

eYSIP2016

GREENHOUSE POWER MONITORING AND APPLIANCE CONTROL

Interns:

Avilash Mohanty

Email: avilashmohanty1920@gmail.com

Mobile: 9560372680

Abhishek Acharya

Email: abhi11796acharaya@gmail.com

Mobile: 8014692750

Mentors:

Saurav Shandilya

Vishwanathan Iyer

Parin Chheda

Duration of Internship: 10/06/2016 – 24/07/2016

2016, e-Yantra Publication

Contents

1	Greenhouse Power Monitoring and Appliance Control	2
1.1	Abstract	2
1.2	Completion status	3
1.3	Hardware parts	4
1.4	Software used	6
1.5	Assembly of hardware	21
1.6	Software and Code	27
1.7	Use and Demo	38
1.8	Future Work	46
1.9	Bug report and Challenges	47
	Appendices	50
A	Measurement board manual	51
A.1	Abstract	51
A.2	Circuit diagram	52
A.3	Description of circuit diagram	53
A.4	DIY	57
A.5	Circuit output the each pin	63

Greenhouse Power Monitoring and Appliance Control

1.1 Abstract

This project aims at developing a monitoring system which will consist of a Wi-Fi enabled device which can measure the basic electrical parameters of greenhouse's electrical appliances. This system will have Web based GUI through which user can monitor real-time graphically visualised data. The monitored parameters will be logged in a database for future reference and plotted on the web GUI logging page. This Web GUI also enables the user to switch and schedule ON/OFF as per requirement.



1.2. COMPLETION STATUS

1.2 Completion status

- Tasks Accomplished
 1. Literature survey of existing Real time Web Monitoring sites.
 2. Developed a Circuit to measure Current,Voltage,Phase,Frequency.
 3. Research and finalisation on Micro-controller board and Softwares to be used.
 4. Measurement of Current,Phase,Frequency,Voltage.
 5. Established a connection between server and micro-controller.
 6. Logged Real time data in database.
 7. Created Real Time updating Charts.
 8. Graphical Visualisation of logged data.
 9. Interfaced relay with the micro-controller.
 10. Created a Webpage to control the connected devices.
 11. Scheduling feature.
 12. 3 device Support for monitoring and controlling.
 13. Feedback system to compare real-device status and system's data.
 14. Terminal based GUI to configure the 'Smart Switchboard/Plug'.
 15. Code Documentation and Project Report.
- Incomplete Tasks
 1. Responsive web design.

Front-end can be improved and made responsive by using bootstrap.



1.3. HARDWARE PARTS

1.3 Hardware parts

1. List of hardware used:-

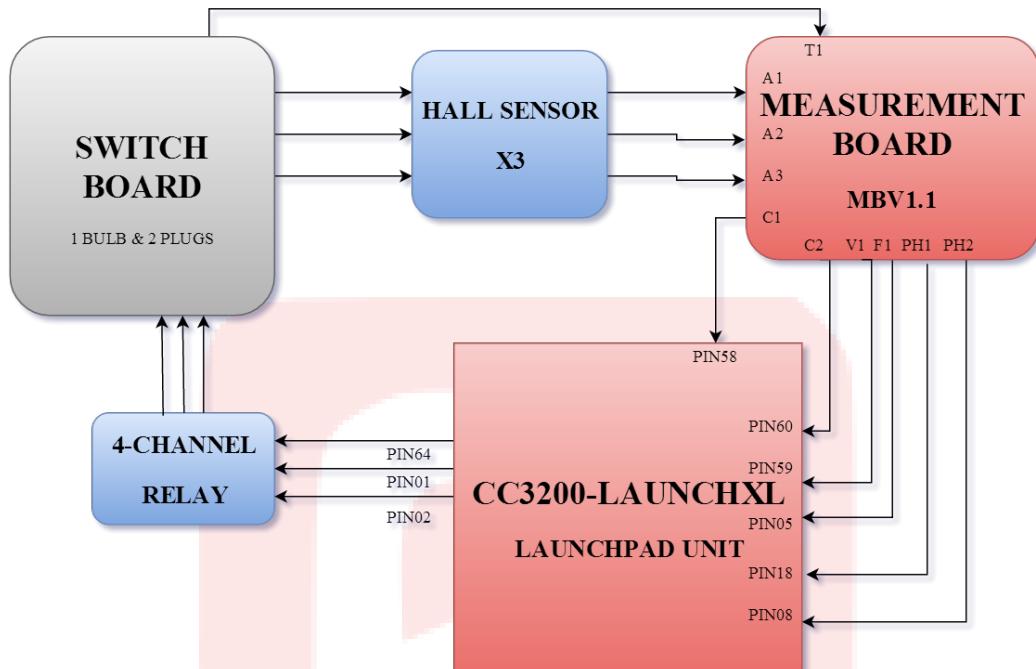
Name of hardware	Specification	Quantity
Electrical Switch Board	2-Plug and 1 Bulb Holder and 3 switches	1
Potential Transformer	230V-9V 100mA	1
Hall Effect Sensor Module	ACS712 (30A)	3
Bulb	100W	1
CC3200-LAUNCHPAD	TI's CC3200-LAUNCHXL <i>(With CC3200-HZ on chip wifi microcontroller)</i>	1
Soldering Iron	220V, 40W	1
Soldering Wire	50gm	1
Jumper Wires	Female to Female Male to Female	
4-Channel Relay Board	230V AC (10Amp)	1
Measurement Board <i>(Designed & Developed By Eyantra during eYSIP-16)</i>	MBV1.1	1

2. Details of hardware used:-

- Electrical Switch Board (Purchased from local market)
- Potential Transformer [Vendor link](#)
- Hall Effect Sensor Module [Datasheet](#), [Vendor link](#),
- Bulb (Purchased from local market)
- CC3200 LAUNCHPAD [Datasheet](#), [Vendor link](#)
- Jumper Wires [Vendor link](#)
- 4-Channel Relay Board [Datasheet](#), [Vendor link](#)
- Measurement Board [A](#)

1.3. HARDWARE PARTS

3. Connection diagram:-



HERE:-

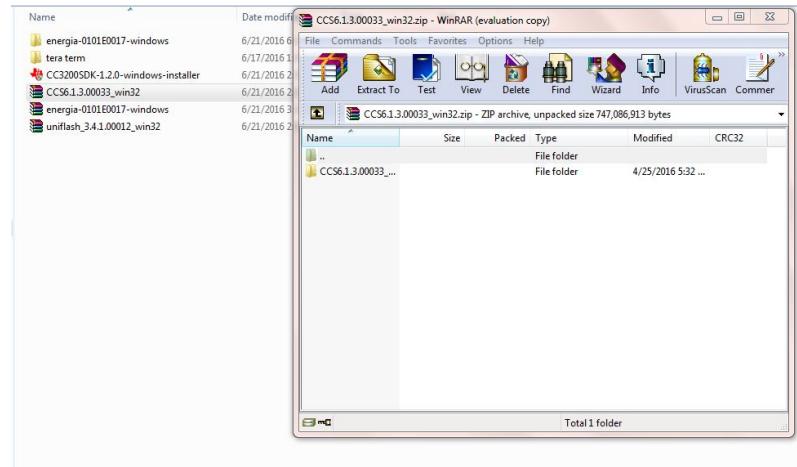
- A1: current sensor input pin 1 (Analog pin)
- A2: current sensor input pin 2 (Analog pin)
- A3: current sensor input pin 3 (Analog pin)
- V1: transformer output pin (Analog pin)
- C1: current1 signal out pin (Analog pin)
- C2: current2 signal out pin (Analog pin)
- V1: voltage signal out pin (Analog pin)
- F1: frequency signal out pin (Digital pin)
- PH1: phase 1 signal out pin (Digital pin)
- PH2: phase 2 signal out pin (Digital pin)
- All the pins used in CC3200 Launchpad are General purpose I/O pins.



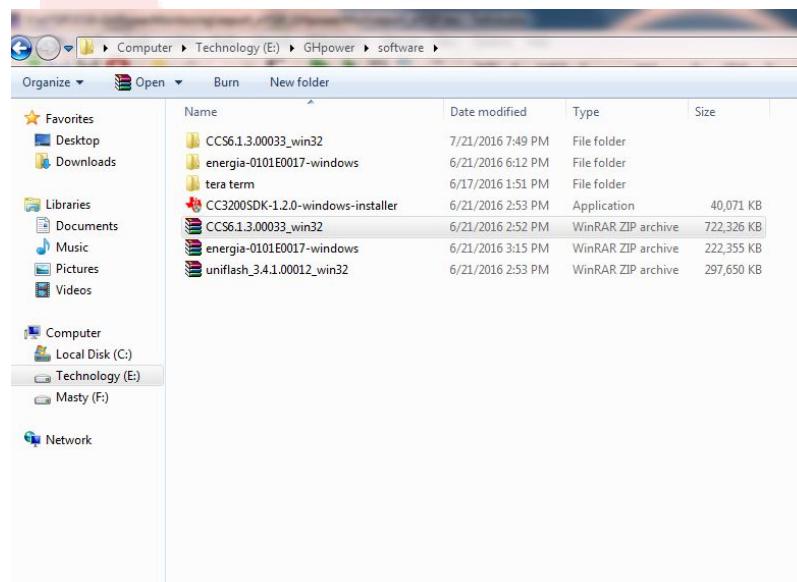
1.4. SOFTWARE USED

1. Code Composer Studio

- Detail of software: version 6.1.3.00033, [download link](#)
- Installation steps:
 - click on the rar file named "*CCS6.1.3.00033_win32*"



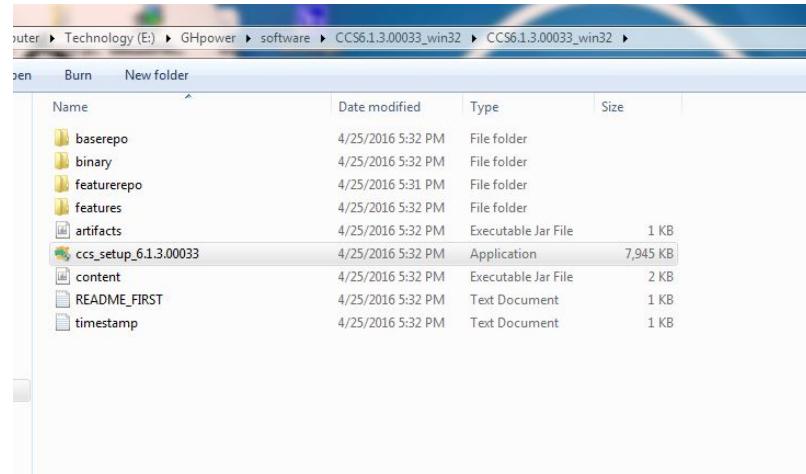
- Now click on the extracted folder



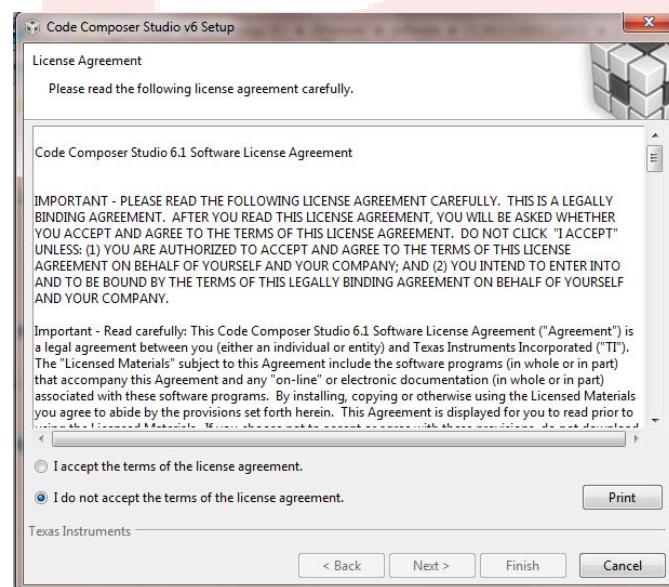


1.4. SOFTWARE USED

- click on the .exe setup file



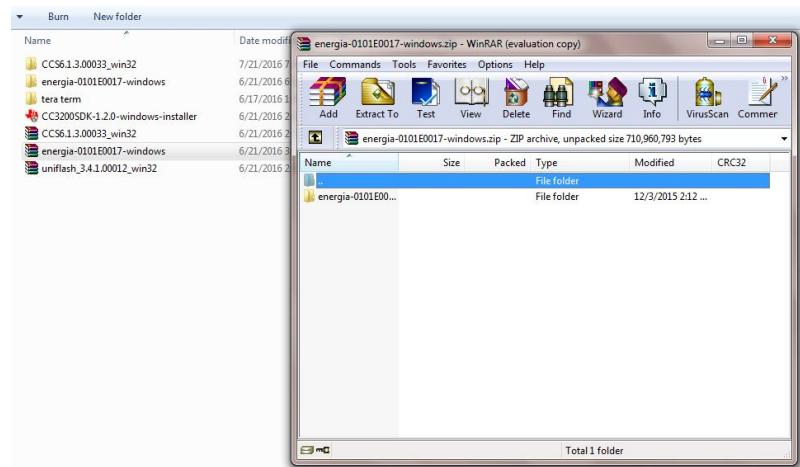
- now continue to click on next till complete installation of Code Composer studio



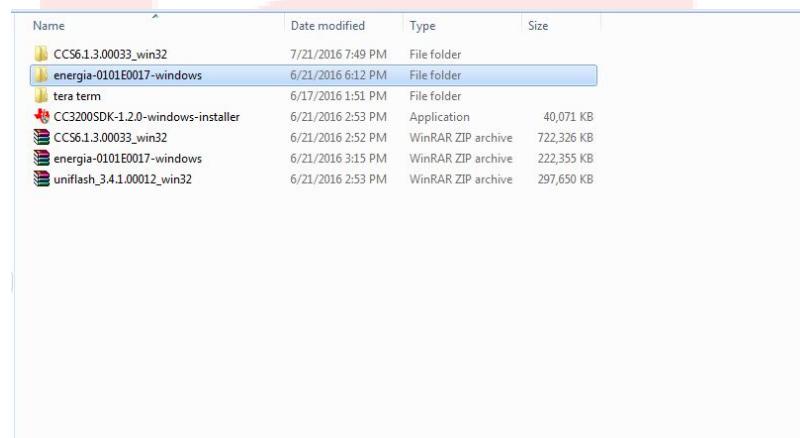


1.4. SOFTWARE USED

- **Energia** – Detail of software: version 6.1.3.00033, [download link](#)
 - Installation steps:
 - click on the rar file named ”*energia-0101E0017-windows*”



- Now click on extracted folder with same name





1.4. SOFTWARE USED

- click on energia .exe application file

A screenshot of a Windows File Explorer window showing the contents of the Energia distribution folder. The folder structure includes subfolders like _MACOSX, drivers, examples, hardware, java, lib, reference, tools, and several DLL files. An 'energia - Shortcut' icon is also present.

Name	Date modified	Type	Size
_MACOSX	12/3/2015 1:36 PM	File folder	
drivers	12/3/2015 1:33 PM	File folder	
examples	12/3/2015 1:36 PM	File folder	
hardware	12/3/2015 1:36 PM	File folder	
java	12/3/2015 2:12 PM	File folder	
lib	12/3/2015 2:12 PM	File folder	
reference	12/3/2015 1:36 PM	File folder	
tools	12/3/2015 1:34 PM	File folder	
cygiconv-2.dll	12/3/2015 1:33 PM	Application extens...	947 KB
cygwin1.dll	12/3/2015 1:33 PM	Application extens...	1,829 KB
energia - Shortcut	6/21/2016 6:14 PM	Shortcut	1 KB
energia	12/3/2015 2:12 PM	Application	1,059 KB
libiconv-2.dll	12/3/2015 1:33 PM	Application extens...	1,213 KB
libusb0.dll	12/3/2015 1:33 PM	Application extens...	43 KB
revisions	12/3/2015 1:33 PM	Text Document	28 KB
rxtxSerial.dll	12/3/2015 1:33 PM	Application extens...	76 KB

- code compile upload and enjoy

A screenshot of the Energia IDE showing a sketch titled 'sketch_jul21a'. The code consists of a standard 'setup' function with a comment about running once, and a 'loop' function with a comment about running repeatedly. The IDE has a red-themed interface.

