



***K J Somaiya Institute of Management***  
***Vidyavihar, Mumbai – 400 077***  
***Masters in Computer Applications***  
***Trimester IV (2021 – 23)***

*This is to certify that Mr. Roshan Dattatray Patil Roll No. 31 of MCA has completed his IOT Journal as per the Syllabus for the Academic year 2022 – 2023. The Journal has also been evaluated by the concerned faculty of K J Somaiya Institute of Management.*

*-----  
Signature of the Faculty-In charge*

*-----  
Signature of the Course Coordinator*

*Date: \_\_\_\_\_*

*-----  
Signature of the External Examiner*

# INDEX

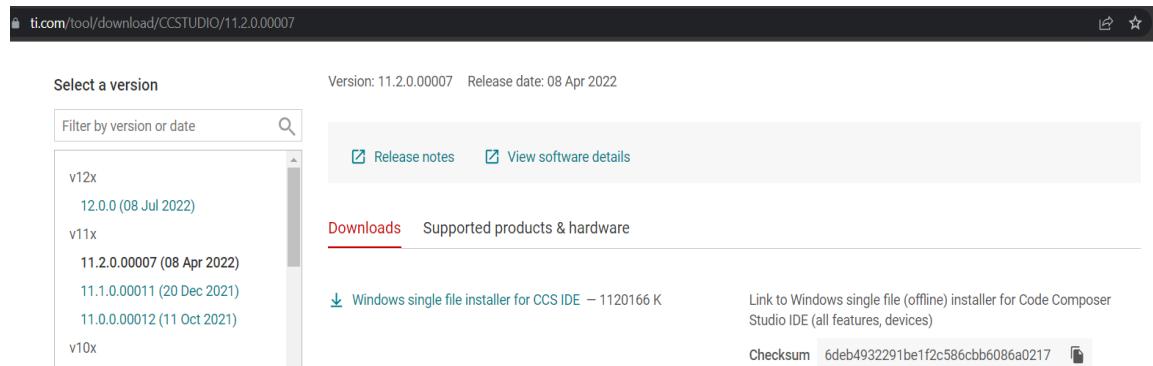
Sr. No	Practical Name	Page No.	Sign
1	CCS Installation, User Interface tour, Project creation, Compiling and debugging with MSP432.	3-13	
2	Hello world of Embedded Systems	14-26	
3	Installing Node.js ,Node-Red and Node Red interface	27-31	
4	Introducing the inject, function, debug and switch nodes	32-35	
5	Random number generator with selection of color. Introducing the HTTP node.	36-41	
6	CPU utilization flow.	42-54	
7	WiFi Setup	55-59	
8	Analog To Digital Converter.	60-65	
9	Sending Raw data to the cloud	66-76	
10	Connecting MSP432 to the cloud.	77-92	

# Practical No. 01

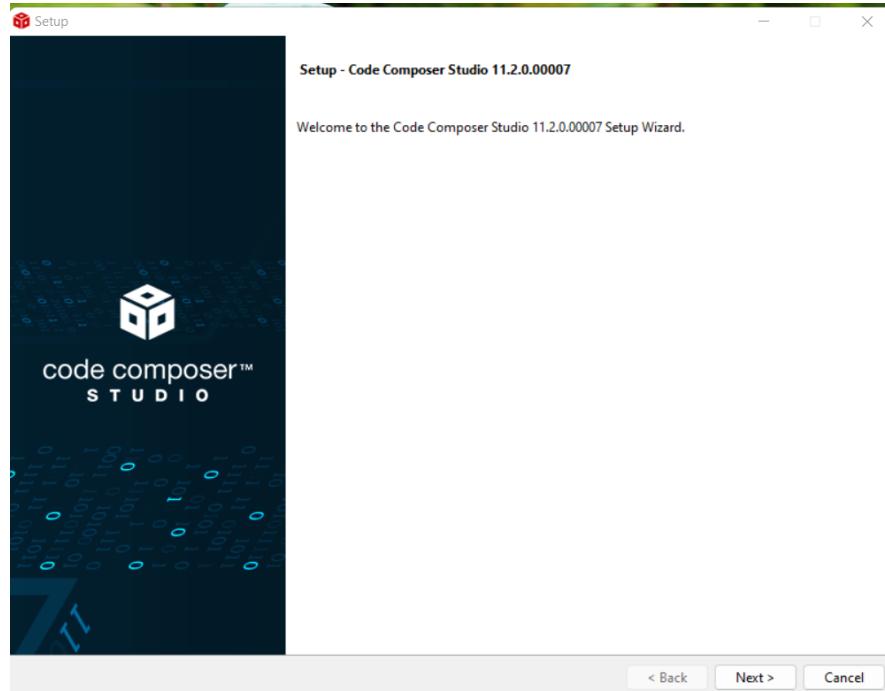
**Practical Name: CCS Installation, User Interface tour, Project creation, Compiling and debugging with MSP432**

1) CCS Installation Steps for CCS :-

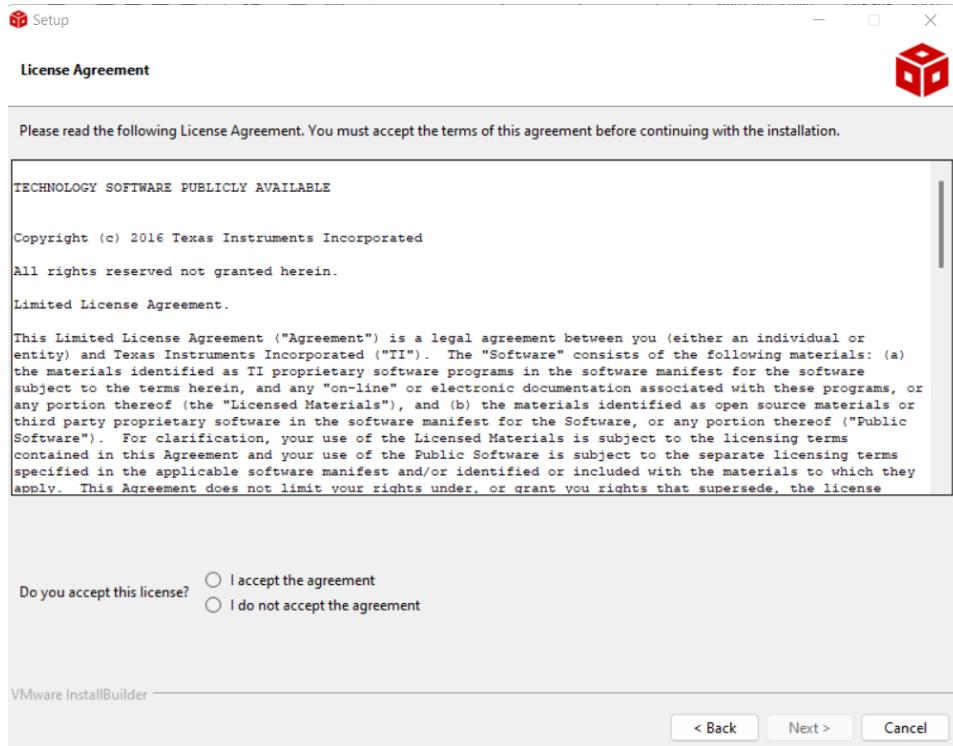
- Download CCS(version 11.2.0) from any browser:



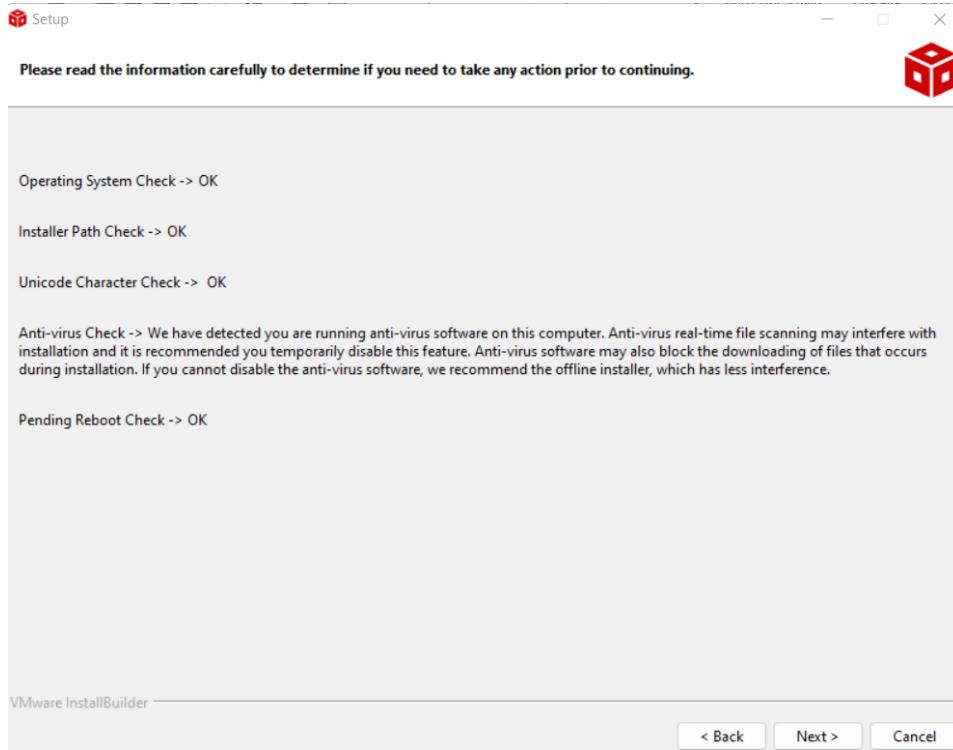
- Click on Install and Click on next:



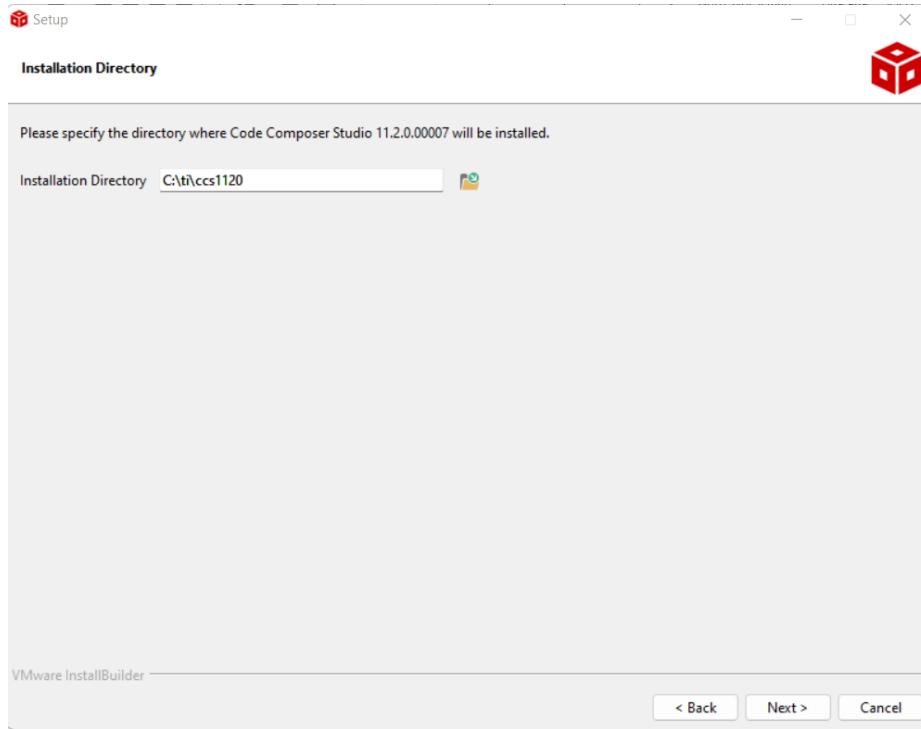
- Accept the agreement and click on next:



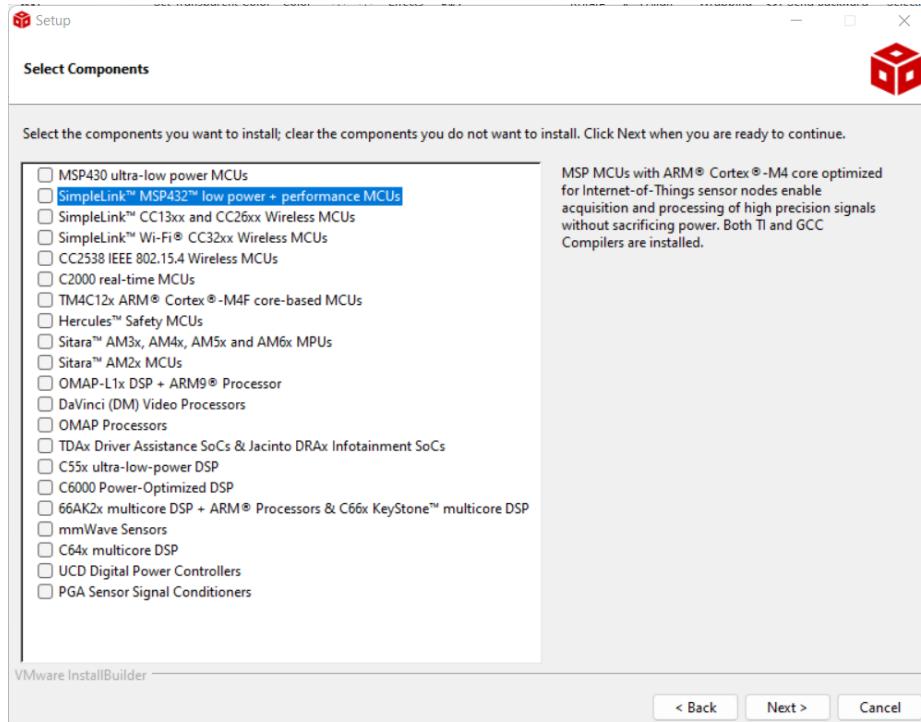
- Check for all OK & Click on Next



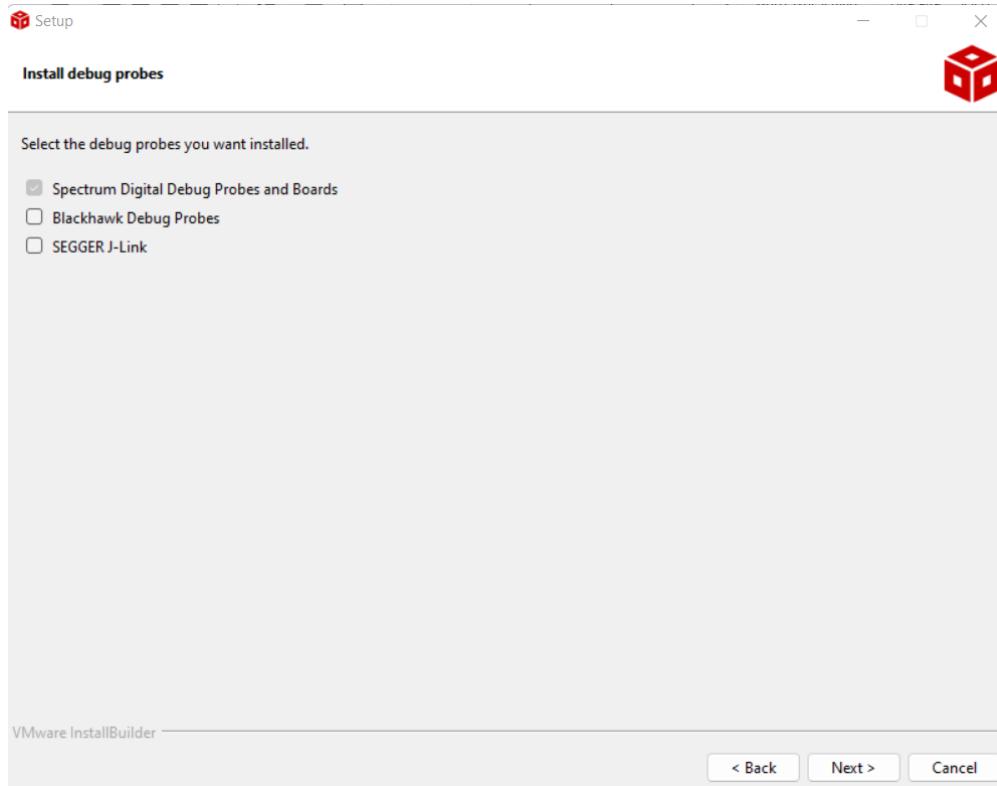
- Select directory for CCS: Mostly prefer C Drive



- Select as custom installation.
- Select the checkbox simplelink MSP432 low power+performer option:

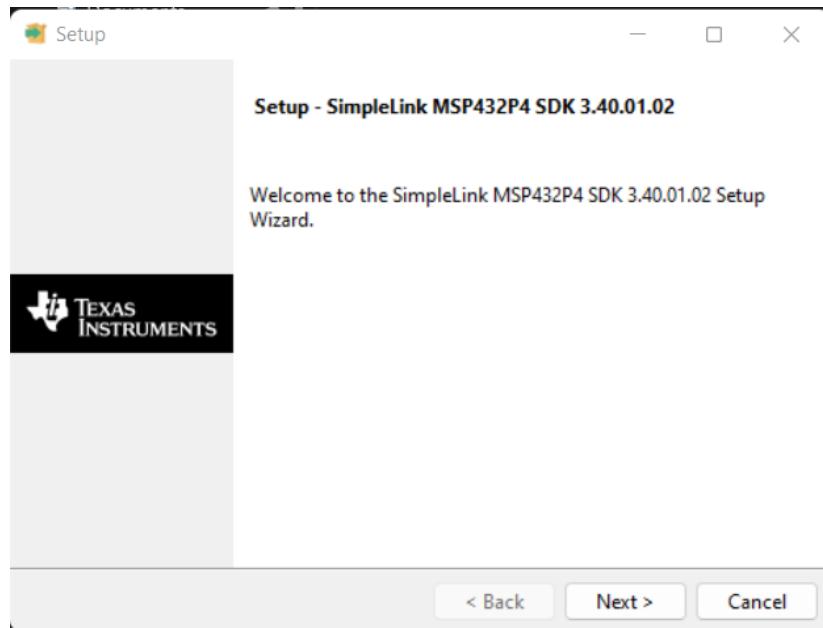


- Click on next to start the installation process:

