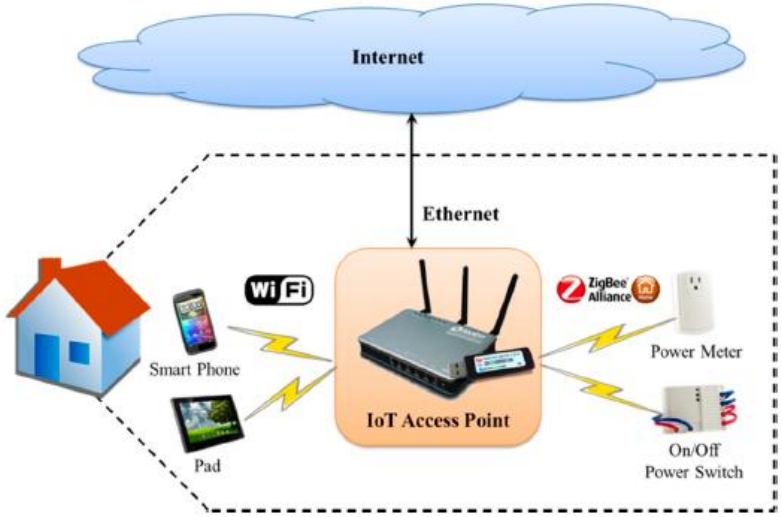
- Abstract
- INTERNET OF THINGS (IoT)
- EMS solutions: MIT App Inventor
- Proposed system
- Schematic wiring diagram
- Hardware & Software details
- GATEWAY:MICRO CONTROLLER (ATMEGA328)
  ESP-8266 WIFI Module
- SENSORS
- HARDWARE SETUP
- CLOUD SERVICES
- Microcontroller program and MIT AI2 application designing
- Monitoring the measured parameters on smart phone using MIT AI2 App

ABSTRACT

- The Internet of Things is a network of ever growing physical objects (such as connected devices and smart devices), embedded with electronics, software, sensors and network connectivity that enables these objects to collect and exchange data.
- IoT is commonly used for smart home, data acquisition, smart energy monitoring, industrial automation , and a variety of platforms.
- This project describes the architecture of an Internet of things based Energy Monitoring System in the power system.
- The goal of this project is to visualize and monitor the power consumption online on a smart phone using mobile application by integrating smart plugs, sensors, Internet of Things (IoT) devices and GATEWAY which enables the communication between the various smart plugs and the web server hosting the monitoring system application, thereby facilitating the user to act accordingly to save power or to provide the reliable power supply by making maximum use of Renewable Energy Sources.

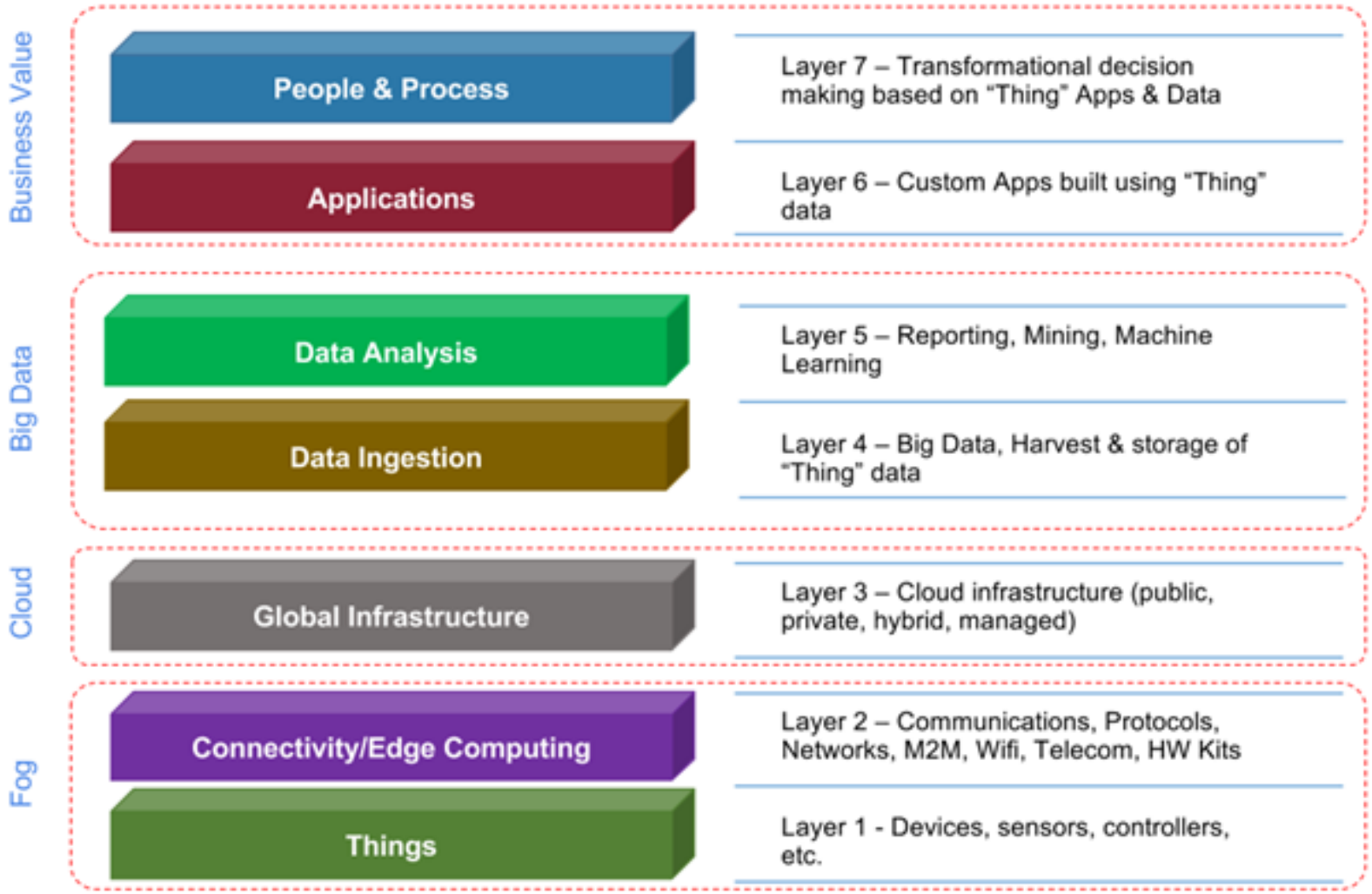
INTERNET OF THINGS (IoT)

- The internet of things is a network of ever growing physical objects (such as connected devices and smart devices),embedded with electronics, software, sensors and network connectivity that enables these objects to collect and exchange data.
- The Internet of Things (IoT) objects features an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems.



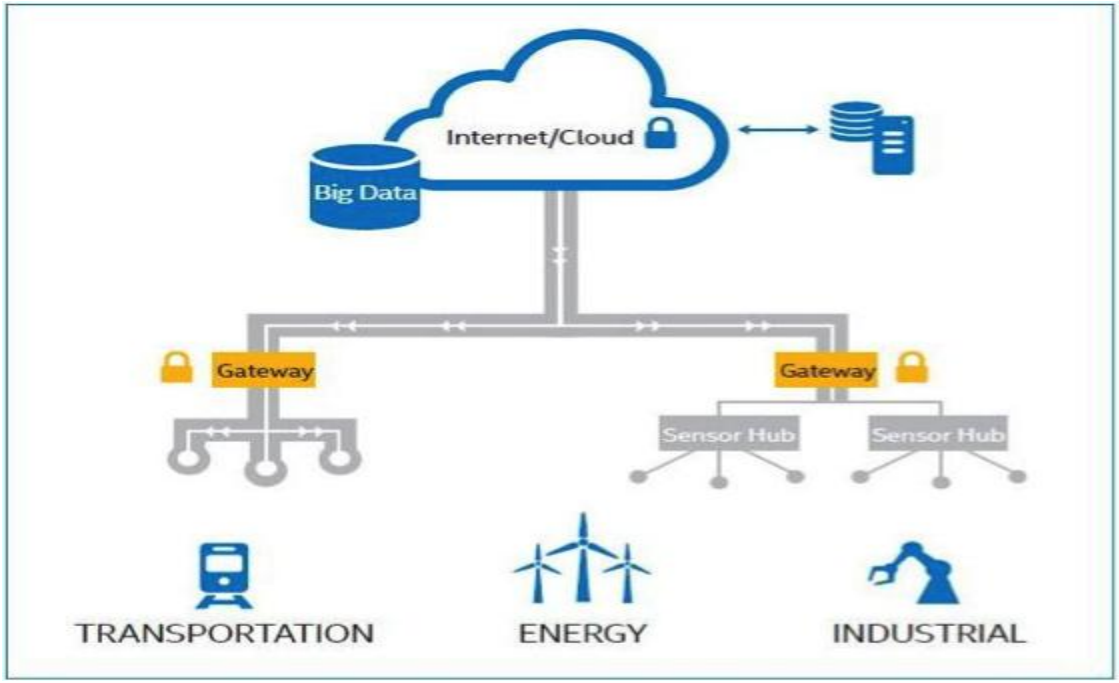
- Besides home based energy management, the IoT is especially relevant to the Smart Grid since it provides systems to gather and act on energy and power-related information in an automated fashion with the goal to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity.