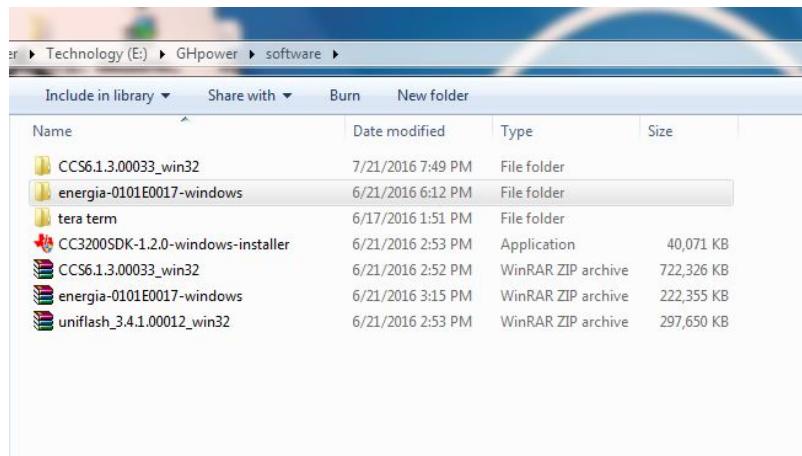
```
sketch_jul21a | Energia 0101E0017
File Edit Sketch Tools Help
sketch_jul21a
void setup()
{
  // put your setup code here, to run once:
}
void loop()
{
  // put your main code here, to run repeatedly:
}
```



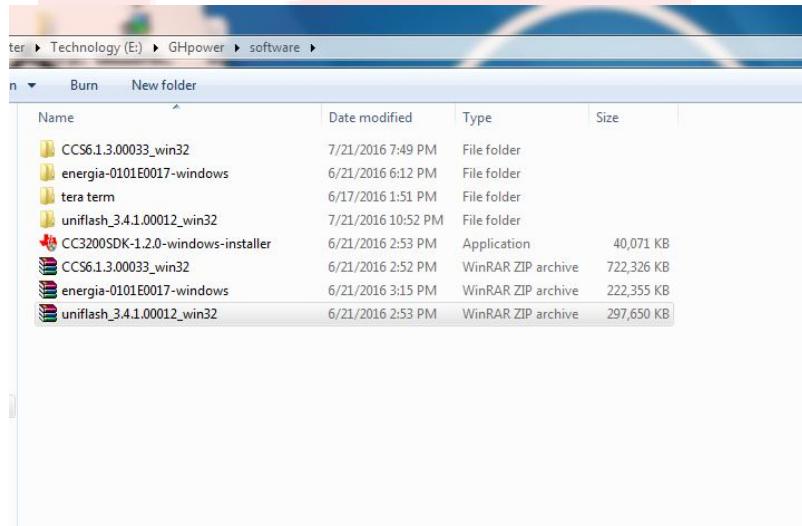
1.4. SOFTWARE USED

2. CCS Uniflash

- Detail of software: version 3.4, [download link](#)
- Installation steps:
 - click on the rar file named "*uniflash_3.4.1.00012_win32*"



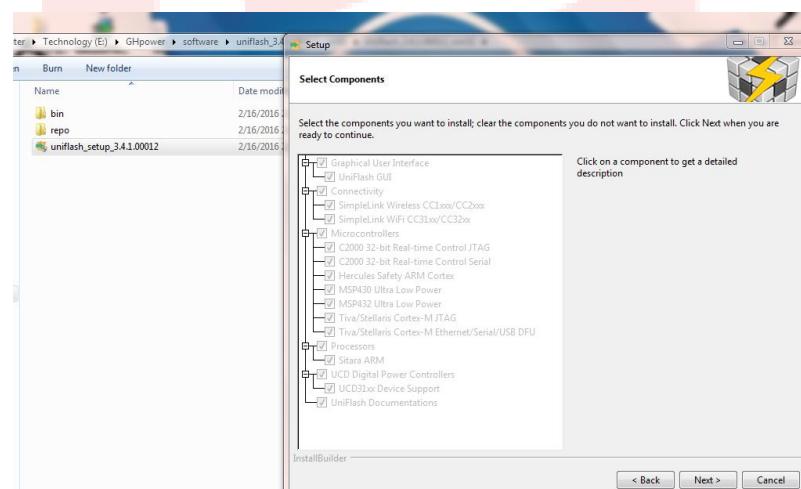
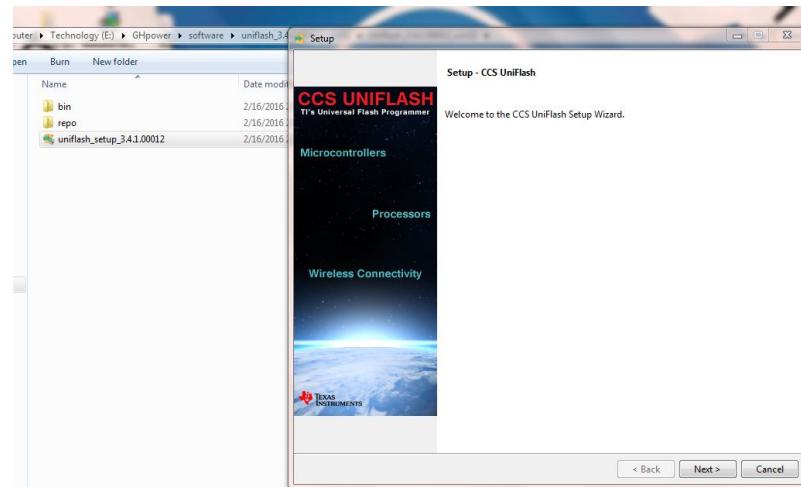
- Now click on extracted folder with same name





1.4. SOFTWARE USED

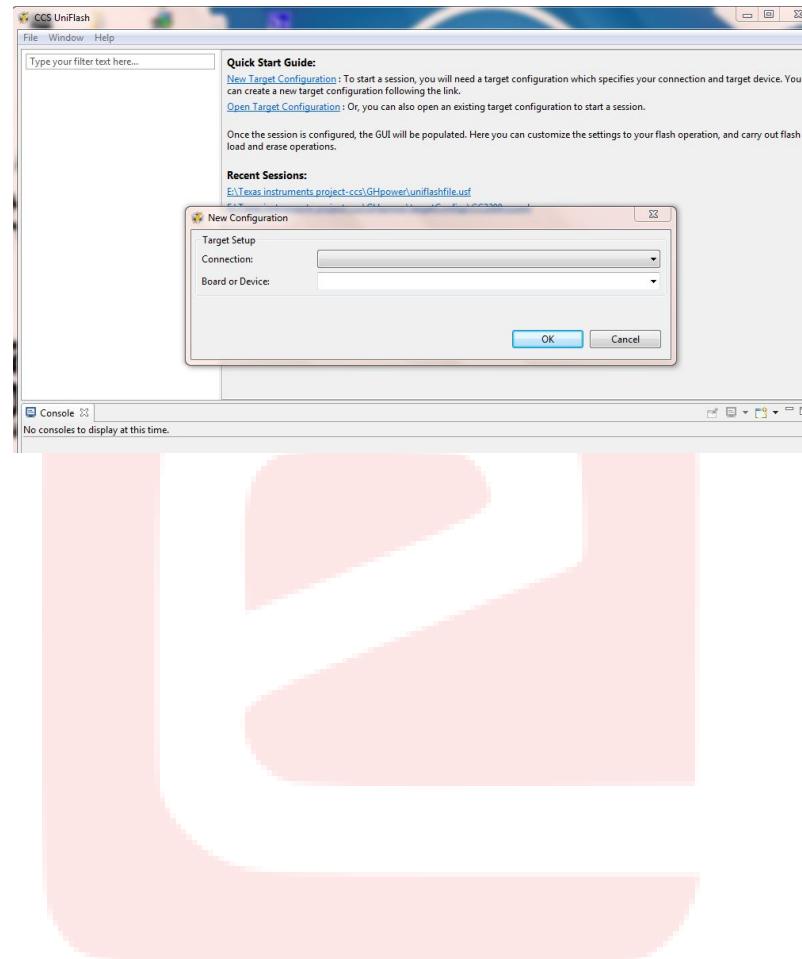
- click on *uniflash-win.exe* application file and continue to click on next till installation complete.





1.4. SOFTWARE USED

- open uniflash, select COM port and make a .ccxml file for cc3200 launchpad

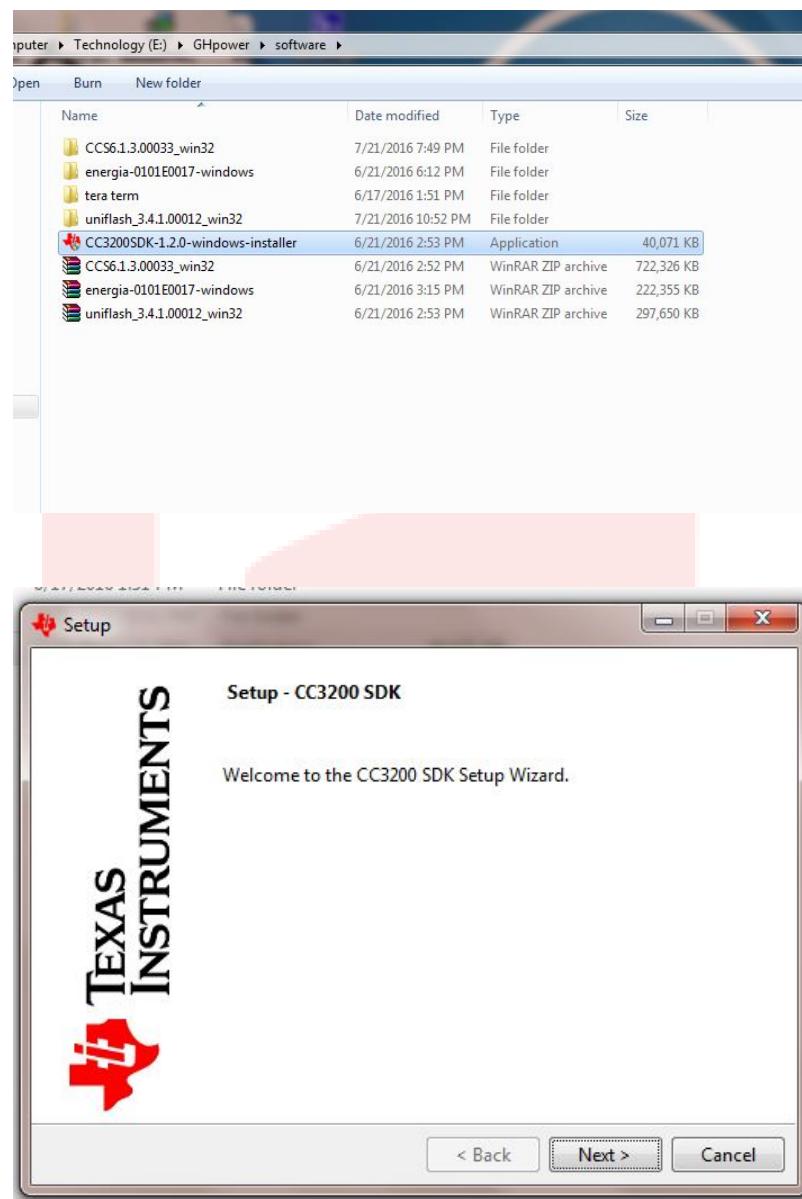




1.4. SOFTWARE USED

3. CC3200 SDK from TI

- Detail of software: version 1.2.0, [download link](#)
- Installation steps:
 - click on CC3200SDK.exe application file and install the sdk by clicking next button till installation complete





1.4. SOFTWARE USED

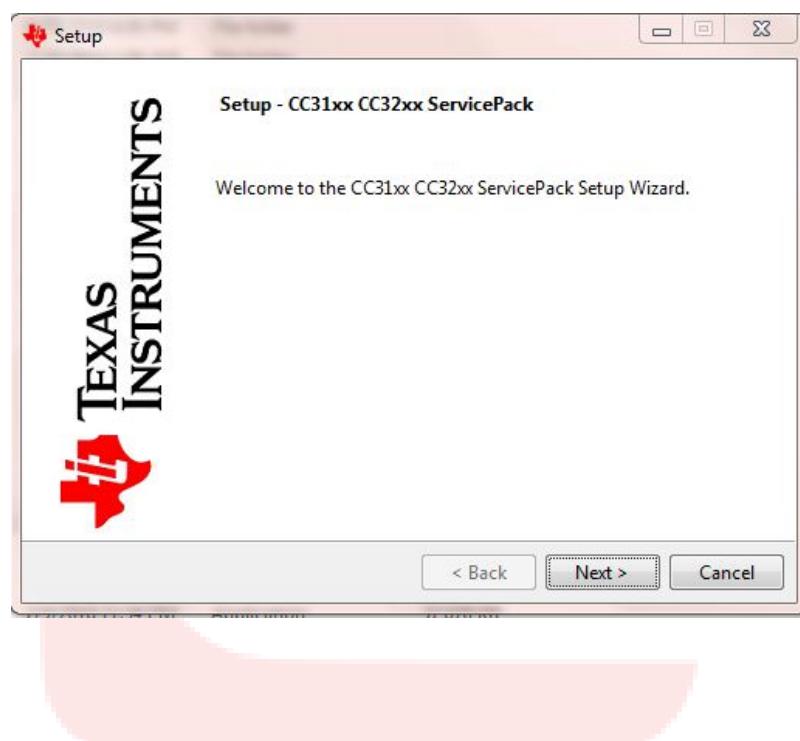
4. CC3200 service pack

– Detail of software: version 1.0.0.1.2, [download link](#)

note:- service pack should be compatible with your launchpad version

– Installation steps:

- click on CC3200servicepack.exe application file and install the sdk by clicking next button till installation complete

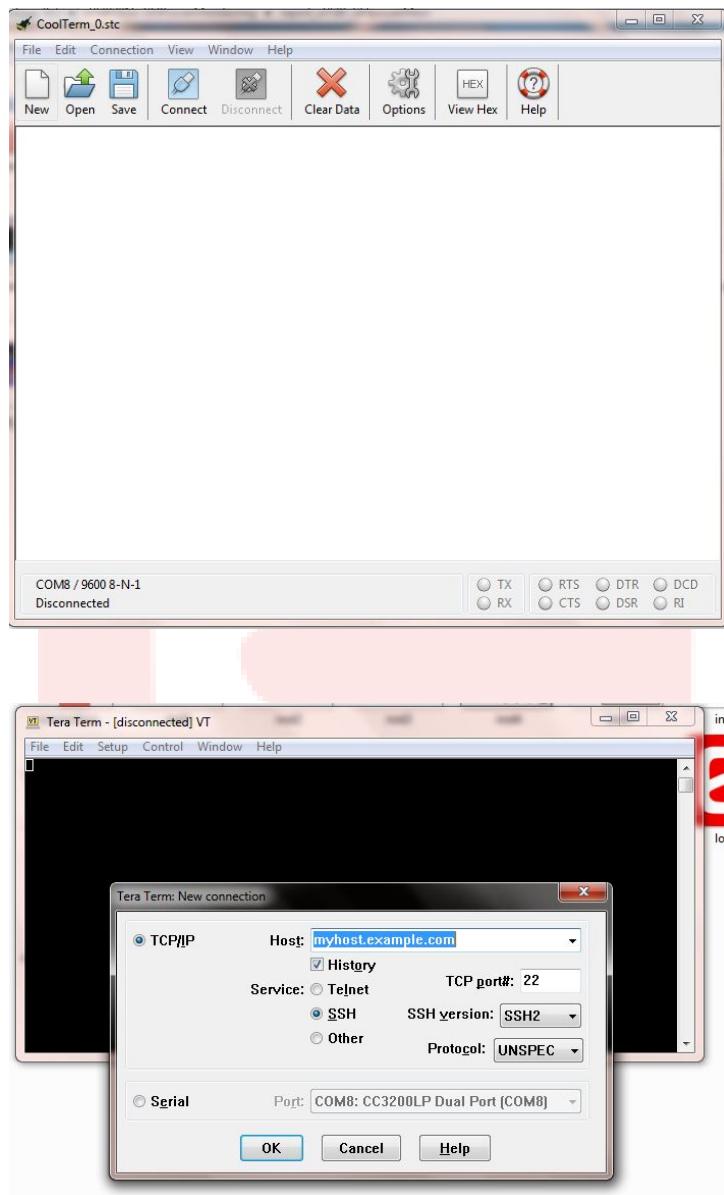




1.4. SOFTWARE USED

5. Terminal softwares: Coolterm & Teraterm

- Detail of software: version 1.4.6, [Coolterm](#)
- Detail of software: version 1.4.6, [Teraterm](#)



1.4. SOFTWARE USED

6. Setting up the environment
 - (a) Software used:Windows OS
 - (b) Version:Windows 10
7. Setting up the Server & Database
 - (a) Software used:XAMPP
 - (b) Version:XAMPP version 3.2.2
 - (c) [XAMPP Download Link](#)
 - (d) [XAMPP Installation steps.](#)