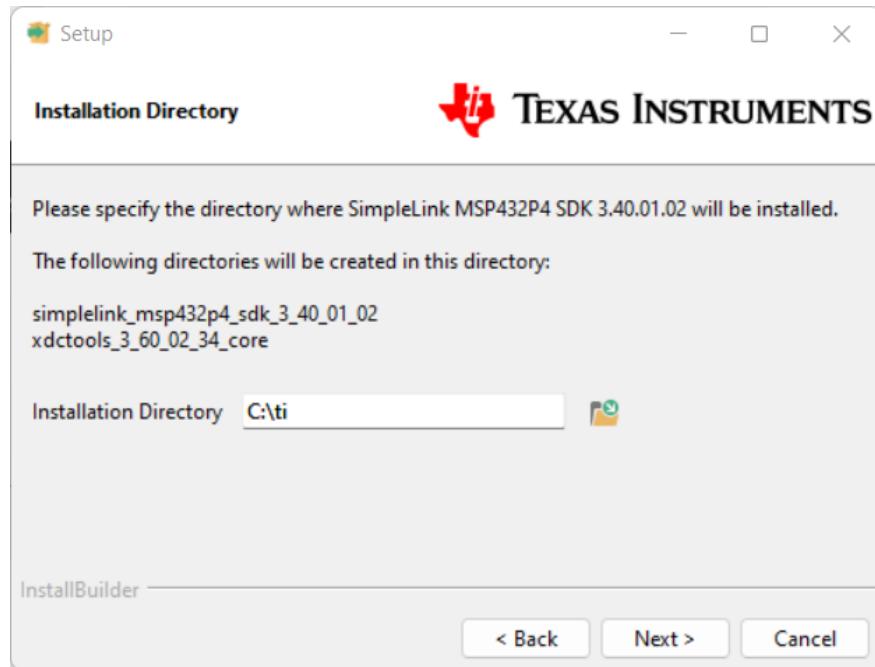


## 2) MSP432 SDK installation :(Simplelink)

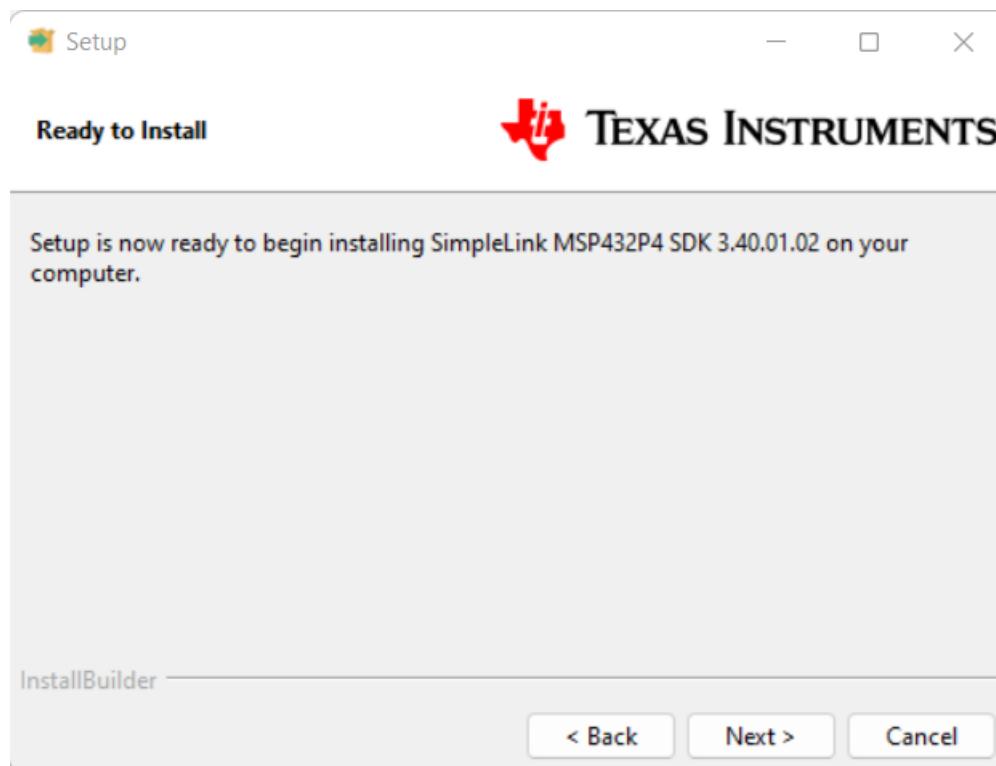
- Download simplelink tool from ti.com:
- Click on install and then click on next:



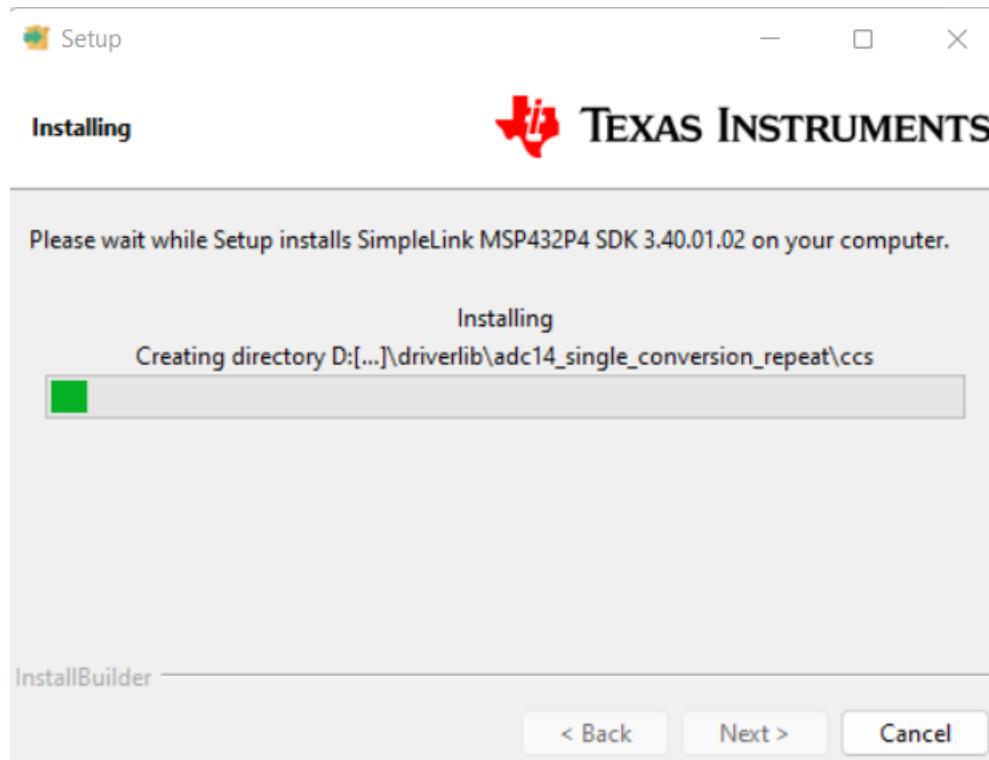
- Select appropriate folder/Directory to install the tool: (Mostly prefer C Drive)

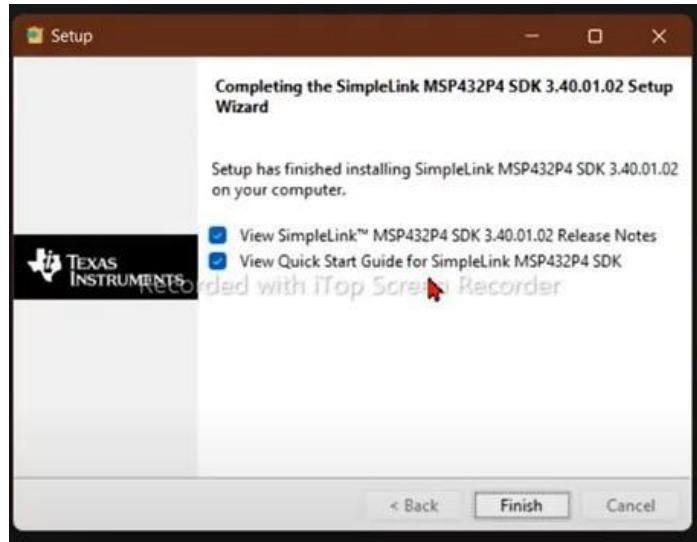


- Click on next:

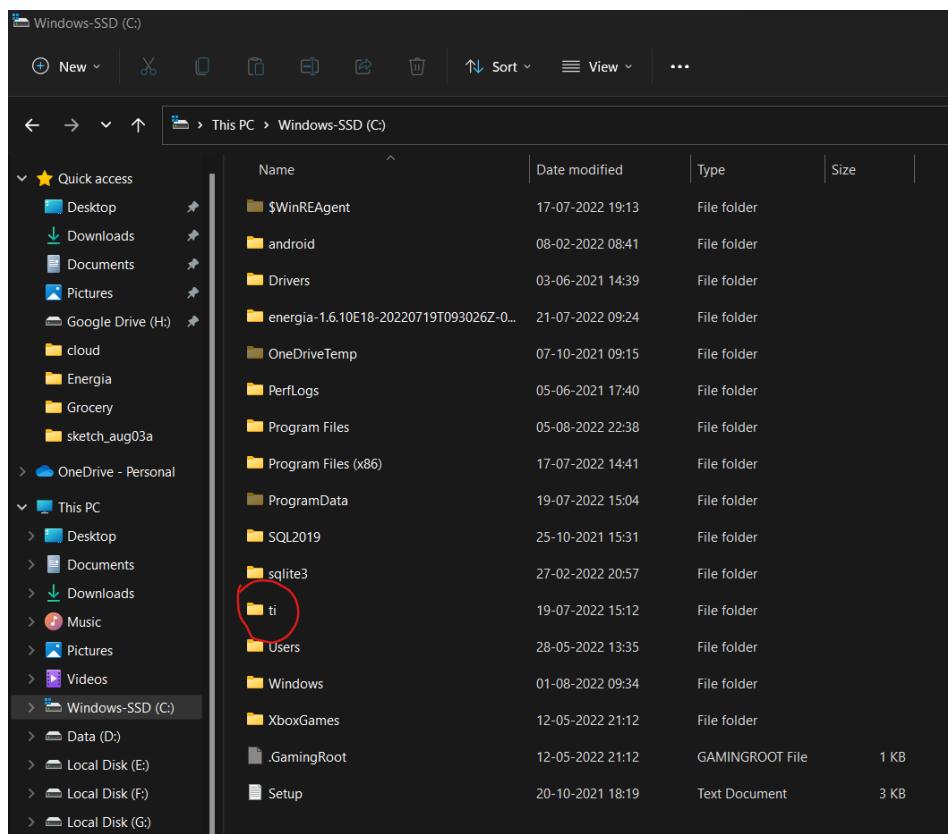


- Installation process is started:



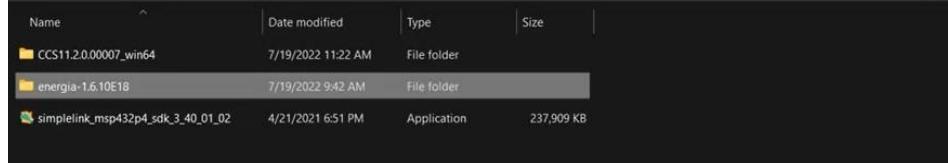


- We can see 'ti' folder inside C drive.

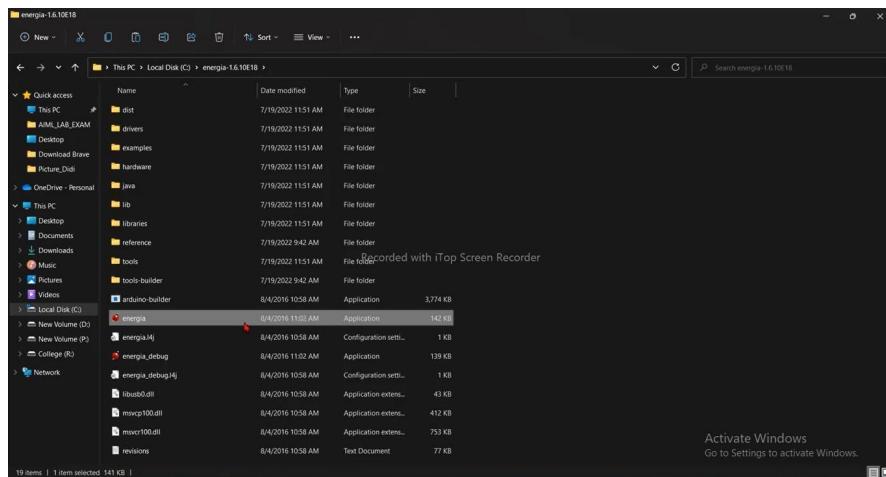


### 3) Eneric Installation With board manger details

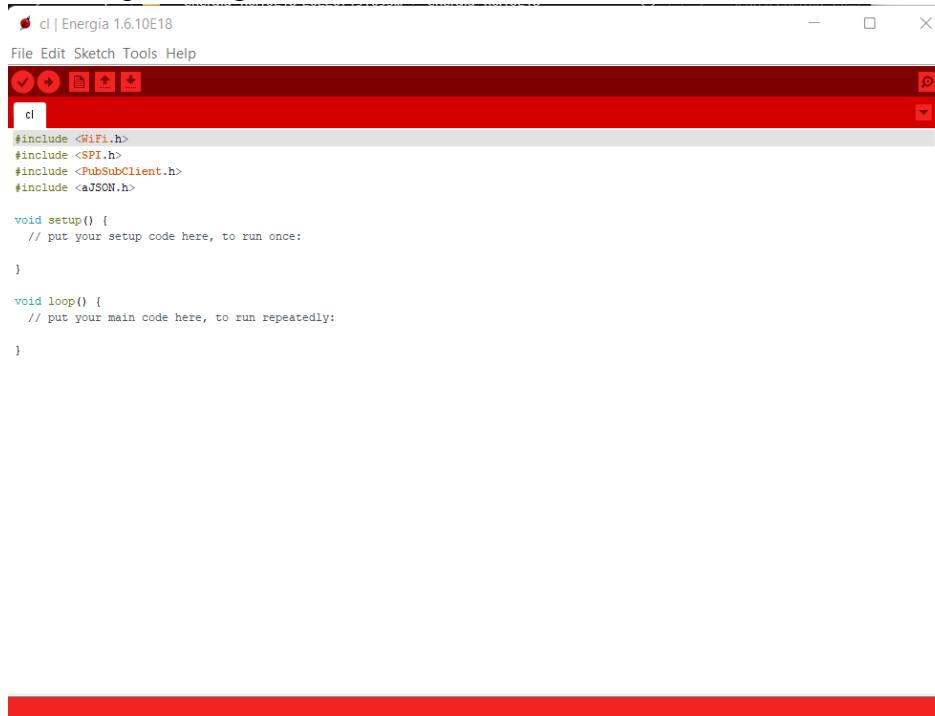
- Unzip energia folder:



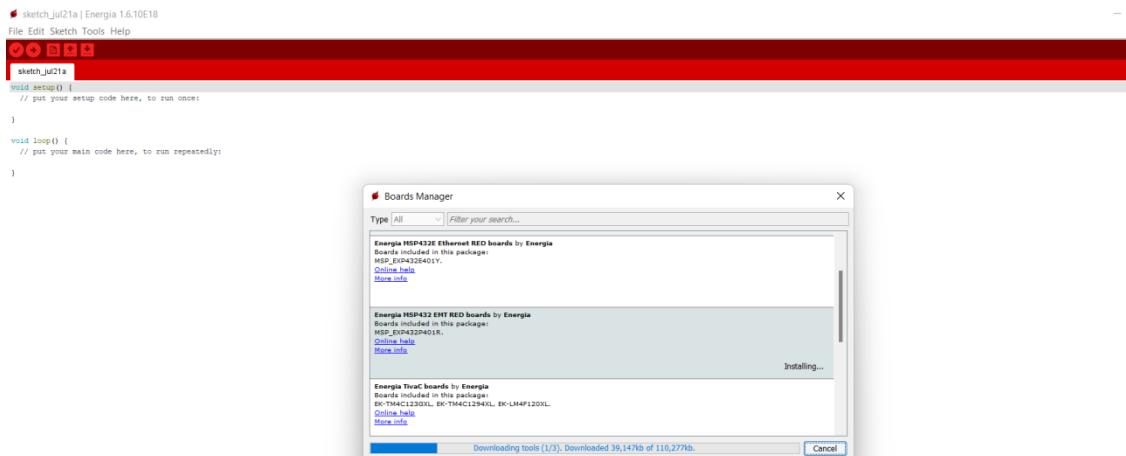
- Paste enegric folder to C Drive



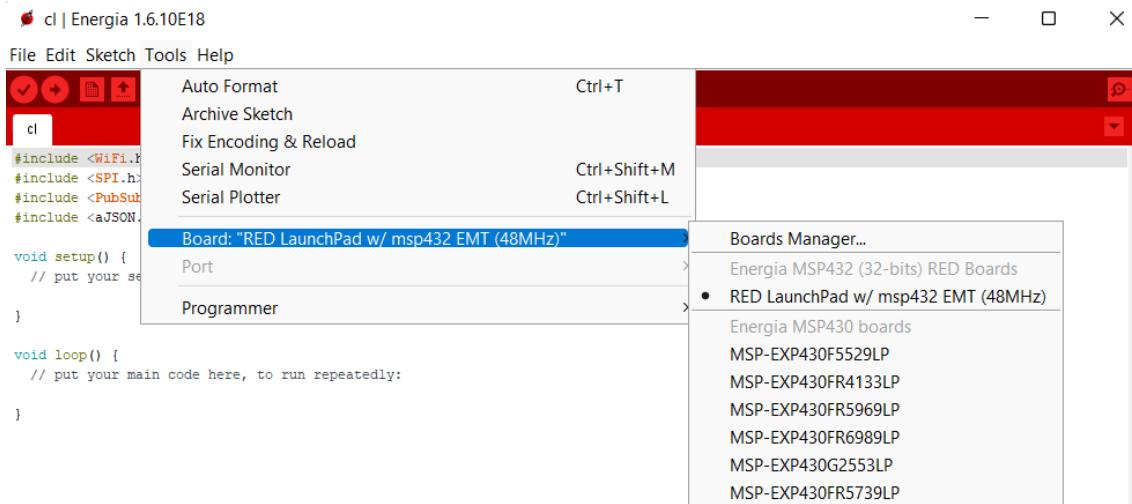
- Open energia:



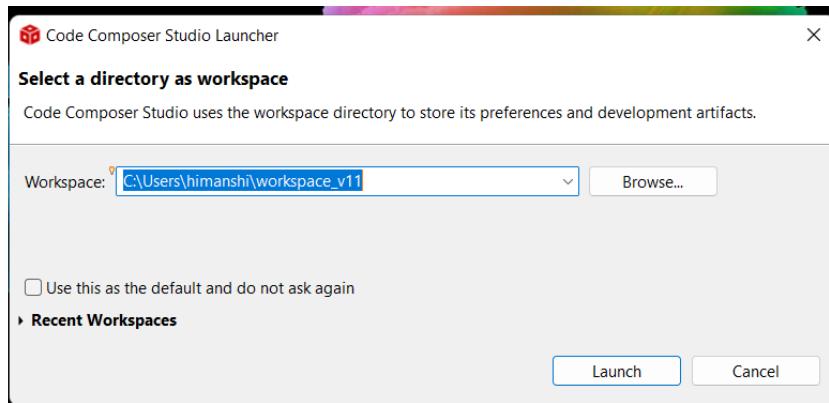
- To work with MSP432 we have to install board:  
Select Energia MSP432 EMT RED Boards by Energia and Select 5.6.1 version then click on install:



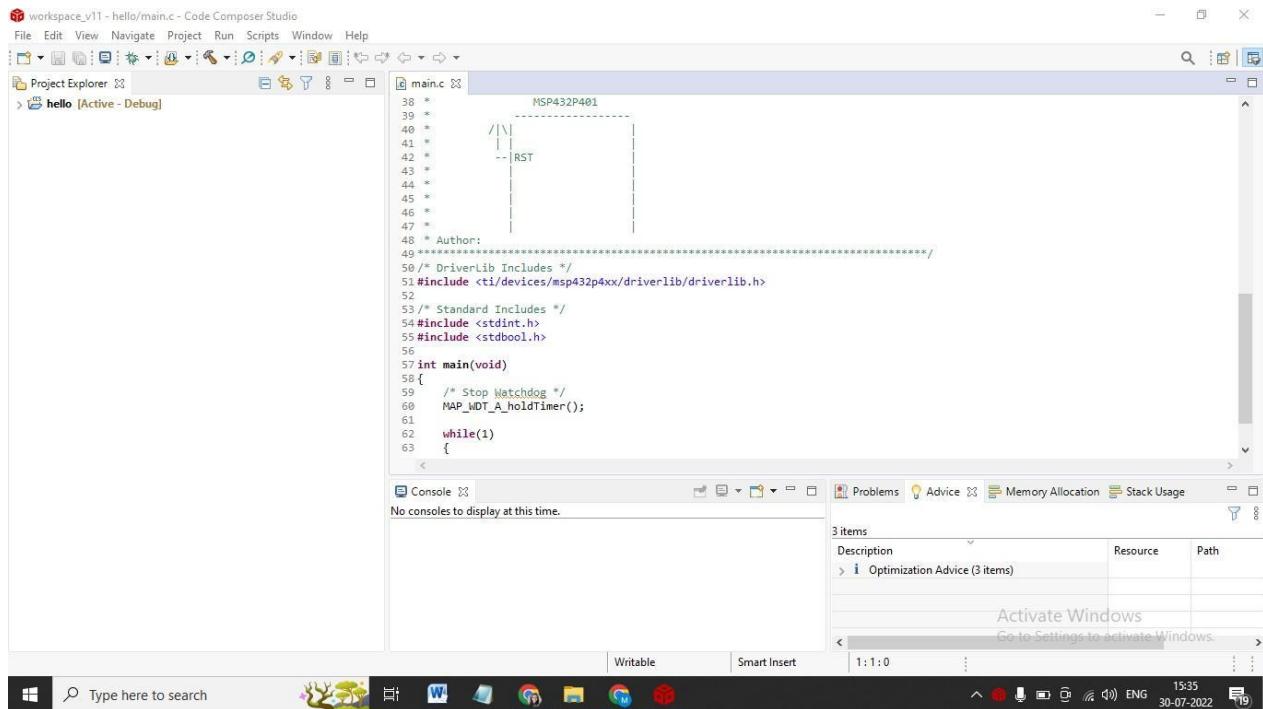
- Select RED LaunchPad w/ msp432 Board from Tool:



- Explain main components of CSS IDE:



- We have debugger option on right up corner:



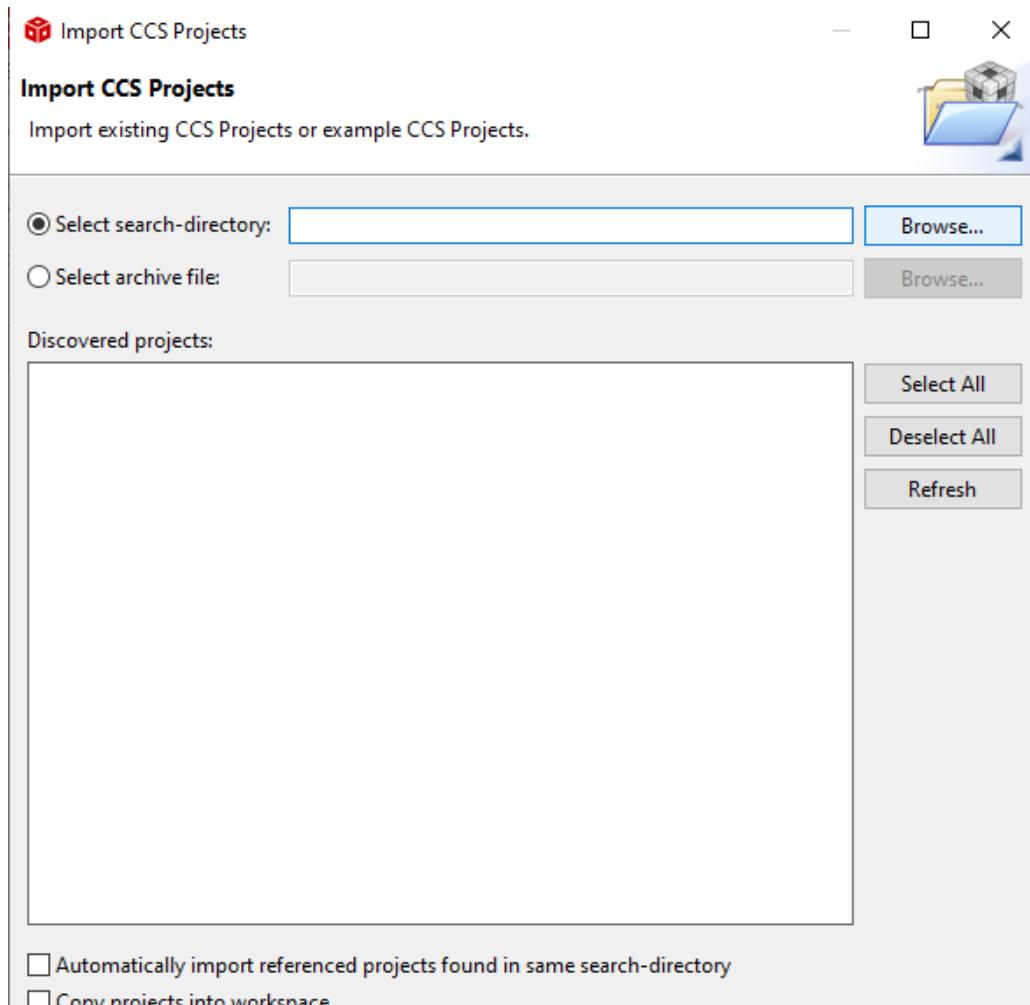
- We can debug the code by clicking on the hammer:
- We can run the code by Run => Debug:

```
workspace_v11 - hello/main.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Project Explorer Build 'Debug' for project 'hello'
hello [Active - Debug]
main.c
38 *          NSP432P401
39 *          -----
40 *          /|\\
41 *          | |
42 *          --| RST
43 *
44 *
45 *
46 *
47 *
48 * Author:
49 ****
50 /* DriverLib Includes */
51 #include <ti/devices/msp432p4xx/driverlib/driverlib.h>
52
53 /* Standard Includes */
54 #include <stdint.h>
55 #include <stdbool.h>
56
57 int main(void)
58 {
59     /* Stop Watchdog */
60     MAP_WDT_A_holdTimer();
61
62     while(1)
63     {
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
79 **** Build of configuration Debug for project hello ****
80 "C:\ti\ccs1120\ccs\utils\bin\gmake" -k -j 4 all -o
81 gmake[1]: 'hello.out' is up to date.
82 **** Build Finished ****
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
99
```

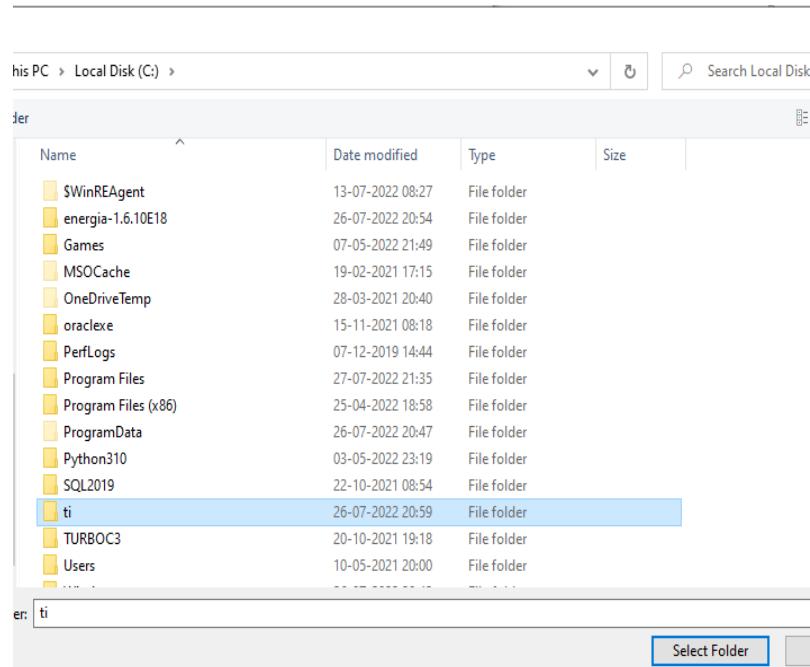
## Practical No. 02

**Practical Name: Explain the project creation process from the very beginning with the proper dialog boxes that pop up at every stage during project creation.**

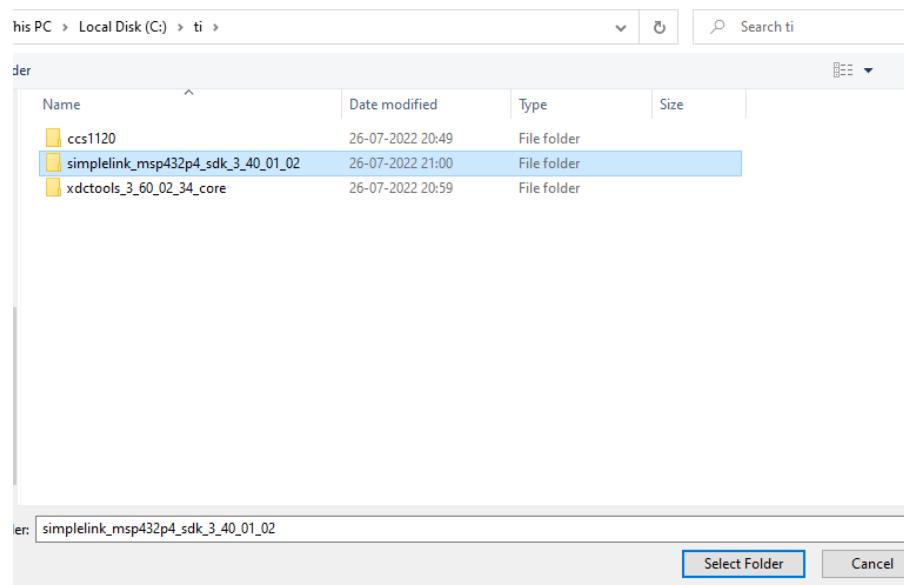
- Step 1: Project => Import CSS Project



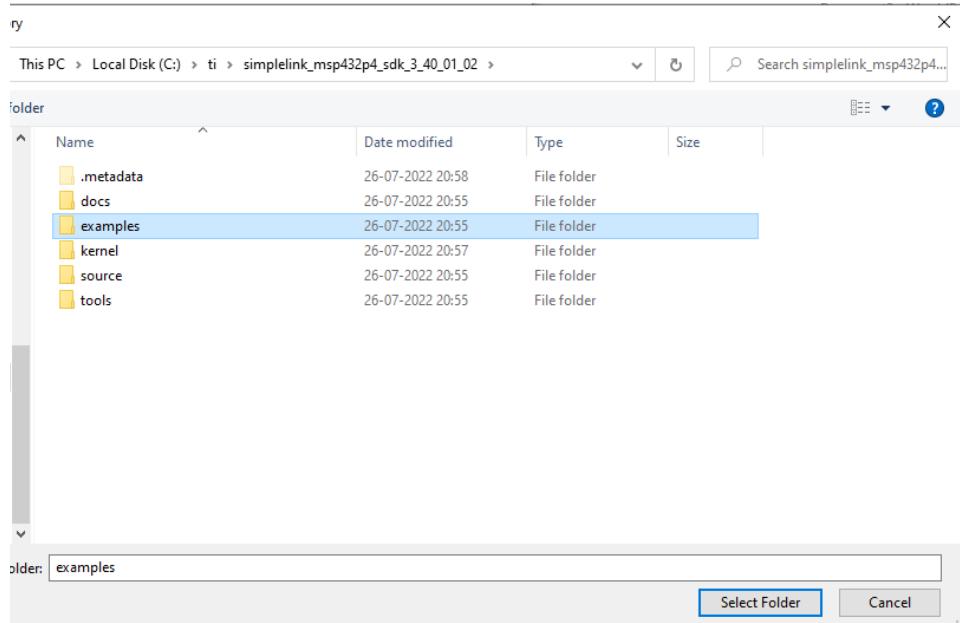
- Select ‘ti’ folder :



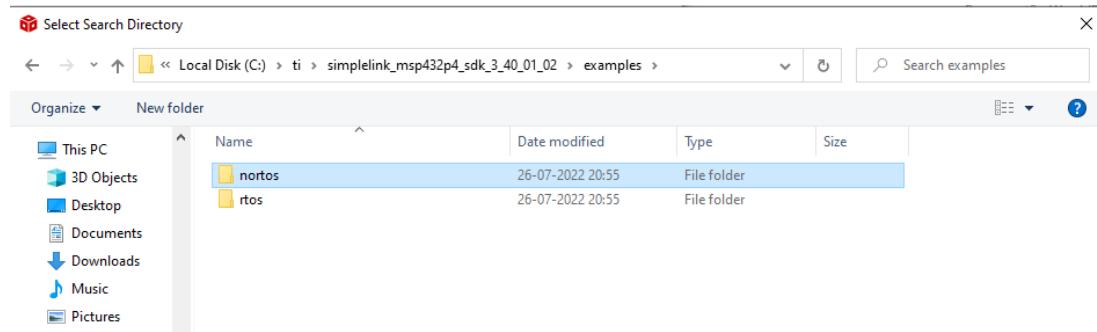
- Select below folder :



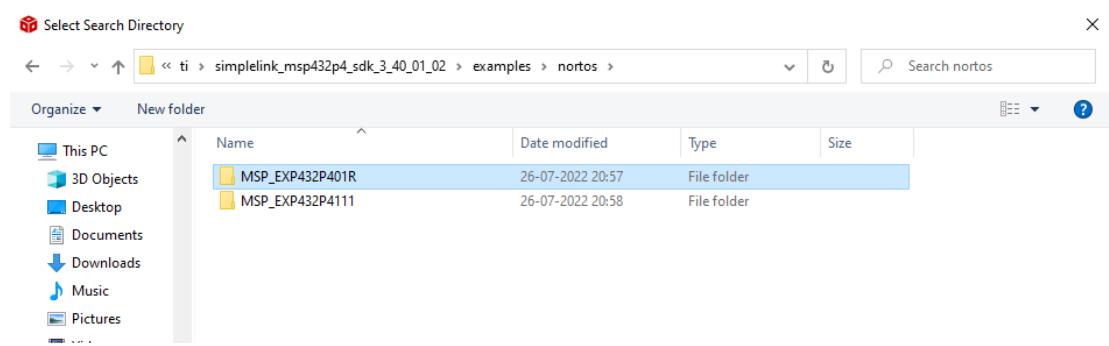
- Select examples folder:



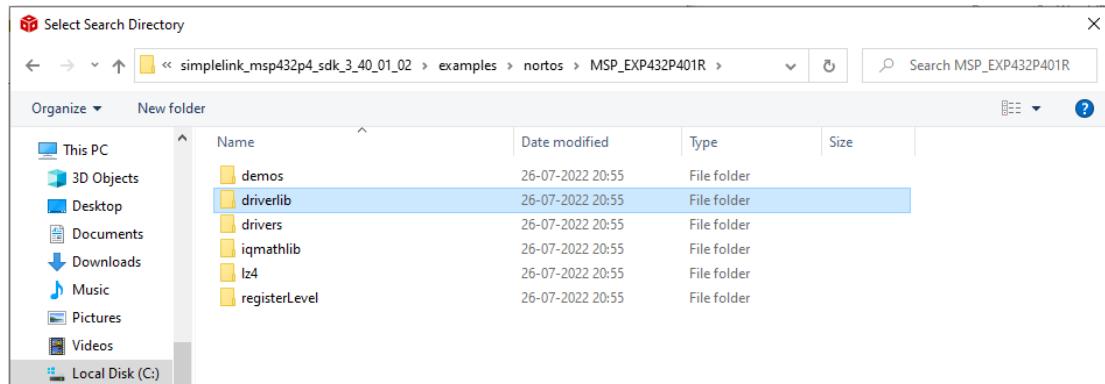
- Select nortos folder :



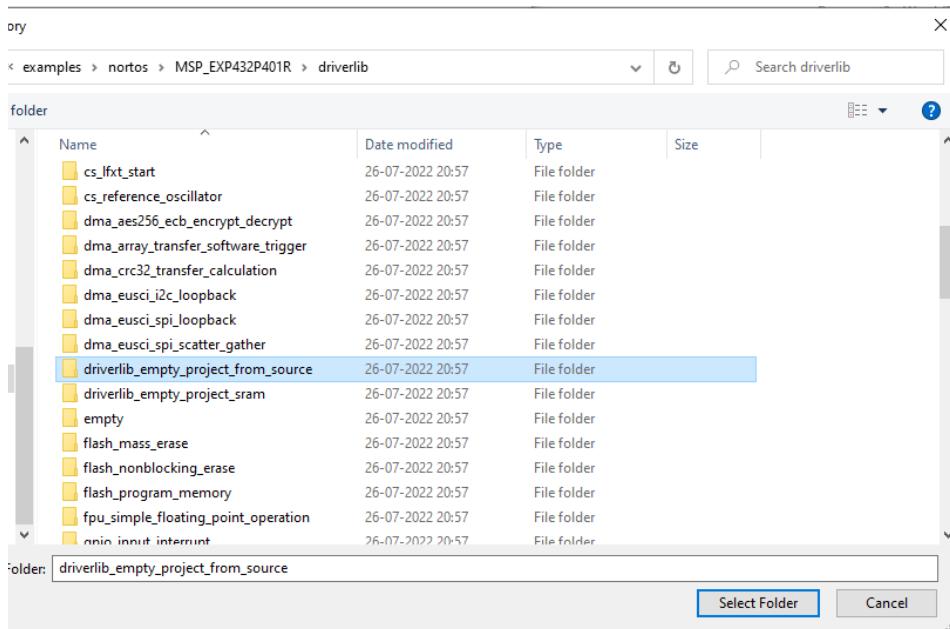
- Select below folder:



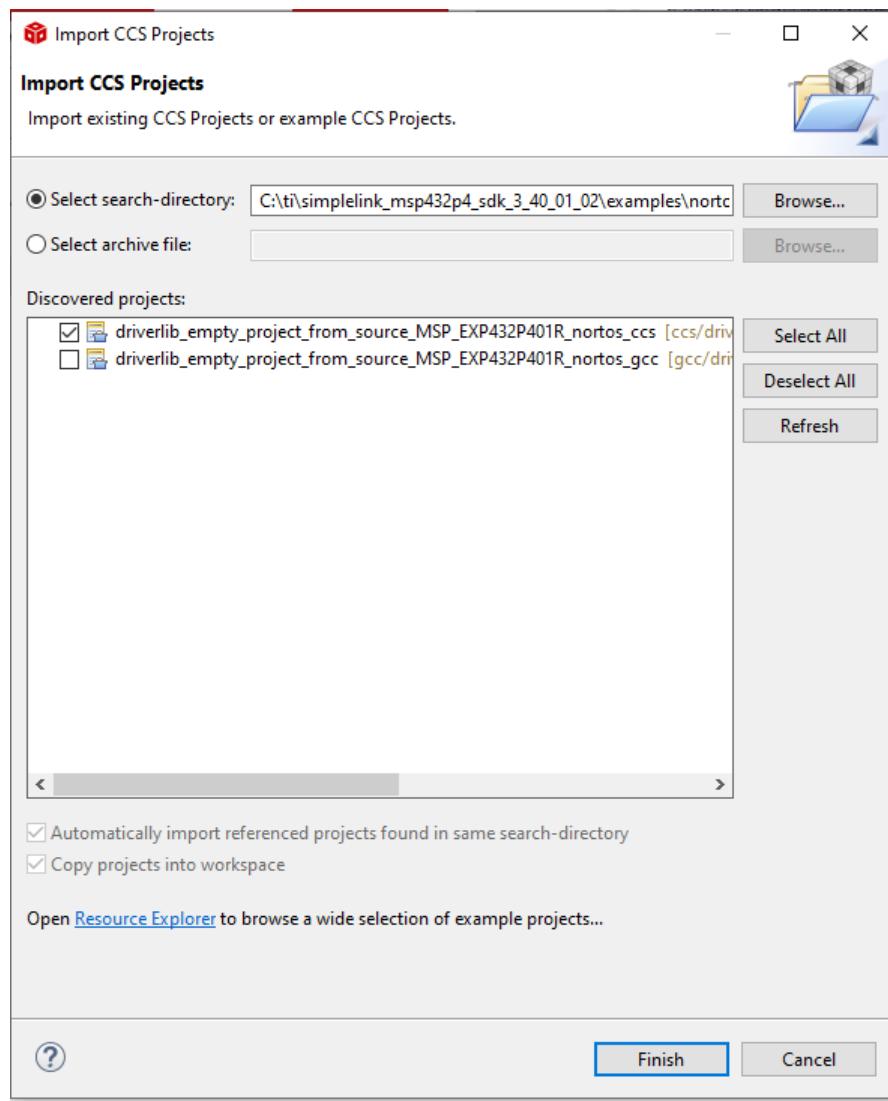
- Select driverlib folder :