7 LAYERS OF IoT



IoT ECOSYSTEM

1. SENSOR DATA: Smart plugs (transducers and Sensors MODULES)
2. GATEWAY (WIFI Modules)
3. CLOUD SERVICES (that will help individual users to post and retrieve sensor data)



Components of a Sensor network

Sensors (Motes)

• Typically, low power, low duty cycle for non-critical ones

Data aggregators (Gateways)

• Could be a collection center or local analytics centre which analyze, compress information for reducing data traffic, storage and upload on demand

Data servers and Display (Server)

• Central data centres such as cloud servers, for large data analysis and dash board display

GATEWAY

- The gateway is a device that enables the communication between the various smart plugs and the web server hosting the monitoring system application.

Gateway components :

- Microcontroller: ATMEGA326P ,Arduino Uno..
- Wireless communication: X-bee shield..
- Internet connectivity: Ethernet shield,ESP8266 WIFI module

EMS SOFTWARES

1. By **Seimens:**

Energy Management Software

a. WinPM.Net Software

b. WinPM.Net.7.0

- http://w3.usa.siemens.com/powerdistribution/us/en/product-portfolio/power-monitoring/energy-management-software/pages/energy-management-software.aspx

2. By **Schneider electric:**

Power Monitoring and Control Software

a. StruxureWare PowerSCADA Expert 8.1

b. Power Monitoring Expert 8.1

- http://www.schneider-electric.co.in/en/product-subcategory/4170-power-monitoring-and-control-software/?parent-category-id=4100

3. By **Phoenix Energy Technologies**

a. EnterpriseDX® Energy Management Software

- http://www.phoenixet.com/solutions/enterprisedx-energy-management

4. BY **GOOGLE ,**DEVELOPING APPLICATION FOR MOBILES : **MIT AI2 COMPANION APP**

Applications of IoT

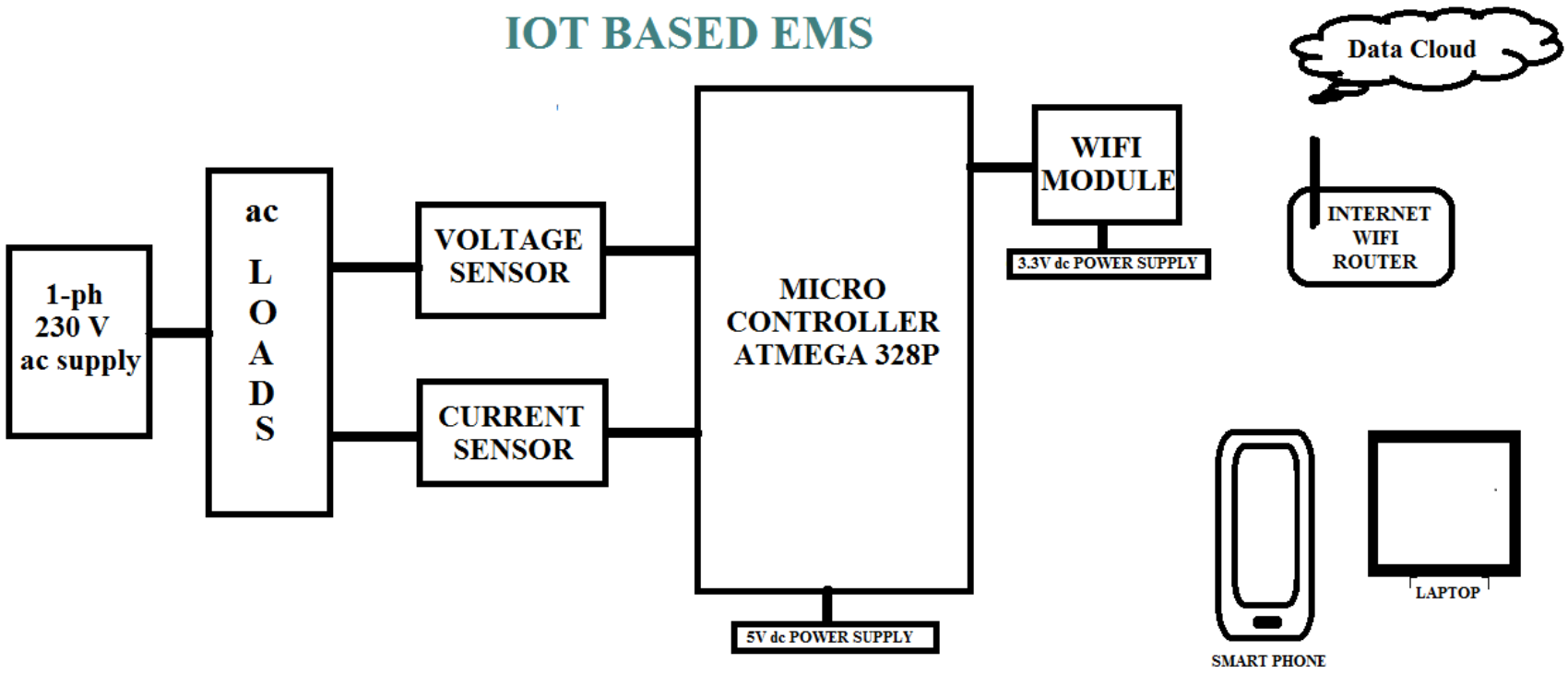
- Building and Home automation
- Manufacturing
- Medical and Healthcare systems
- Media
- Environmental monitoring
- Infrastructure management
- Energy management
- Transportation
- Better quality of life
- ***So on.....***

Application of IoT In Electrical Field

- Smart Grid
- Smart Home
- SCADA
- Smart Energy Monitoring
- Solar control and monitoring
- Industrial Internet
- Smart Street lighting
- Energy Usage and Efficiency
- Power quality monitoring
- Building automatiom
- Smart Solar Tracking

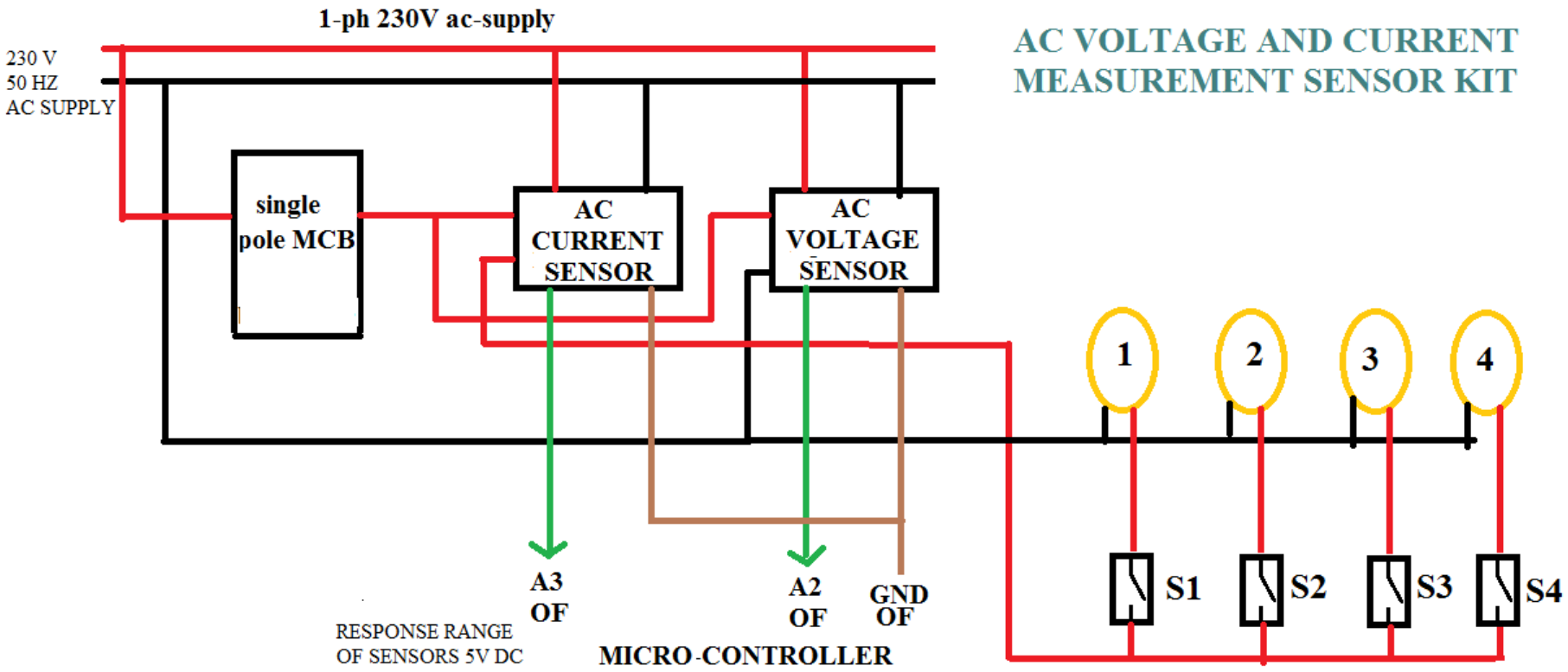
PROPOSED SYSTEM

IOT BASED EMS:

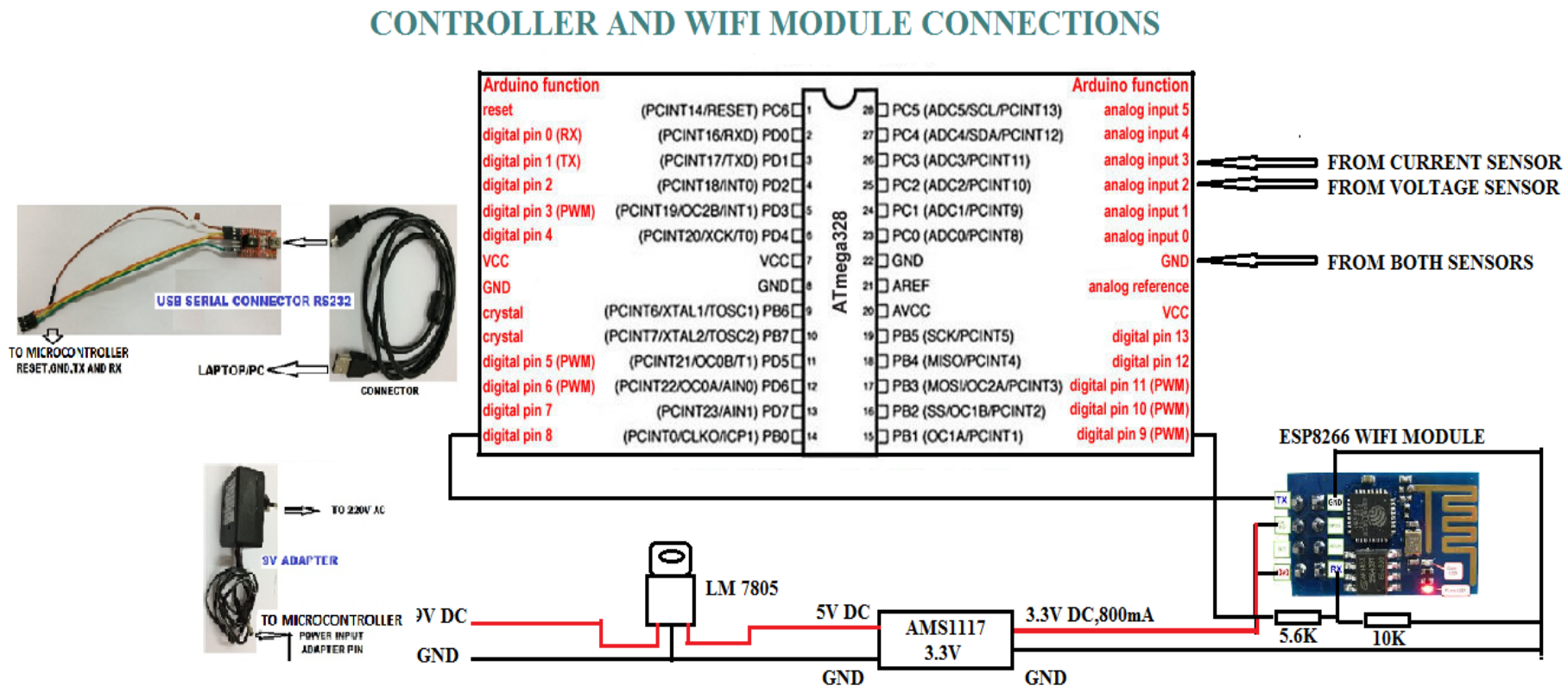


IOT BASED EMS BLOCK DIAGRAM

AC VOLTAGE AND CURRENT MEASUREMENT SENSOR KIT



CONTROLLER AND WIFI MODULE CONNECTION



Hardware DETAILS

A. Microcontroller

AVR microcontroller-ATmega328 with arduino boot loader.

B. Input devices/sensors

-Hardware input sensors:

a. AC Voltage sensor

b. AC Current sensor

c. RS232 USB programmer for communication

C. Output devices/relays

a.5V operated relay with NO and NC (normally open and close) pins brought out to the connecter.

b. LED's for user defined diagnostics and some resistors.

c. 230 V,100W AC Load bulbs with SPST switches

D. Connectivity modules

a. ESP-01 Wifi module

E. AC Voltage and current measurement sensor KIT with switches to vary loads.

F. Power resources

a. 230V ac power supply

b.9V dc ADAPTER

c.7805 regulator for on-board 5v dc power

d.LM1117 LDO for on-board 3.3V dc generation.

Software DETAILS

1. Microcontroller programming :

a. The microcontroller can run code developed using the Arduino IDE.

The serial interface between the micro controller and the Wifi module is realized using software.

b. Library function (**#include<SparkFunESP8266WiFi.h>**) with AT command set and set firmware loaded for ESP-01 Wifi module.

Library file (SparkFunESP8266WiFi.h)

2. Cloud services:

The cloud services helps the user to post their sensor data to the cloud and retrieve the data.

The two services offered by a cloud provider are:

Pushing data service (setsensordata.php)

Pulling data service (getsensordata.php)

3. Notification services

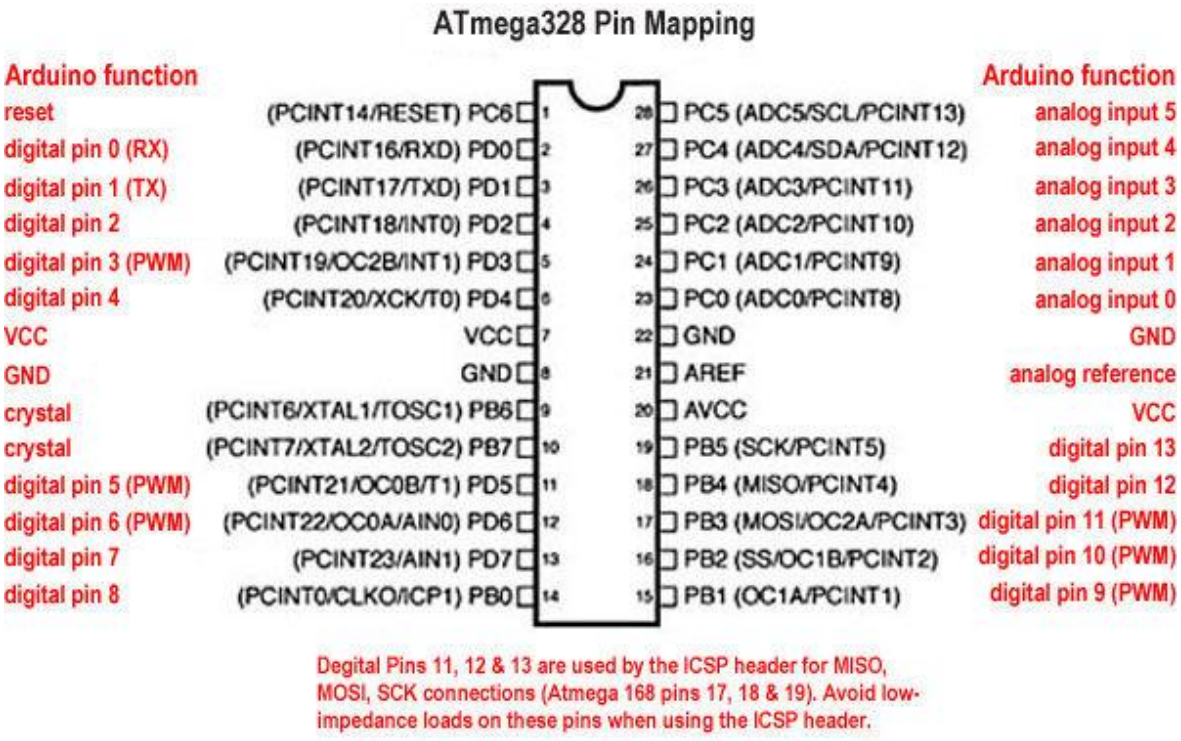
a. General notification services (through g-mails, twitter etc)

Ex: PUSHING BOX

b. Mobile notification applications

Ex: AI2 (App Inventor) mobile app for smart phones

Micro controller

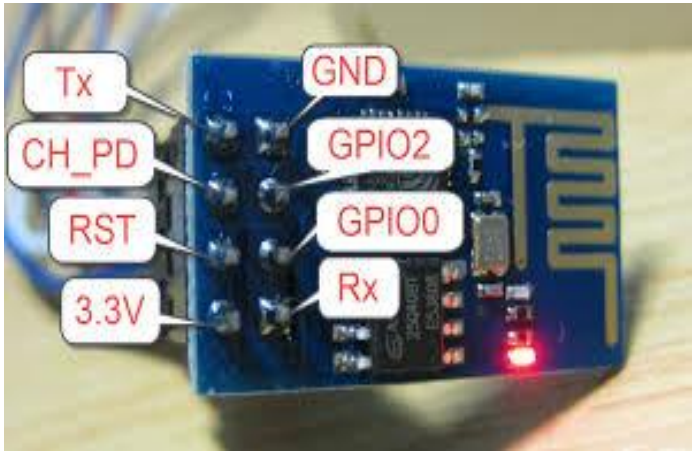


ARDUINO UNO

ESP8266 WIFI Module

- The ESP-01 Wifi module is used to get connected to a wifi network for internet access. The ESP-01 is used as a Wifi adapter to give wireless internet access to the microcontroller.
- The ESP8266 comes loaded with firmware that can accept AT commands over the serial interface to do various functions.
- These AT commands are sent on the serial interface by the microcontroller.

.



AT commands of ESP-01 Wifi module

Basic	
Command	**Description**
AT	Test AT startup
AT+RST	Restart module
AT+GMR	View version info
AT+GSLP	Enter deep-sleep mode
ATE	AT commands echo or not
AT+RESTORE	Factory Reset
AT+UART	UART configuration, [@deprecated]
AT+UART_CUR	UART current configuration
AT+UART_DEF	UART default configuration, save to flash
AT+SLEEP	Sleep mode
AT+RFPOWER	Set maximum value of RF TX Power
AT+RFVDD	Set RF TX Power according to VDD33

ESP-01 Wifi module Contd..

- The hardware interface between the microcontroller and the ESP-01 is a serial interface.
- The RX and TX pins of the ESP-01 module are connected to pin 9 and 8 (arduino nomenclature). Since these pins are not the hardware UART pins, the SoftwareSerial library is used to create a soft serial port.
- These AT commands have to be issued by the microcontroller over the serial interface. An Arduino library <SparkFunESP8266WiFi> from the SparkFun gives library functions that will issue the AT commands, receive and pass-on/interpret the response,thus providing a convenient way of calling these fuctions.