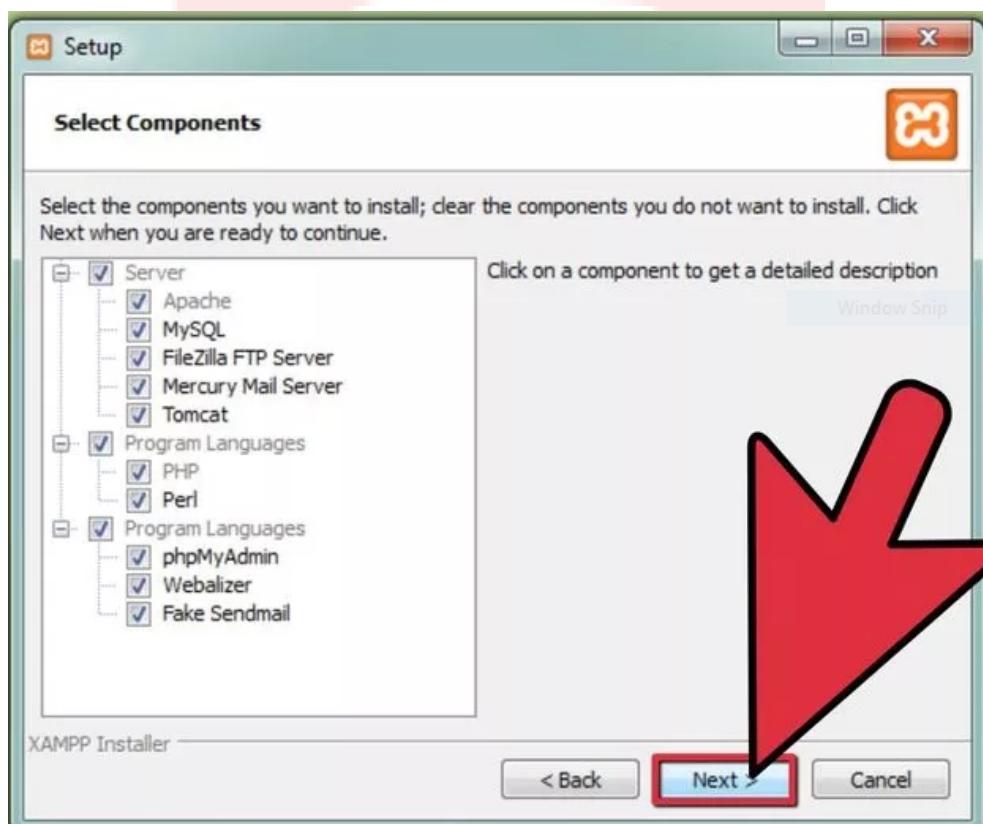


Figure 1.1: Component Selection

1.4. SOFTWARE USED

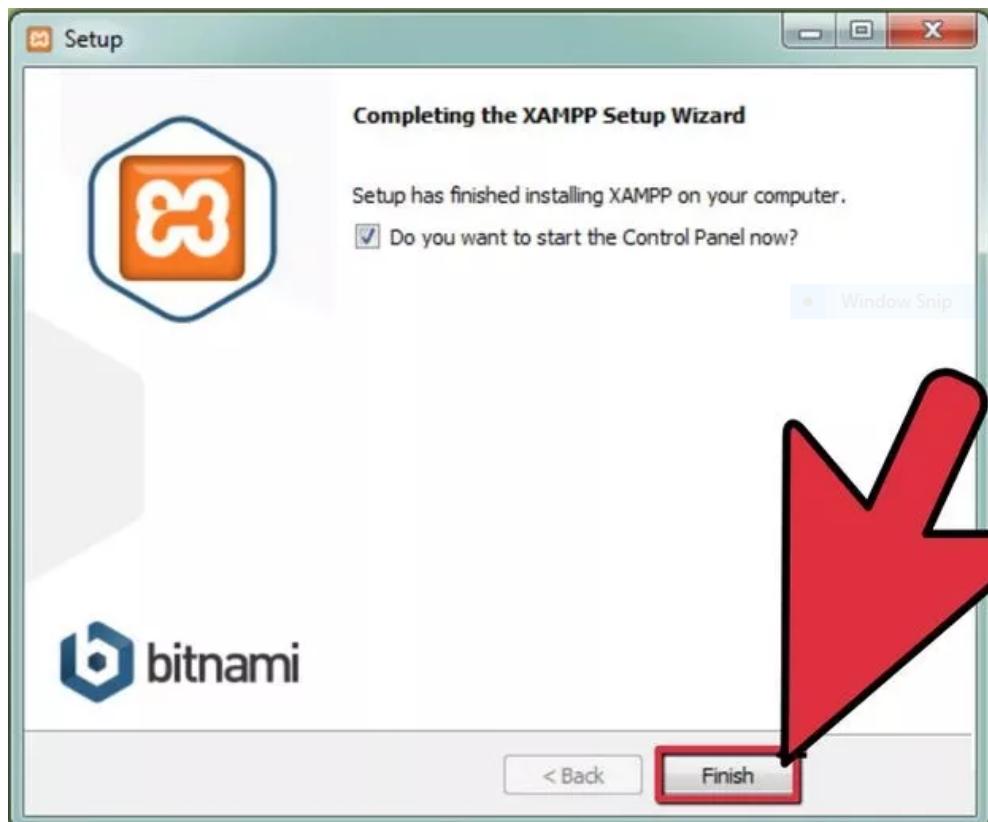


Figure 1.2: Finish Installation



1.4. SOFTWARE USED

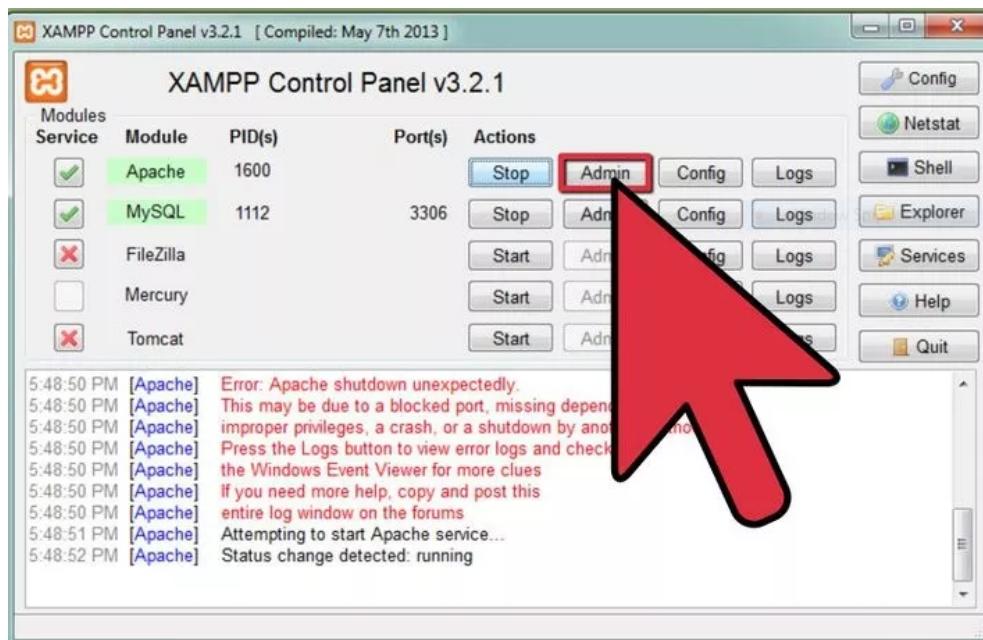


Figure 1.3: Open XAMPP Control Panel

NOTE: If Server rejects connection requests by micro-controller, follow these steps.



1.4. SOFTWARE USED

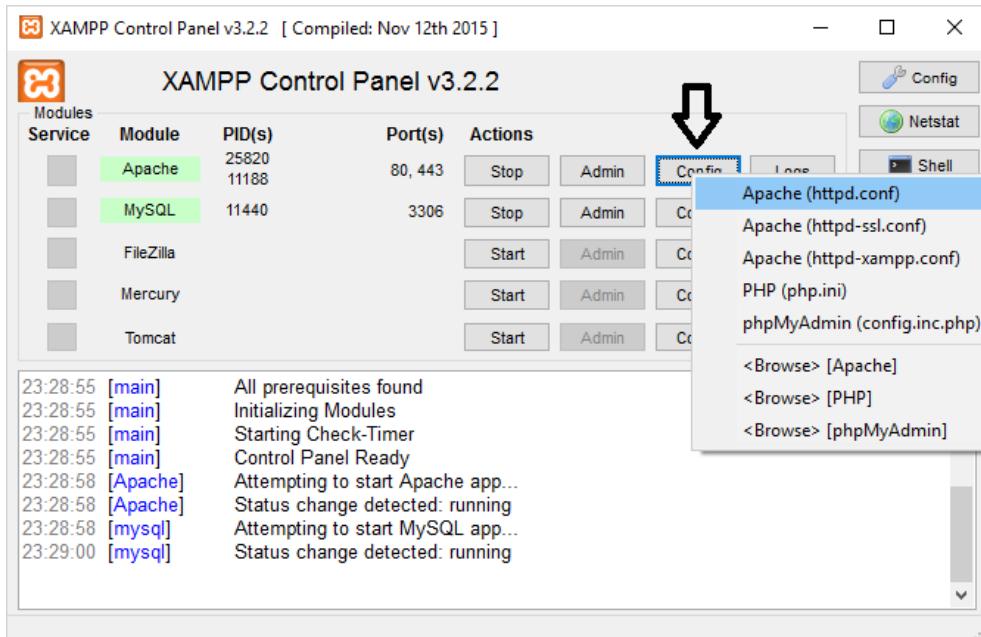


Figure 1.4: Open httpd.conf

- (e) Open httpd.conf file in configuration setting for Apache Server in XAMPP.

```
httpd.conf - Notepad
File Edit Format View Help
# virtual host being defined.
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
# ServerAdmin postmaster@localhost
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost:80
#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
<Directory />
    AllowOverride All
    Options All
    Order allow,deny
    Allow from All
</Directory>
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.

```

Figure 1.5: Change httpd.conf

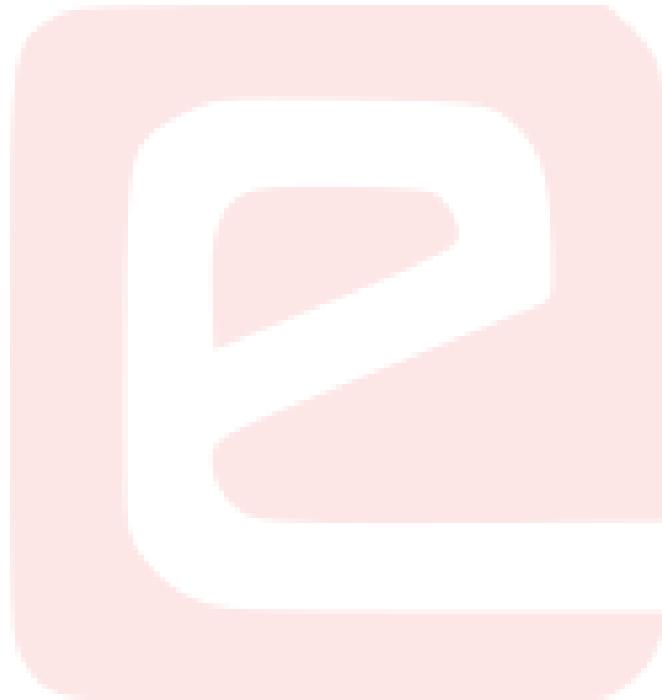
- (f) Set
AllowOverrideAll



1.4. SOFTWARE USED

*OptionsAll
Orderallow, deny
AllowfromAll*

8. Setting up the Editor
 - (a) Software used : [Notepad++ v6.9.2](#)



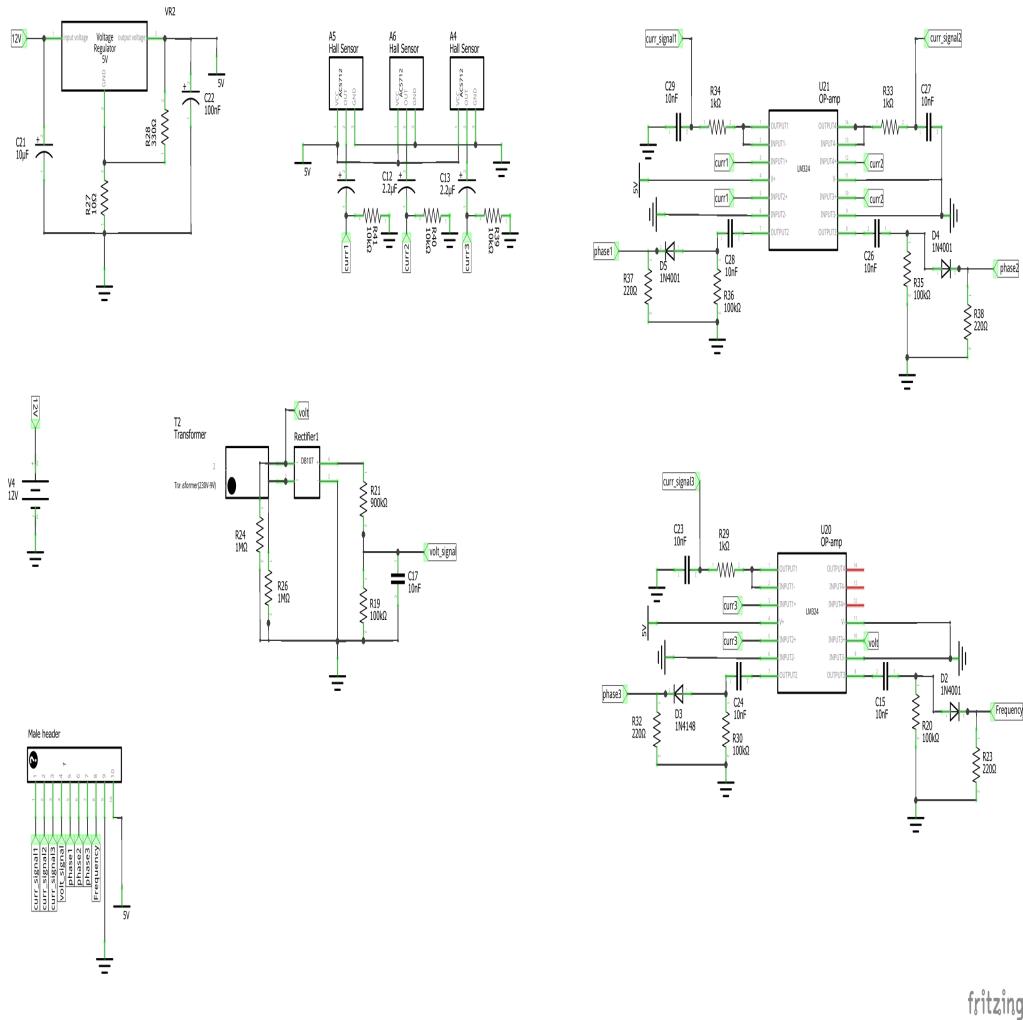


1.5. ASSEMBLY OF HARDWARE

1.5 Assembly of hardware

Circuit diagram and Steps of assembly of hardware with pictures for each step. For Detailed information about *Measurement Board A*

Circuit Diagram



fritzing

1.5. ASSEMBLY OF HARDWARE

Step 1

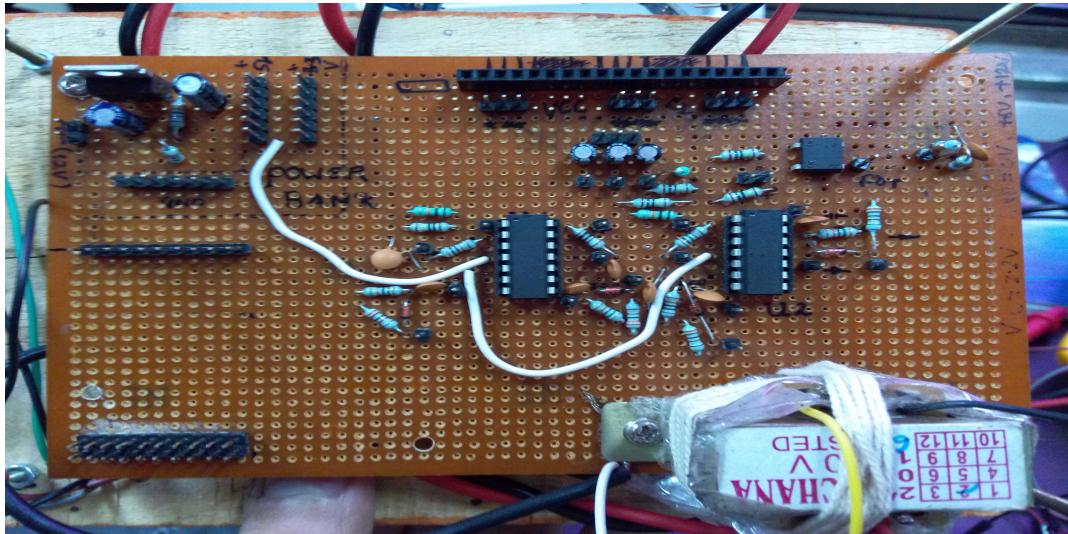
Make a basic switch board which has two plug and one bulb holder mounted on it as shown in figure below



1.5. ASSEMBLY OF HARDWARE

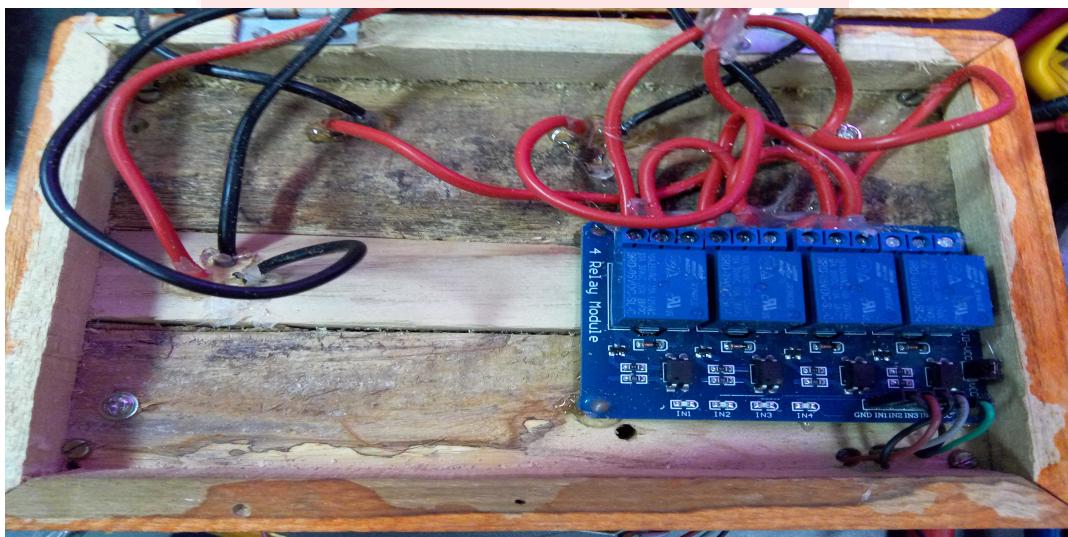
Step 2

Now assemble the "measurement board" with the switch board at the location of your convenience. Make sure you have fixed the measurement board firmly with switch board as shown in figure below.



