CS-684-2018 Final Report

Smart Fault Monitoring Device

Team-6

Abhishek Pal, 173074015
Anil Garg, 173074018

Table of Contents

1. Introduction

Consider the scenario, when you are out of your house (for work or travel) & no body is at your home. At late night, when you reach the home exhausted from daily work, but you find that there is no electricity in your house… Reason could be there is short circuit, some appliance fault, or it could be that there is incoming utility supply itself… Now you have to rush to electrician late night.

To help out in such scenarios, we have come up with a Smart Fault Monitoring Device (SFMD), which will send an alert to user as soon as any fault occurs (MCB tripped, short-circuit, no electricity etc.).

This Smart Fault Monitoring Device can be used for monitoring the trip/fault status across an MCB connected in all domestic and industrial applications. This will log trip information and intimate to user using IoT infrastructure. Device can be fitted across existing MCB mounted in your home or factory installations.

Since this device has a wide impact on all residential and commercial establishments, it can also be extended to be a part of Smart City initiative.

2. Problem Statement

Aim of this project is to design a Smart Fault Monitoring Device, which can be used for monitoring the trip/fault status across MCB in domestic and industrial applications.

- Unit will monitor the mains supply & MCB trip contact status. There can be any one of the following states:
    - Power OK: means voltages on Primary & Secondary side of MCB are OK.
    - No Power: means incoming Mains supply is not present.
    - MCB Tripped: means MCB has got tripped due to overload or short circuit current etc.
- Unit will log above trip information and intimate to user using IoT device along with current & voltage values.

3. Requirements

3.1. Functional Requirements

- ADC Acquisition: Acquire the signals from voltage & current sensors.
- Measurement: Compute the RMS values of incoming voltage & current.
- Status Monitoring: Derive the power & MCB status from above measurements.
- Communication: Communicate the measurements to AWS server.
- Alert Message: Send alert message to user mentioning MCB trip status & measured values.

3.2. Non-Functional Requirements

- Safety: No live connection should be hanging to ensure user safety.
- Packaging: All components should be suitably packaged in a unit/box

3.3. Hardware Requirements

Following components were identified to monitor the voltage/current & derive the power status.

- Node MCU (ESP8266 based Wireless module)
- Tiva Launchpad (TM4C123GXL Microcontroller Board)
- AC Voltage Sensor (Step down Transformer)
- AC Current Sensor (ACS712 Hall Sensor)
- Signal Conditioning Board
- 5VDC Supply

3.4. Software Requirements

- TI CCS 6.1.2 (for Tiva launchpad Programming)
- Mongoose Operating System (for ESP8266 programming & MQTT implementation)
- e-Yantra platform (to access AWS (Amazon Web Service) for Dashboard & Alert Rules)
- Serial Console (for debugging)

4. System Design

4.1. Overall Scheme (Block Diagram)

The block level scheme of the system is shown below. The SFMD unit will interface to the exisiting MCB of your household. Primary & secondary contacts of the MCB are tapped to measure the voltage on both sides of the MCB. Also hall current sensor ACS712 is connected in series at the MCB output to measure the current.

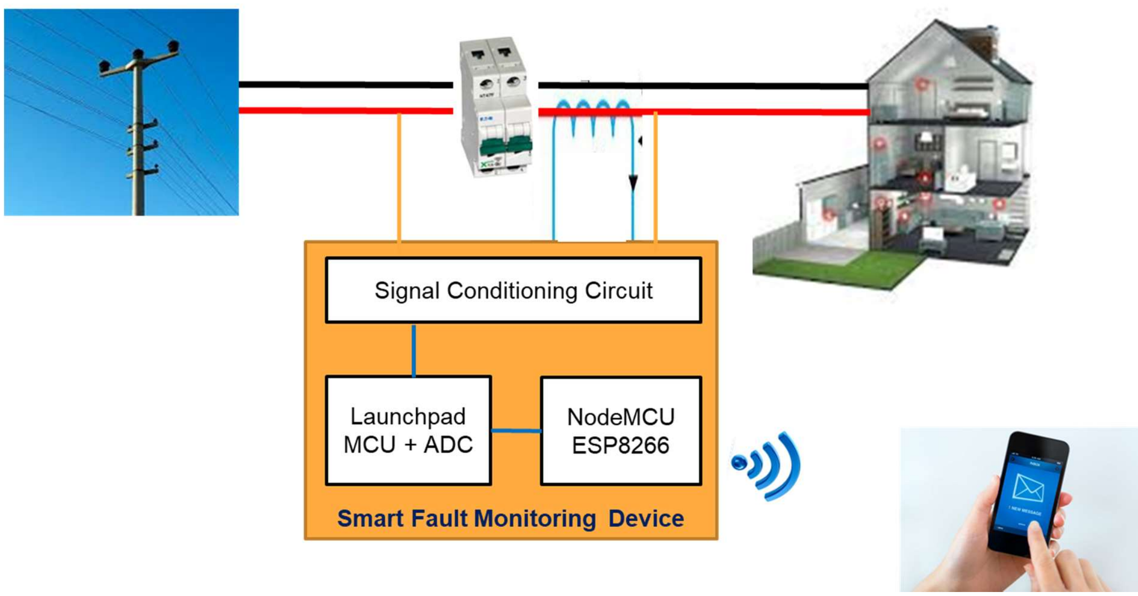


Fig – Block Diagram of Overall Scheme

The SFMD unit is designed & packaged into a box as shown ahead. The figure below is marked to show major components. Following are components of the unit:

1. Analog signal conditioning board
2. Hall Current Sensor
3. Voltage Sensor (Step-down Transformer)
4. Tiva Launchpad board
5. ESP8266 Node MCU Wifi Module
6. MCB

7. e-Yantra AWS Server for Dashboard display
8. Email interface for alert messaging

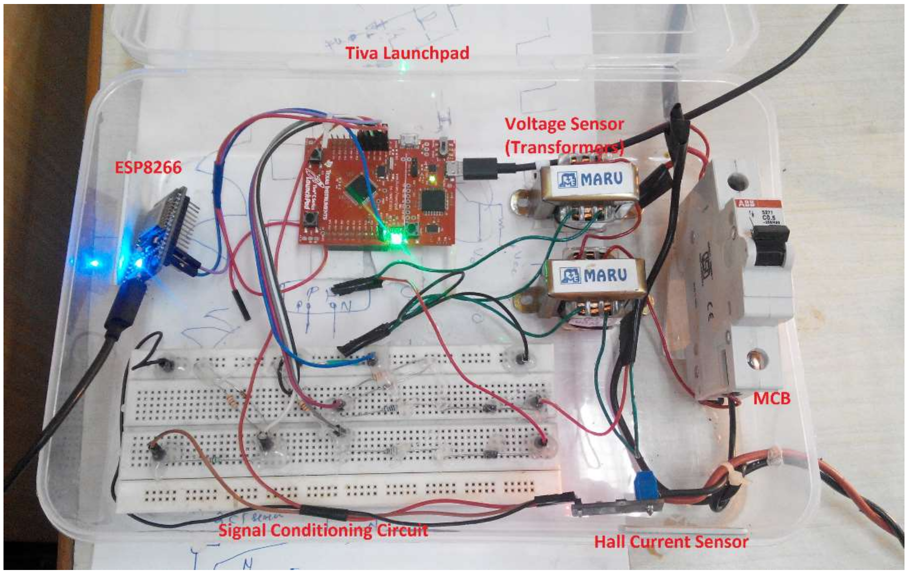


Fig: SFMD Unit (Packaged inside a Box)

4.2. Analog Signal Conditioning Circuit

A signal conditioning circuit is designed & assembled on breadboard. In order to measure the Primary & secondary voltages of MCB, the circuit below takes the stepped down voltage signal from transformer outputs. This is further divided to bring the levels within the input range of the ADC of Tiva controller.