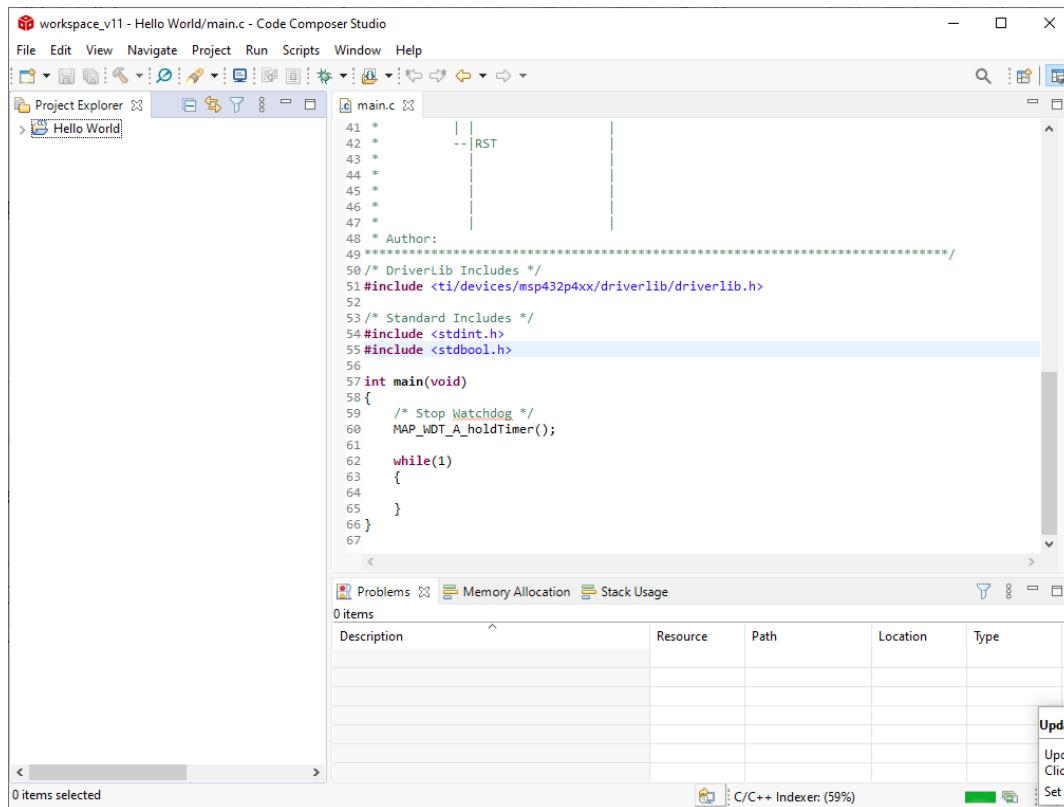
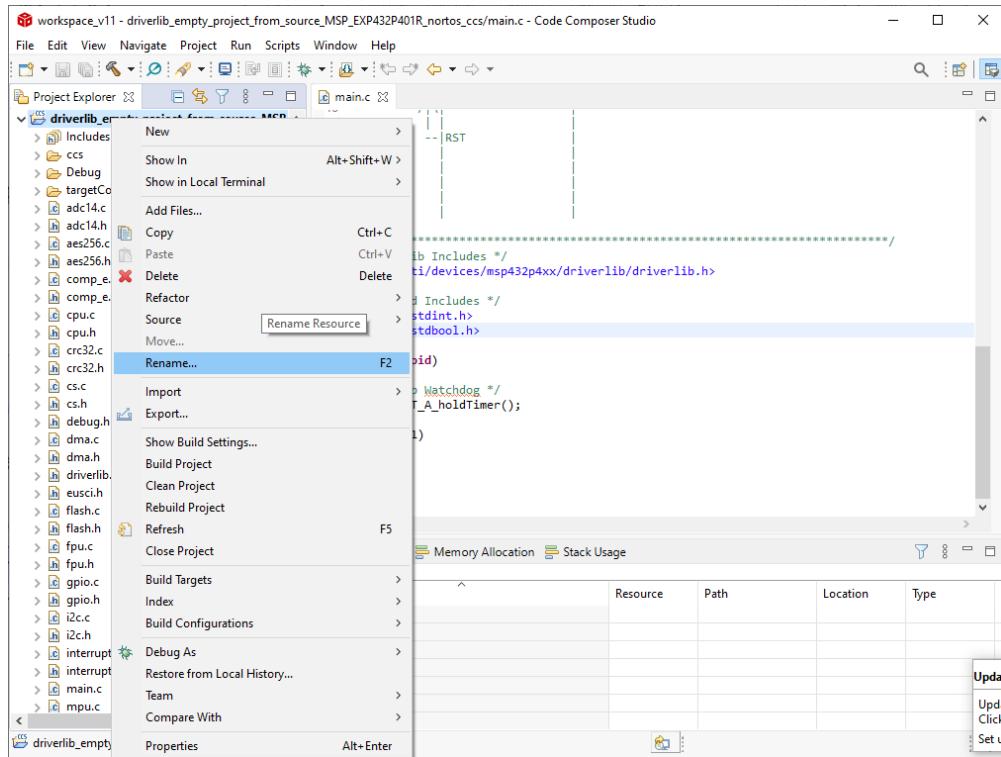
- Select below folder :



- Select the first ccs checkbox:



- Rename the created project as your preference:



- Put the full program for blinking LED.

Main.c

```
#include <ti/devices/msp432p4xx/driverlib/driverlib.h>
/* Standard Includes */
#include <stdint.h>
#include <stdbool.h>
int main(void)
{
/* Stop Watchdog */
MAP_WDT_A_holdTimer();
int i;
GPIO_setAsOutputPin(GPIO_PORT_P2, GPIO_PIN0); while(1)
{
GPIO_setOutputHighOnPin(GPIO_PORT_P2, GPIO_PIN0); for(i=0;i<10000;i++); //random
delay
GPIO_setOutputLowOnPin(GPIO_PORT_P2, GPIO_PIN0); for(i=0;i<10000;i++); // random
delay // GPI General Purpose Input Output
}
}
```

- Explain the full process of entering Debug mode and then starting the program run on the MSP432 board.

Create a new project by importing css project and rename it:

Click on Hammer icon to Build the project

```
workspace_v11 - HelloWorld/main.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Project Explorer main.c adc14.c sketch_jul21a.ino main.c cl.ino project
48 * Author:
49 ****
50/* DriverLib Includes */
51#include <ti/devices/msp432p4xx/driverlib/driverlib.h>
52
53/* Standard Includes */
54#include <stdint.h>
55#include <stdbool.h>
56
57int main(void)
58{
59    /* Stop_Watchdog */
60    MAP_WDT_A_holdTimer();
61    int i;
62    GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0);
63    while(1)
64    {
65        GPIO_setOutputHighOnPin(GPIO_PORT_P2,GPIO_PIN0);
66        for(i=0; i<10000; i++);
67        GPIO_setOutputLowOnPin(GPIO_PORT_P2,GPIO_PIN0);
68        for(i=0; i<10000; i++);
```

- Step 2 : In Energia write include library



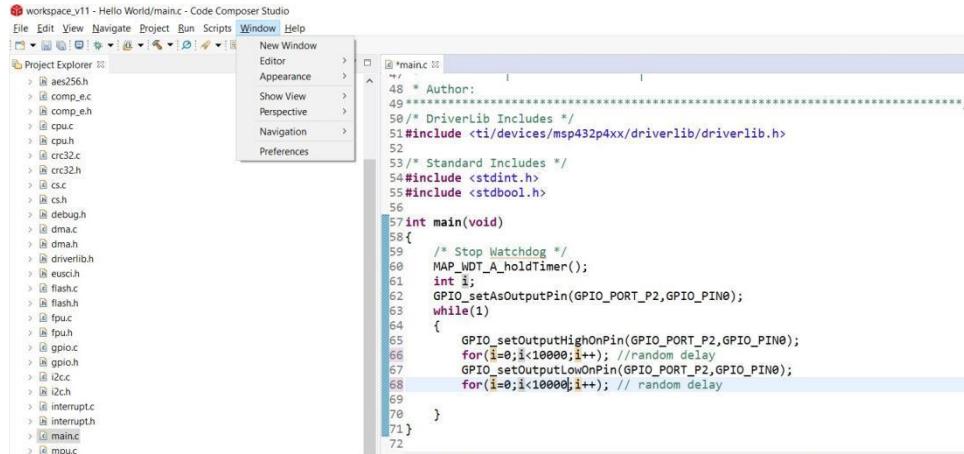
```
#include<WiFi.h>
#include<SPI.h>
```

```
void setup() {
    // put your setup code here, to run once:
}

void loop() {
    // put your main code here, to run repeatedly:
}
```



- Go on Windows => Preferences:



workspace\_v11 - Hello World/main.c - Code Composer Studio

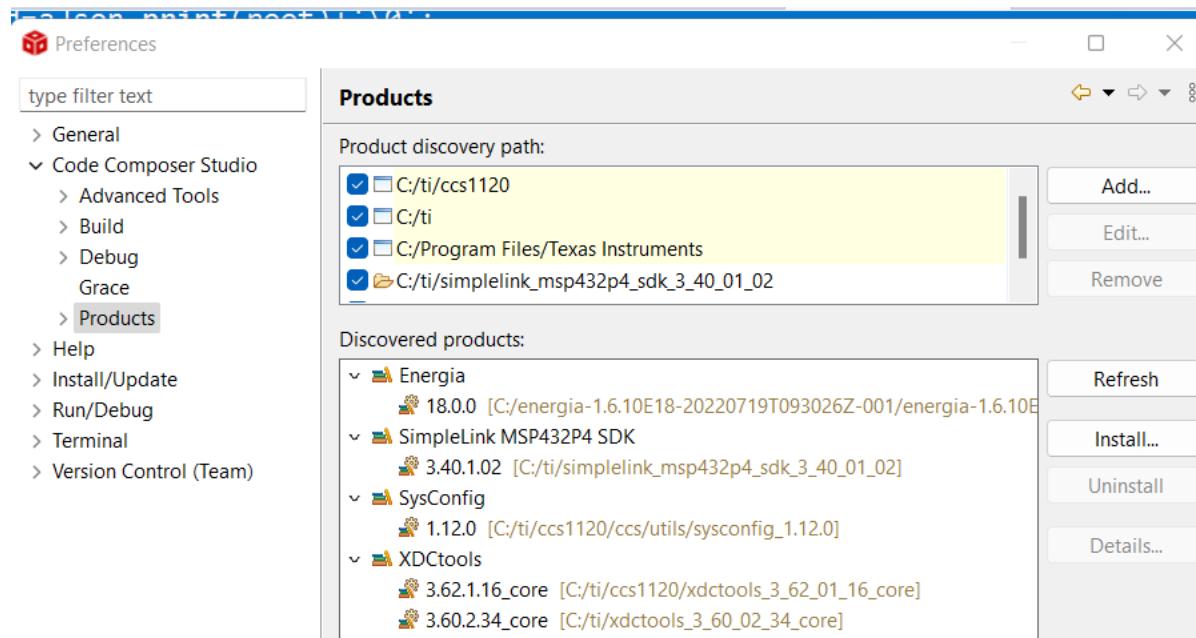
File Edit View Navigator Project Run Scripts Window Help

Project Explorer

New Window Editor Appearance Show View Perspective Navigation Preferences

```
/* * Author: ****
50 /* DriverLib Includes */
51 #include <ti/devices/msp432p4xx/driverlib/driverlib.h>
52
53 /* Standard Includes */
54 #include <stdint.h>
55 #include <stdbool.h>
56
57 int main(void)
58{
59    /* Stop Watchdog */
60    MAP_WDT_A_holdTimer();
61    int i;
62    GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0);
63    while(1)
64    {
65        GPIO_setOutputHighOnPin(GPIO_PORT_P2,GPIO_PIN0);
66        for(i=0;i<10000;i++); //random delay
67        GPIO_setOutputLowOnPin(GPIO_PORT_P2,GPIO_PIN0);
68        for(i=0;i<10000;i++); // random delay
69    }
70}
71}
```

- Click on Code Composer Studio => Products => Add XDCTools by selecting ti folder



- Now, go to Projects =>

Import Engeria in CSS file we created : Go to Windows :

Write below code in the project.

```
#include <WiFi.h>
#include <SPI.h>
IPAddress shieldIP,subnetMask,gatewayIP;
uint8_t rssi;
uint8_t networkId;
byte macAddr[6];
byte encryptionType;
char ssid[]="Opp";//my cellphone
//char password[]="";
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial.print("connecting to wifi");
```

```

while(WiFi.begin(ssid)!=WL_CONNECTED)//for cellphone
//while(WiFi.begin(ssid,password)!=WL_CONNECTED)
{
    Serial.print(".");
    delay(1); //delay in ms
}
Serial.println("");
Serial.print("Wifi connected ,fetching wifi sheild's IP Address: ");
while(WiFi.localIP()==INADDR_NONE)
{
    Serial.print(".");
    delay(1); //delay in ms
}
shieldIP = WiFi.localIP();
Serial.println(shieldIP);
Serial.print("Access point name: ");
Serial.println(ssid);
Serial.print("Signal strength: ");
rssI=WiFi.RSSI();
Serial.println(rssI); //RSSI Received Signal Strength Indication.
uint8_t networkId=WiFi.scanNetworks();
Serial.print("Number of access points in Range:");
Serial.println(networkId);
for(int i=1;i<=networkId;i++)
{
    Serial.print("Name of Access Points and Encryption type:");
    Serial.print(WiFi.SSID(1));
    Serial.print(",");
    encryptionType=WiFi.encryptionType(1);
    //Serial.println(encryptionType,Hex);
    Serial.println(encryptionType,DEC);
}

```

```
}

subnetMask=WiFi.subnetMask();
Serial.println("Subnet mask: ");
Serial.print(subnetMask);

gatewayIP=WiFi.gatewayIP();
Serial.println("Gateway IP Address: ");
Serial.print(gatewayIP);

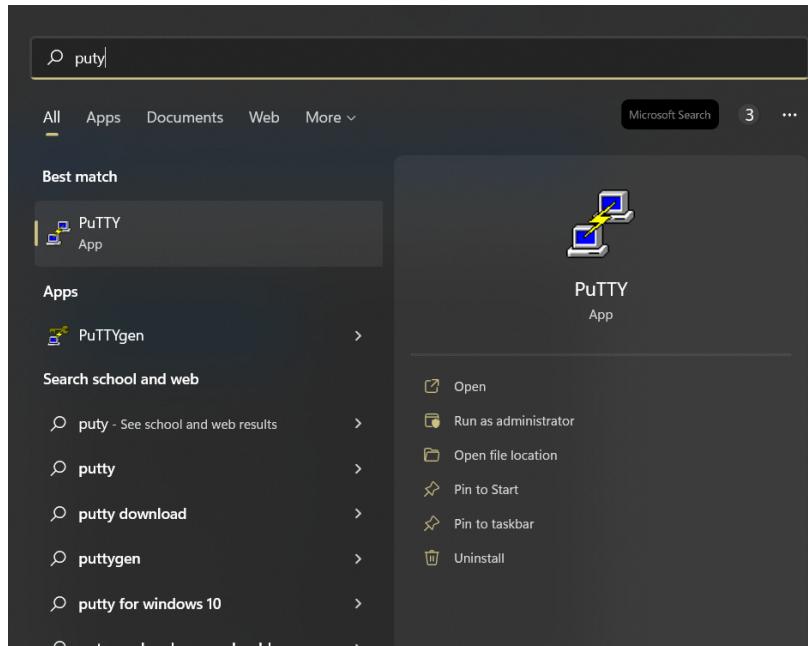
WiFi.macAddress(macAddr);
Serial.println("Mac Address of Sheild:");
for(int i=0;i<6;i++)
{
    Serial.print(macAddr[i],HEX);
    if(i<=4)
        Serial.print(':');
    else
        Serial.println();
}
}

void loop() {
    // put your main code here, to run repeatedly:
}
```

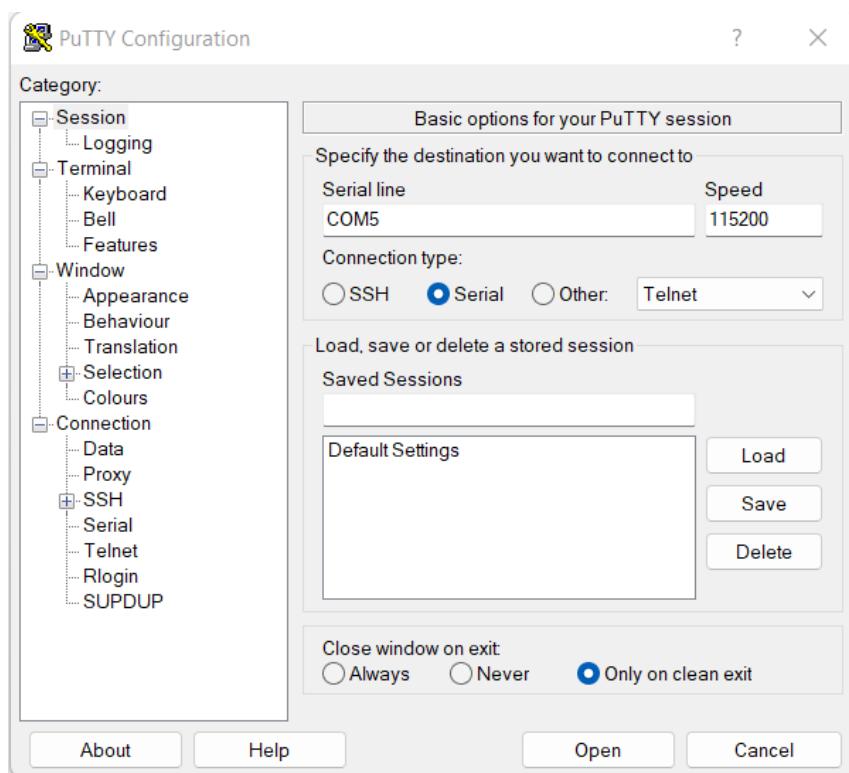
Now build the project. And then debug the project.

Download Putty for terminal:

Putty :



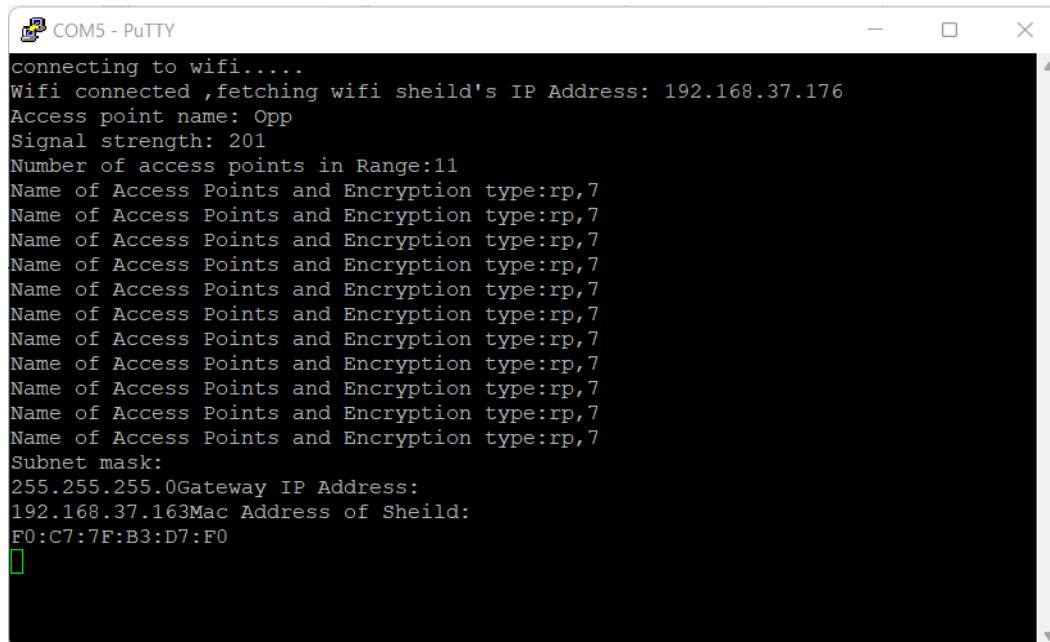
Go to Serial => Add serial line same as your boards port number(COM5)  
Add Speed as 115200



Click on Open and resume the project:

Wait for the wifi to be connected..