Step 3

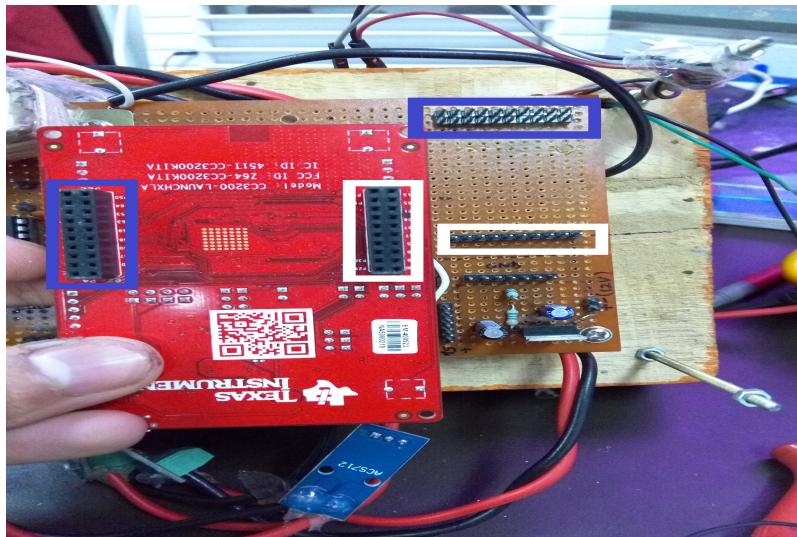
Now connect and fix relay board with switch board at the location of your convenience as show in figure below



1.5. ASSEMBLY OF HARDWARE

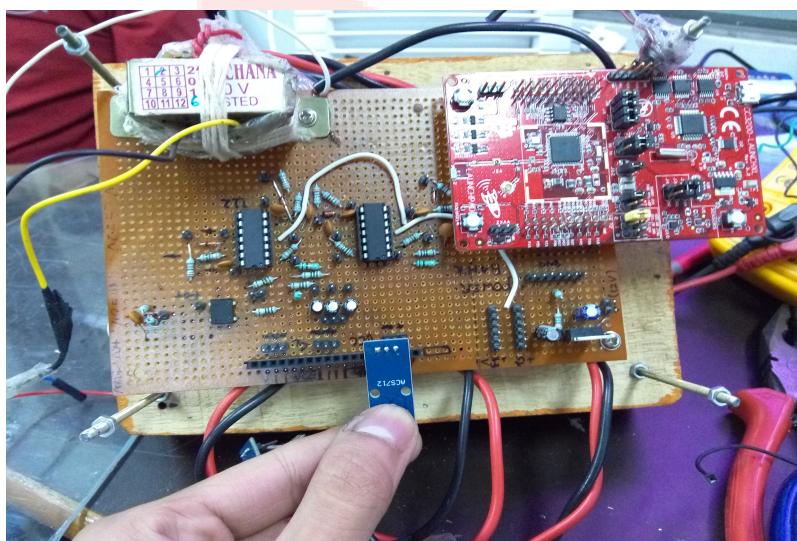
Step 4

Now connect the *CC3200 – launchpad* at the male headers 21 and 22. For more detail see board manual [A](#)



Step 5

Now connect all the three hall sensor at pin number 1,2 and 3 appropriately on measurement board. for more detail see board manual [A](#)





1.5. ASSEMBLY OF HARDWARE

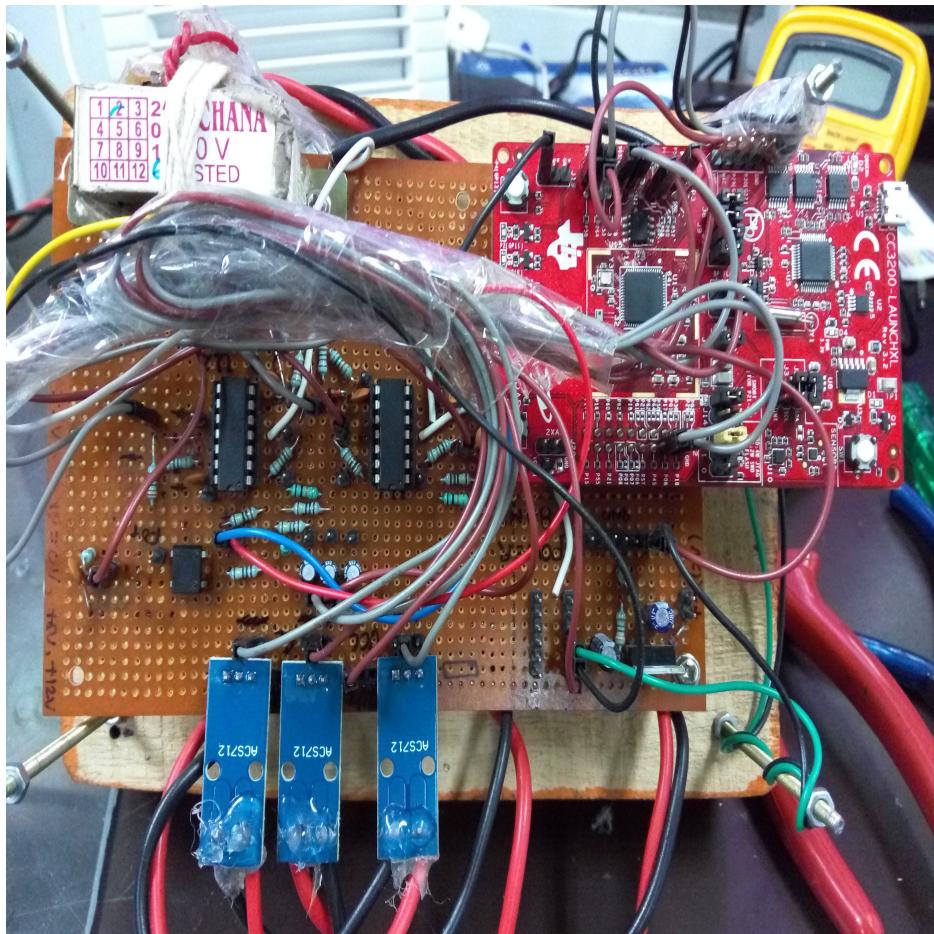
Step 6

Now connect *cc3200-launchpad* and *Transformer* with measurement board using female jumper as described below:

- connect (pin1,pin4),(pin2,pin5) and (pin3,pin7) on the measurement board.
- connect pin7 and pin8 with the transformer output jumper.
- connect pin9 of measurement board to *PIN59* of cc3200 launchpad.
- connect pin17 of measurement board to *PIN05* of cc3200 launchpad.
- connect pin10 of measurement board to *PIN58* of cc3200 launchpad.
- connect pin12 of measurement board to *PIN18* of cc3200 launchpad.
- connect pin14 of measurement board to *PIN60* of cc3200 launchpad.
- connect pin15 of measurement board to *PIN08* of cc3200 launchpad.
- connect 5V and 3.3V pins of measurement board to 5V and *Vcc* pins of cc3200 launchpad.
- connect "Vcc" and "GND" of relay board with "5V" and "GND" of measurement board.
- connect the input pins of relay with *PIN64*, *PIN01* and *PIN02* of CC3200 launchpad.



1.5. ASSEMBLY OF HARDWARE



for more detail about pin connection see board manual [A](#)

Step 7

Now connect some more accessories to improve mechanical design and your **”Smart switch board”** is ready to use.



1.6. SOFTWARE AND CODE

1.6 Software and Code

[Github link](#) for the repository of code.

The wifi enabled device is designed in such a way, that it will POST a particular string in every 4 secs and reads the response from the server as shown in the figure below.

The screenshot shows the CoolTerm software interface. The menu bar includes File, Edit, Connection, View, Window, and Help. The toolbar contains New, Open, Save, Connect, Disconnect, Clear Data, Options, HEX, and Help buttons. The main window displays a terminal session. The session starts with system information: Date: Sat, 23 Jul 2016 07:05:42 GMT, Server: Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.21, X-Powered-By: PHP/5.6.21, Content-Length: 3, Connection: close, Content-Type: text/html; charset=UTF-8. It then shows two lines of text input: '111' and 'hii'. A message follows: Starting connection to server... connected to server. Below this, a red box highlights a GET request: GET /GHPowermon/action.php/?cur=0.42&cur2=0.35&cur3=0.00&pha=355.8&pha2=227.2&pha3=000.0&vol=240.54&fre=49.9176&ds1=1&ds2=1&ds3=0&c=0 HTTP/1.1. The response to this request is shown below the red box: HTTP/1.1 200 OK, Date: Sat, 23 Jul 2016 07:05:46 GMT, Server: Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.21, X-Powered-By: PHP/5.6.21, Content-Length: 3, Connection: close, Content-Type: text/html; charset=UTF-8. At the bottom left, a yellow box highlights the text '111 RESPONSE TEXT'. The bottom status bar shows COM8 / 9600 8-N-1 and Connected 00:12:29. On the right, there are status indicators for TX, RX, RTS, CTS, DTR, DSR, and RI.

Figure 1.6: Sending Request after successfull connection.

- *action.php* is the page accessed by the micro-controller, which in turn updates the database according to the data acquired.
- CC3200 uses queries to POST data and updates current values for devices 1,2,3 using *cur*, *cur2*, *cur3*, updates voltage using *vol*, frequency

1.6. SOFTWARE AND CODE

using *fre*, phase of three devices using *pha*, *pha2*, *pha3*.

- logging the data
 - *action.php* is also capable of logging in database, it mainly updates two tables in database.
 - * Table name - *reading*, It holds data under the columns *id*(Auto Increment), *Current1*, *Current2*, *Current3*, *Voltage*, *Frequency*, *Phase1*, *Phase2*, *Phase3*, *reg_date* (Auto time stamp)

	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	id	Current1	Current2	Current3	Voltage	frequency	Phase1	Phase2	Phase3	reg_date
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12476	0	0	0	232.68	49.99	0	0	0	2016-07-19 15:21:22
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12477	0	0	0	232.68	50.0751	0	0	0	2016-07-19 15:21:26
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12478	0	0	0	232.74	49.7363	0	0	0	2016-07-19 15:21:30
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12479	0	0	0	232.73	50.0075	0	0	0	2016-07-19 15:21:34
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12480	0	0	0	232.58	50	0	0	0	2016-07-19 15:21:38
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12481	0	0	0	232.6	49.97	0	0	0	2016-07-19 15:21:42
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12482	0	0	0	232.56	49.9925	0	0	0	2016-07-19 15:21:46
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12483	0	0	0	232.6	50	0	0	0	2016-07-19 15:21:50
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12484	0	0	0	232.8	50.0625	0	0	0	2016-07-19 15:21:54
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12485	0	0	0	232.63	50.02	0	0	0	2016-07-19 15:21:58
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12486	0	0	0	232.69	49.9375	0	0	0	2016-07-19 15:22.02
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12487	0	0	0	232.6	50.0225	0	0	0	2016-07-19 15:22.06
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12488	0	0	0	232.87	50.0575	0	0	0	2016-07-19 15:22:10
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12489	0	0	0	232.76	49.97	0	0	0	2016-07-19 15:22:14
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12490	0	0	0	0	0	0	0	0	2016-07-19 15:27:40
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12491	0	0	0	0	0	0	0	0	2016-07-19 15:27:43

Figure 1.7: Database table - "reading"

- * Table name - *para*, It only keeps the latest value under columns *RT*, in front of different row ids assigned to it.



1.6. SOFTWARE AND CODE

		id	parameters	reg_date	▲ 1	RT
<input type="checkbox"/>		1	Current	2016-07-19 15:27:43		0
<input type="checkbox"/>		2	Current2	2016-07-19 15:27:43		0
<input type="checkbox"/>		3	Current3	2016-07-19 15:27:43		0
<input type="checkbox"/>		4	Voltage	2016-07-19 15:27:43		0
<input type="checkbox"/>		5	Frequency	2016-07-19 15:27:43		0
<input type="checkbox"/>		6	Phase	2016-07-19 15:27:43		0
<input type="checkbox"/>		7	Phase2	2016-07-19 15:27:43		0
<input type="checkbox"/>		8	Phase3	2016-07-19 15:27:43		0

Check all With selected:

Show all | Restore column order | Number of rows: 25 ▾ Filter rows:

Figure 1.8: Database table - "para"



1.6. SOFTWARE AND CODE

- Real Time Plotting Graphs
 - *current.html* makes AJAX requests to *refresh.php* which connects to database and updates it with latest data.
 - With the latest data made available continuously, Real Time updating graphs can be plotted.
 - As the *chart.js* will need data in JSON format, it requests to *action_page.php* to fulfill this requirement.