Similarly to measure the current, a hall current sensor (ACS712) is used. ACS712 output is 0 to 5V with 2.5V DC offset. So with no current (zero current input), the sensor output is 2.5V. ACS output is passed through a voltage divider so as to keep maximum output within ADC's input range.

Component values are shown in the schematic shown ahead:

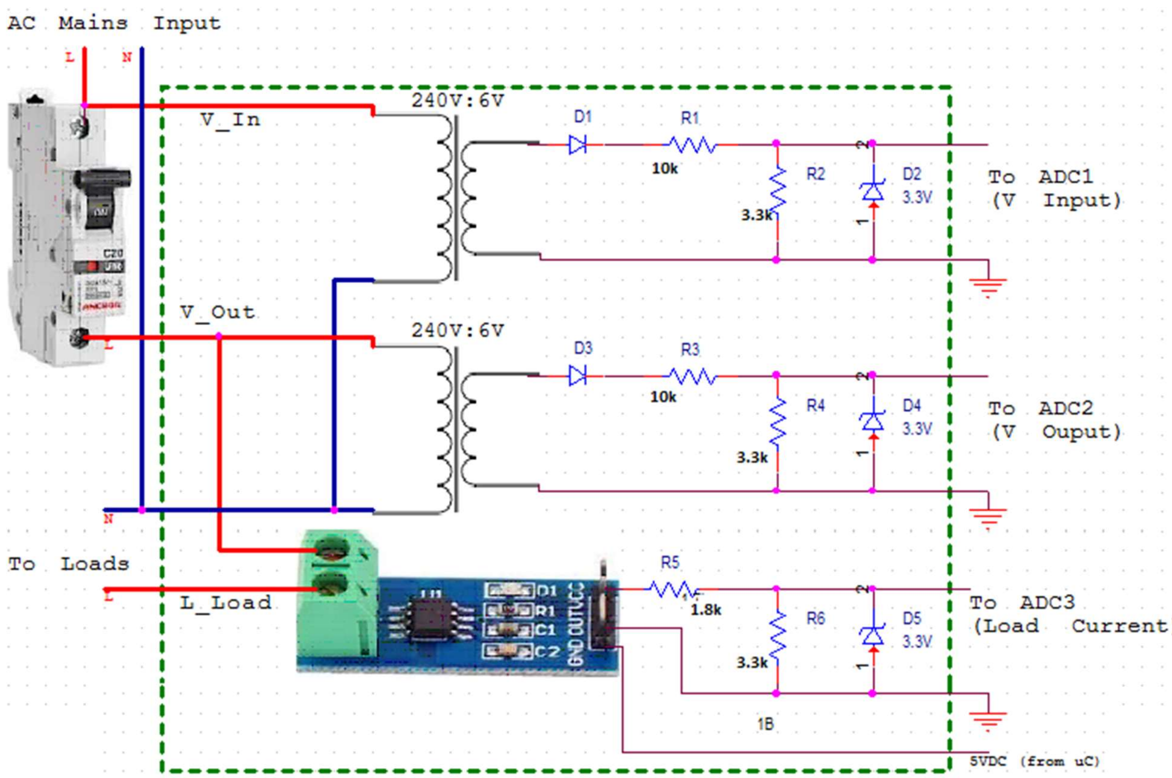


Figure – Circuit-Diagram of Analog Signal Conditioning before ADC

4.3. Tiva Launchpad Microcontroller:

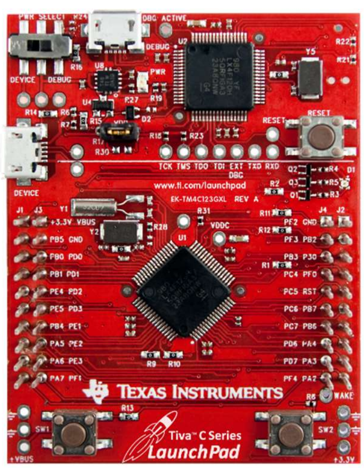


Figure – TI Tivaware Launchpad used for sensor acquisition

The Tiva launch is used to interface 3 sensors with 3 ADC channels of the microcontroller. ADCs are triggered every 1.25ms, so as to acquire the 16 samples od each 50Hz voltage/current cycles. These 16 samples are used to calculate root mean squared values of current & voltages.

The measured voltage & currents are used to derive the MCB trip contact status. Whenever a MCB is found to be tripped, last current & voltages are latched status meassages.

The status messages are sent every 5 second to ESP8266 microcontroller over a serial channel.

4.4. Node MCU (ESP8266) Wifi Module:

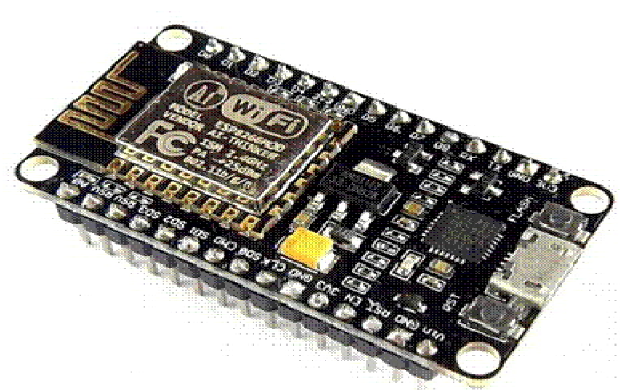


Fig: Node MCU ESP8266 Wifi Module

The Node MCU is ESP8266 based wifi moduled, which can be configured as access point as well as client. The ESP8266 is configured using Mongoose OS using mJS programming. ESP is configured to connect to an wifi SSID & password to access the e-yantra AWS endpoint over internet.

ESP's UART0 is connected to Tiva microcontroller's UART for serial communication between them. Each of the ESP module can be uniquely identified using its device ID in AWS server.

ESP is programmed to receive the serial frame containing following format:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
Status	Current MSB	Current LSB	Voltage Primary	Voltage Secondary	EOL (New Line Character)

Above frame is received from Tiva to ESP. The ESP then parses the parameters from the received packets, & formats it into a JSON string format to be published as an MQTT topic.