IDE PROGRAMMING commands for ESP8266

WiFi adapter Initialisation using Integrated Development Environment (IDE)

Initialisation of Wifi Adapter :

While using the wifi adapter we have to use two libraries :

#include<SparkFunESP8266WiFi.h>

#include<SoftwareSerial.h>

Steps while using Wifi Adapter :

1. Associate the wifi network

esp8266.begin()

Verifies that the ESP8266 is operational and sets it up for the rest of the sketch.Returns true or false indicating whether communication with the module is successful or not.

2.Associating with Wifi Network

esp8266.connect([“SSID”],[“password”])

The ssid and password is your wifi id and password required for connecting to wifi network.

After ESP8266 gets connected to wifi

a. On success return value is > 0

b. On failure return value <0 or

c. Fail could not connect to network.

esp8266client

This library provides a ESP8266Client class for instantiating a TCP Client.

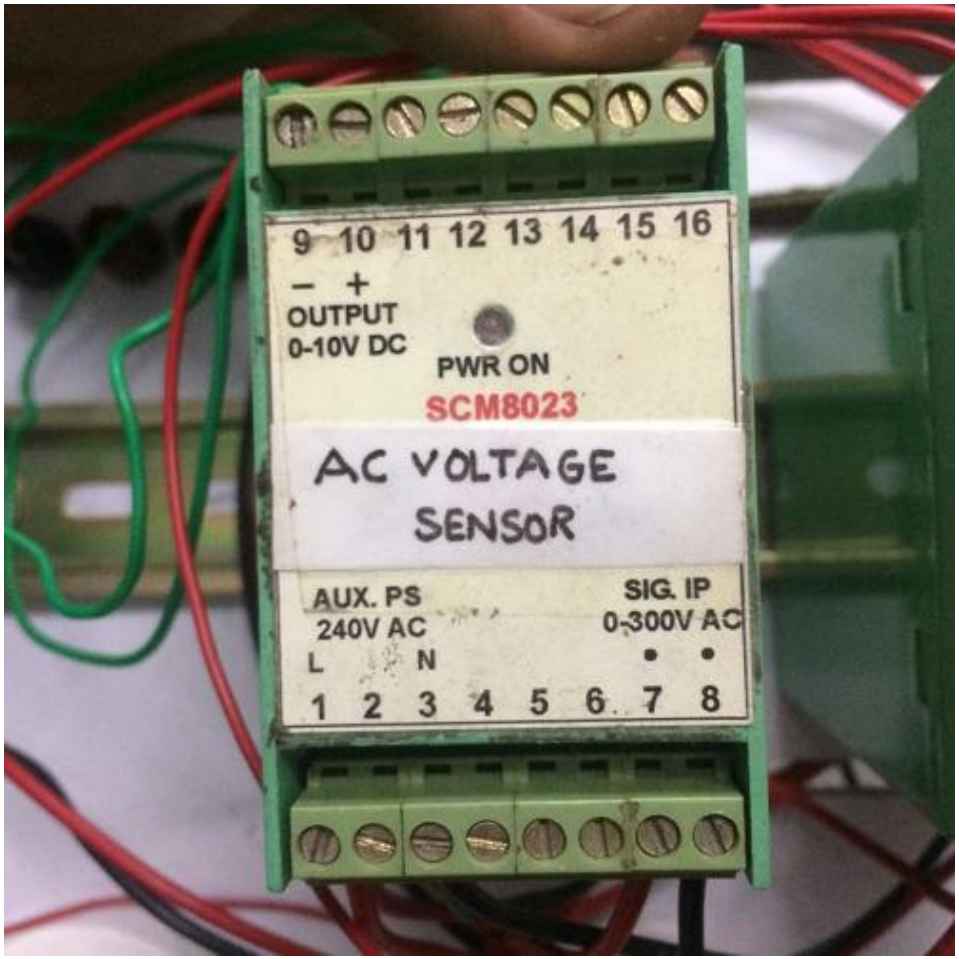
(**ESP8266Client.connect([server],[password])**

eg : “www.mywebsite.in”,80); is connected to the server on a specified port,

return value >0 then it is already connected

-1,-2 then connection is failure / timeout on unsuccessful connection.

Voltage sensor



Specifications:

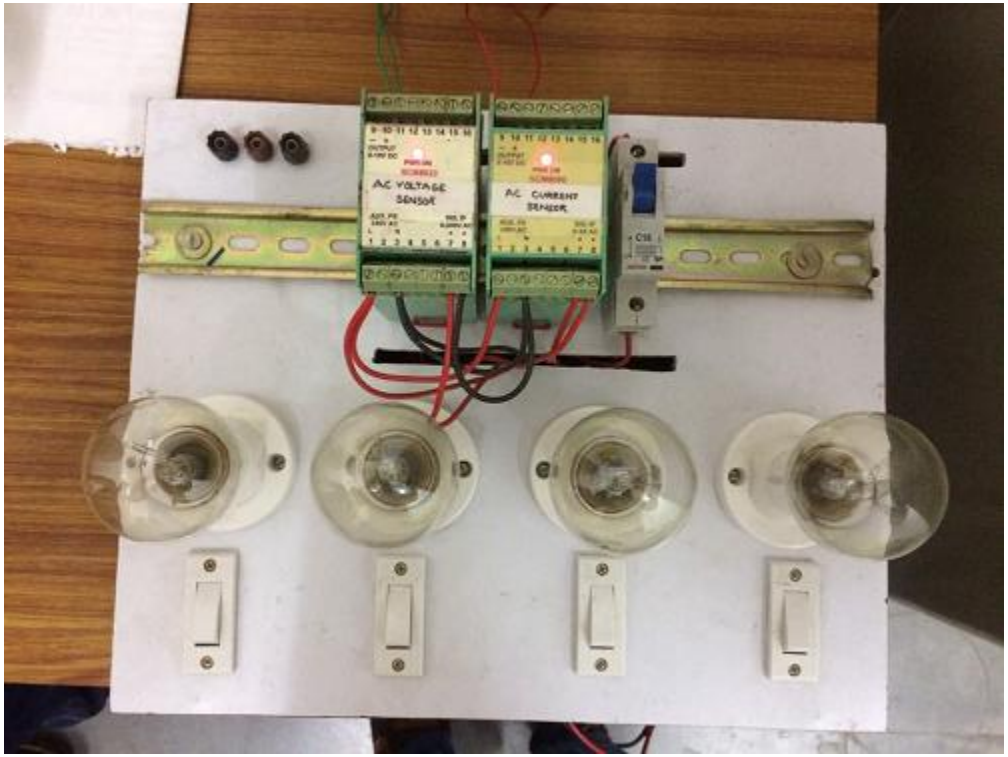
Model number: SCM8023
Signal input = 0-300 V ac
Auxiliary power supply= 240V ac
Ouput signal= 0-10 V dc

Current sensor

Specifications:

Model number: SCM8006
Signal input = 0-300 V ac
Auxiliary power supply= 240V ac
Ouput signal= 0-10 V dc

HARDWARE SETUP



CLOUD SERVICES

- The cloud services helps the user to post their sensor data to the cloud and retrieve the data.
- The services will be available for the registered users.
- A utility is available for the registered users.
- The maximum number of posts per user is restricted 1000 per day. Posts received after the maximum limit is reached will be discarded.
- The two services offered are:

1. **Pushing data service** (setsensordata.php)
2. **Pulling data service** (getsensordata.php)

CLOUD SERVICES

The string assigned to the sensor parameter will have the following fields separated by colons :

- Valid user ID
- Sensor ID
- Sensor value

1. Pushing data service:

www.websitename.in/services/setsensor.php?sensor=userid:sensorid:sensorvalue

Example:

www.websitename.in/services/setsensor.php?sensor=mjts:1:230

2.Pulling data services

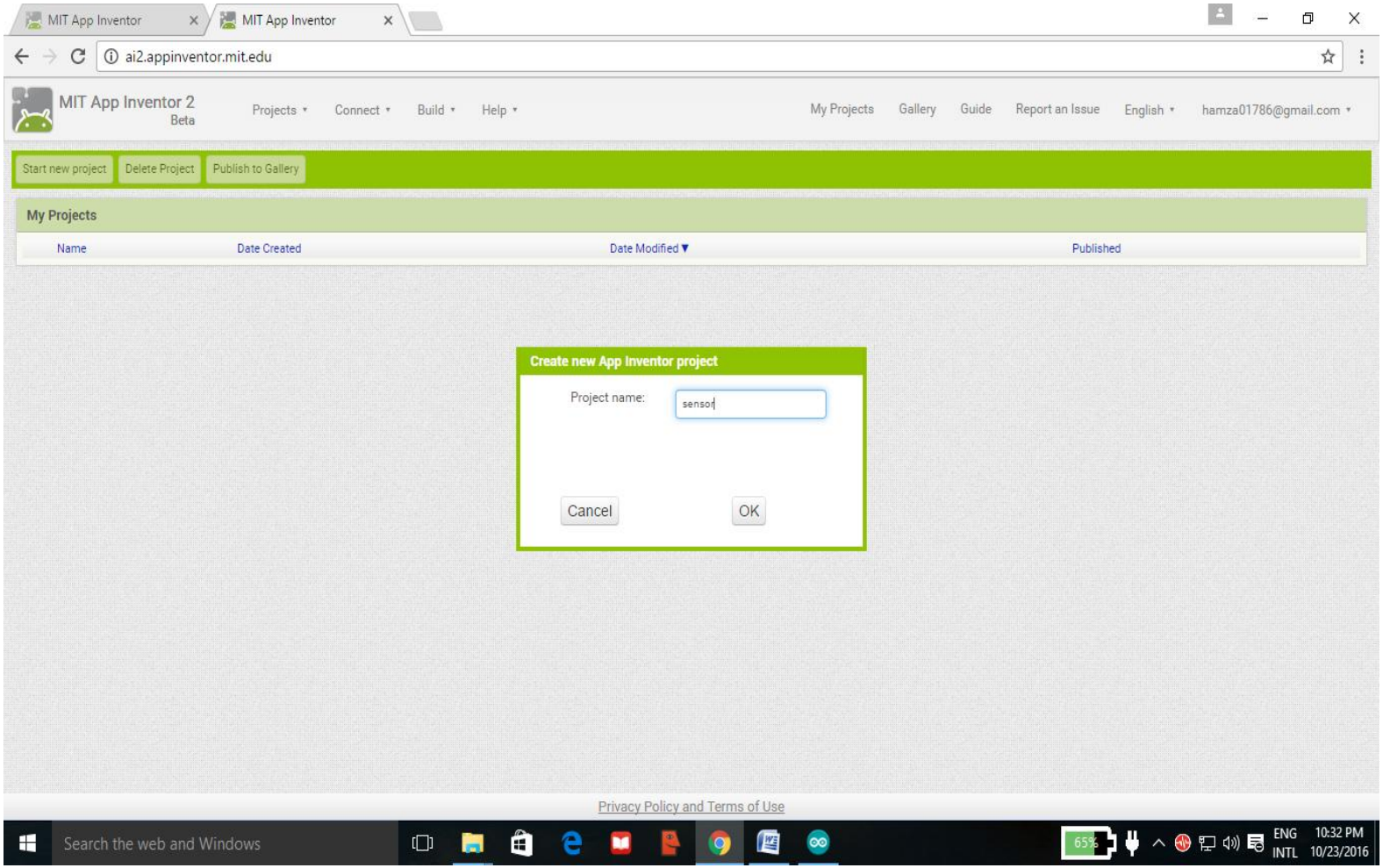
www.websitename.in/services/getsensor.php?format=json&uid=mjts&sensorid=1

MIT AI2 APP INVENTOR

- APP Invertor2 is a cloud based utility for developing mobile applications for android phones. It is free utility and log in can be done using your Gmail Id.
- MIT AI2 COMPANION APP is developed by Google.
- By using MIT AI2 can develop code in our laptop and further it will run our respective mobile.
- The app is developed using components:
    1. User interface (textbox, buttons, list box, checkbox).
    2. Layout.
    3. Sensors (accelerometer, proximity sensor, GPS).
    4. Connectivity (Bluetooth server, Bluetooth client, web component).
- App window can be designed using two views :
    1. Designer View
    2. Block View

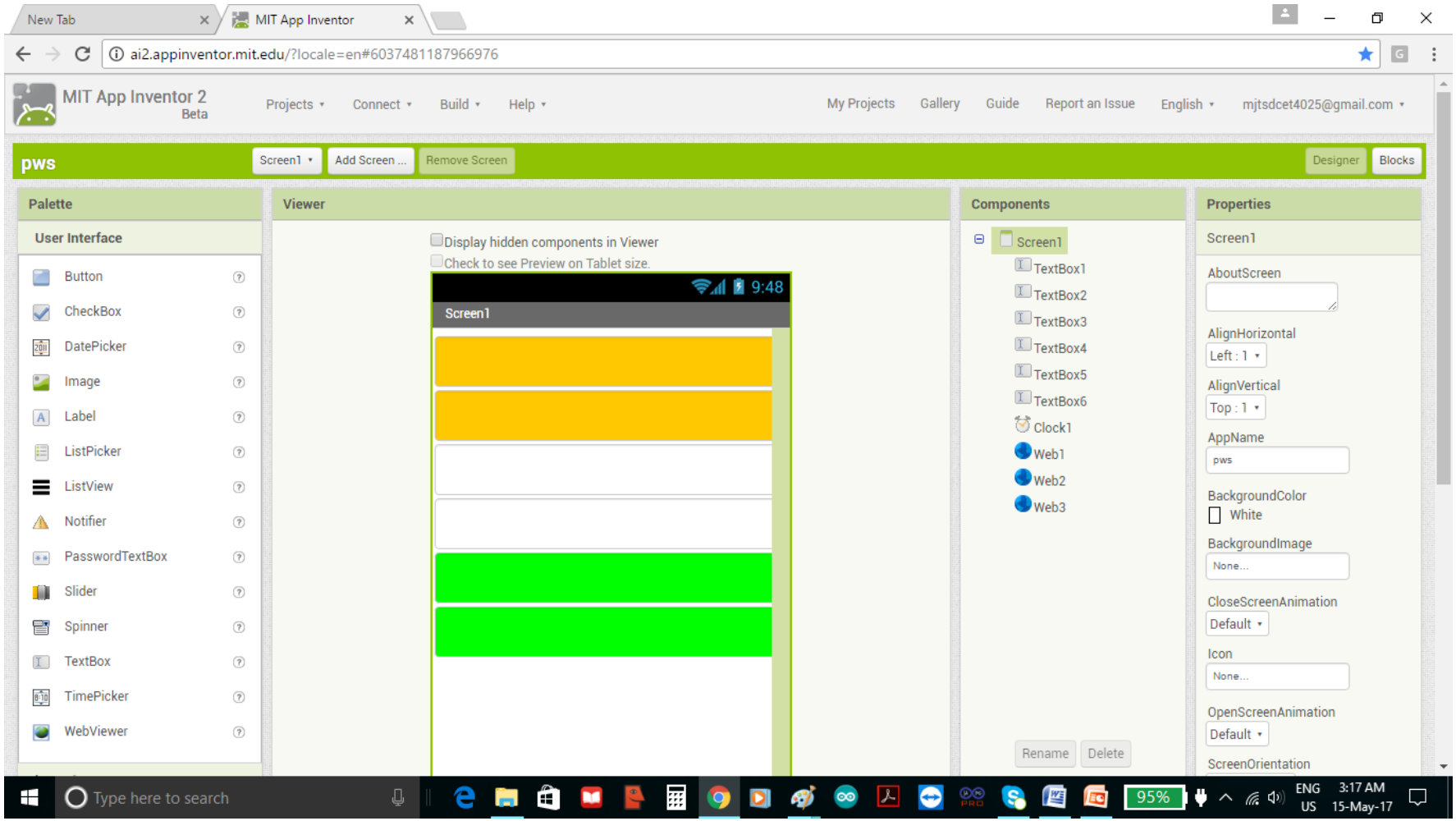
Designing The MIT AI2 Mobile View For Displaying 3 Sensor Values

Create a new project by naming it and press ok.



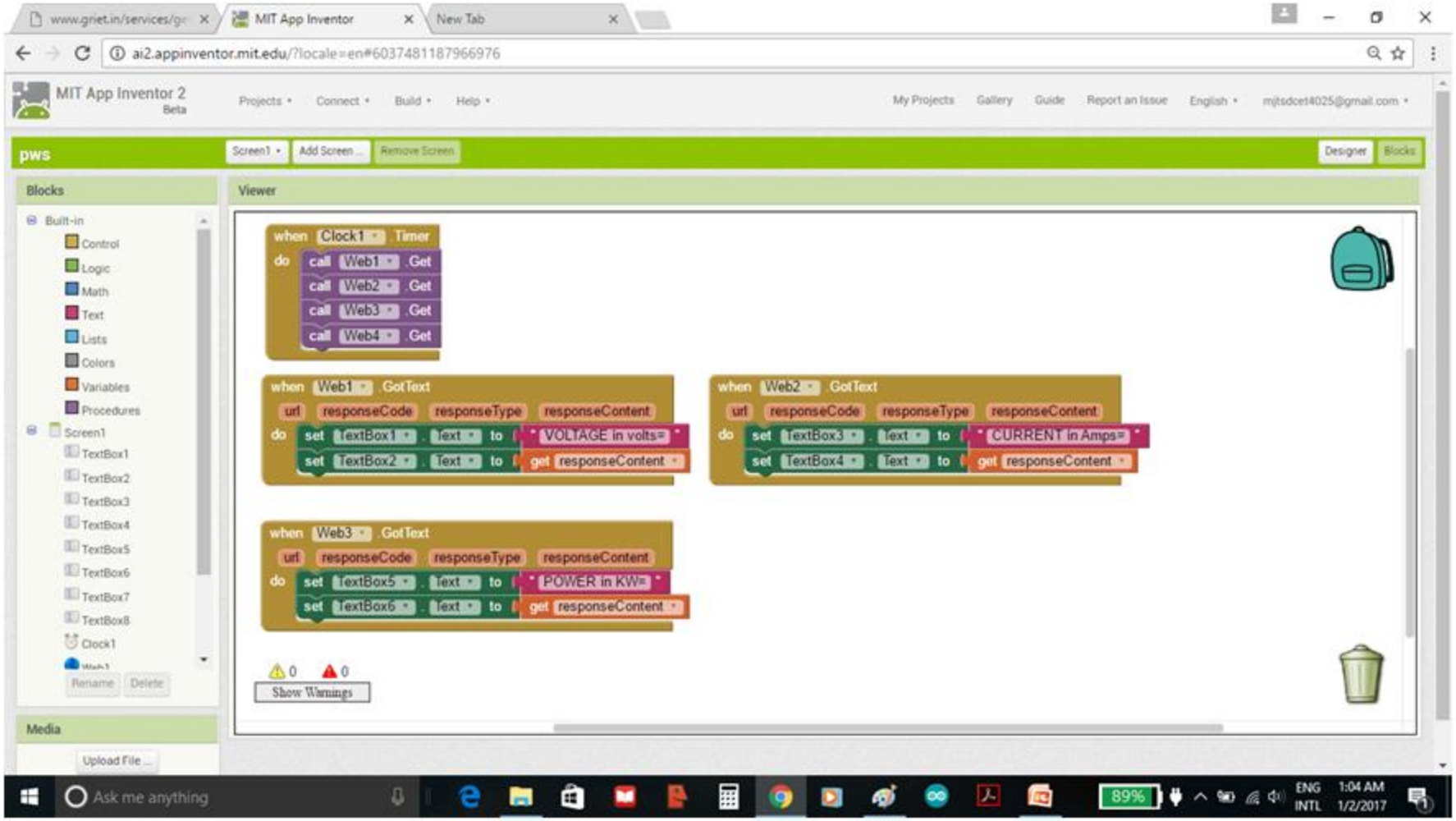
MIT AI2 APP DESIGNING

DESIGNER VIEW:



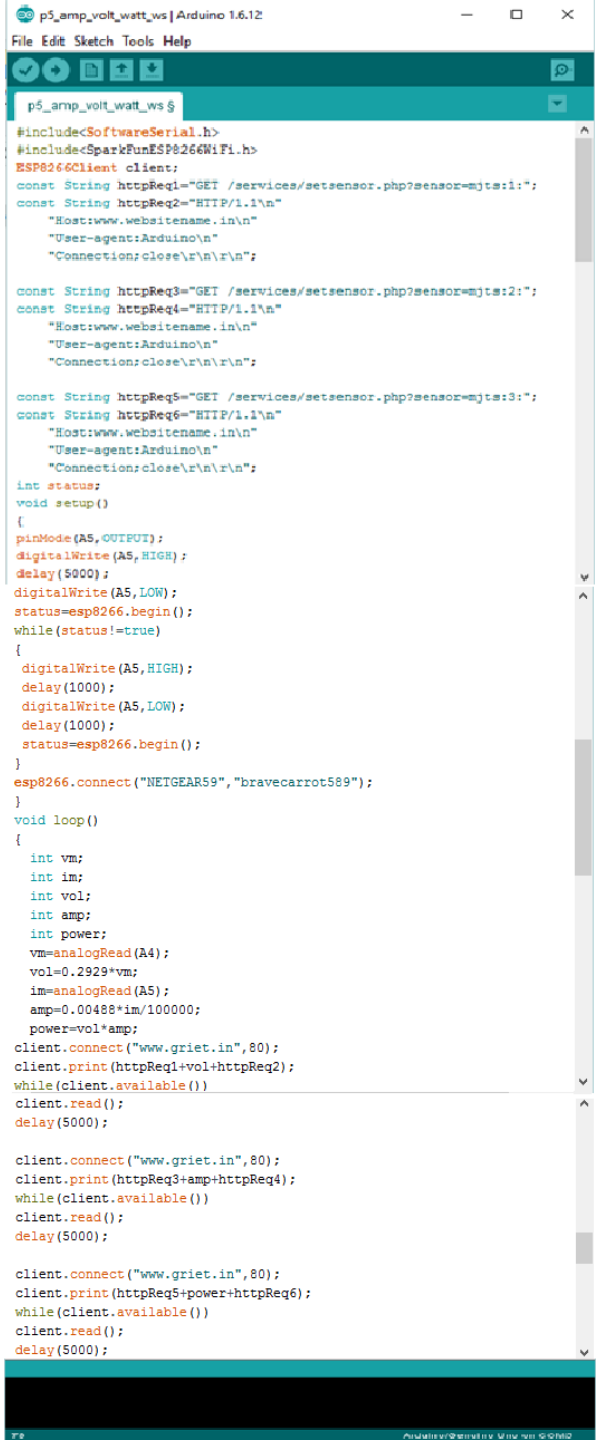
MIT AI2 APP DESIGNING

BLOCK VIEW:



ARDUINO PROGRAM:

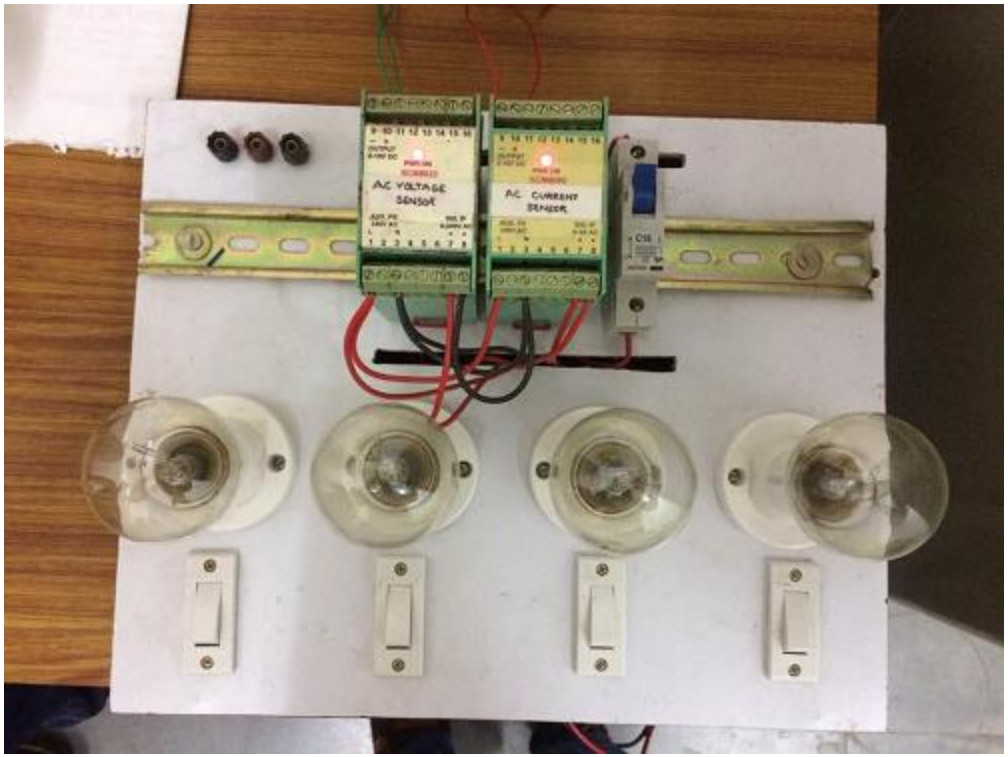
Program to display the voltage, current and power on smart phone.

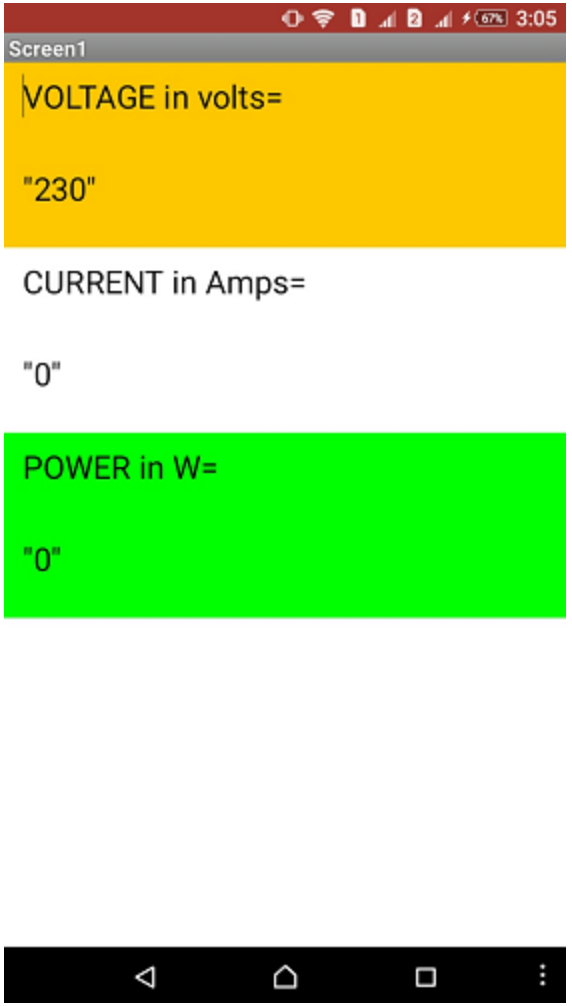



```
#include<SoftwareSerial.h>
#include<SparkFunESP8266WiFi.h>
ESP8266Client client;
const String httpReq1="GET /services/setsensor.php?sensor=mjts:1:";
const String httpReq2="HTTP/1.1\n"
    "Host:www.websitename.in\n"
    "User-agent:Arduino\n"
    "Connection:close\r\n\r\n";

const String httpReq3="GET /services/setsensor.php?sensor=mjts:2:";
const String httpReq4="HTTP/1.1\n"
    "Host:www.websitename.in\n"
    "User-agent:Arduino\n"
    "Connection:close\r\n\r\n";

const String httpReq5="GET /services/setsensor.php?sensor=mjts:3:";
const String httpReq6="HTTP/1.1\n"
    "Host:www.websitename.in\n"
    "User-agent:Arduino\n"
    "Connection:close\r\n\r\n";
int status;
void setup()
{
pinMode(A5,OUTPUT);
digitalWrite(A5,HIGH);
delay(5000);
digitalWrite(A5,LOW);
status=esp8266.begin();
while(status!=true)
{
 digitalWrite(A5,HIGH);
 delay(1000);
 digitalWrite(A5,LOW);
 delay(1000);
 status=esp8266.begin();
}
esp8266.connect("NETGEAR59","bravecarrot589");
}
void loop()
{
  int vm;
  int im;
  int vol;
  int amp;
  int power;
  vm=analogRead(A4);
  vol=0.2929*vm;
  im=analogRead(A5);
  amp=0.00488*im/100000;
  power=vol*amp;
client.connect("www.griet.in",80);
client.print(httpReq1+vol+httpReq2);
while(client.available())
client.read();
delay(5000);

client.connect("www.griet.in",80);
client.print(httpReq3+amp+httpReq4);
while(client.available())
client.read();
delay(5000);

client.connect("www.griet.in",80);
client.print(httpReq5+power+httpReq6);
while(client.available())
client.read();
delay(5000);
```