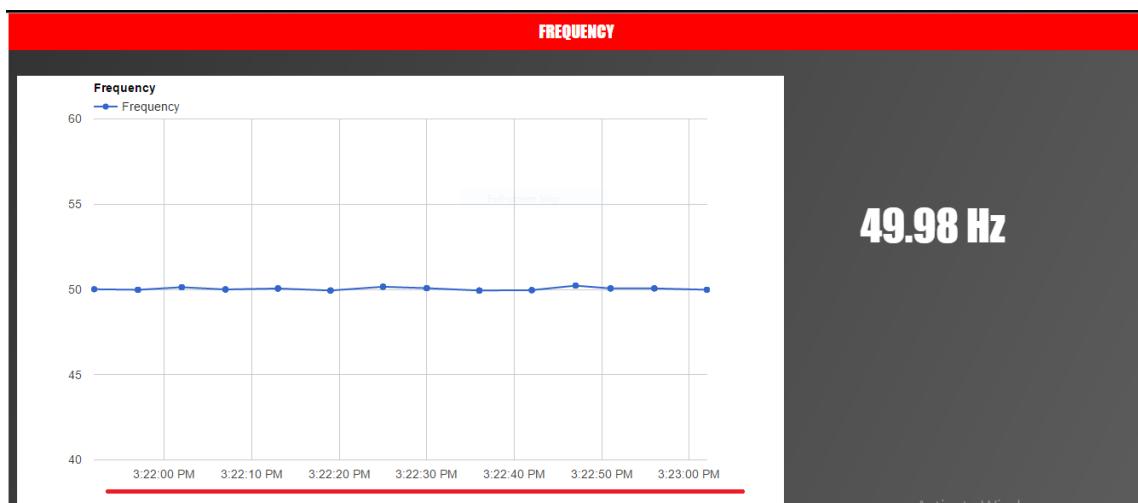


Figure 1.9: Real Time Frequency plot 1



1.6. SOFTWARE AND CODE



Figure 1.10: Real Time Frequency plot 2

- Plotting the logged data
 - The logged data is gathered and converted into JSON by *action_graph_week.php*



Figure 1.13: Logged Voltage Data



1.6. SOFTWARE AND CODE

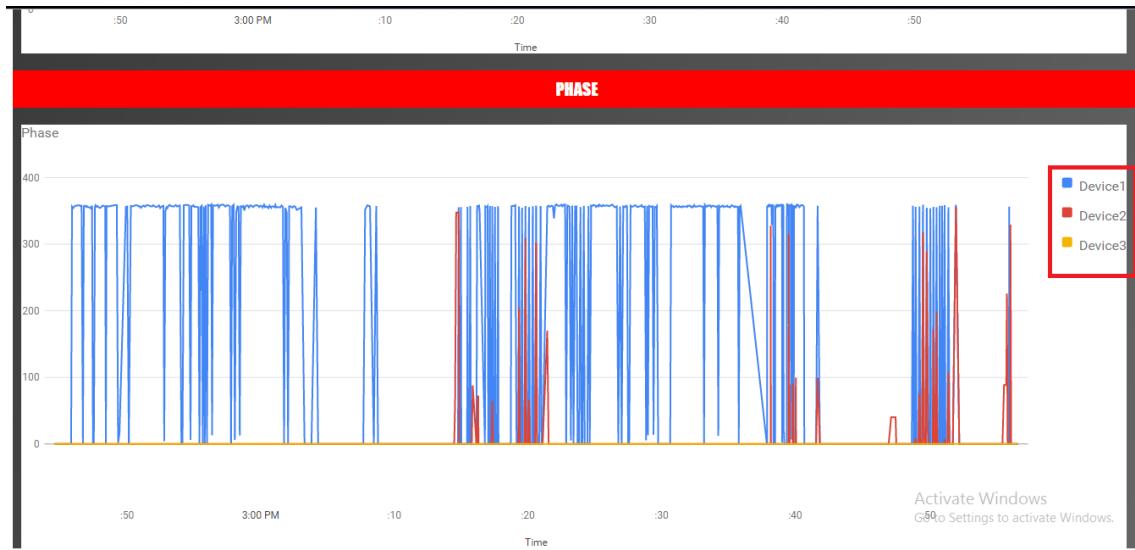


Figure 1.11: Logged Phase data



Figure 1.12: Logged Frequency Data



1.6. SOFTWARE AND CODE



Figure 1.14: Logged Current Data

- The graph is plotted by *stable_chart_loop.js* which creates a graph according to datatable generated after user selected a particular period.

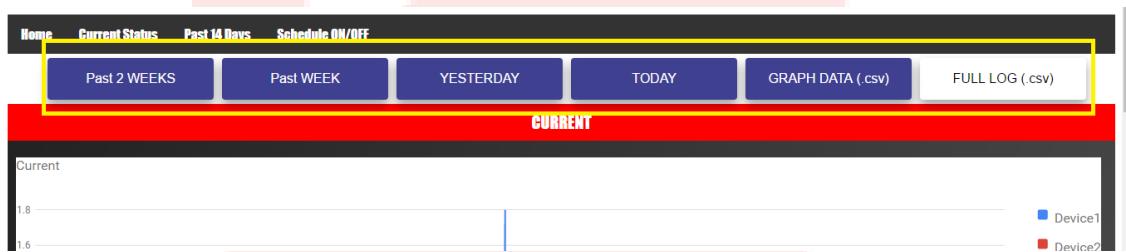


Figure 1.15: Selection Buttons

- As shown in above image, user can select the period ,which will then accordingly request and generate JSON data required for *stable_chart_loop.js*
- The graph can be selectively highlighted and observed with timestamp and value on tool-tip.

1.6. SOFTWARE AND CODE



Figure 1.16: Selective view of graphs for multiple devices.

- Feedback System.
 - In request cc3200 also provides feedback using $ds1, ds2, ds3$
 - In response server will update it with a boolean sequence for eg. 100, 110, 000... indicating switch status for each one of the 3 devices
 - Then the server waits for feedback in next request, which are provided by cc3200 in variables $ds1, ds2, ds3$

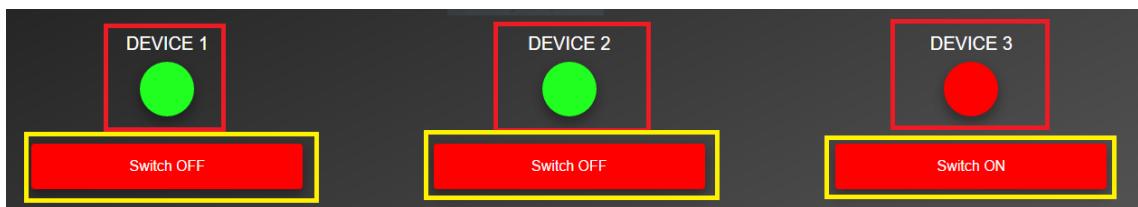


Figure 1.17: Feedback system

- The above image shows circular blocks which shows current device status according to the board, obtained in request.



1.6. SOFTWARE AND CODE

If a device is showing OFF when it should be ON or showing ON when it should be off, then fallacy can be detected. It indicates that maybe there is no device or it is not functioning properly

- Scheduling and Control
 - The scheduling page allows user to schedule ON and OFF (as Radio buttons) and selection 'FROM' time and 'TO' time using input text field and datetimepicker.

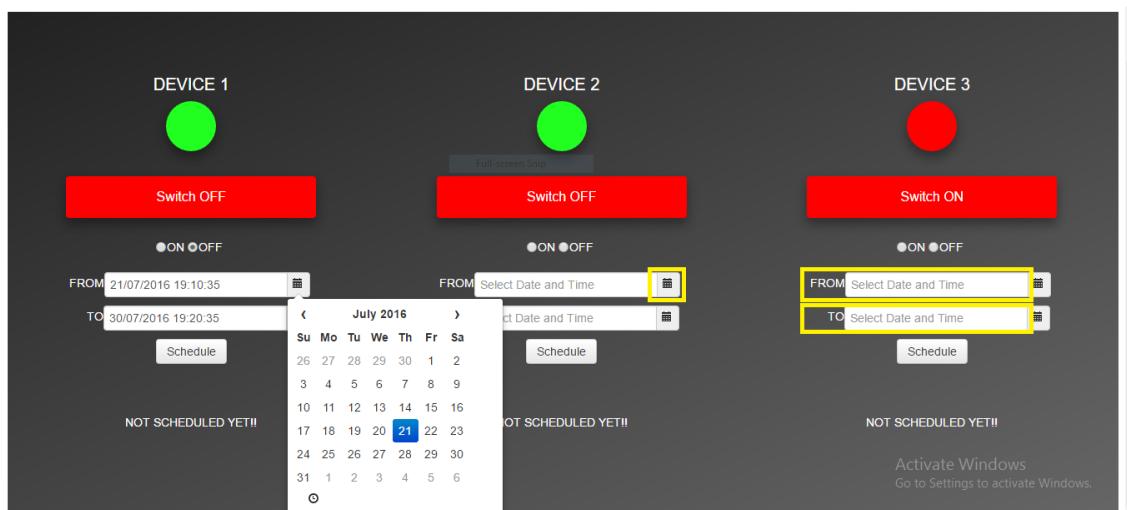


Figure 1.18: ON/OFF Scheduler

- The Scheduling datetime is submitted to *selection.php* which update values in database, which are then further compared with every timestamp and a boolean string response for connected devices



1.6. SOFTWARE AND CODE

	Edit	Copy	Delete	id	Switch_off	Switch_on	reg_date	Button_change
<input type="checkbox"/>	Edit	Copy	Delete	1	2016-07-19 14:30:46	2016-07-19 14:29:46	2016-07-19 14:50:09	1
<input type="checkbox"/>	Edit	Copy	Delete	2	2016-07-11 22:19:15	2016-07-30 22:19:15	2016-07-18 22:31:52	1
<input type="checkbox"/>	Edit	Copy	Delete	3	2016-07-11 22:19:26	2016-07-11 22:19:26	2016-07-18 22:37:24	1

[Check all With selected:](#) [Edit](#) [Copy](#) [Delete](#) [Export](#)

Show all | Number of rows: 25 | Filter rows:

Query results operations

Figure 1.19: Database Table *scheduled*

- Initial status of *button_change* is 1, which means scheduling is inactive.

	Edit	Copy	Delete	id	Switch_off	Switch_on	reg_date	Button_change
<input type="checkbox"/>	Edit	Copy	Delete	1	2016-07-21 19:10:35	2016-07-30 19:20:35	2016-07-21 19:11:00	1
<input type="checkbox"/>	Edit	Copy	Delete	2	2016-07-11 22:19:15	2016-07-30 22:19:15	2016-07-18 22:31:52	1
<input type="checkbox"/>	Edit	Copy	Delete	3	2016-07-11 22:19:26	2016-07-11 22:19:26	2016-07-18 22:37:24	1

[Check all With selected:](#) [Edit](#) [Copy](#) [Delete](#) [Export](#)

[Show all](#) | Number of rows: 25 | Filter rows:

Query results operations

Figure 1.20: Database Table updated values in table

- Initial status of *button_change* is 0, with updated values of *Switch_on* time and *Switch_off* time(Scheduled values on left frame for reference).
- As *Switch_off* Time is smaller than *Switch_on* time, the Scheduler will automatically Switch OFF the device and wait for next Switch ON command.
- Buttons are available for all three devices, which have the capability to switch devices off/on.



1.6. SOFTWARE AND CODE

	id	button	button_2	button_3	d_status1	d_status2	d_status3	Choice
	1	1	1	1	0	0	0	4

Figure 1.21: Database table "Power"

- The button data is submitted to *button_status.php*, *button_status2.php*, *button_status.php3* with a value for variable *q*. which upadates the above shown table.





1.7. USE AND DEMO

1.7 Use and Demo

- Configure the Smart Switch-board using terminal based GUI

Connection: direct
Content-Type: text/html; charset=UTF-8

```
111
*****
WELCOME TO EYANTRA'S SMART PLUG PORTAL
*****
Choose following options for configuration:-
1) Wifi-Configuration
2) Email-Configuration
3) Mode Configuration
4) Security Options
5) Signout from portal
Choose option>
```

COM8 / 9600 8-N-1
Connected 00:06:17

TX	RTS	DTR	DCD
RX	CTS	DSR	RI

Figure 1.22: Interactive Interface - Initial Options

```
111
*****
WELCOME TO EYANTRA'S SMART PLUG PORTAL
*****
Choose following options for configuration:-
1) Wifi-Configuration
2) Email-Configuration
3) Mode Configuration
4) Security Options
5) Signout from portal
Choose option>
Choose following options for Wifi-configuration:-
1) Change SSID
2) Change Password
3) Change Network Mode
4) Go to main menu
Choose option>
```

COM8 / 9600 8-N-1
Connected 00:06:51

TX	RTS	DTR	DCD
RX	CTS	DSR	RI

Figure 1.23: Select an option by entering an Integer value.



1.7. USE AND DEMO

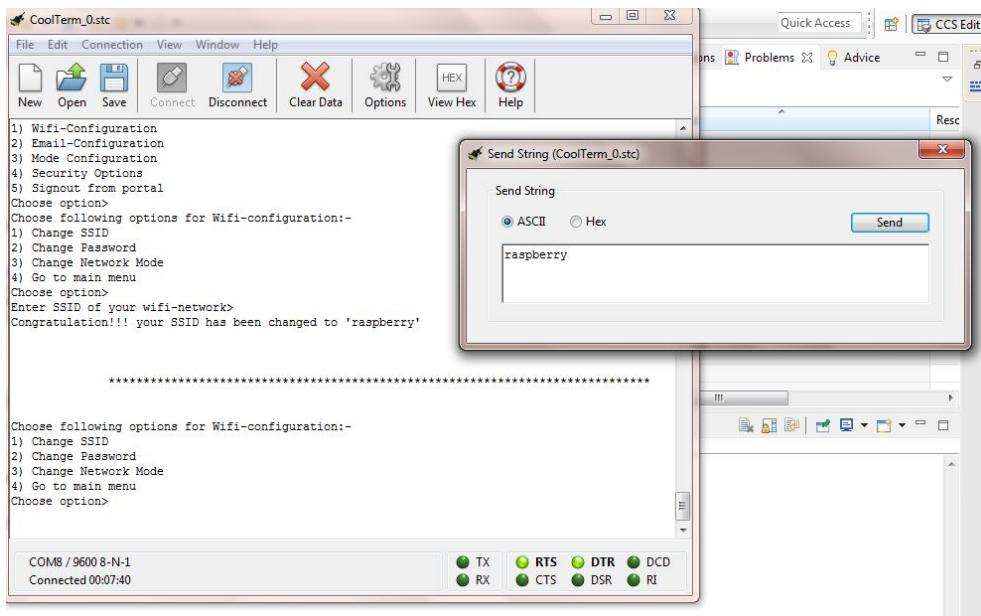


Figure 1.24: Enter Wi-Fi SSID and Password

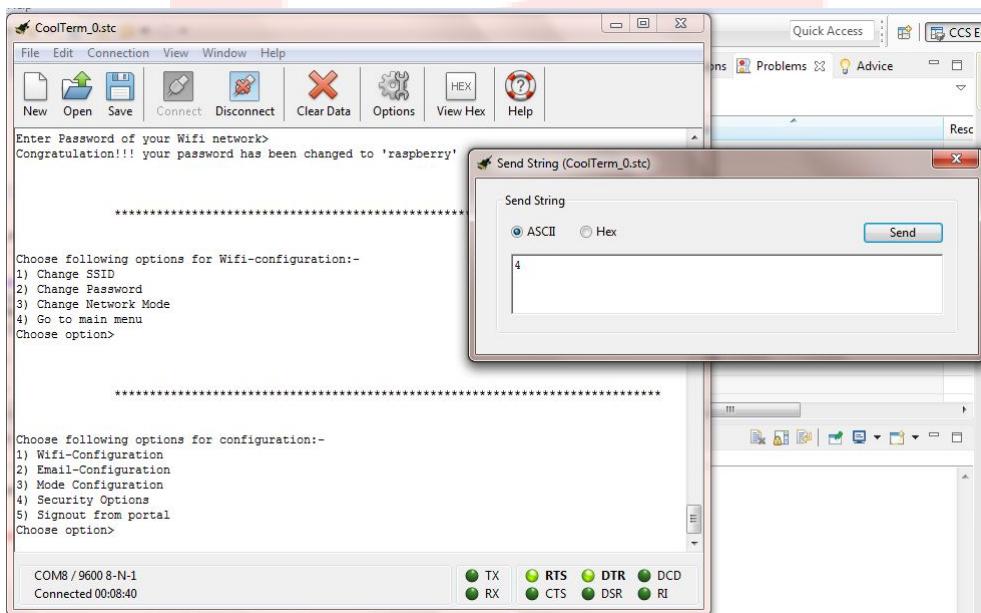


Figure 1.25: Return to Main Menu



1.7. USE AND DEMO

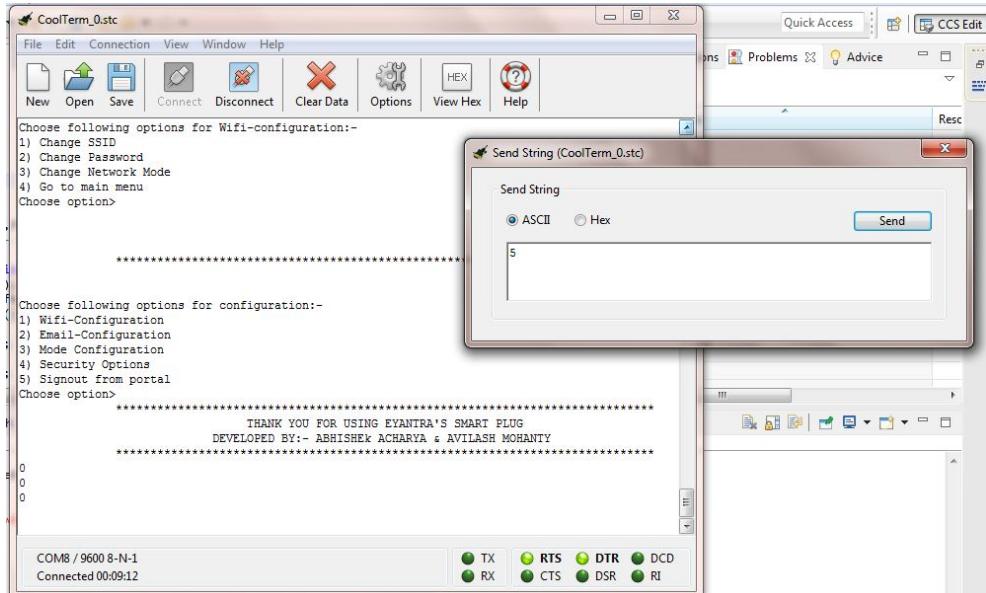


Figure 1.26: Sign Out

- Open the webpage. (NOTE: The Web Site was developed on a local-host)

The screenshot shows a web browser window titled 'postermymwall - Google Search' with the URL 'localhost/GHpowermon/'. The page has a green header bar with the text 'GREEN HOUSE POWER MONITORING'. Below the header is a navigation menu with links: 'Home', 'Current Status', 'Past 30 Days', and 'Schedule ON/OFF'. The main content area has a dark background. On the left, there is a section titled 'ABOUT' containing text about the project's goal and components. On the right, there are three large orange buttons with white text: 'START MONITORING', 'GET LOG DATA', and 'SCHEDULE ON/OFF'.