Also it checks for the change in the MCB status flag. Whenever a transition is detected, ESP sets a flag ‘sendEmail’ to high only for one cycle. All the parameters are updated with a 5 second timer.

4.5. MCB:

An MCB or Miniature Circuit Breaker is an electromagnetic device. It is enclosed completely in a moulded insulating material. The main function of an MCB is to open the circuit (which has been connected to it) automatically, when the current through MCB exceeds the value for which it is set. It can also be manually switched ON and OFF, if needed.

MCB used here for demo purpose is rated for 0.5A current. Although higher rated MCB also can be used without changing the program of the SFMD unit.

4.6. AWS e-Yantra Server & Dashboard:

e-Yantra AWS platform is used as Dashboard to display the measured current & voltages of the MCB in real-time. The ESP publishes the MQTT topics over AWS IoT Infrastructure. MQTT connection details for Client as obtained on server is shown below. ESP is programmed to publish the topic on the same ID’s.

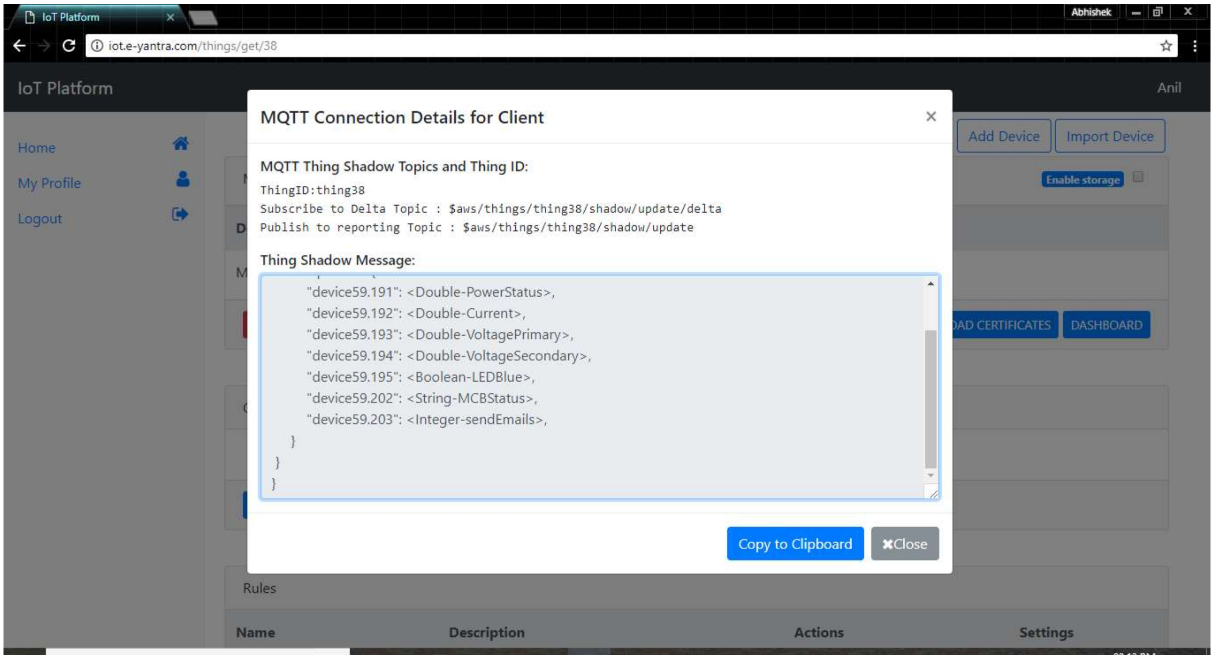


Fig: MQTT Connection details

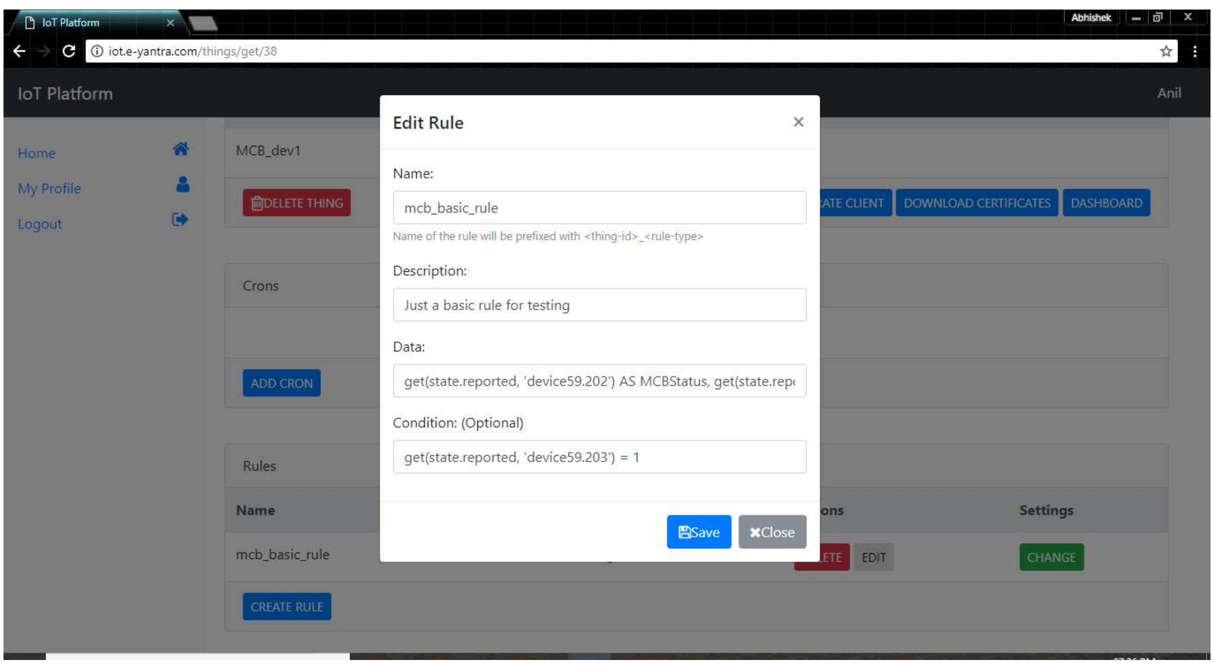


Fig: AWS SNS Rule for email

The e-Yantra screenshot above shows one device named 'MCB_dev1'. An associated rule created for AWS SNS (Simple Notification Service) is displayed with the rule to trigger an email at the time of fault detection.

Email Trigger Condition is configured as:

get(state.reported, 'device59.203') = 1

'device59.203' corresponds to the 'sendEmail' flag inside ESP8266, which is set whenever a change in current state is detected.

Data to be sent is configure with an SQL query:

get(state.reported, 'device59.202') AS MCBStatus, get(state.reported, 'device59.192') AS Current_mA, get(state.reported, 'device59.193') AS VoltagePrimary, get(state.reported, 'device59.194') AS VoltageSecondary

The above statement formats the data to be sent via email. Email ID is configured in settings as shown ahead.