We can see Access point name, Signal strength, Number of access points in range, Subnet Mask, Gateway IP Address, Mac Address of Shield.



The screenshot shows a PuTTY terminal window with the title 'COM5 - PuTTY'. The window displays the following text output:

```
connecting to wifi.....
Wifi connected ,fetching wifi sheild's IP Address: 192.168.37.176
Access point name: Opp
Signal strength: 201
Number of access points in Range:11
Name of Access Points and Encryption type:rp,7
Subnet mask:
255.255.255.0Gateway IP Address:
192.168.37.163Mac Address of Sheild:
F0:C7:7F:B3:D7:F0
```

# Practical No. 03

## Practical Name: Installing Node.js ,Node-Red and Node Red interface

### Node.js installation

Installing Node.js ,Node-Red and Node Red interface Points to be covered in the following order:

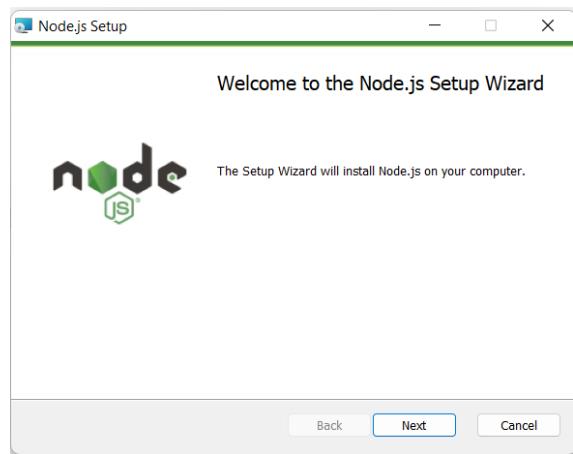
- Start with the steps for installing Node.js, Node-red.

Download Node.js from any browser:

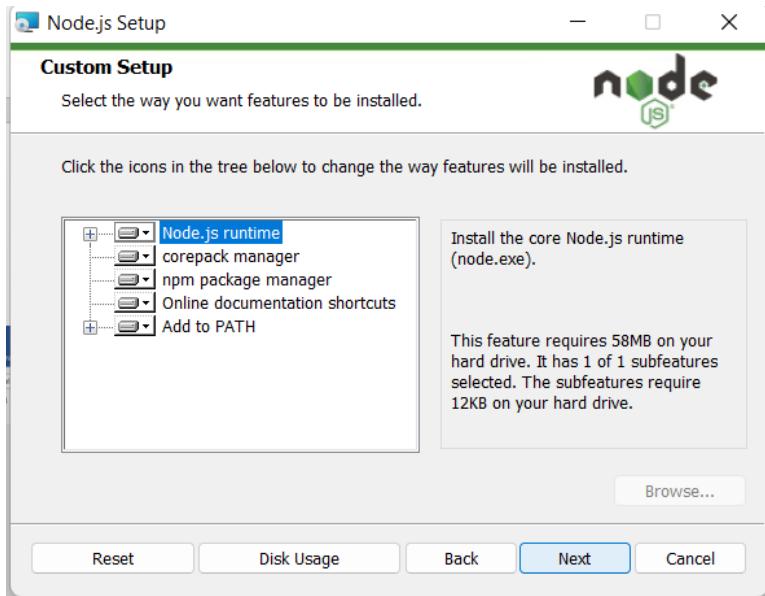
The screenshot shows the official Node.js download page. At the top, it says "Latest LTS version: 16.16.0 (includes npm 8.1.0)". Below this, there are two main sections: "LTS" (Recommended For Most Users) and "Current" (Latest Features). Under "LTS", there are links for "Windows Installer" (node-v16.16.0-x64.msi), "macOS Installer" (node-v16.16.0.pkg), and "Source Code" (node-v16.16.0.tar.gz). Under "Current", there is a table showing binary distributions for different platforms:

	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Binary (.tar.gz)	64-bit / ARM64	
Linux Binaries (x64)	64-bit	ARM64
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v16.16.0.tar.gz	

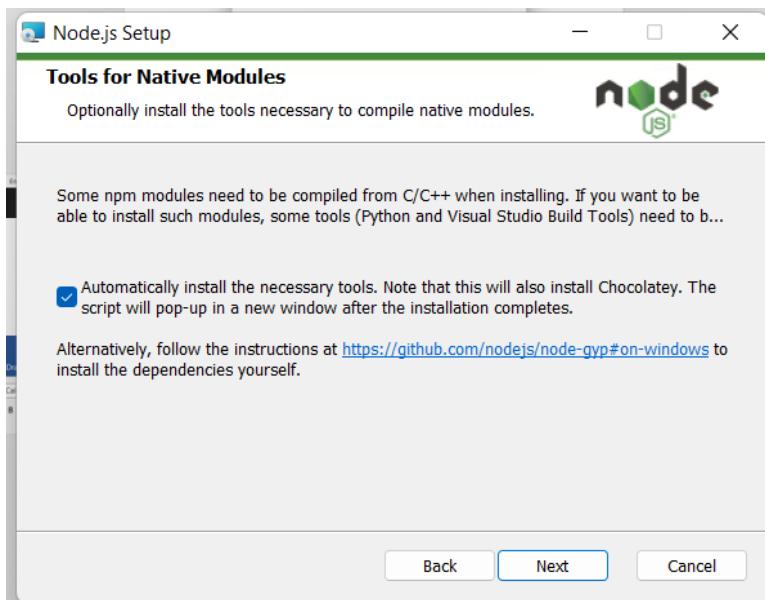
### Install the Node.js



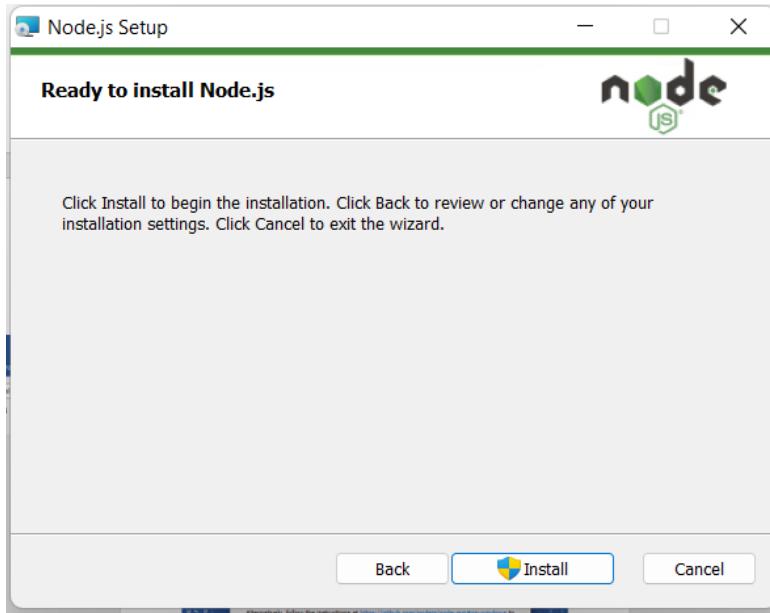
Click on Next:



Click on Next:



Click on Install:



Setup for node.js:

```
C:\Users\Owner>node --version && npm --version
v16.16.0
npm [WARN] config global `--global`, `--local` are deprecated. Use `--location=global` instead.
8.11.0

C:\Users\Owner>npm install -g --unsafe-perm node-red
npm [WARN] config global `--global`, `--local` are deprecated. Use `--location=global` instead.
npm [WARN] config global `--global`, `--local` are deprecated. Use `--location=global` instead.

added 294 packages, and audited 295 packages in 2m

38 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 8.11.0 → 8.15.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.15.0
npm notice Run npm install -g npm@8.15.0 to update!
npm notice

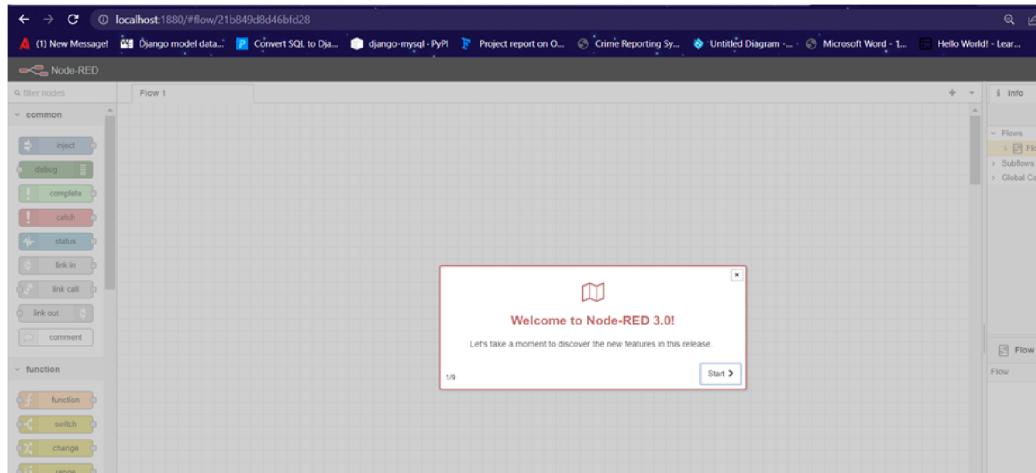
C:\Users\Owner>
```

To check the version of node.js we use **node –version** command.

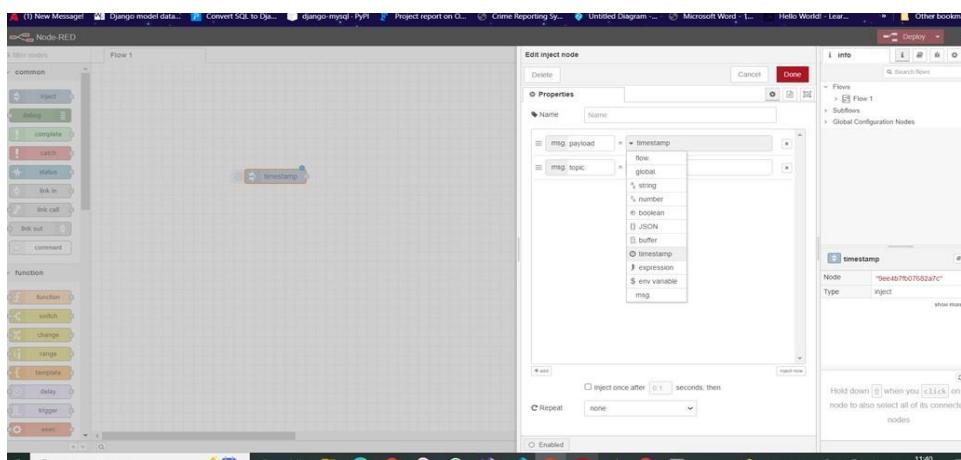
- Start node red using **node-red** command:

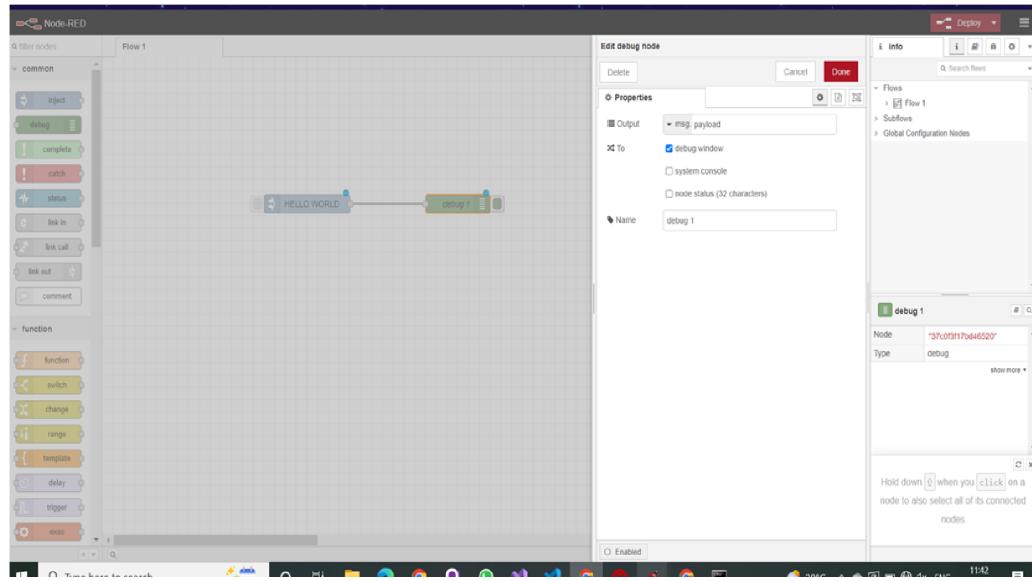
```
C:\>node-red
3 Aug 22:26:03 - [info]
Welcome to Node-RED
=====
3 Aug 22:26:03 - [info] Node-RED version: v3.0.1
3 Aug 22:26:03 - [info] Node.js version: v16.16.0
3 Aug 22:26:03 - [info] Windows_NT 10.0.22000 x64 LE
3 Aug 22:26:06 - [info] Loading palette nodes
3 Aug 22:26:09 - [info] Dashboard version 3.1.7 started at /ui
3 Aug 22:26:09 - [info] Settings file : C:\Users\ROSHA\.node-red\settings.js
3 Aug 22:26:09 - [info] Context store : 'default' [module=memory]
3 Aug 22:26:09 - [info] User directory : \Users\ROSHA\.node-red
3 Aug 22:26:09 - [warn] Projects disabled : editorTheme.projects.enabled=false
3 Aug 22:26:09 - [info] Flows file : \Users\ROSHA\.node-red\flows.json
3 Aug 22:26:09 - [info] Server now running at http://127.0.0.1:1880/
3 Aug 22:26:09 - [warn]
```

- Now, open the browser => Write in query string => **localhost:1880**



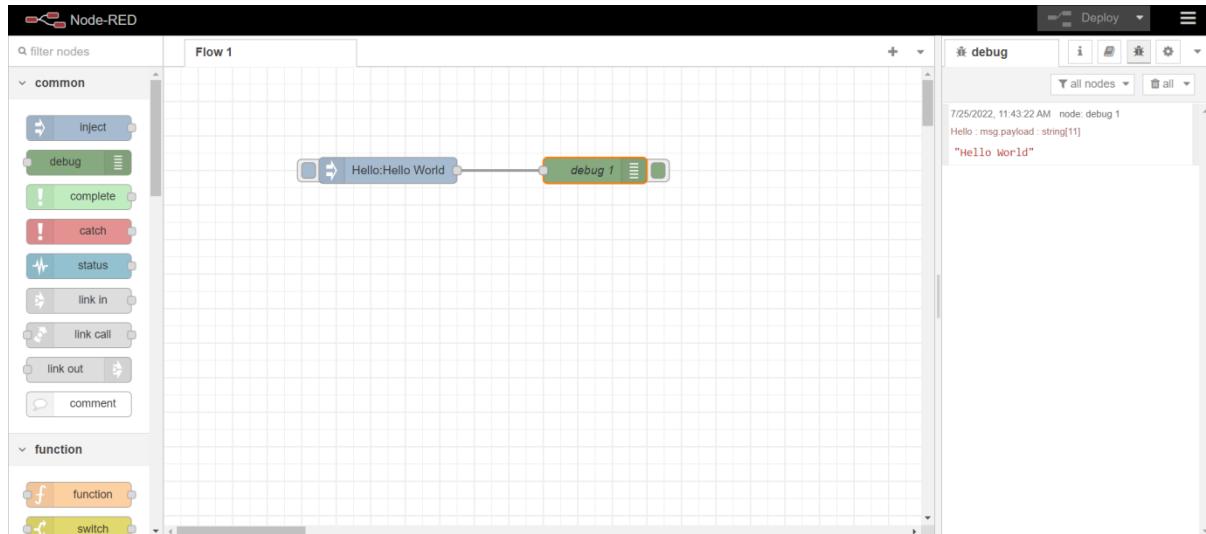
- Mark all the parts of the node red interface and explain each section of the IDE including the Menu.





Select one inject and one debug node:

Inject node used to trigger the flow.

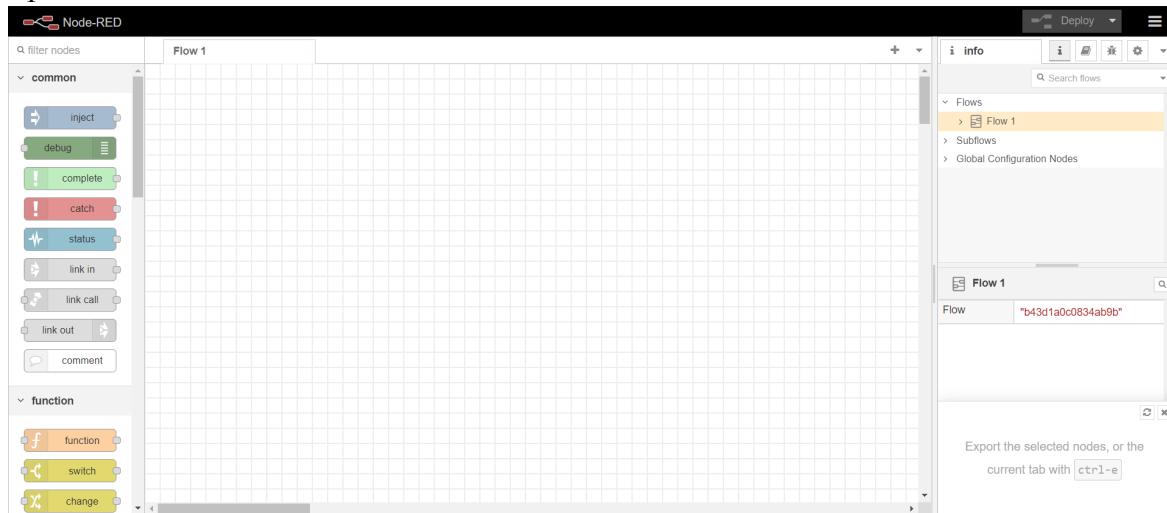


# Practical No. 04

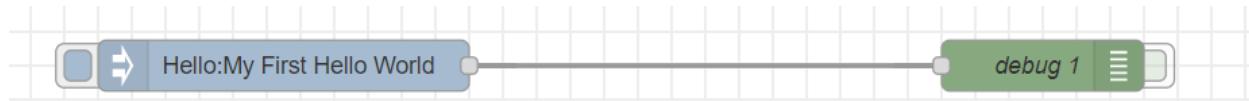
**Practical Name: Introducing the inject, function, debug and switch nodes**  
**Points to be covered in the following order:**

- Use the inject and debug nodes.(Explain the input and the output for the ‘Hello’ flow. Put the screen shots with screenshots of the settings and label correctly with explanations)

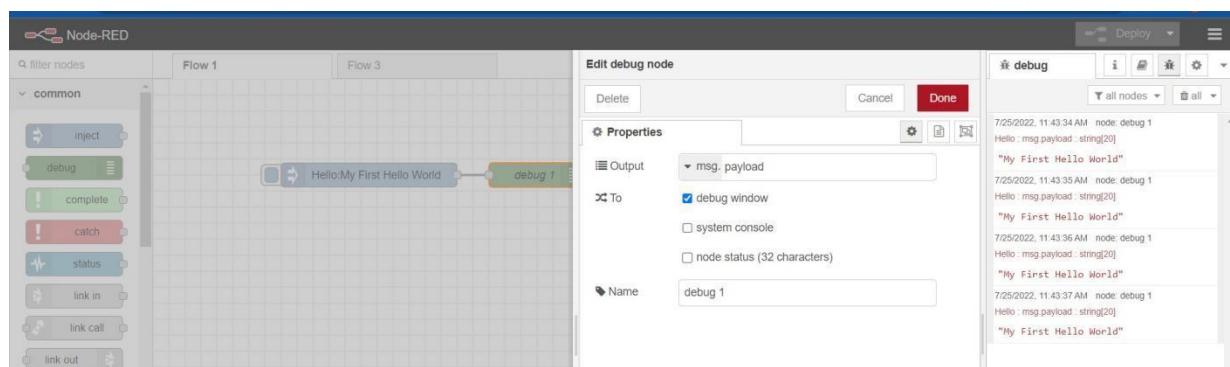
Open node red:



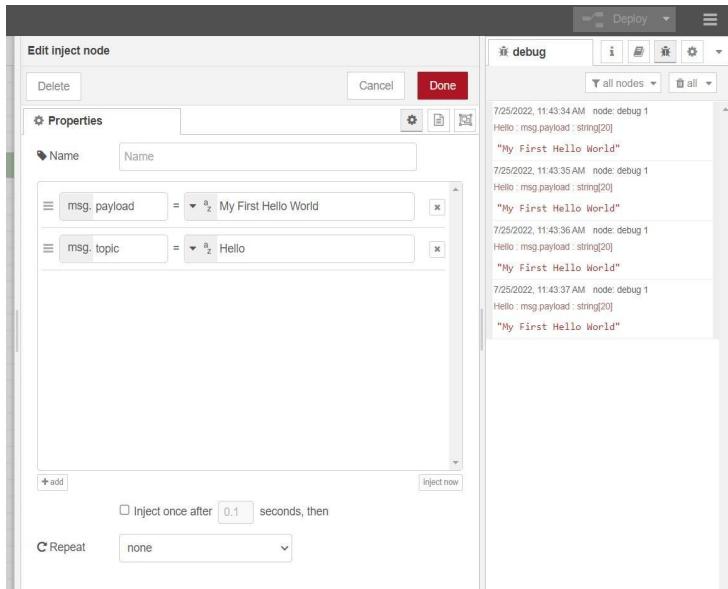
Select one Inject and one Debug Node:



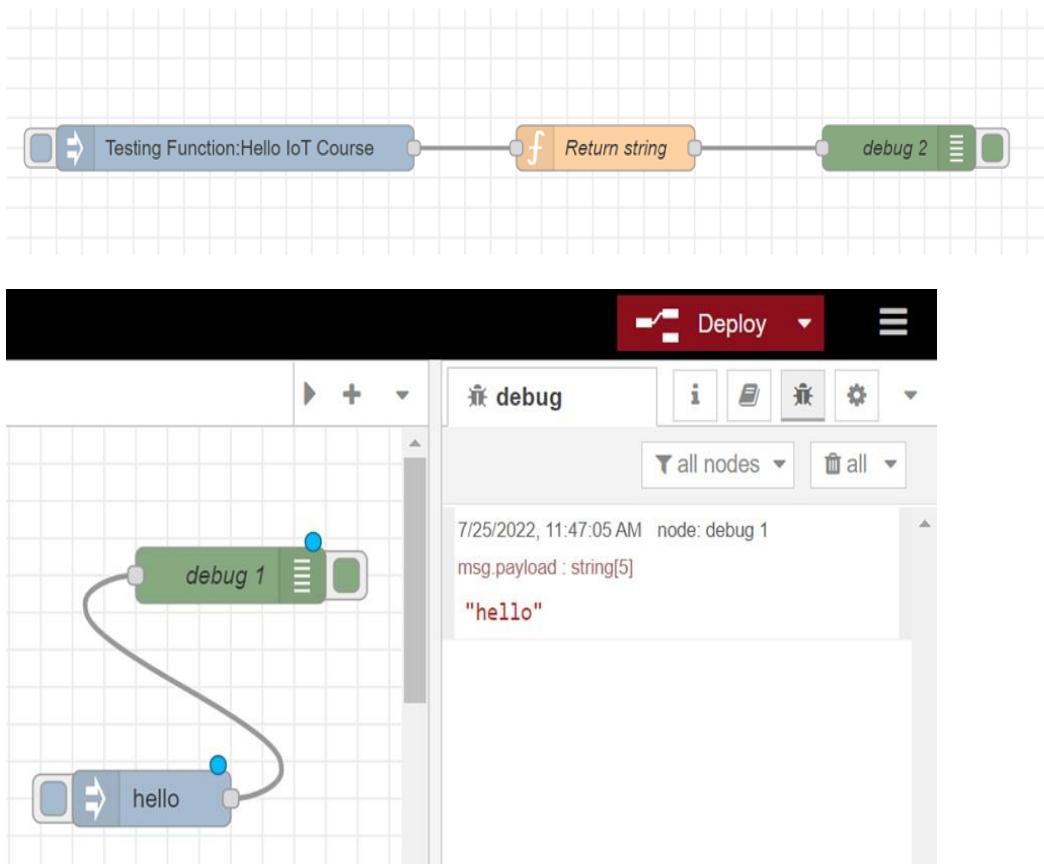
Select payload in Debug:



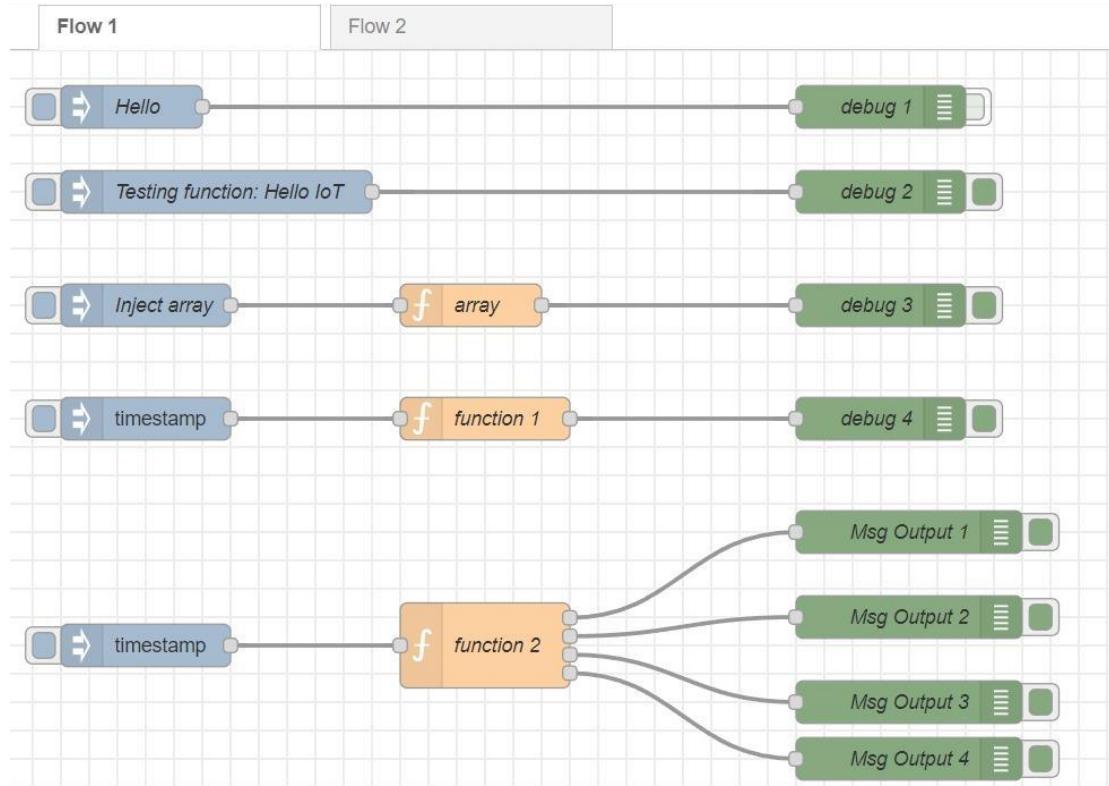
In inject write msg payload and topic:  
Deploy and run:



Function in node-red:



Generate Random numbers:



---

7/29/2022, 10:19:38 PM node: debug 9

msg.payload : number

**2.4509020525988556**

---

7/29/2022, 10:19:38 PM node: debug 9

msg.payload : number

**0.0037134059370935724**

---

7/29/2022, 10:19:52 PM node: debug 9

msg.payload : number

**2.756009661456671**

---

7/29/2022, 10:19:52 PM node: debug 9

msg.payload : number

**4.381563918409041**

---

7/29/2022, 10:19:52 PM node: debug 9

msg.payload : number

**2.919006925600851**

---

7/29/2022, 10:19:53 PM node: debug 9

msg.payload : number

**3.1149702760624454**

---

7/29/2022, 10:19:53 PM node: debug 9

msg.payload : number

**4.7133856903726095**

---

7/29/2022, 10:19:53 PM node: debug 9

msg.payload : number

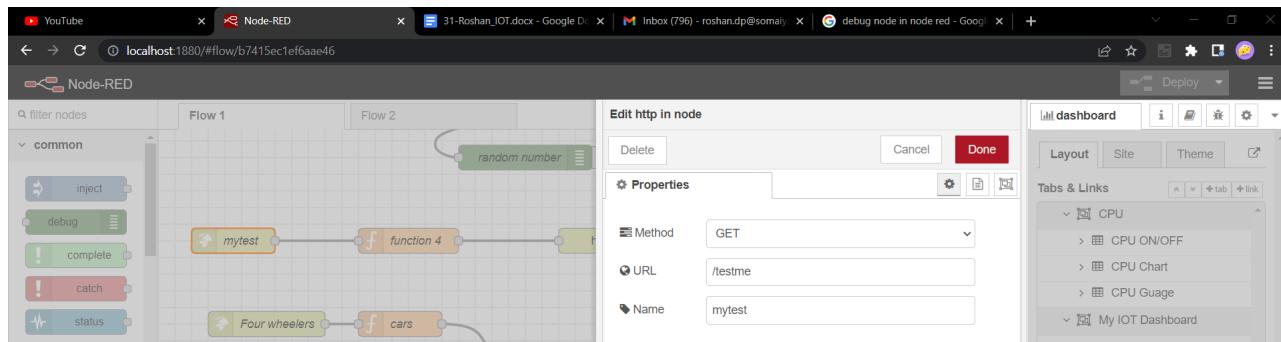
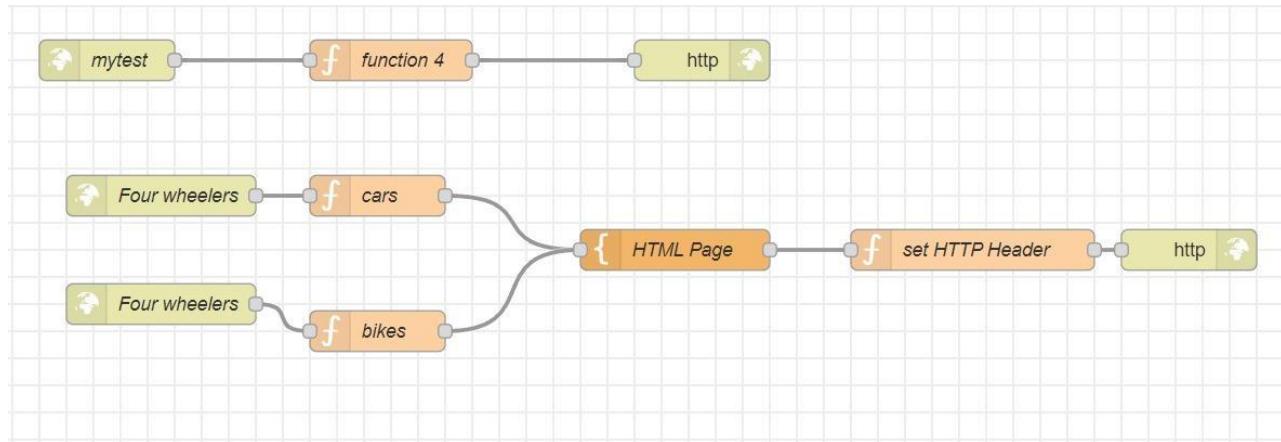
**5.564814477155191**

---

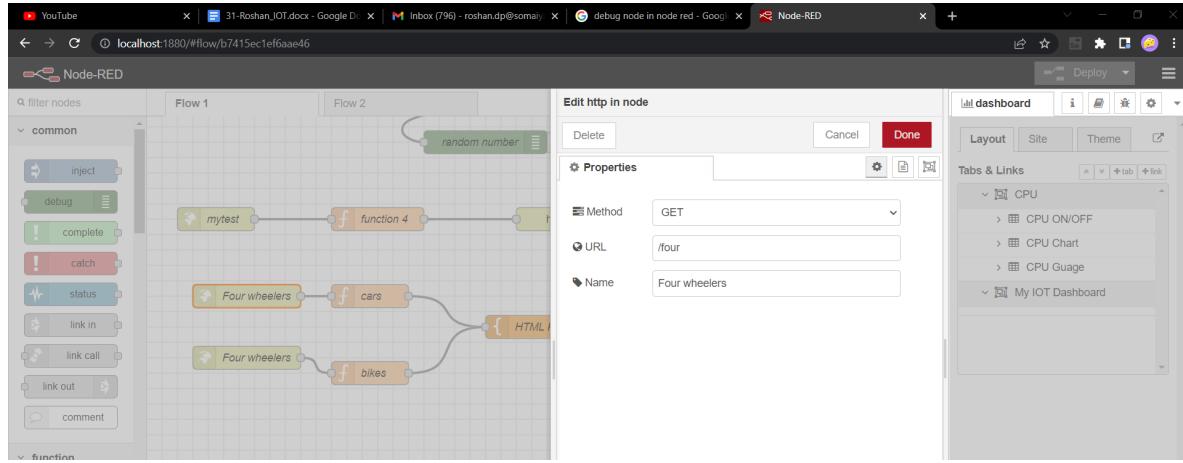
# Practical No. 05

**Practical Name:** Random number generator with selection of color.  
**Introducing the HTTP node.** Points to be covered in the following order:

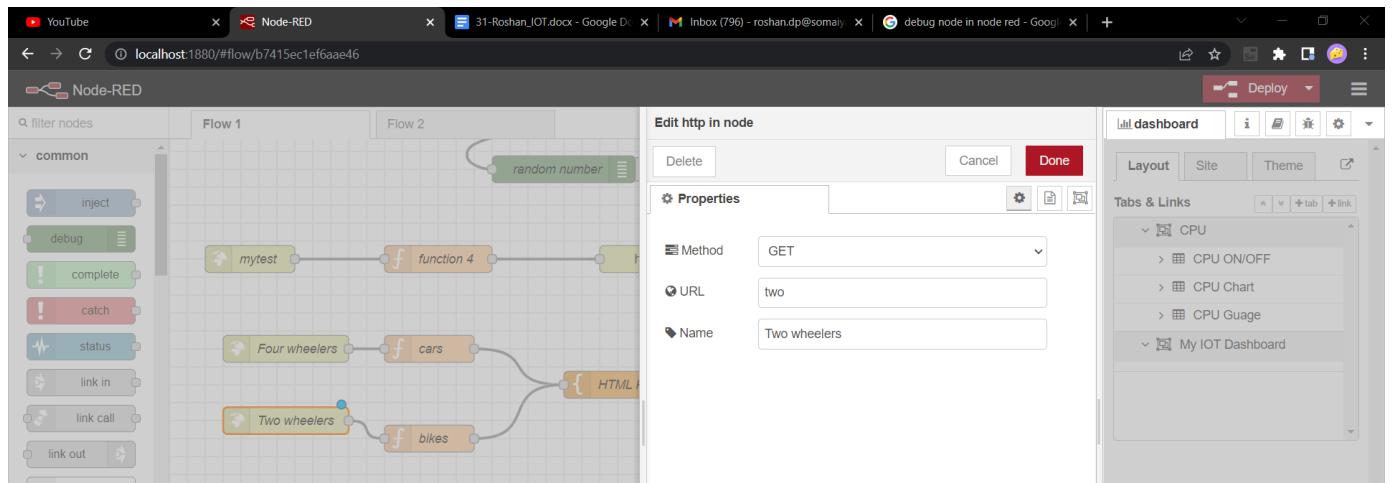
- This below flow shows the combination of http input and http output nodes.



- In first http input node i.e Four wheelers put url as four:

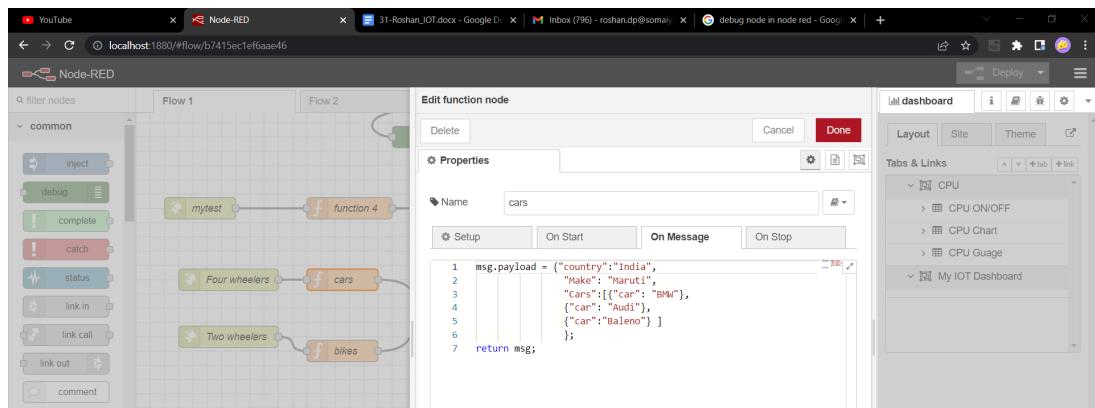


- In second http input node i.e Two wheelers put url as two:



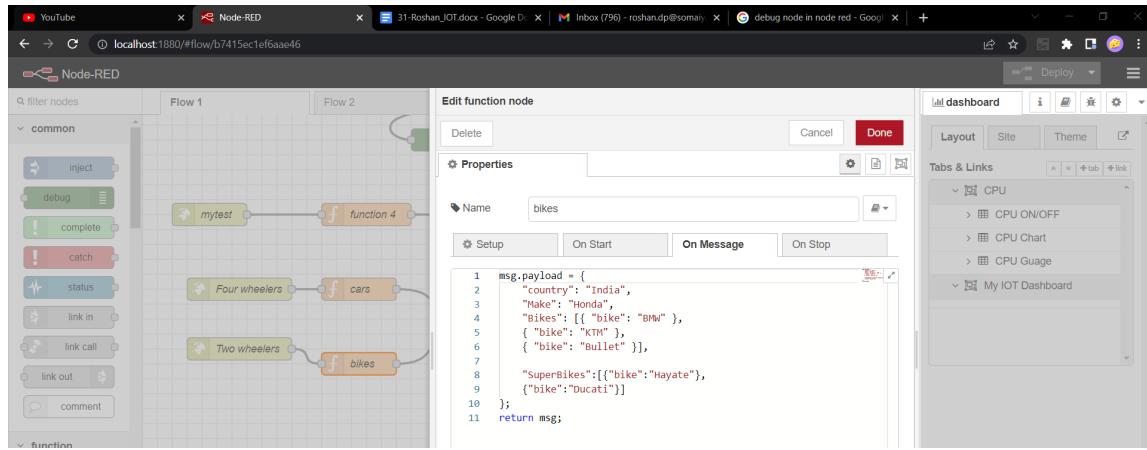
- In the first function, give the name of the cars and write a message as below.

We have created an array of objects which contains car names.



- To the second function, give name as bikes and write a message as below.

We have created an array of objects which contains bike names.



- In the html node we will write the format or design of the webpage.

HTML nodes used a mustache template.