Figure 1.27: Home Page

- This is the home page of the web based GUI, with 3 buttons redirecting

1.7. USE AND DEMO

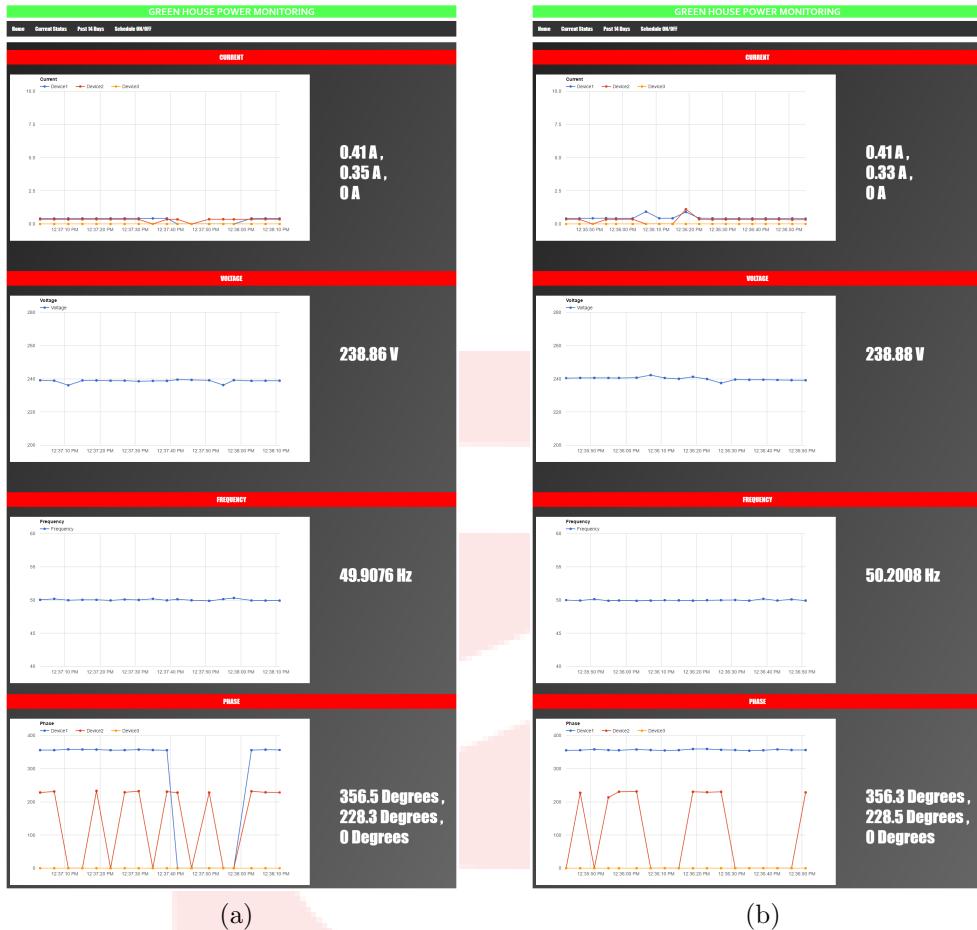


Figure 1.28: Real time updating graph with different stamp

buttons.

- Under the title of page, a navigation bar is present.
- Next is the current status page. See [Figure 1.28](#).
The first Division shows the current(in A) flowing in the 3 devices.
BLUE for Device 1
RED for device 2
YELLOW for device 3
- Divisions for voltage(in V)and frequency(in Hz) follow Current's division.
Only one line chart is being made for these two parameters as they are same for all three devices.



1.7. USE AND DEMO

- Last division represents phase difference(in deg) of the three connected devices. See [Figure 1.28](#) for reference .
BLUE for device 1
RED for device 2
YELLOW for device 3

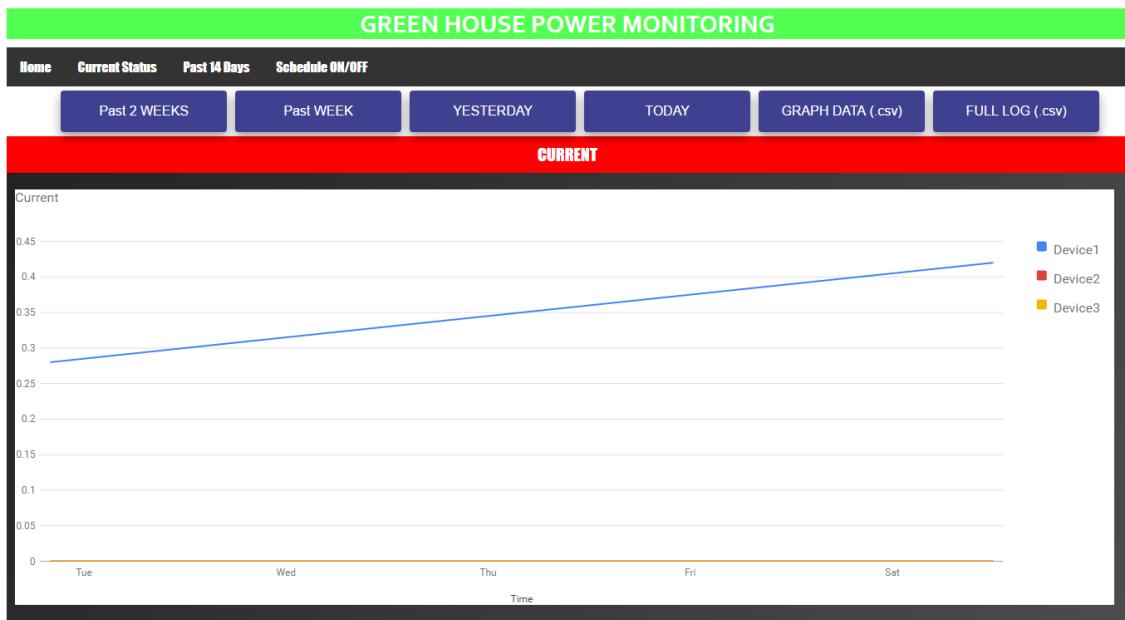


Figure 1.29: Logging page

- [Figure 1.29](#) shows the logging page.
- As shown in the [Figure 1.15](#) the first four buttons provided are for period selection, and the last two are download links for .csv format for graph data and complete log data respectively.
- By clicking any of the four buttons, graph will be plotted for the given period, and *Graph Data* download link will generate a log for that specific period in .csv format.



1.7. USE AND DEMO

Font	Paragraph	Insert	Editing
3 · · · 2 · · · 1 · · · 1 · · · 2 · · · 3 · · · 4 · · · 5 · · · 6 · · · 7 · · · 8 · · · 9 · · · 10 · · · 11 · · · 12 · · · 13 · · · 14 · · · 15 · · · 16 · · · 17 · · · 18 ·	· · · 1 · · · 2 · · · 3 · · · 4 · · · 5 · · · 6 · · · 7 · · · 8 · · · 9 · · · 10 · · · 11 · · · 12 · · · 13 · · · 14 · · · 15 · · · 16 · · · 17 · · · 18 ·	· · · 1 · · · 2 · · · 3 · · · 4 · · · 5 · · · 6 · · · 7 · · · 8 · · · 9 · · · 10 · · · 11 · · · 12 · · · 13 · · · 14 · · · 15 · · · 16 · · · 17 · · · 18 ·	· · · 1 · · · 2 · · · 3 · · · 4 · · · 5 · · · 6 · · · 7 · · · 8 · · · 9 · · · 10 · · · 11 · · · 12 · · · 13 · · · 14 · · · 15 · · · 16 · · · 17 · · · 18 ·
0,0,0,229.64,49.9425,0,0,0,"2016-07-21 15:50:55" 0,0,0,229.64,49.9425,0,0,0,"2016-07-21 15:50:57" 0,0,0,229.64,49.9425,357.8,0,0,"2016-07-21 15:51:00" 0,0,0,228.86,50.03,0,0,0,"2016-07-21 15:51:03" 0,0,37,0,228.86,50.03,0,0,0,"2016-07-21 15:51:05" 0,0,37,0,228.86,50.03,0,0,0,"2016-07-21 15:51:08" 0,0,37,0,228.86,50.03,358.5,6.731,0,"2016-07-21 15:51:11" 0,0,38,0,228.43,50.0425,0,0,0,"2016-07-21 15:51:15" 0,0,38,0,228.43,50.0425,0,0,0,"2016-07-21 15:51:21" 0,0,38,0,228.43,50.0425,0,0,0,"2016-07-21 15:51:24" 0,0,38,0,228.43,50.0425,355.6,108.3,0,"2016-07-21 15:51:28" 0,0,0,228.28,49.9675,0,0,0,"2016-07-21 15:51:33" 0,0,0,228.28,49.9675,0,0,0,"2016-07-21 15:51:38" 0,0,0,228.28,49.9675,0,0,0,"2016-07-21 15:51:48" 0,0,0,228.28,49.9675,359.1,355.9,0,"2016-07-21 15:52:02" 0,0,0,228.87,50.0175,0,0,0,"2016-07-21 15:52:17" 0,0,0,228.87,50.0175,0,0,0,"2016-07-21 15:52:20" 0,0,0,228.87,50.0175,0,0,0,"2016-07-21 15:52:25" 0,0,0,228.87,50.0175,0,0,0,"2016-07-21 15:52:29" 0,0,0,0,0,0,0,0,"2016-07-21 15:55:31" 0,0,0,0,0,88.41,0,"2016-07-21 15:55:38" 0,0,0,0,0,88.41,0,"2016-07-21 15:55:42" 0,0,0,0,0,88.41,0,"2016-07-21 15:55:45" 0,0,0,0,0,88.41,0,"2016-07-21 15:55:48" 0,0,0,0,0,226,0,"2016-07-21 15:55:51" 0,0,38,0,228.65,49.9725,0,0,0,"2016-07-21 15:55:54" 0,0,38,0,228.65,49.9725,0,0,0,"2016-07-21 15:55:58" 0,0,38,0,228.65,49.9725,356.4,0,0,"2016-07-21 15:56:01" 0,0,0,228.99,49.8603,0,0,0,"2016-07-21 15:56:03" 0,0,0,228.99,49.8603,0,0,0,"2016-07-21 15:56:04"			

Figure 1.30: Graph Data Log

- *Today* button will generate graph with Only That day's data.
Yesterday button will generate graph for previous data only.
Past week will generate filtered hourly data in the past week.
Past 2 Weeks will generate filtered hourly data in the past 2 weeks.
These are defined in *action_graph_week.php*.
- Full log in .csv format is always available for the user to download .

1.7. USE AND DEMO

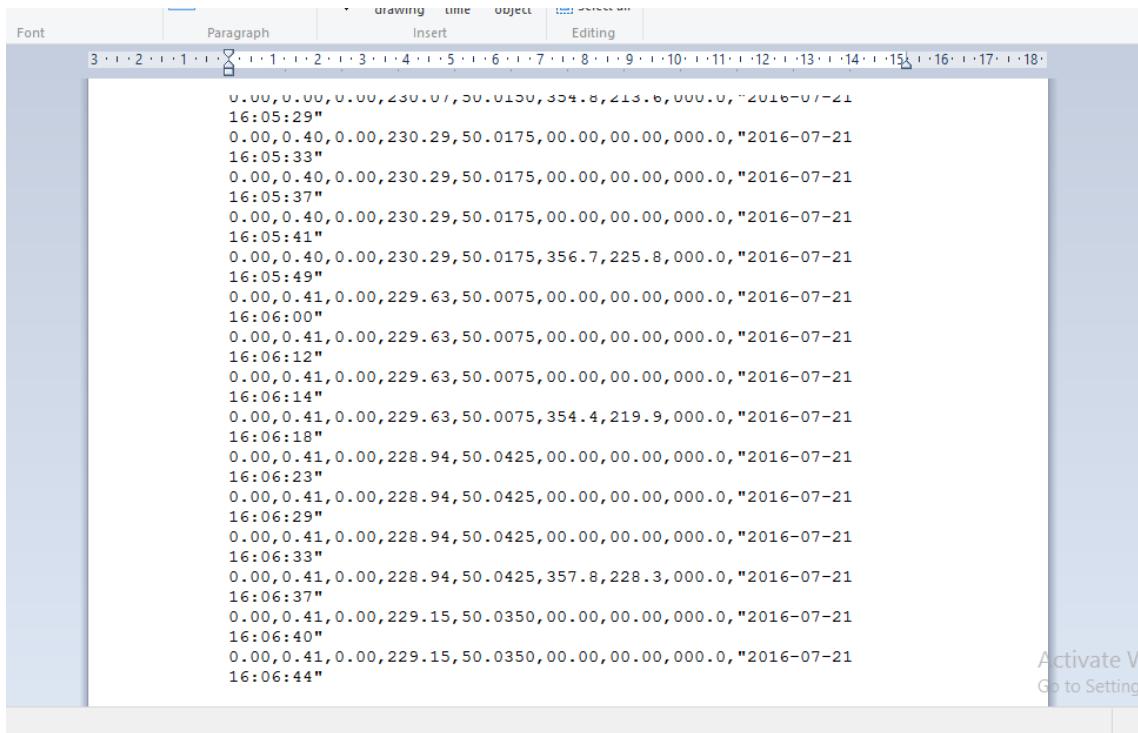


Figure 1.31: Full Log

- The last page is *Schedule ON/OFF*. See [Figure 1.18](#). This is the controlling page with features like scheduling and switching ON/OFF. The button highlighted in the image is the switch. It can be noted that the text on it changes according to current button status.
Above switch, there is a circular block which indicates the actual device status.
GREEN represents ON status.
RED represents OFF status. As depicted in figure 1.17.
- Just below every switch button there is a scheduler which can schedule ON/OFF.



1.7. USE AND DEMO

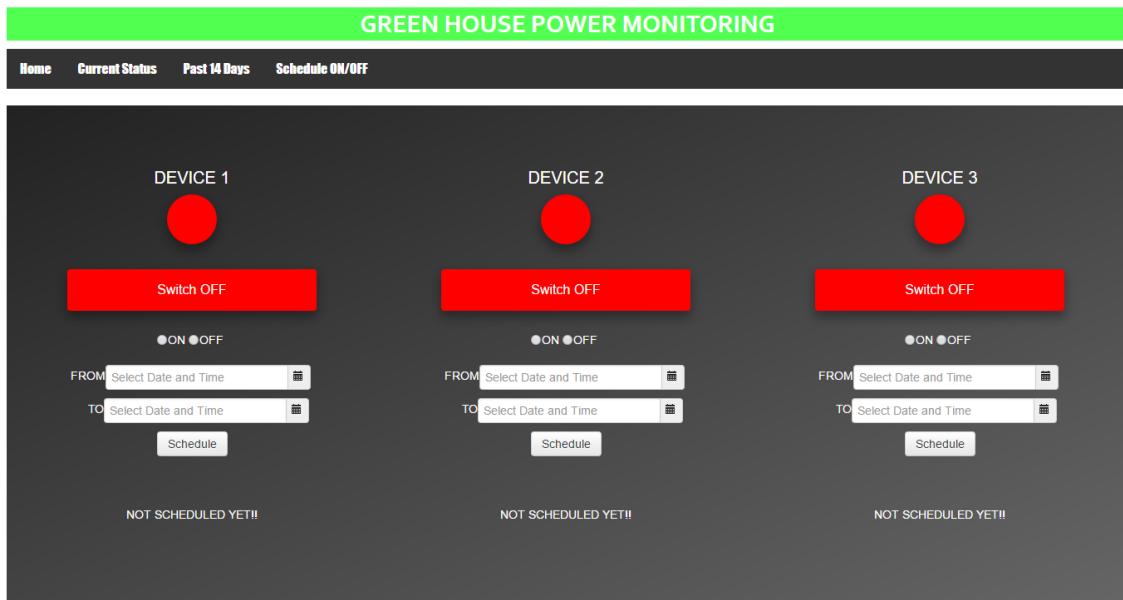


Figure 1.32: Control page

The user first have to make a choice between ON/OFF radio buttons. If no selection is made it will be automatically counted as OFF .