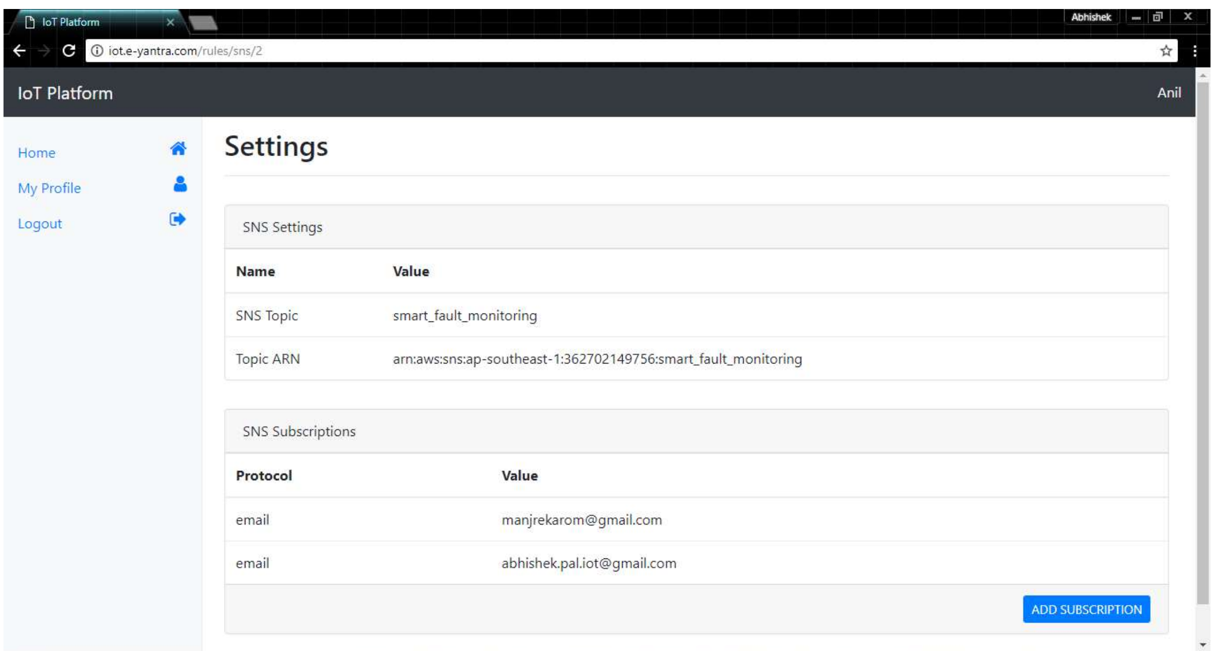


Fig: AWS SNS page showing email-ID & topic ARN setting

5. Working of the System and Test results

Individual modules are tested, before integrated testing to ensure that each module functions as per requirements. Tiva program sets the MCB status flag as per following conditions:

- **Power OK**: Voltages on Primary & Secondary side of MCB are OK.
- **No Power**: Mains supply itself is not present.
- **MCB Tripped**: MCB got tripped due to overcurrent, short circuit etc.

5.1. Test Results:

Hall Current Sensor:

Test results measured at output of ACS712-30A Hall-Effect current sensor are listed below. Output is sinusoidal waveform with 2.5V DC offset & peak to peak voltage as below for various AC current conditions:

Ip (Input Current), Amp (rms)	Sensor O/P Volt (Vpp)
0.25	56.8mV
0.5	101mV
1	220 mV
2	420 mV
3	620 mV
4	820 mV
5	980 mV
6	1.18 V
7	1.38 V
8	1.58 V
9	1.76 V
10	1.94 V

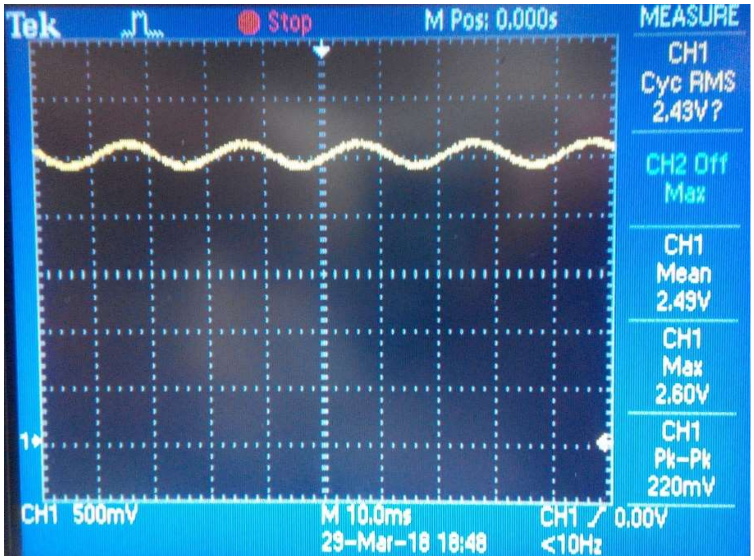
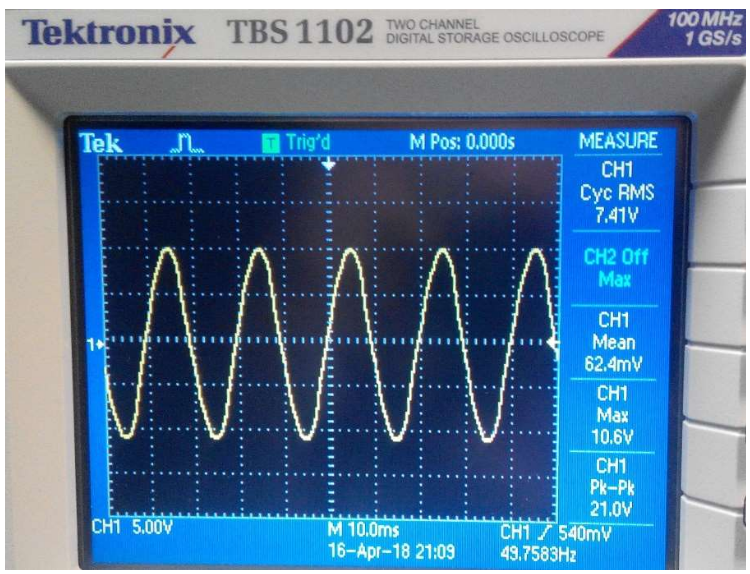


Fig: ACS712 current sensor output

Step-down Transformer Output

The output of step-down transformer with 240Vrms AC input is found to be +/- 10.6V as shown below:



Sensor Interface Circuit Testing & Calibration

The output of stepdown transformer is first converted to positive cycles only using half-wave rectifier (series diode) circuit. This is then brought down to be within the ADC input range of 0 to 3V. Below is the corresponding output of the signal conditioning circuit, which can be directly connected to ADC input