RESULTS: MEASURED PARAMETERS ON SMART PHONE USING MIT AI2 APPLICATION

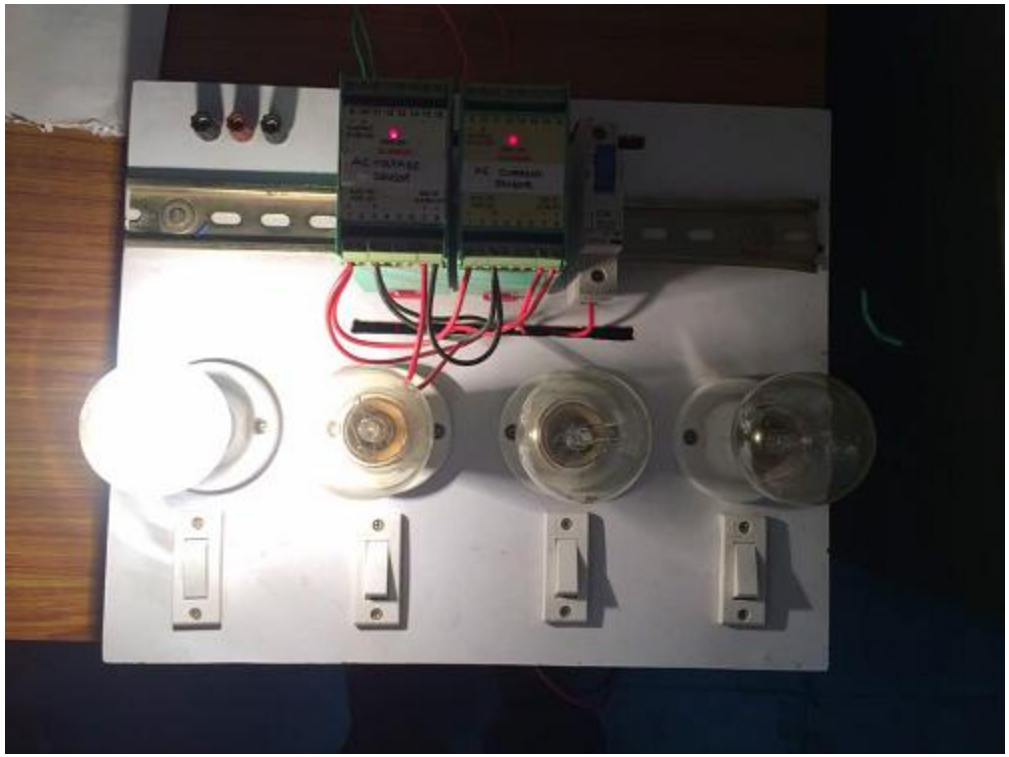
WHEN SETUP IS POWERED ON , NO LOAD BULB IS ON
THEN VOLTAGE=230 V, CURRENT=0 A, POWER= 0 W

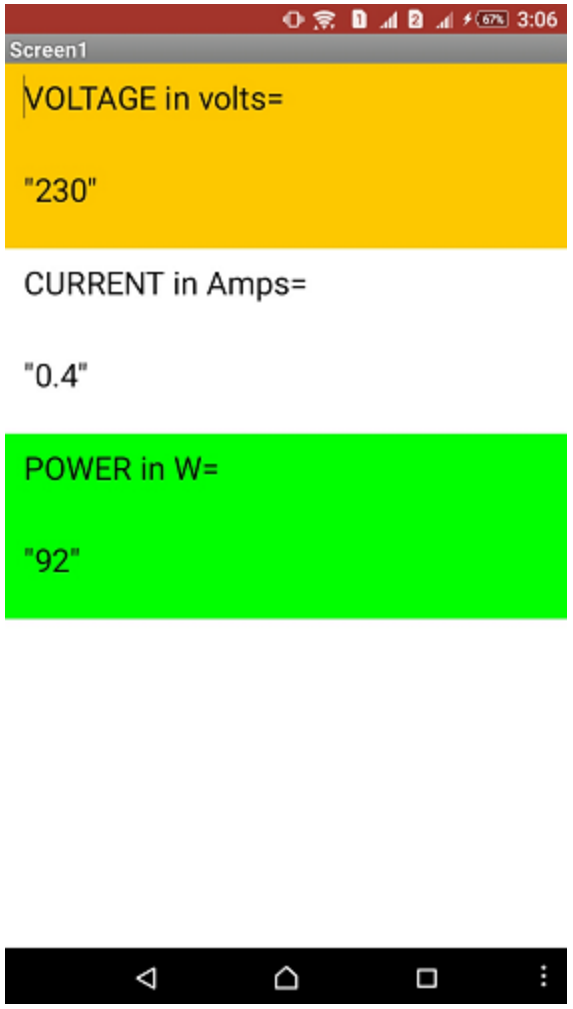




CASE 1: WHEN **1** LOAD BULB IS ON

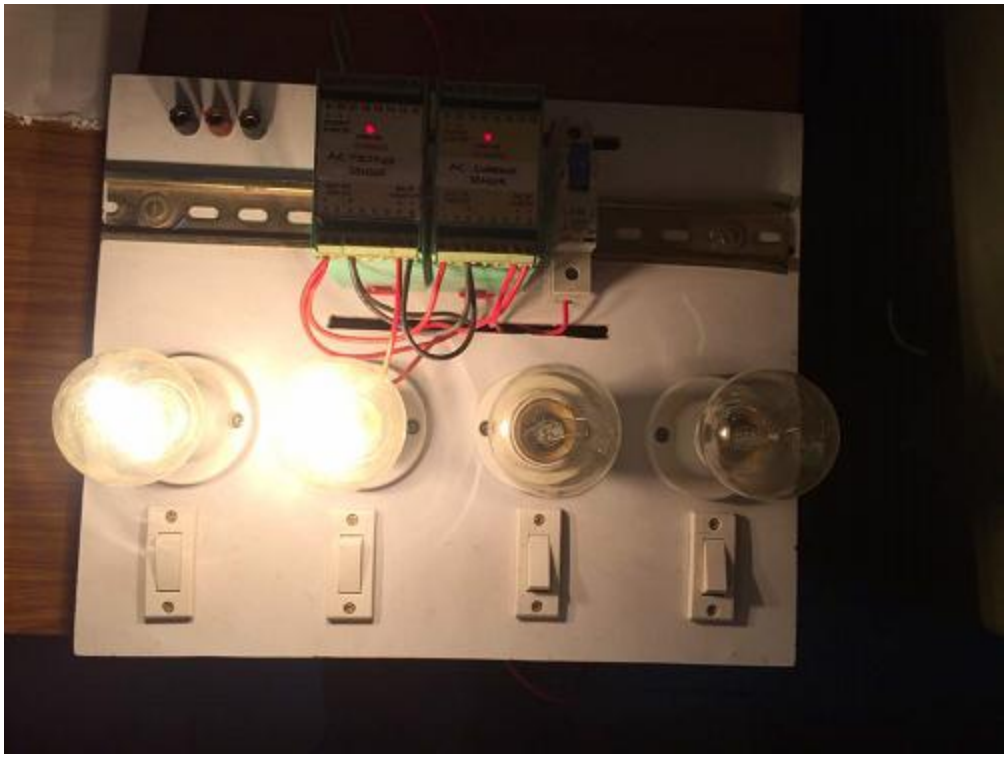
THEN VOLTAGE=230 V, CURRENT=**0.4 A**, POWER= **92 W**