- Then the user have to fill the two input fields namely *FROM* and *TO* as per requirement.See [Figure 1.18](#)
- Scheduled Time-interval and Type will be notified as text.See [Figure 1.19](#)



1.8. FUTURE WORK

1.8 Future Work

- Current Web GUI is not responsive and will not work properly on mobile phones. Using Bootstrap (HTML,CSS framework) it can be done.
- Zoom Function can be added in charts for better user experience.
- Features to control frequent or repeated Scheduling can be added.
- Support for more devices can be added even when monitoring is available for few devices due to limited ADCs.
- Circuit can be miniaturized by using Redbear Wifi-mini or cc3200 micro, which can increase portability
- Phase measurement accuracy can be improved.



1.9 Bug report and Challenges

Bug report

- For accurate measurement of phase please calculate the phase difference created by coupling capacitor applied to current wave for removing offset voltage.
- Use **TI-RTOS & Free-RTOS** for fast processing and transient analysis of the electrical parameters.
- SQL Injection - A code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution. Blacklisting some characters can prevent attacks. For more details [W3schools SQL injection](#)
- Lightweight protocols Like MQTT can be used for better performance.
- Security can be improved by adding Cryptography techniques like AES techniques.

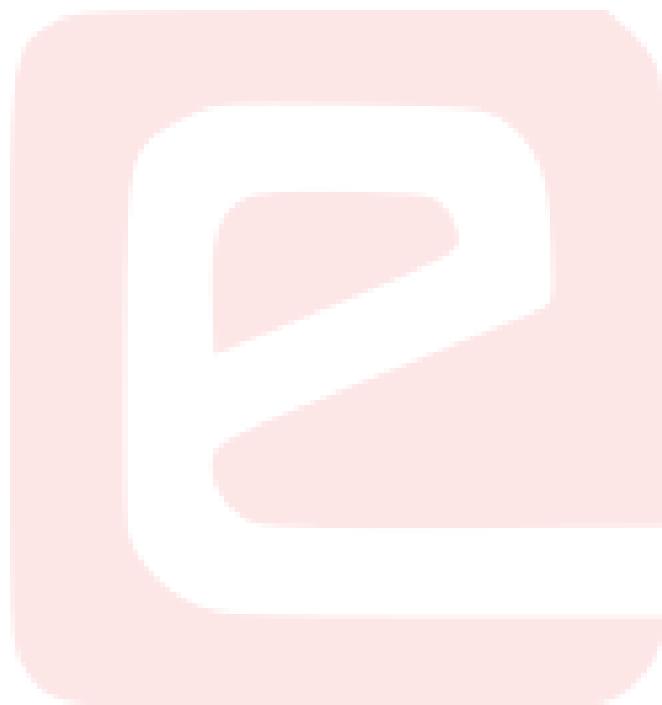
Challenges

- To add header files to the code composer studio project right click on project ,go to properties than select linked resources and than add **CC3200_sdk**. After adding sdk to linked resources than again right click on project, go to GNU compiler and include directories to it.
- The debug mode of CCS load the program into RAM of CC3200 launchpad so to load program into ROM use **Uniflash**.
- Before loading the program into flash please follow the below procedure
 - Format the whole CC3200 launchpad's ROM using **format** tab.
 - Load **service pack**, which is compatible with your launchpad's version.
 - Now load the .bin file of project.
- For loading program into ROM select **mcu.bin** tab in Uniflash and than add the .bin file of the project. After adding the .bin file program MCU using program tab.



1.9. BUG REPORT AND CHALLENGES

- To write program for CC3200-launchpad please take help from **Driver-lib API's, Simplelink API's** and all the API module files present inside CC3200sdk folder. For more details [page 49](#)



Bibliography

- [1] Stack Overflow www.stackoverflow.com
- [2] TutorialsPoint www.tutorialspoint.com/
- [3] GitHub www.github.com
- [4] W3Schools www.w3schools.com/
- [5] TI E2E Community <https://e2e.ti.com/>
- [6] Instructables www.instructables.com
- [7] Driver Library and Simplelink Library

Appendices

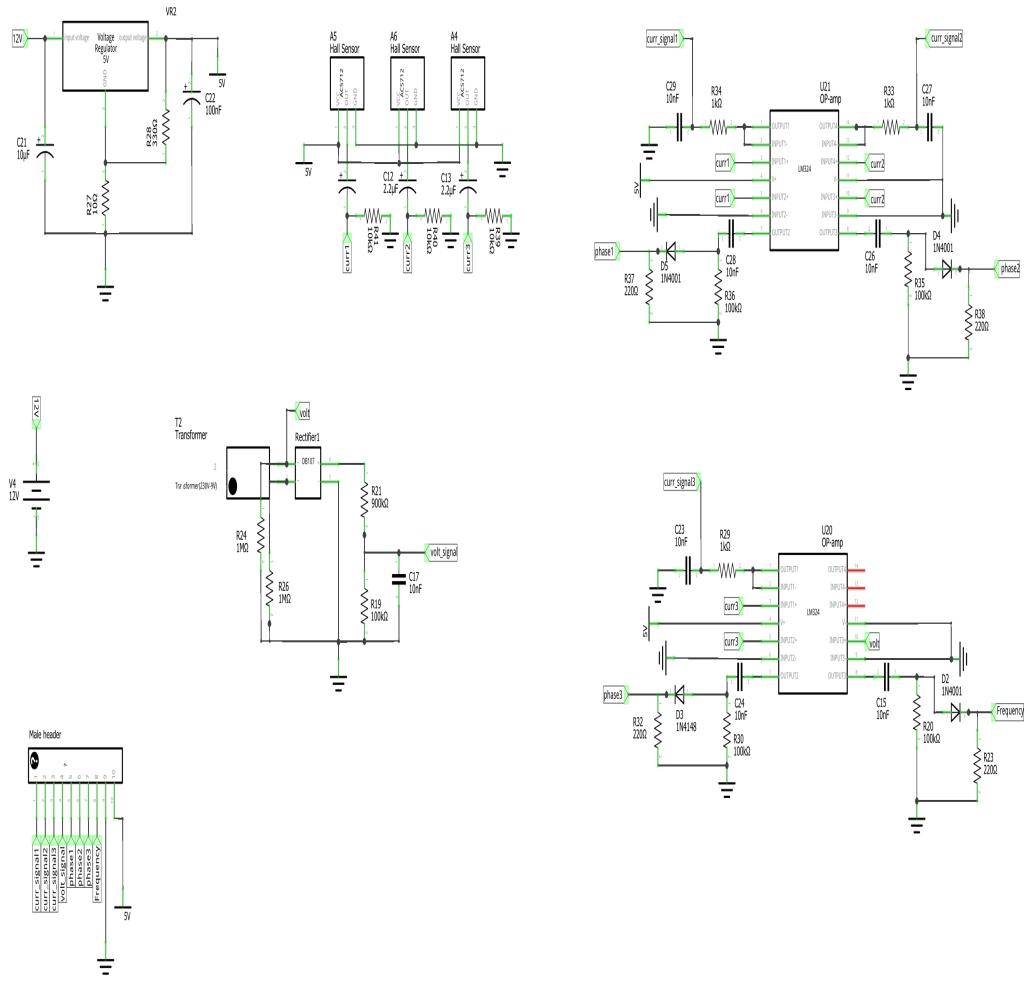
Measurement board manual

A.1 Abstract

The measurement board is an analog front end, which is used to manipulate the input signal according to the MCU unit. The signals generated by this board will be processed by MCU in the further sections of hardware. Basically this board provides a common platform where microcontroller unit can be able understand the actual electrical signal. This measurement board has a set of small circuits such as zero crossing detector, buffers, voltage dividers and filters etc.

A.2 CIRCUIT DIAGRAM

A.2 Circuit diagram

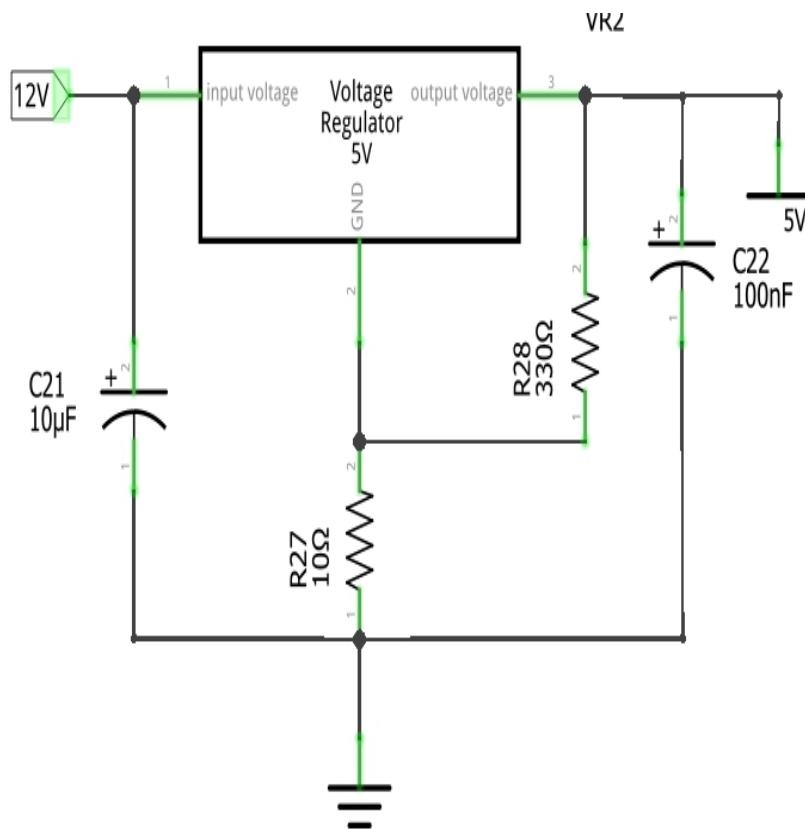


A.3. DESCRIPTION OF CIRCUIT DIAGRAM

A.3 Description of circuit diagram

1. Power block:

This circuit block is based on 7805 (5V regulator IC). It has a 10uF electrostatic capacitor at input side. A 10 ohm resistor is used at the ground terminal of voltage regulator to protect it by limiting its output current. A first order low pass circuit is applied at output side to remove ripple present in output voltage.

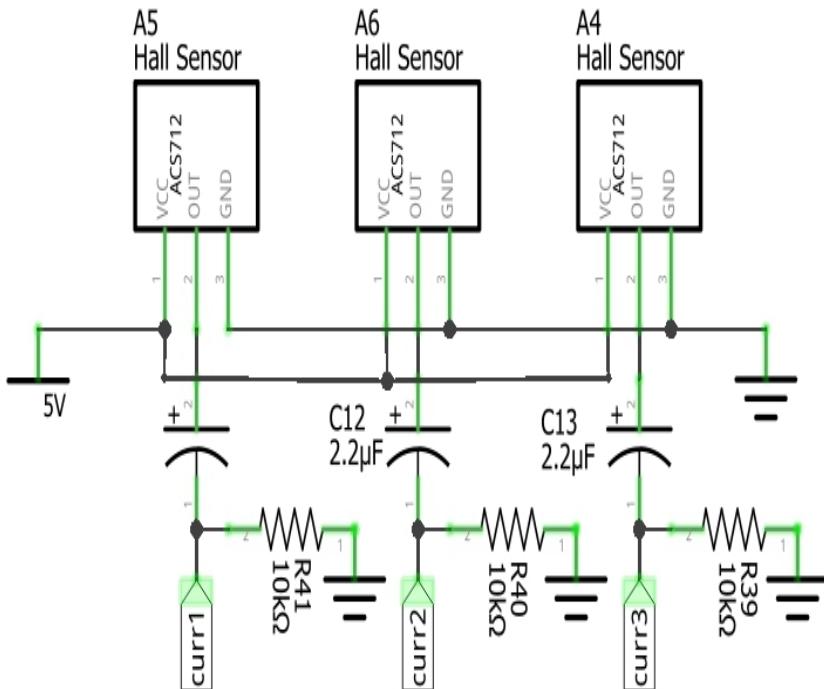


2. Hall sensor block:

This circuit block is used to convert the current wave of all the three electrical appliances into a proportional voltage wave used for the measurement. Three coupling capacitor is applied to remove DC offset

A.3. DESCRIPTION OF CIRCUIT DIAGRAM

voltage which is coming because of Hall sensor module for detailed explanation go through data sheet of ACS712 [Datasheet](#). Three 10K ohm resistor are applied to change the output impedance of the circuit.

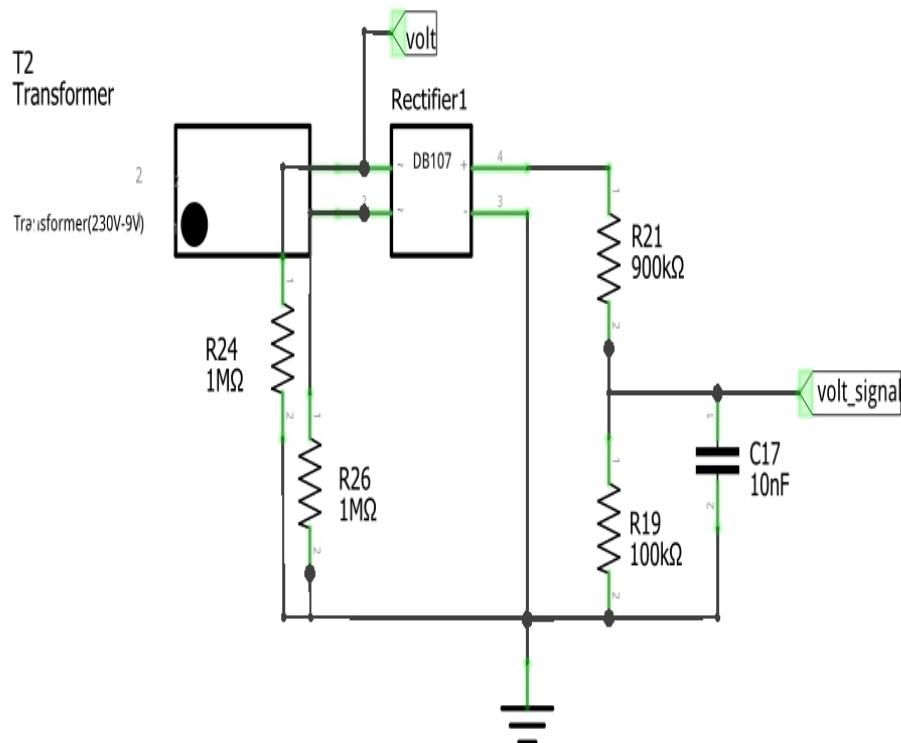


3. Transformer block:

This circuit block is used to step down and manipulate the voltage wave according to reference voltage (1.467V) of CC3200 launchpad. For this purpose a high impedance voltage divider circuit is applied to step down the voltage by 10:1. We choose to apply high impedance voltage divider to decrease loading effect and power loss. A 0.01uF ceramic cap is applied at the output side to provide pickup to the ADC channel of cc3200 MCU. For detailed explanation please go through [ADC Appnote cc32xx](#). Two 1M ohm resistor is applied to eliminate

A.3. DESCRIPTION OF CIRCUIT DIAGRAM

attenuation coming in rectified voltage wave.



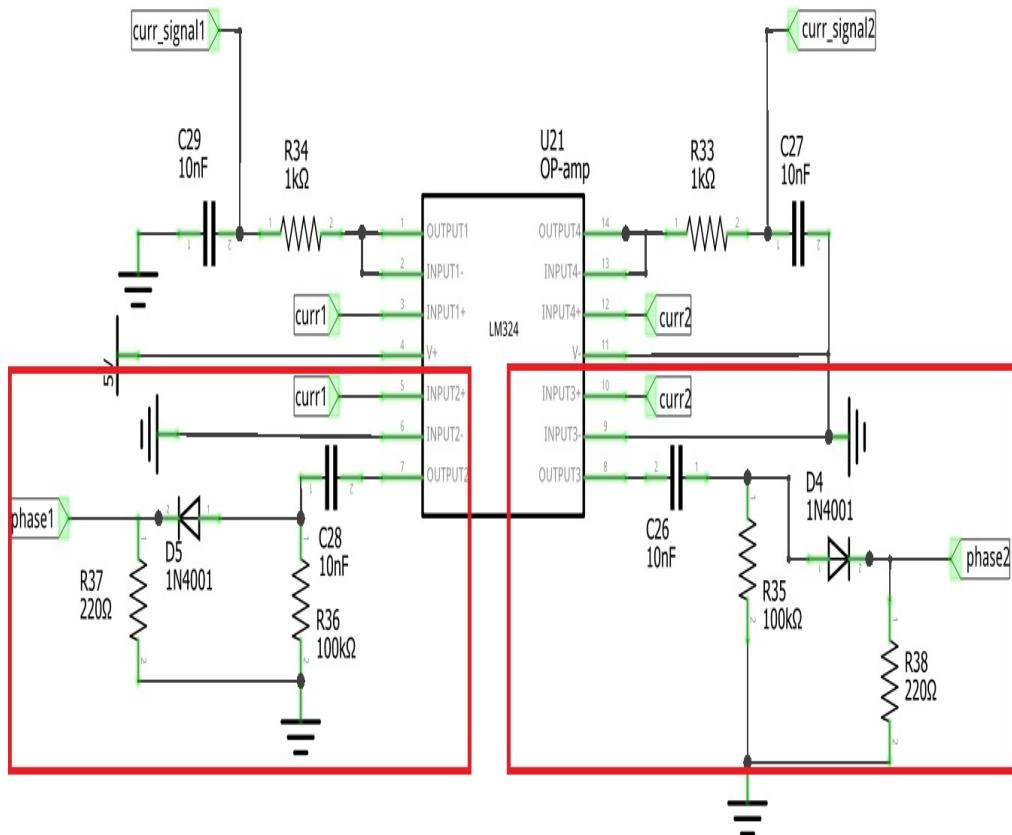
4. Zero crossing detector block:

This circuit block is used to generate a pulse whenever the AC signal crosses the zero level. This circuit is basically a comparator followed by first order RC circuit. Output is taken across the resistor. The logic behind this circuit is that whenever the input wave crosses the zero level the comparator output will either go from 5V to 0V or from 0V to 5V, so we can consider this phenomena as switching of first order RC circuit. During transient, output voltage expression across the resistor is [section A.5](#). The time constant (τ) of circuit is used to determine the width of pulse.