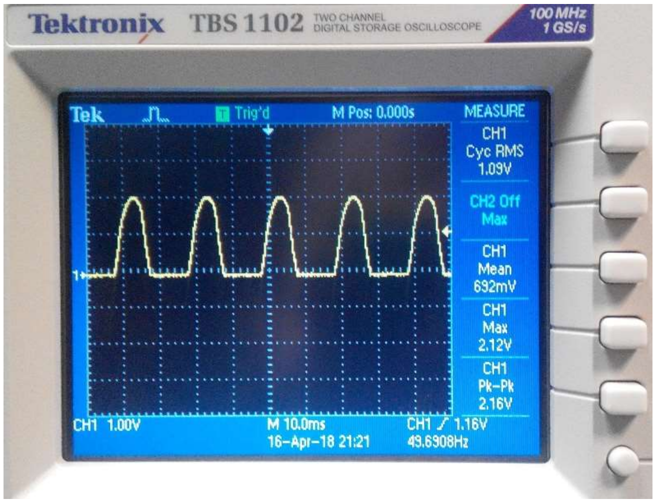
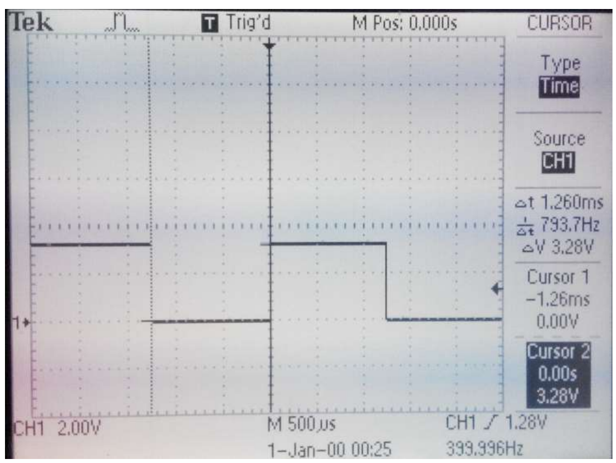


Fig: Voltage waveform at ADC Input after Signal Conditioning

Tiva ADC Acquisition & RMS Calculation

A 1.25ms Timer is configured in Tiva program to calculate the RMS value the current & voltages. As Main supply is 50Hz (~ 20ms), to get 16 samples, we need to sample it every 1.25ms. Below is one GPIO output triggered with 1.25ms timer, measurement confirms the 1.25ms timer operation.



Tiva Serial Interface

The measured results are sent though the UART interface to PC for debugging purpose. Same serial output is connected to ESP's UART to send these measurements to ESP. Below is output of Tiva serial port.

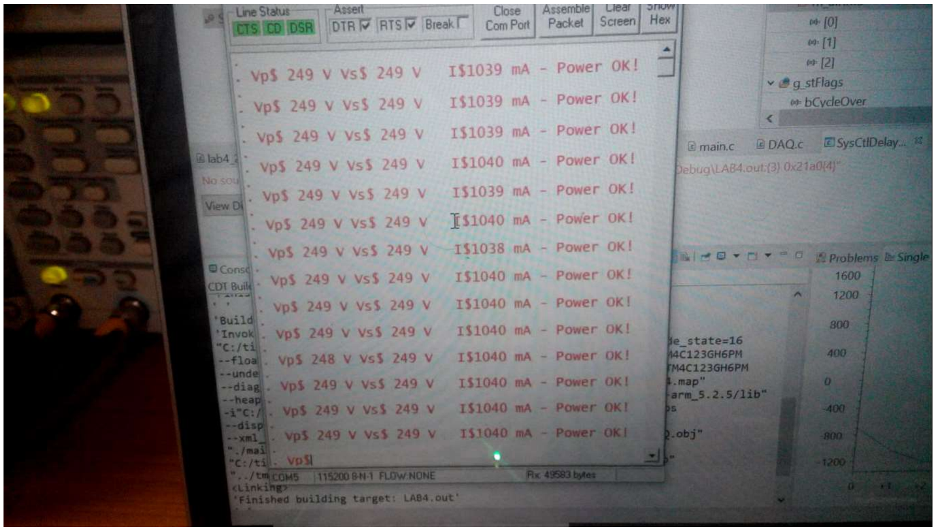


Fig – Serial Ouptut from Tiva Board displaying measured parameters & MCB status

ESP8266 Serial Interface

ESP 8266 serial functionality is tested with PC first. ESP is programmed to receive the serial frame containing following format:

Byte0	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5**
Status	Current MSB	Current LSB	Voltage Primary	Voltage Secondary	EOL (New Line Character)

Tiva is configured to send above frame to ESP using UART. The ESP then parses the parameters from the received packets, & formats it into a JSON string format to be published as an MQTT topic.

ESP8266 Wireless connectivity with AWS

ESP connectivity with AWS/Internet over wifi. A dashboard on e-Yantra is configured to display status, voltage & current parameters. Known values are sent to ESP from serial port in the previously mentioned format. And same values are verified on e-yantra Dashboard.

Integrated Test Setup

Integrated test setup is shown below. Entire unit is packaged in a small Box. To demonstrate various load current situations, multiple Bulbs of 200Ws were used. Unit connects to Internet with wireless hotspot. A laptop connected to AWS server (e-Yantra) over internet is used to displaythe Dashboard & email reception.