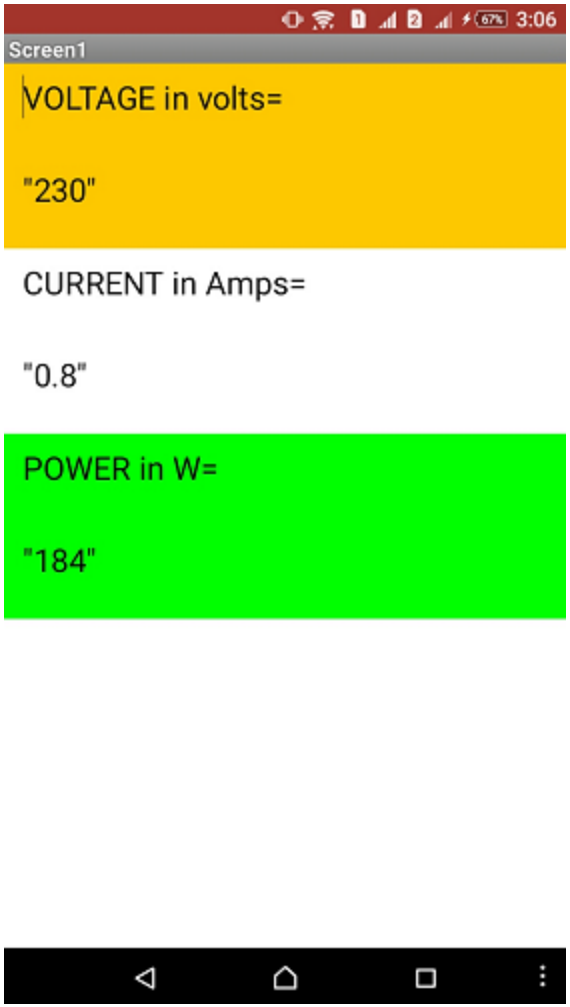




CASE 2: WHEN **2** LOAD BULB IS ON

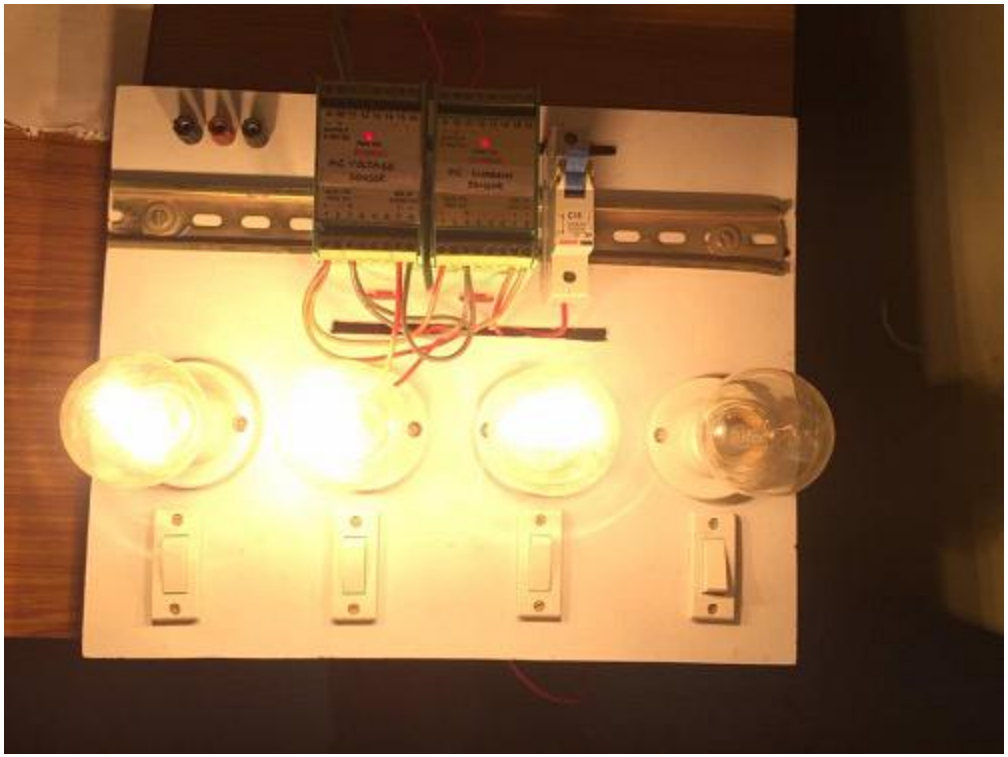
THEN VOLTAGE=230 V, CURRENT= **0.8 A**, POWER= **184 W**

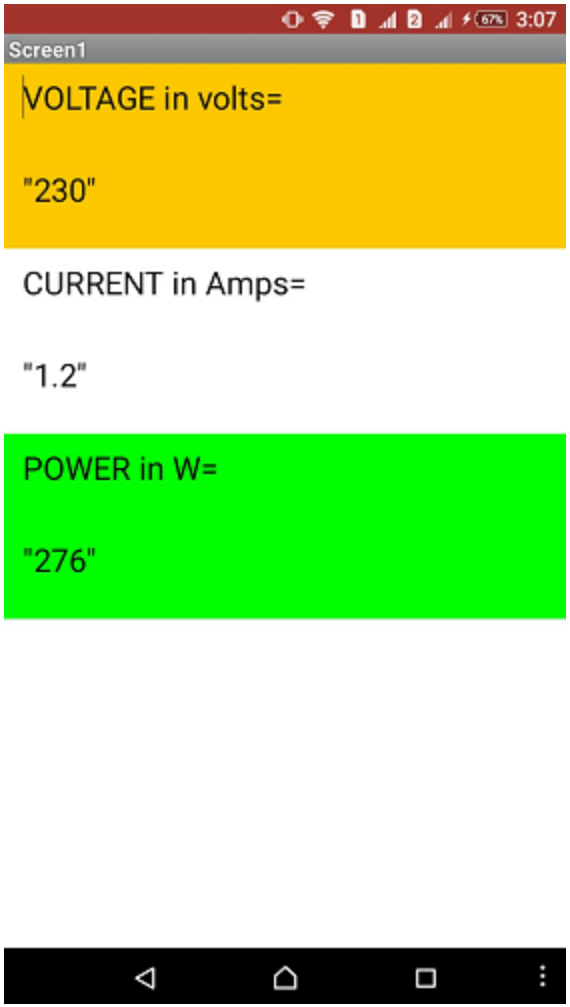




CASE 3: WHEN **3** LOAD BULB IS ON

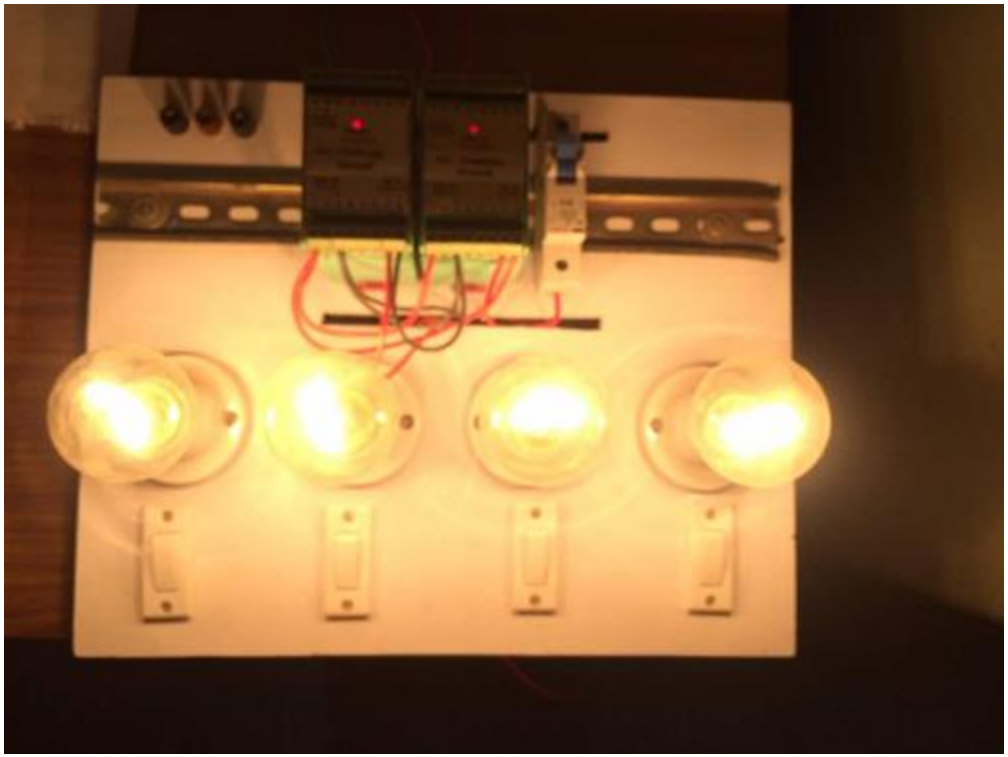
THEN VOLTAGE=230 V, CURRENT= **1.2 A**, POWER= **276 W**

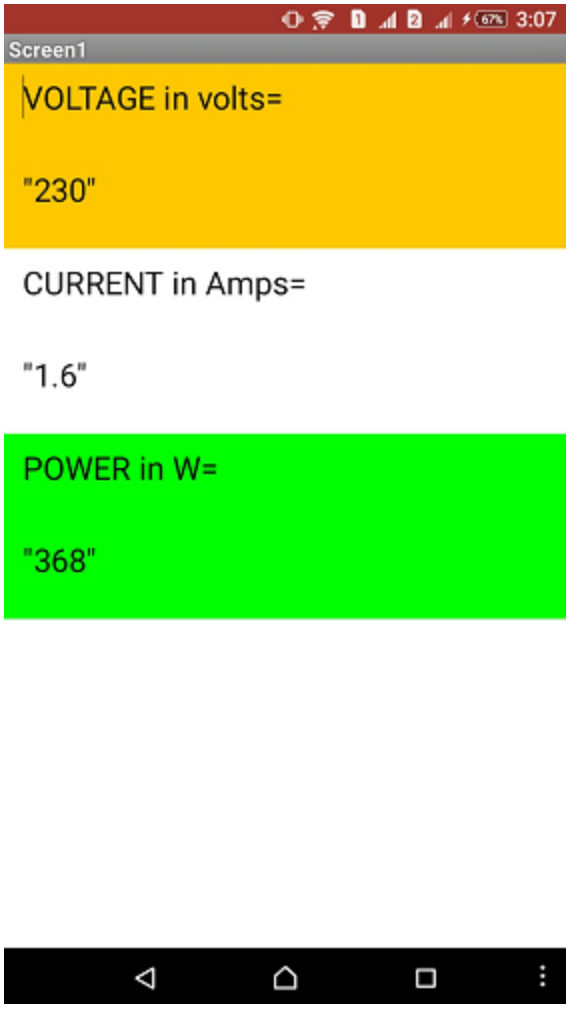




CASE 4: WHEN **4** LOAD BULB IS ON

THEN VOLTAGE=230 V, CURRENT= **1.6 A**, POWER= **368 W**





THANK YOU