This circuit is used four times in the measurement board for generating

A.3. DESCRIPTION OF CIRCUIT DIAGRAM

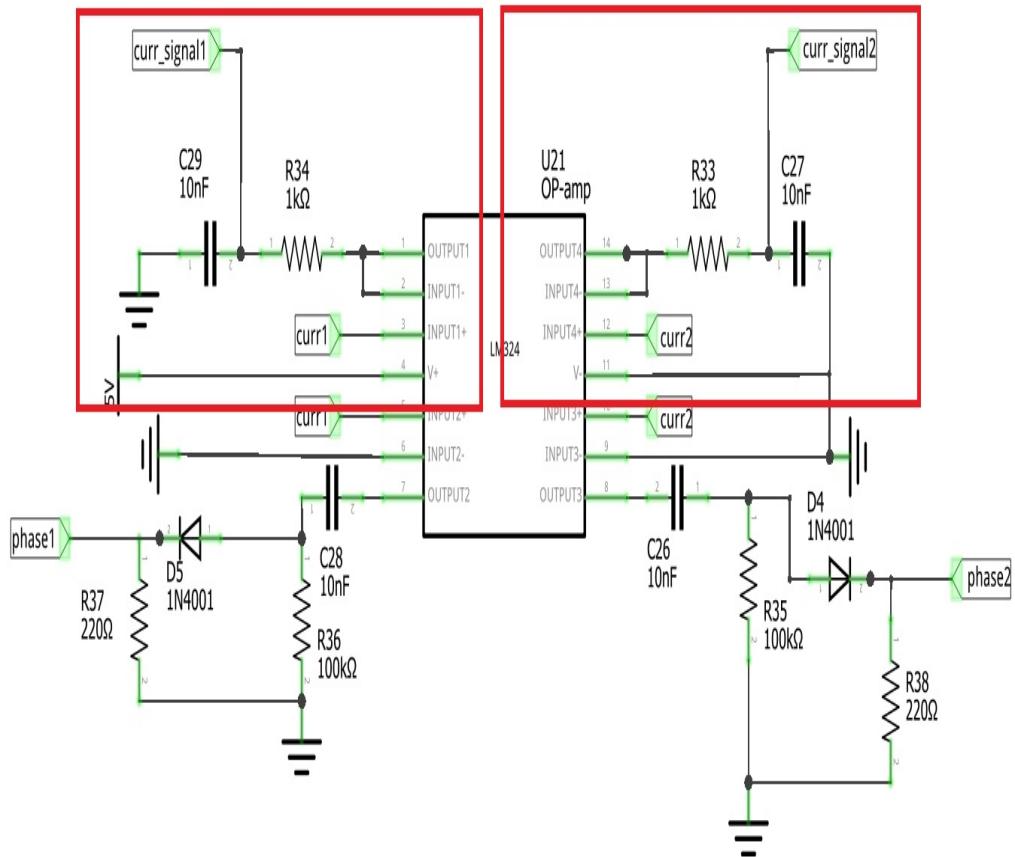
phase and frequency signals. A high speed diode is applied at output side to remove negative pulse. High speed diode is used because it has less amplitude reverse recovery current, so the circuit output doesn't have unusual pulses.



5. Buffer block:

This block is used to clip the current wave and manipulate it according to the ADC channel of CC3200 MCU. Buffer is also used to protect the ADC channel from electrostatic surge by isolating it from the electrolytic coupling capacitor. This block is used three times to generate three current signal.

A.4. DIY



A.4 DIY

Here we are including a special feature in the manual. By reading the manual now you can make this board by your self. To make it your self please follow the following instruction:



A.4. DIY

List of components

Name of component	Specification	Quantity
Resistor	10 ohm (1/4W)	1
	330 ohm (1/4 W)	1
	1K ohm (1/4W)	3
	10K ohm (1/4W)	3
	33K ohm (1/4W)	4
	100K ohm (1/4W)	5
	900K ohm (1/4W)	1
	1M ohm (1/4W)	2
Electrostatic Capacitor	0.1uF (50V)	1
	0.22uF (10V)	3
	10uF (50V)	1
Ceramic capacitor	0.01uF (103)	8
Diode	IN4148 (high speed diode)	4
Operational Amplifier	LM324 <i>Quard opamp IC</i>	2
Rectifier	DB-107	1
Berg strip	Male(40x1)	2
	Female(40x1)	1
Breadboard & Wires	-	1
Soldering iron and wire	40W	1
General purpose PCB	15cmx15cm	1

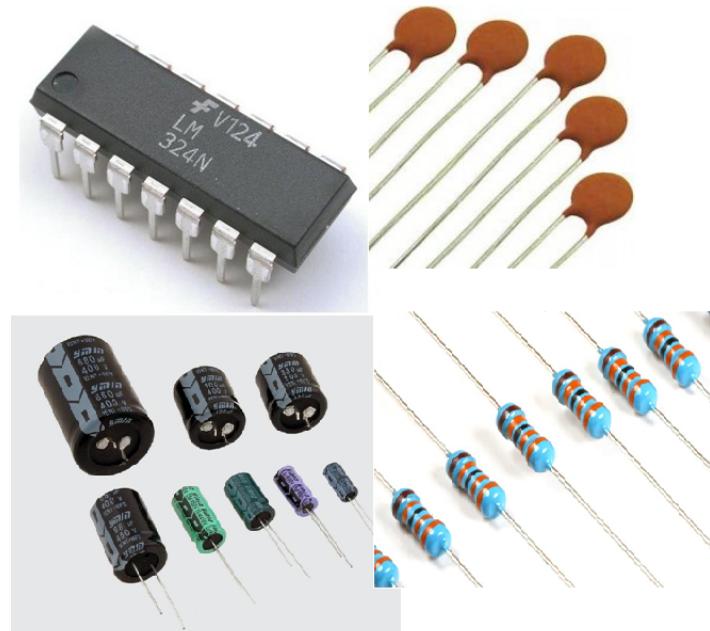
Procedure

- Before starting to build the circuit make sure you have gone through the circuit description and understand the circuit completely.



A.4. DIY

- Collect all the component, listed above if you are not able to find them than you may apply components of approximate value.



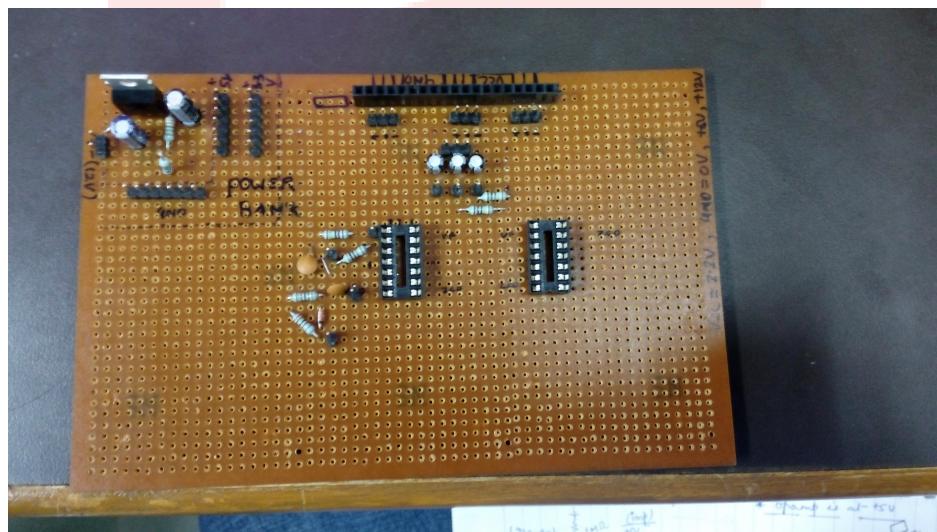
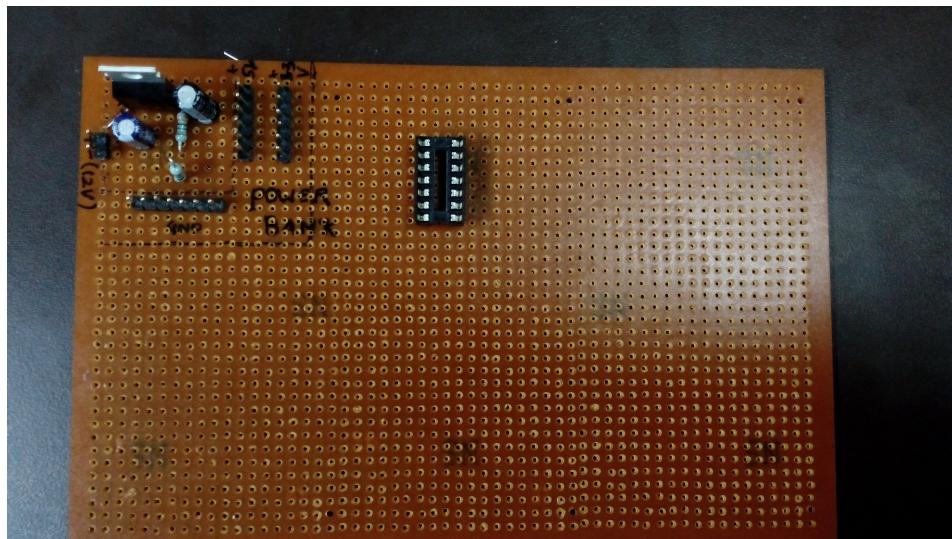
- First of all check the circuit on the breadboard whether it is working or not, After that u can solder the circuit on PCB board. We recommend you to use IC holders for mounting IC, because soldering temperature may damage the IC.





A.4. DIY

- Start making small circuit block on different the part of PCB as explained in the description of circuit diagram. for more details [section A.3.](#)

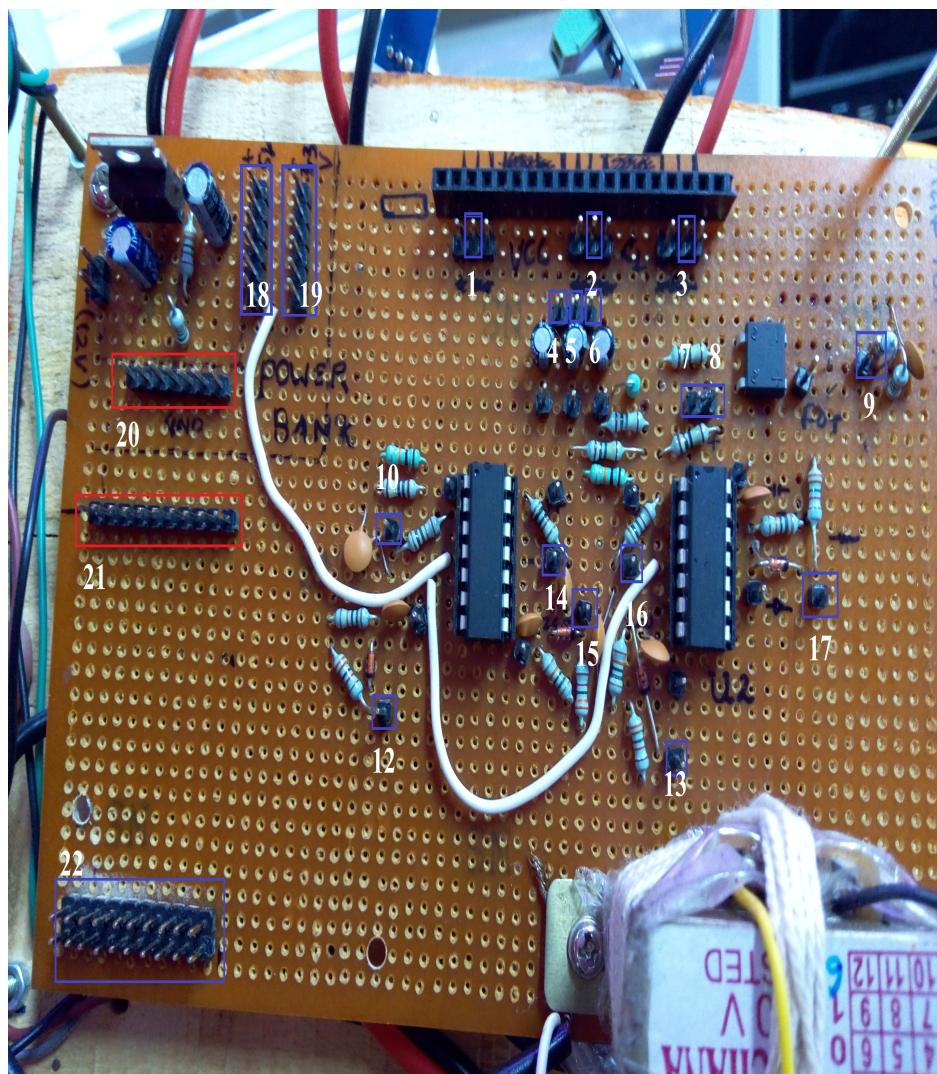


- When small section of circuit schematic got soldered on PCB than start connecting those small section according to label provided in schematic.
- After connecting all the circuit blocks please double check your circuit using multimeter connection check, whether there is any short circuit or not.

A.4. DIY

- Cheers!!! your measurement board is ready to use, now you can connect CC3200 launchpad to it and start software part.

Pin Description



- **PIN 1:** Hall sensor 1 input pin
- **PIN 2:** Hall sensor 2 input pin
- **PIN 3:** Hall sensor 3 input pin
- **PIN 4:** DC blocked current signal 1



A.4. DIY

- **PIN 5:** DC blocked current signal 2
- **PIN 6:** DC blocked current signal 3
- **PIN 7 & PIN 8:** Transformer(230V/9V) output pins
- **PIN 9:** processed voltage signal that will going into ADC pin
- **PIN 10:** processed current signal that will going into ADC pin
- **PIN 12:** Phase 1 signal
- **PIN 14:** processed current signal that will going into ADC pin
- **PIN 15:** Phase 2 signal
- **PIN 16:** processed current signal that will going into ADC pin
- **PIN 13:** Phase 3 signal
- **PIN 17:** Frequency signal
- **PIN 18:** 5V header pins
- **PIN 19:** Vcc header pins
- **PIN 20:** Gnd header pins
- **PIN 21 & PIN22:** Headers for mounting of CC3200 launchpad



A.5. CIRCUIT OUTPUT THE EACH PIN

A.5 Circuit output the each pin

At Pin number 1,2 & 3

At these pins current wave with 2.5V offset will appear. The offset is because of hall sensor module used for current sensing unit.

for further details go through datasheet of ACS712 module [Datasheet](#)

$$\text{OUT} = 2.5 + a \sin(wt)$$

Where:

a = amplitude of *sin* wave in V.

w = angular frequency of current signal in (rad/sec).

t = time in sec.

At Pin number 4,5 & 6

At these pins current wave without offset will appear.

$$\text{OUT} = a \sin(wt)$$

Where:

a = amplitude of *sin* wave in V.

w = angular frequency of current signal in (rad/sec).

t = time in sec.

At Pin number 12,13,15 & 17

At these pins output of zero crossing circuit will appear.

$$\text{OUT} = 5e^{-(t/}$$

Where:

τ = time constant of zero crossing detector circuit.

t = time in sec.

At Pin number 9,10,14 & 16

Voltage and current signal manipulated according to reference voltage of CC3200 launchpad

A.5. CIRCUIT OUTPUT THE EACH PIN

At Pin number 18,19 & 20

5V, Vcc and ground pins on the measurement board.

At Pin number 21 & 22

Headers for connecting CC3200-launchpad.