Fig: Integrated Test Setup with SFMD unit & Load

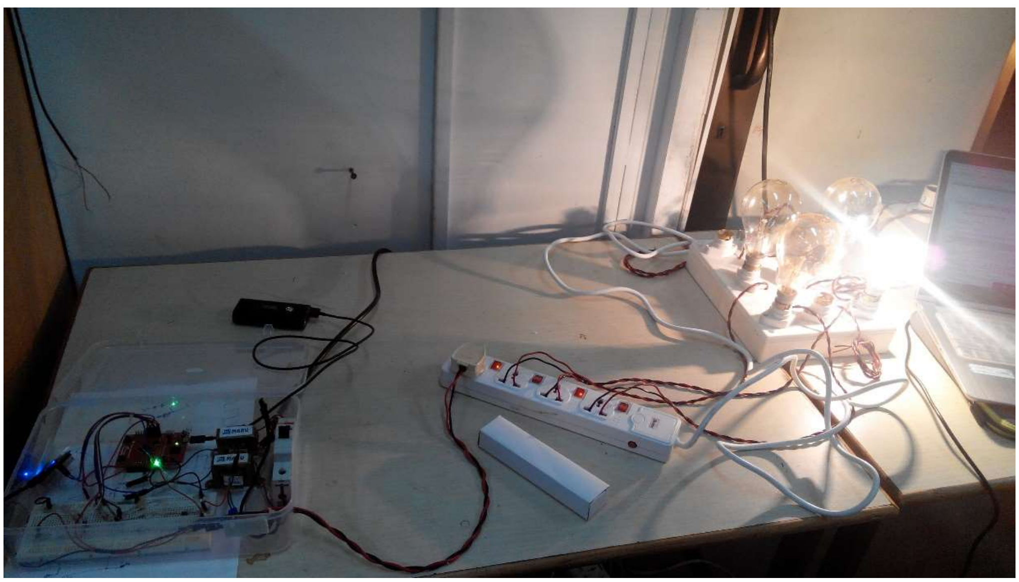


Fig: SFMD unit with Normal Load & Laptop for Dashboard display

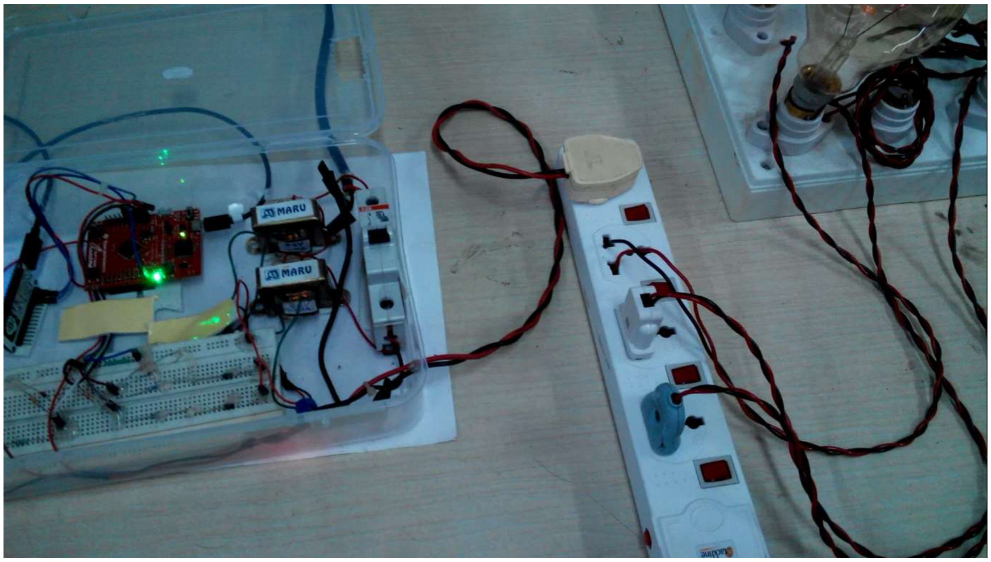


Fig: MCB in tripped condition due to overload

AWS Dashboard

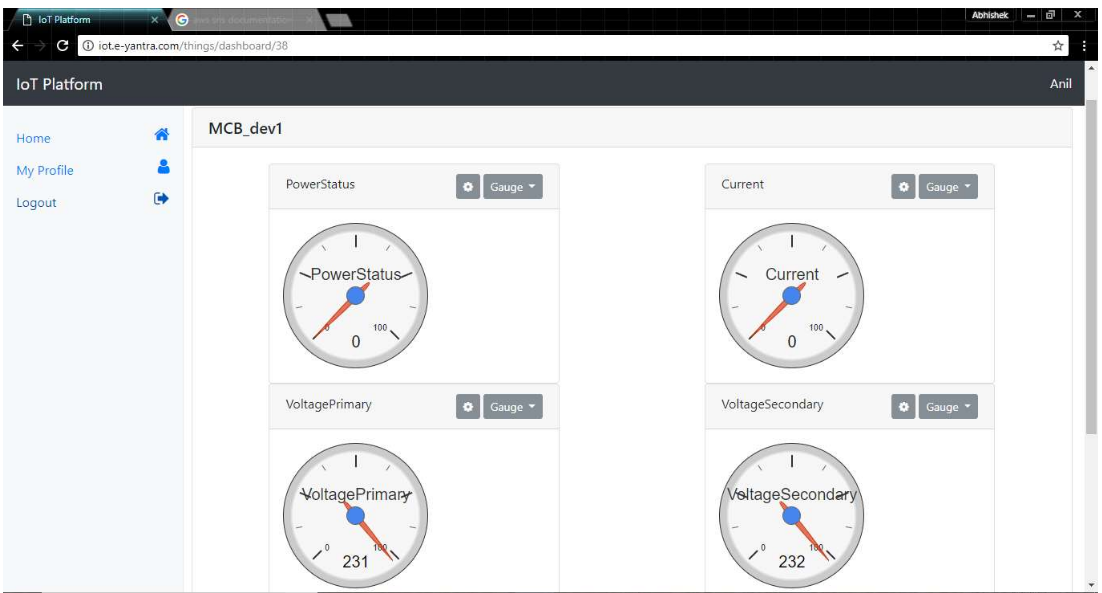


Fig: Dashboard during Power OK state (Normal Condition)

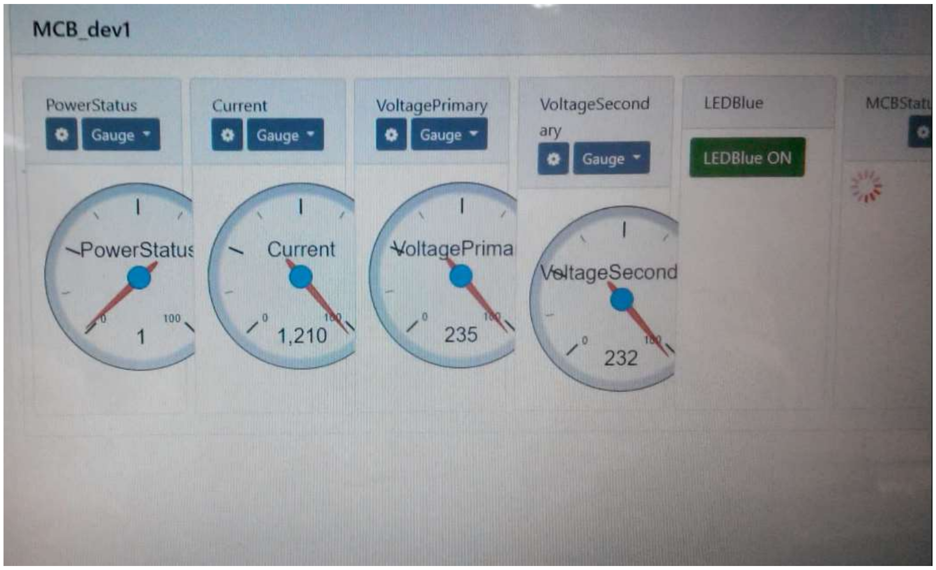


Fig: Dashboard during MCB tripped state due to overload

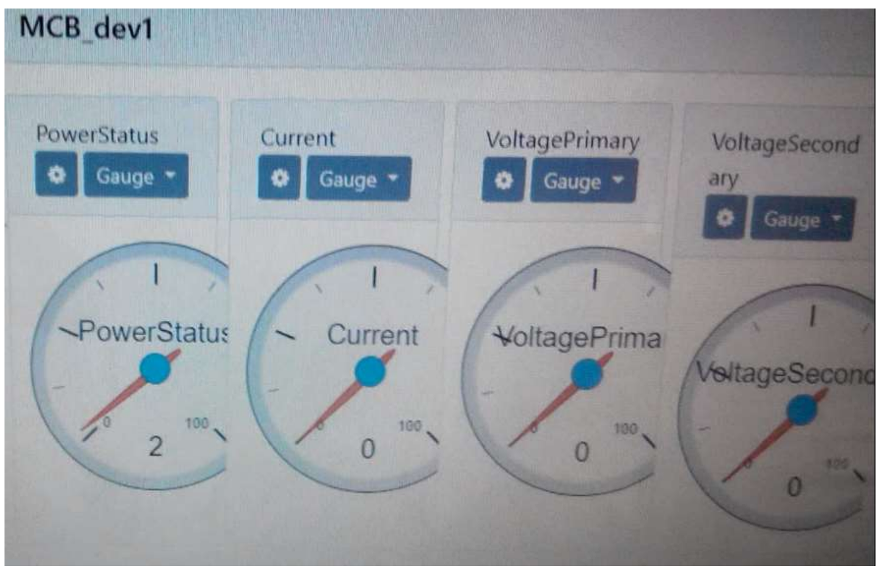


Fig: Dashboard during No Power state (Incoming Mains Off)

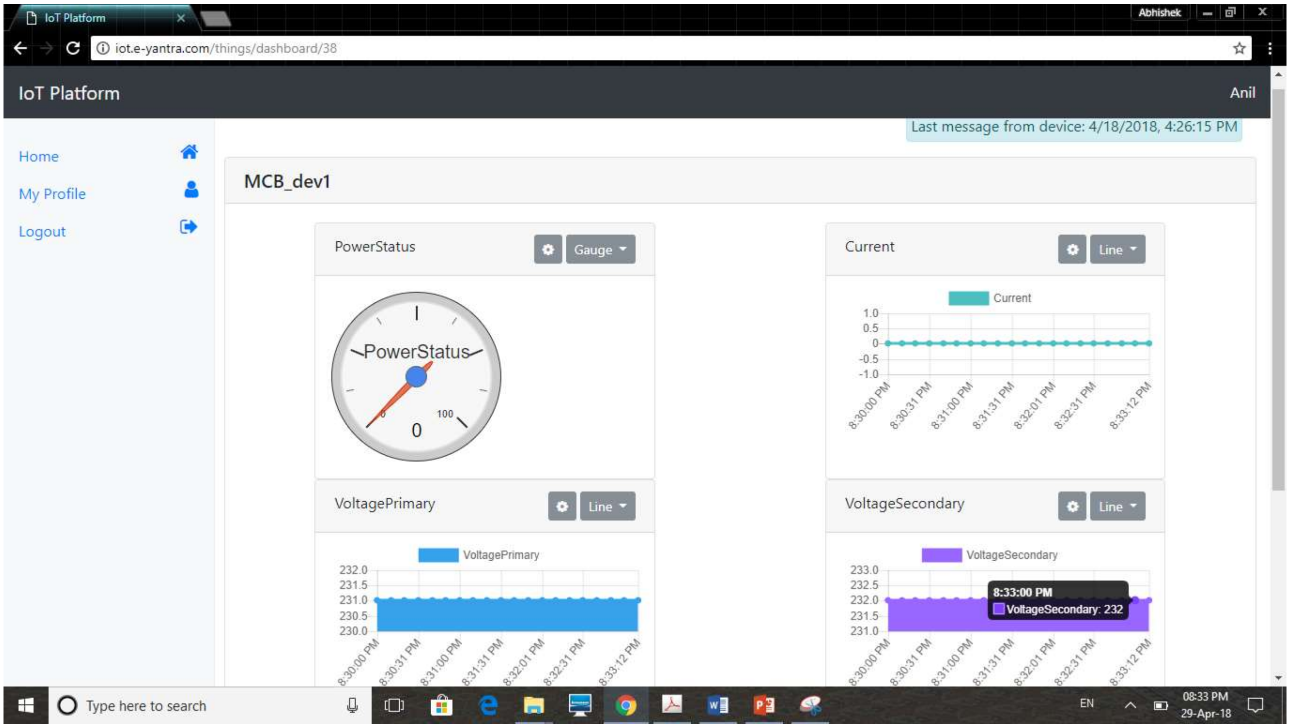


Fig: Alternate Dashboard with Time-Tagged History (Power OK state)

Email Interface

Below are emails received, whenever change in MCB/Power status is detected:

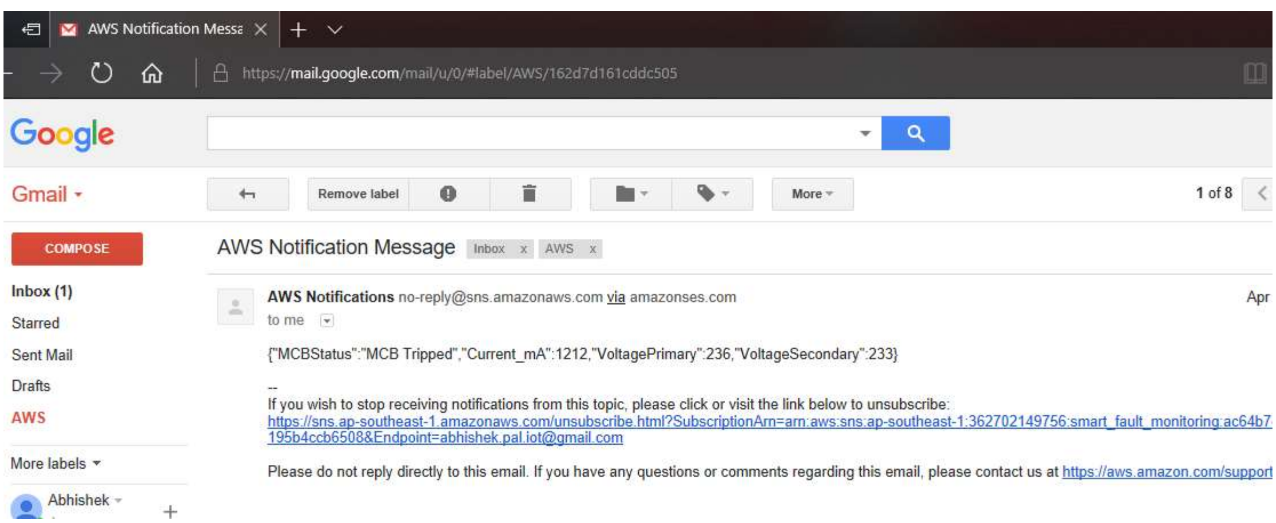


Fig: Email received when MCB is tripped

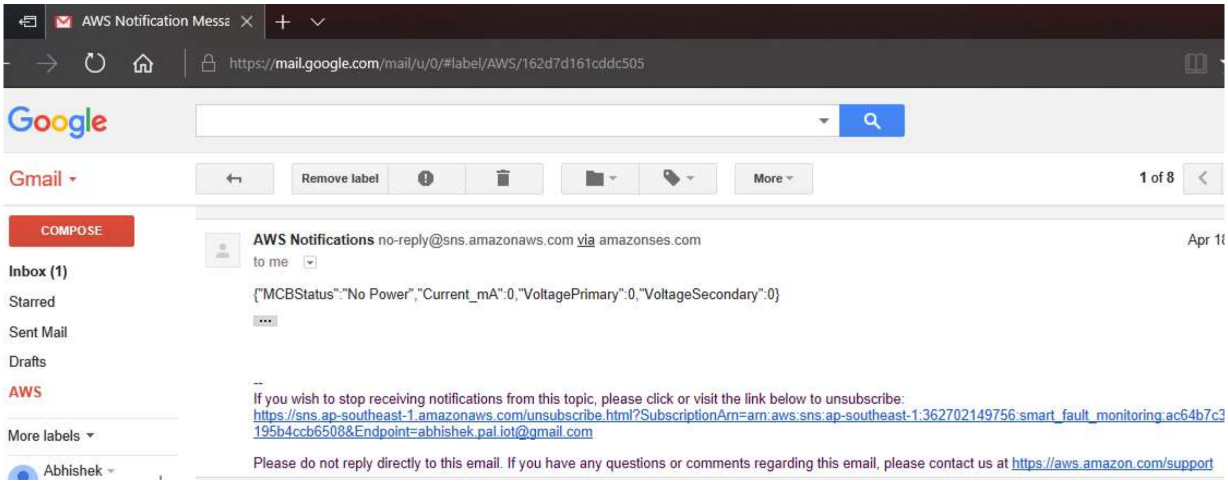


Fig: Email received when incoming Mains supply is off

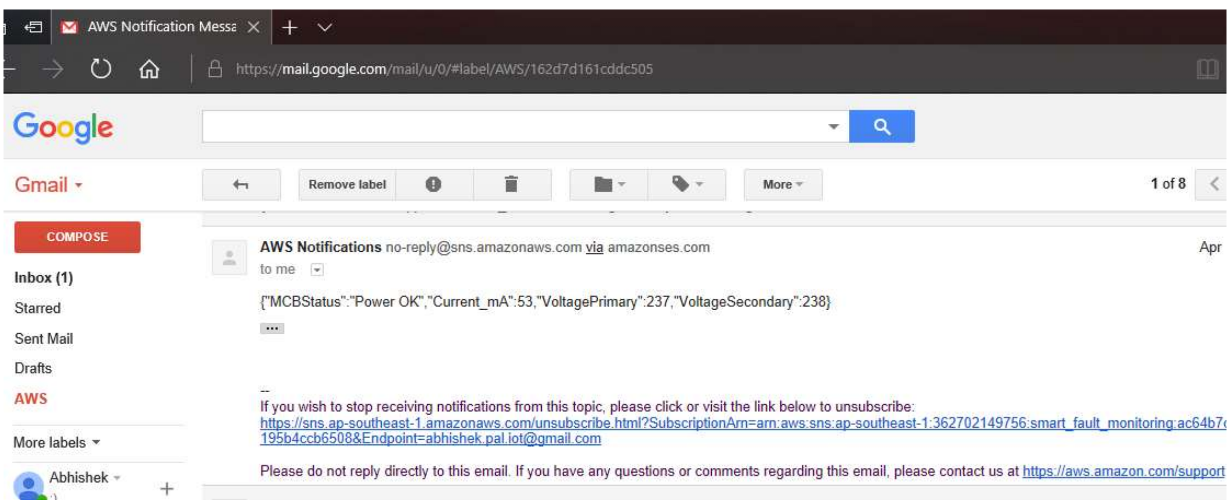


Fig: Email received when power restored back

6. Discussion of System

a) What all components of your project worked as per plan?

Although initially we faced issues with working of some of the modules. But finally we were able to interface all modules. Overall all components worked as were planned. Following are the main modules:

- ADC Interface
- Signal Conditioning Interface
- Serial Communication between Tiva & ESP8266
- AWS Communication with ESP8266
- Email/SMS Notification from e-Yantra

b) What we added more than discussed in SRS?

<u>Addition of Dashboard:</u>
Initially we had planned only to send a user notification along with current & voltages values. However during project review, it was suggested by project guide to add a Dashboard to display the current & voltage history.

<u>Latching of previous measurements whenever MCB trips</u>:
The latched current was getting zero reading after MCB trips. So the latching of current measurement at the time of fault had to be synchronized with fault detection task. This way current being driven at the time of fault is displayed on dashboard & in email notification both.

<u>Event based Email Notification</u>:
Initially multiple emails were getting triggered till fault condition is true. We changed logic to trigger single email, only when there is change in power/MCB state.

c) Changes made in plan from SRS:

Initially we had planned for using Arduino IDE for ESP8266 programming but we didn't any references to interface with AWS & MQTT on Arduino.

So we used Mongoose OS (MOS) for ESP8266 programming, as it has many online references/documentation for AWS & MQTT based topic publication/subscription. However for MOS programing environment, we had to learn mJS (Minimal Java-Script).

Addition of Dashboard: e-Yantra server was used to interface with AWS platform. A Dashboard is created to display current, voltage, power/MCB status along with previous history.

7. Future Work

In future, additional functionalities can provided as an add-on features to this device, such as:

- Android application for easily accessibility
- Universal device for all rating of MCBs
- OTA (Remote firmware upgrade)
- Oscillography during tripping
- Data logging
- Higher precise power analyser for metering application
- Earth fault monitoring system

8. Conclusions

We were able to meet all the product requirements as per plan, although with some hiccups during integration phase. Rugged testing of the product was carried out to ensure that all functionalities are working as per expectation. Unit sends an email alert to user, whenever there is any change in power status (No Power, MCB Tripped or Power OK).

This work can be generalised as smart measurement unit & where appropriate actions can be taken based on signal measurements. For AWS platform integration, the e-Yantra server developed by IITB e-Yantra team helped us to interface our unit with a nice looking dashboard.

9. References

1. Tiva microcontroller tm4c123gh6pm datasheet. http://www.ti.com/lit/ds/symlink/tm4c123gh6pm.pdf (Last viewed on: 29-Apr-2018)
2. ESP8266 Node MCU documentation: https://nodemcu.readthedocs.io/ (Last viewed on: 29-Apr-2018)
3. Mongoose Operating System Reference: https://mongoose-os.com/docs/reference/api.html (Last viewed on: 29-Apr-2018)
4. AWS SNS Notification: https://docs.aws.amazon.com/sns/latest/dg/GettingStarted.html (Last viewed on: 29-Apr-2018)
5. IITB e-Yantra Portal: http://iot.e-yantra.com/login (for Dashboard display)