Write below code in template:

<h1>Cars and Bikes available in {{payload.Country}}</h1>

<ul>

  {{#payload.Cars}}

    <li>{{car}}</li>

  {{/payload.Cars}}

</ul>

<h1>

  Two Wheelers

</h1>

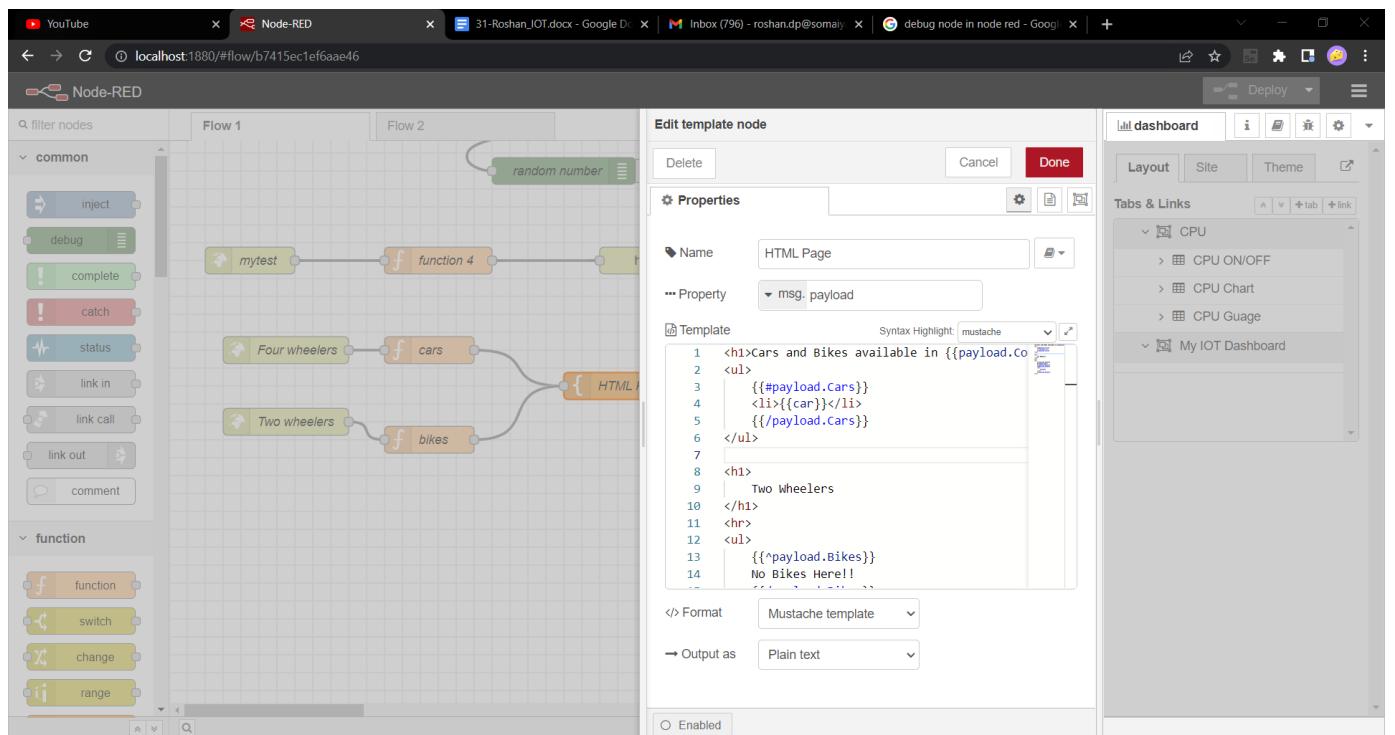
<hr>

<ul>

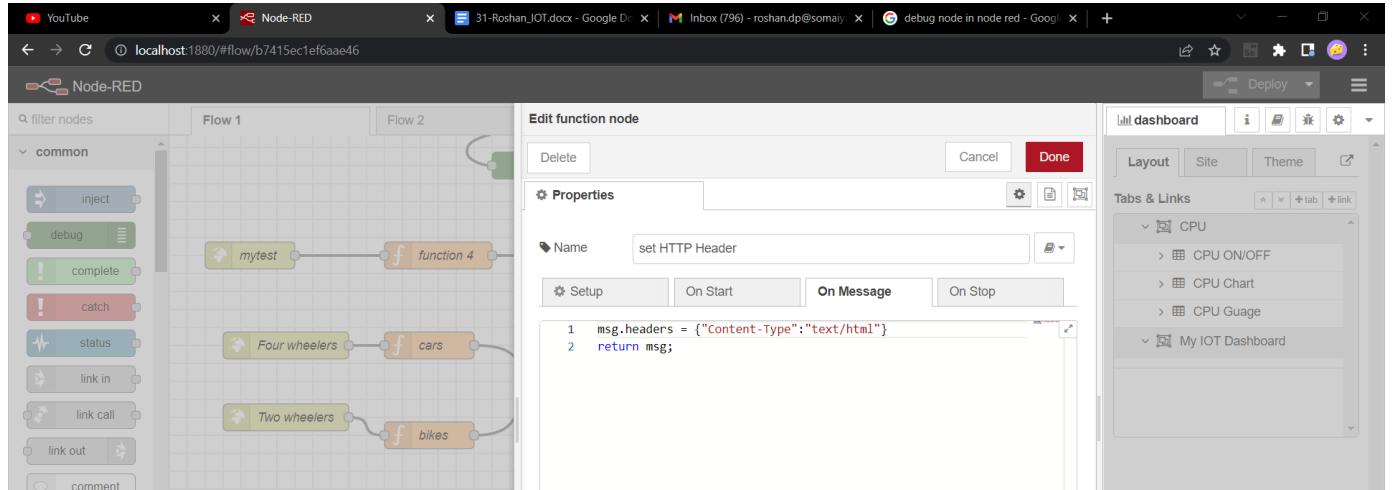
  {{^payload.Bikes}}

    No Bikes Here!!

```
{{/payload.Bikes}}  
{{#payload.Bikes}}  
<li>  
{{bike}}  
</li>  
{{/payload.Bikes}}  
</ul>
```



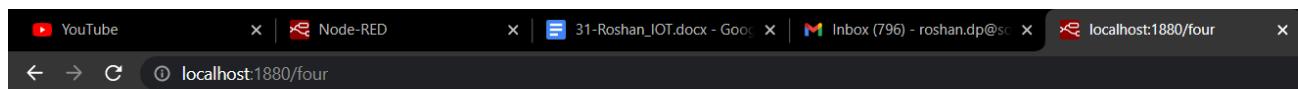
- In the set http header node we will give msg type.



- In google chrome:
- Write localhost:1880/testme



- Write **http://localhost:1880/four** in google and we will get the details of four wheelers:



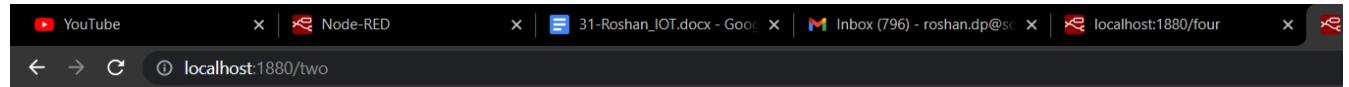
## Cars and Bikes available in

- BMW
- Audi
- Baleno

## Two Wheelers

No Bikes Here!!

- Write **http://localhost:1880/two** in google and we will get the details of two wheelers:



## Cars and Bikes available in Two Wheelers

---

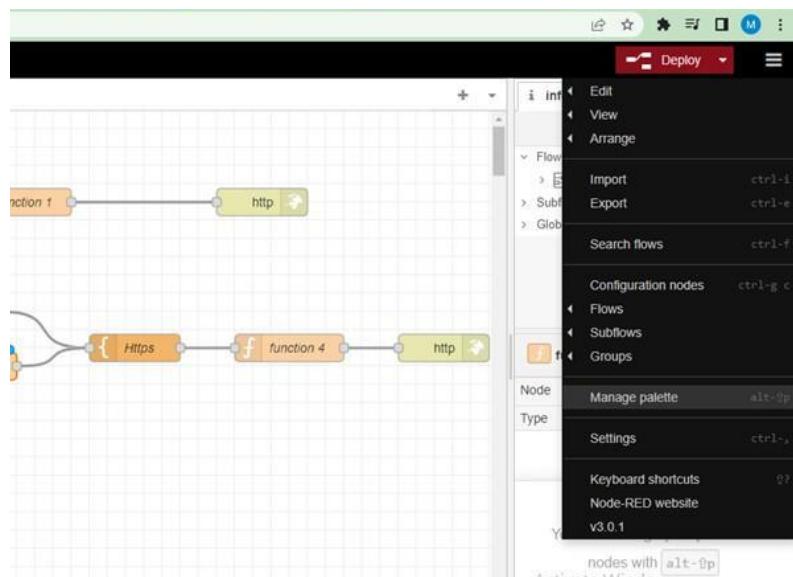
- BMW
- KTM
- Bullet

# Practical No. 06

**Practical Name: CPU utilization flow Points to be covered in the following order:**

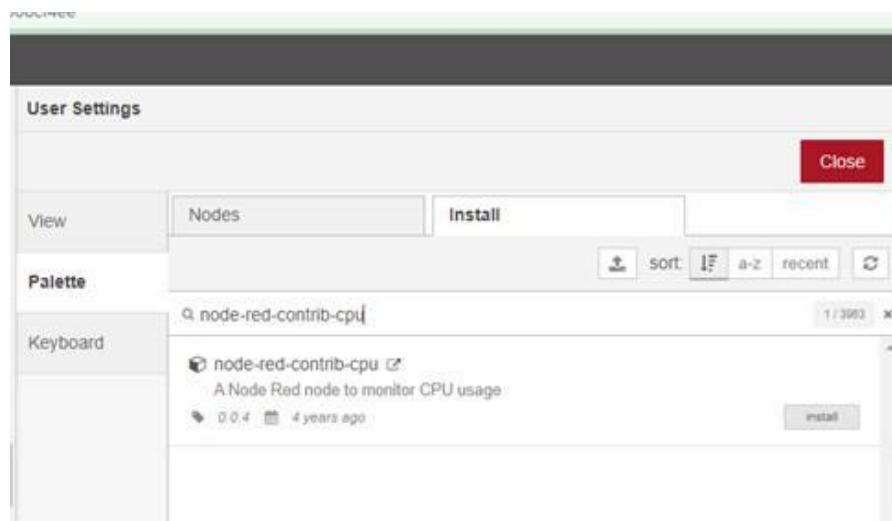
- Installation of node-red-dashboard and node-red-contrib-cpu nodes from the Manage pallet option of the menu.(Explain the steps involved in the installation with screenshots)

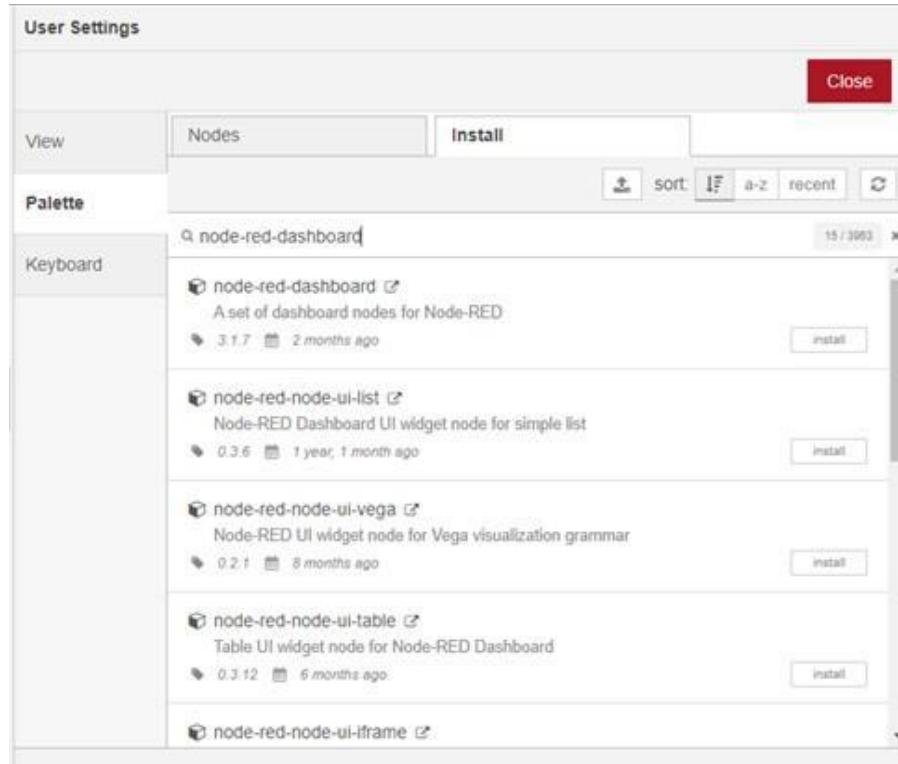
Open Manage Palette:



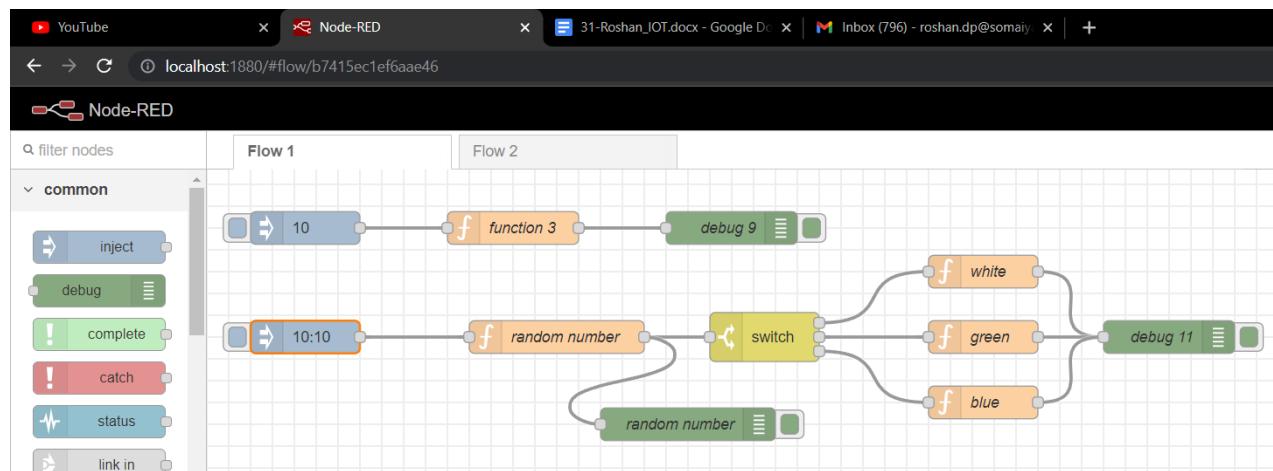
Click on Install and install 2 packages:

1. node-red-contrib-cpu
2. node-red-dashboard

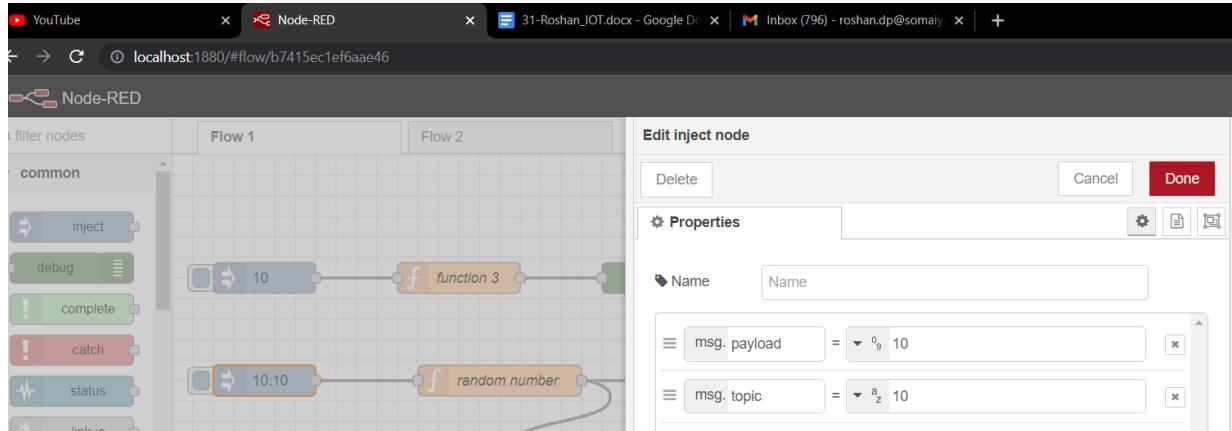




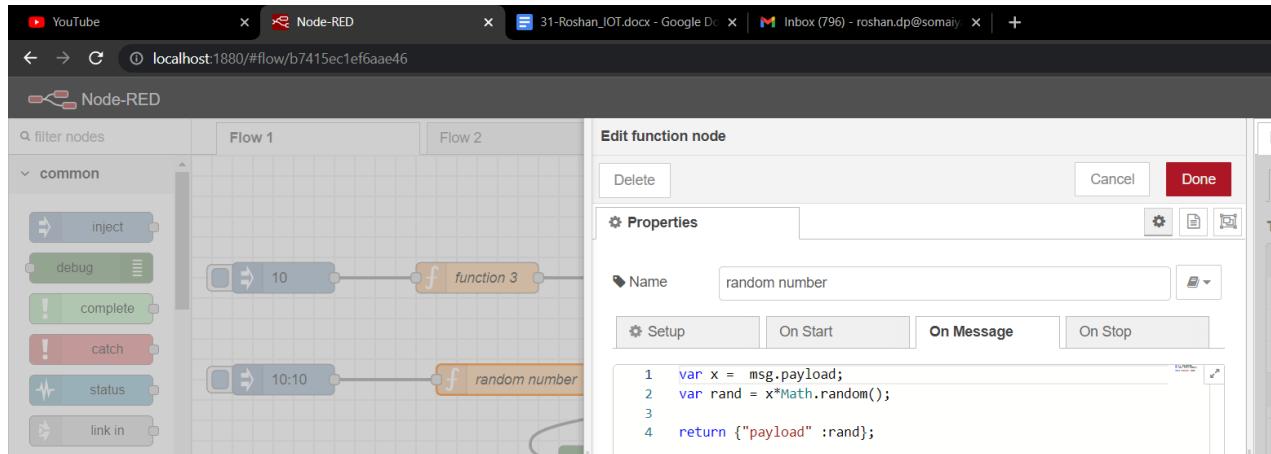
- Now, create following flow for getting colors:



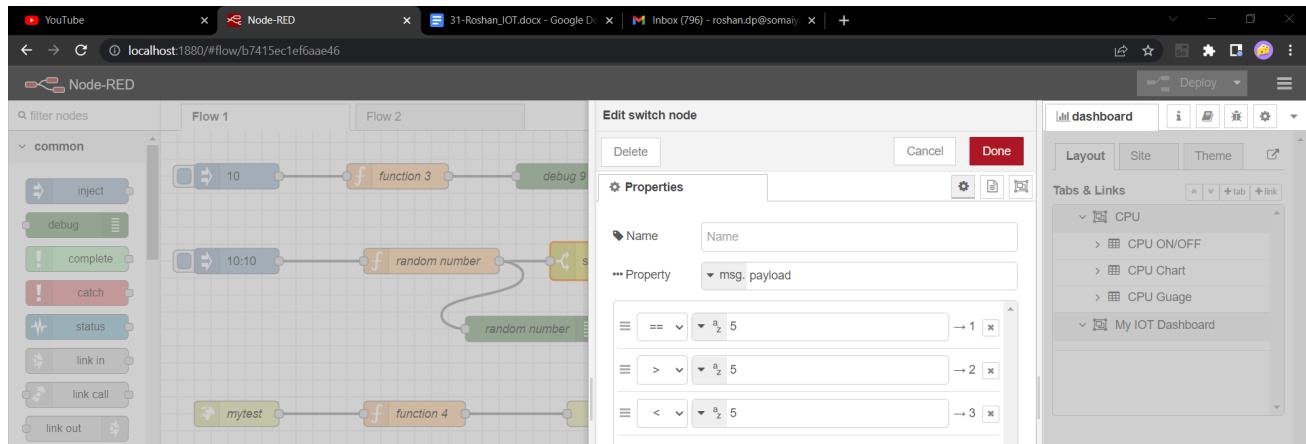
- In the inject node give payload as 10 and give datatype string and topic as 10 and give datatype string:



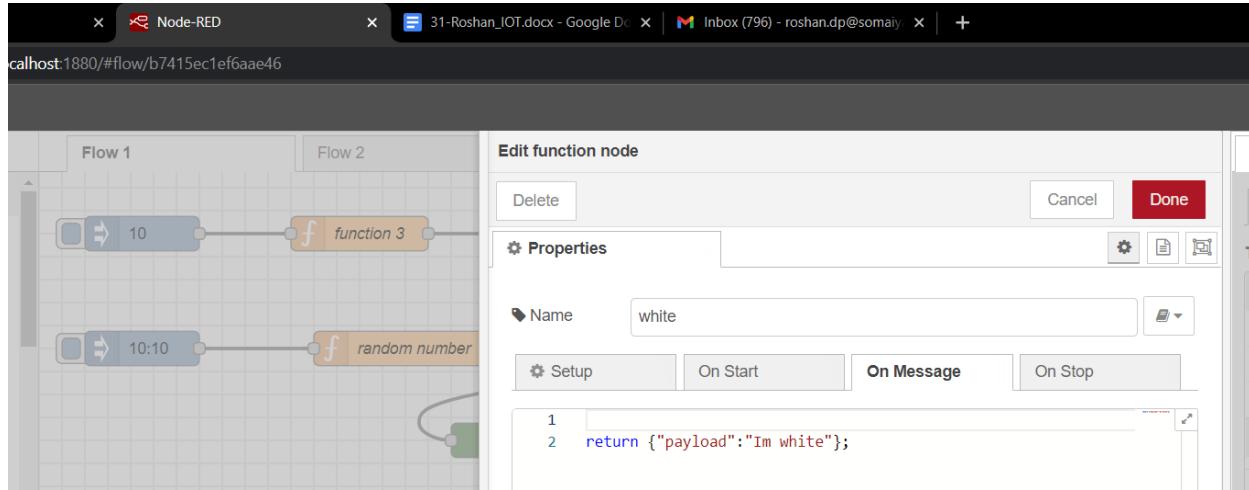
- In the function node write the following function:



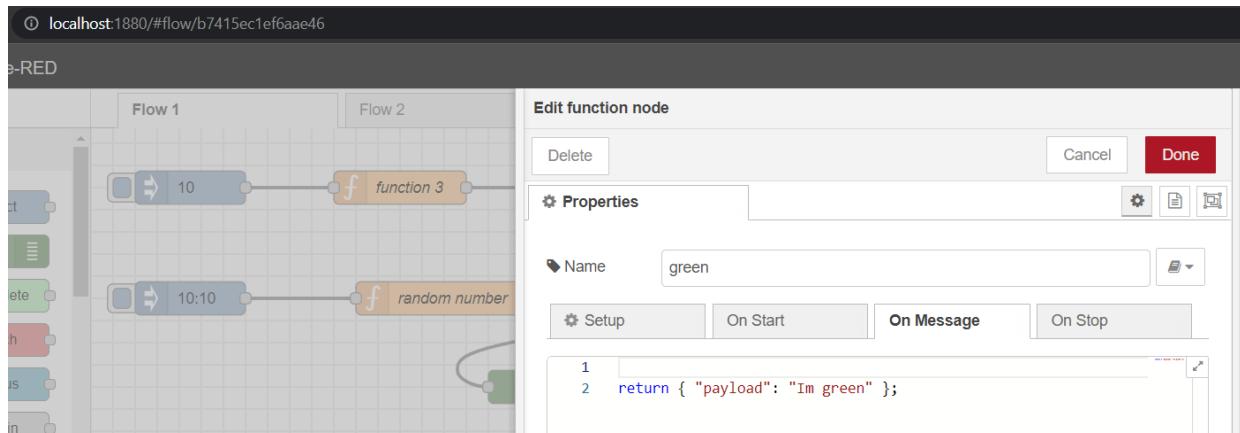
- In switch node give following conditions:



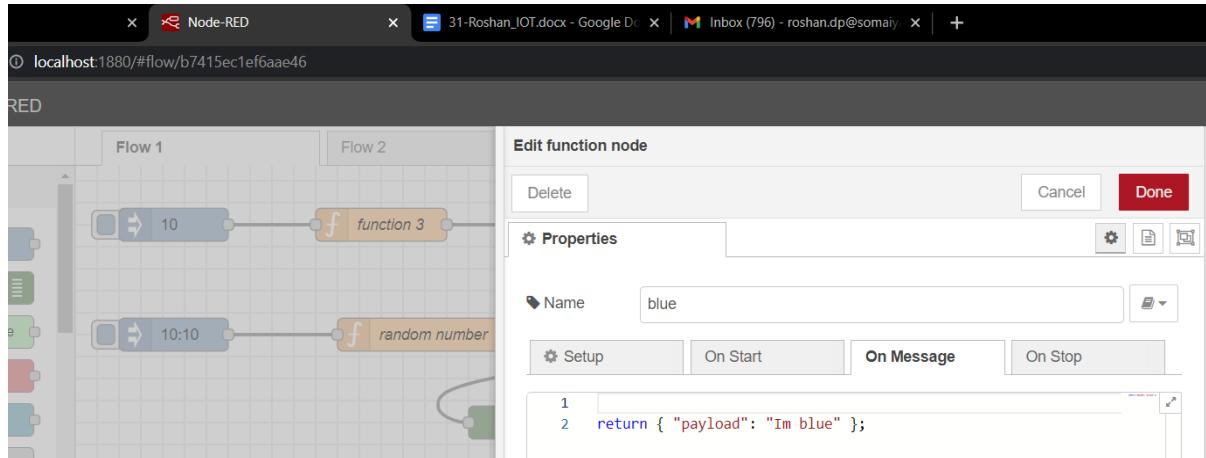
- In the first function return the msg as White. i.e if voltage==5 then give output as 'Im white'.



- In the second function return the msg as Green. i.e if voltage>5 then give output as 'Im Green'.



- In the third function return the msg as Blue. i.e if voltage<5 then give output as 'Im Blue'.



- Now, according to the voltage the output message in the debug section will change.
- If the voltage = 5, then the white function will execute.
  - If the voltage > 5, then the green function will execute.
  - If the voltage < 5, then the blue function will execute.

```

7/29/2022, 10:20:29 PM node: random number
msg.payload : number
4.101779011031259

7/29/2022, 10:20:29 PM node: debug 11
msg.payload : string[7]
"Im blue"

7/29/2022, 10:20:30 PM node: random number
msg.payload : number
3.0984742117198283

7/29/2022, 10:20:30 PM node: debug 11
msg.payload : string[7]
"Im blue"

7/29/2022, 10:20:31 PM node: random number
msg.payload : number
1.112289699573339

7/29/2022, 10:20:31 PM node: debug 11
msg.payload : string[7]
"Im blue"

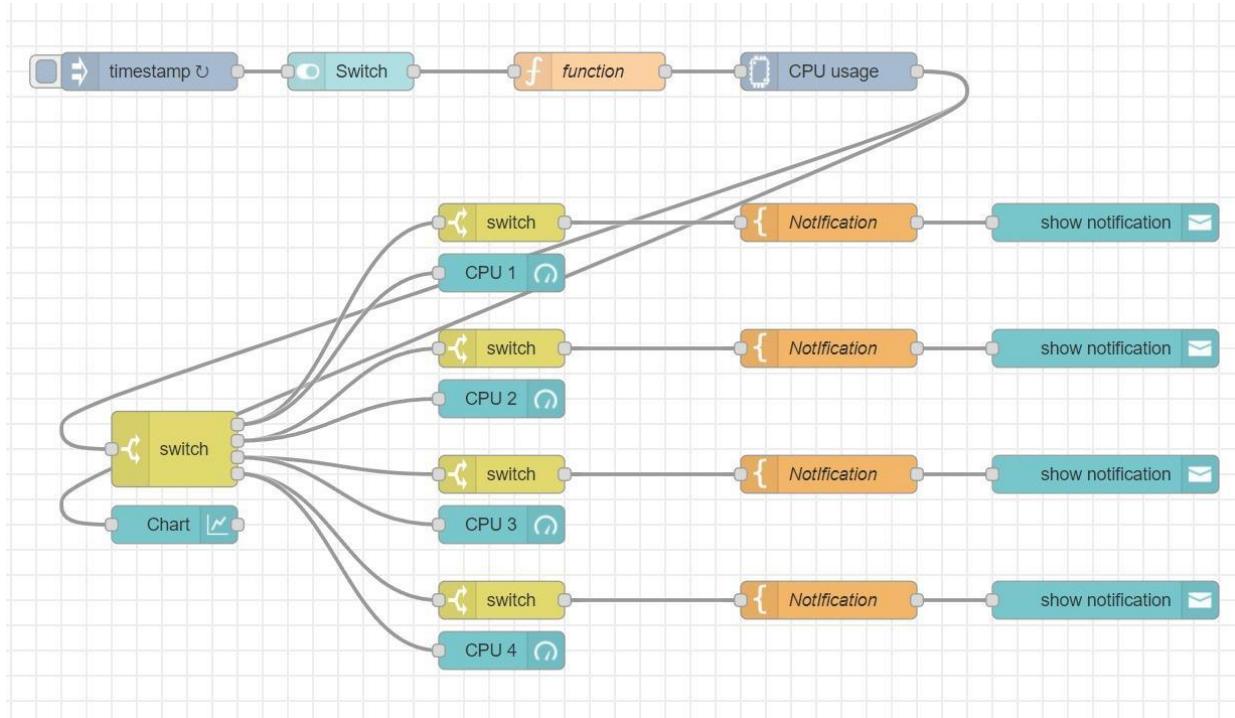
7/29/2022, 10:20:32 PM node: random number
msg.payload : number
9.590091289321641

7/29/2022, 10:20:32 PM node: debug 11
msg.payload : string[8]
"Im green"

```

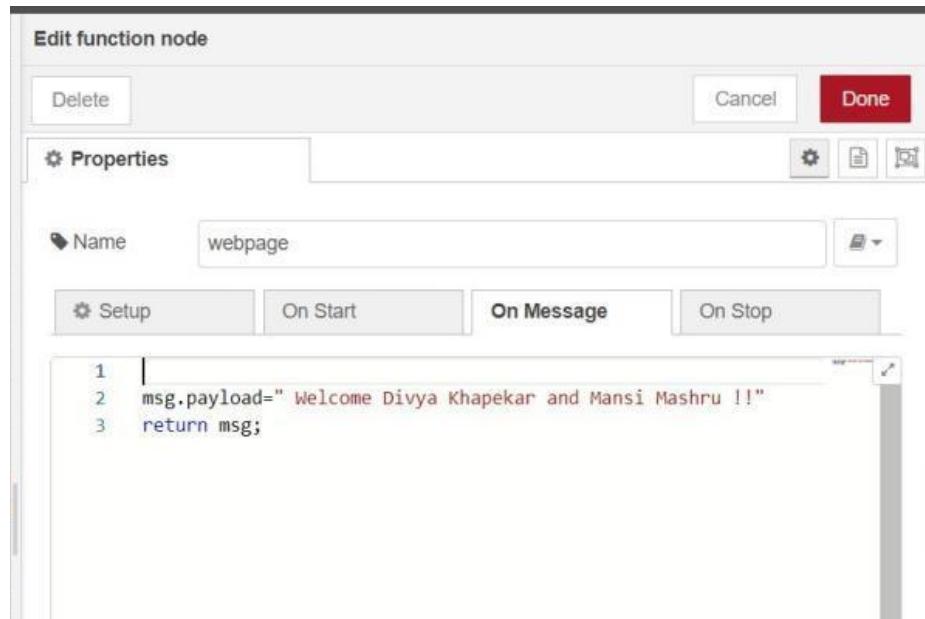
## ● Create the dashboard

Create following flow:



- Click on dashboard => Click on layout => Click on + and add dashboard

In function node:



- Create dashboard Groups:

**Edit dashboard tab node**

**Properties**

**Name:** CPU

**Icon:** dashboard

**State:** Enabled

**Nav. Menu:** Visible

The **Icon** field can be either a Material Design icon (e.g. 'check', 'close') or a Font Awesome icon (e.g. 'fa-fire'), or a Weather icon (e.g. 'wi-wu-sunny'). You can use the full set of google material icons if you add 'mi-' to the icon name, e.g. 'mi-video-game\_asset'.

- CPU on/off group:

**Edit dashboard group node**

**Properties**

**Name:** CPU on/off

**Tab:** CPU

**Class:** Optional CSS class name(s) for widget

**Width:** 6

Display group name

Allow group to be collapsed

Edit dashboard group node

**Properties**

Name: CPU guage

Tab: CPU

Class: Optional CSS class name(s) for widget

Width: 6

Display group name

Allow group to be collapsed

- Add group in switch node

Edit switch node

**Properties**

Group: Add new dashboard group... [CPU] CPU on/off

Size: [CPU] CPU chart  
[CPU] CPU guage  
[CPU] Group 4  
Add new dashboard group...

Label:

Tooltip: optional tooltip

Icon: Default

Pass through msg if payload matches valid state.

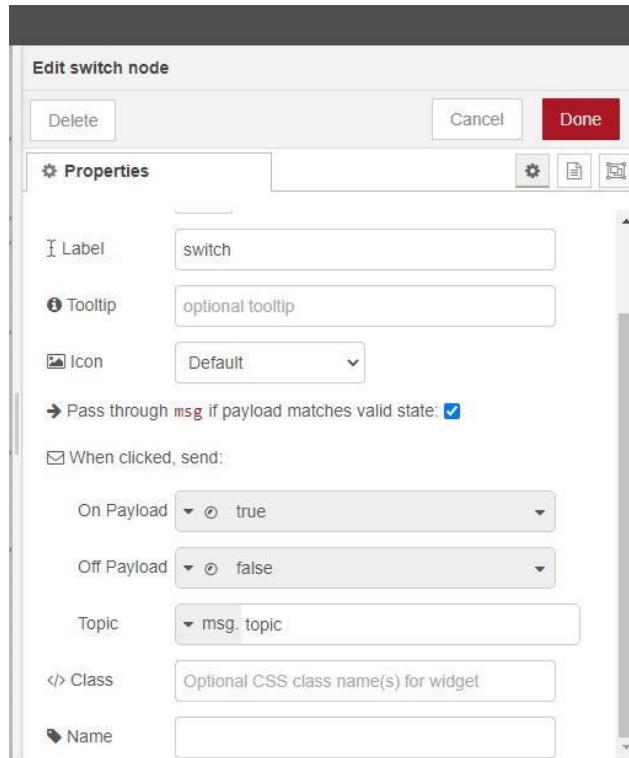
When clicked, send:

On Payload: true

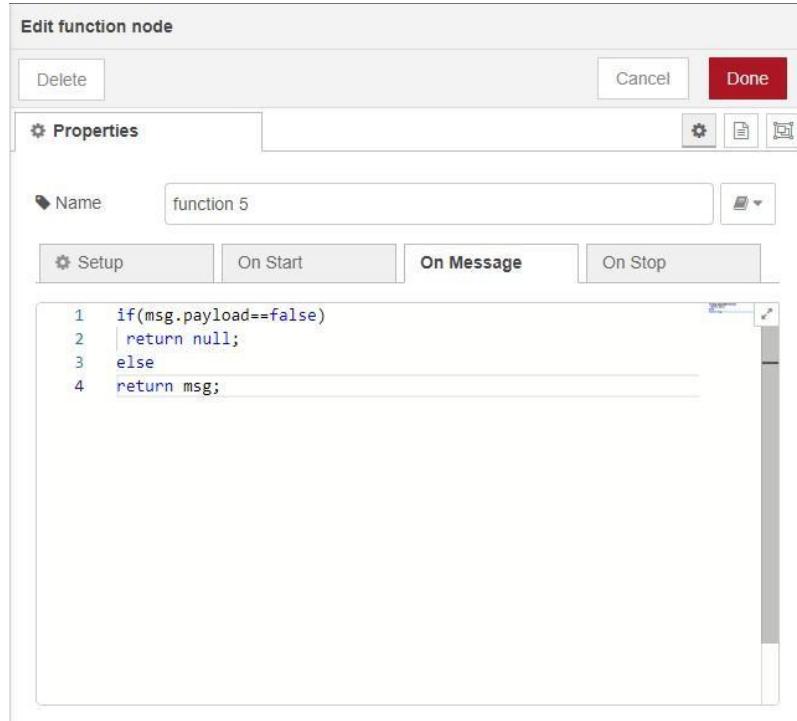
Off Payload: false

Topic: msg. topic

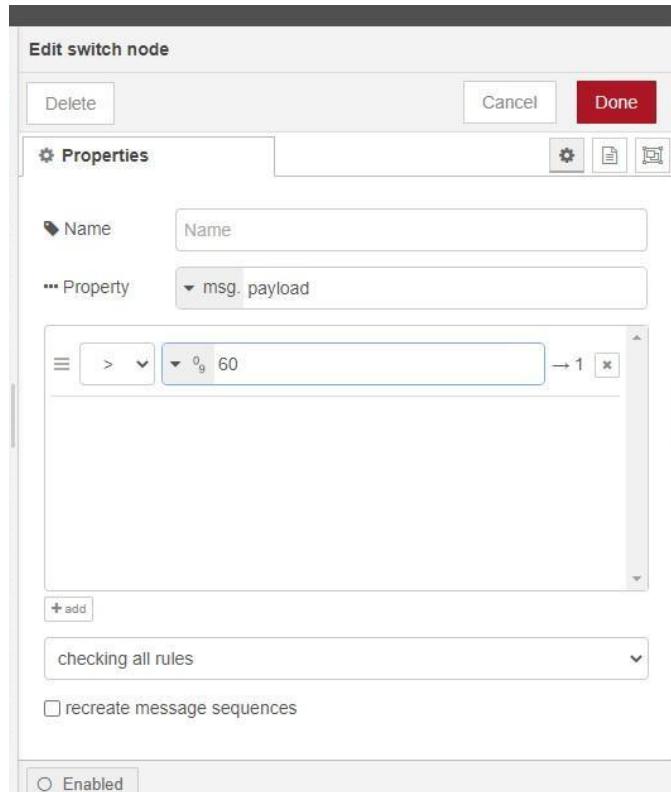
- Edit switch node as below:



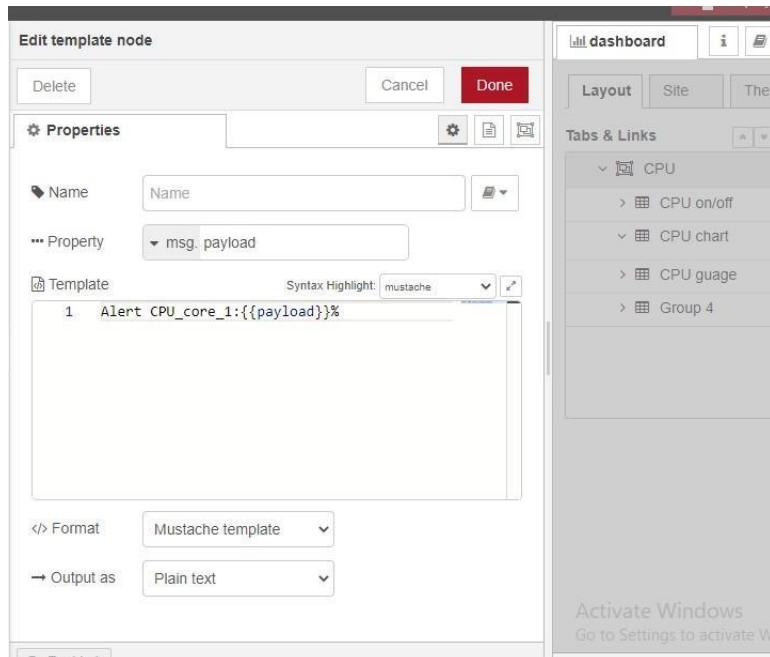
- Write function in function node:



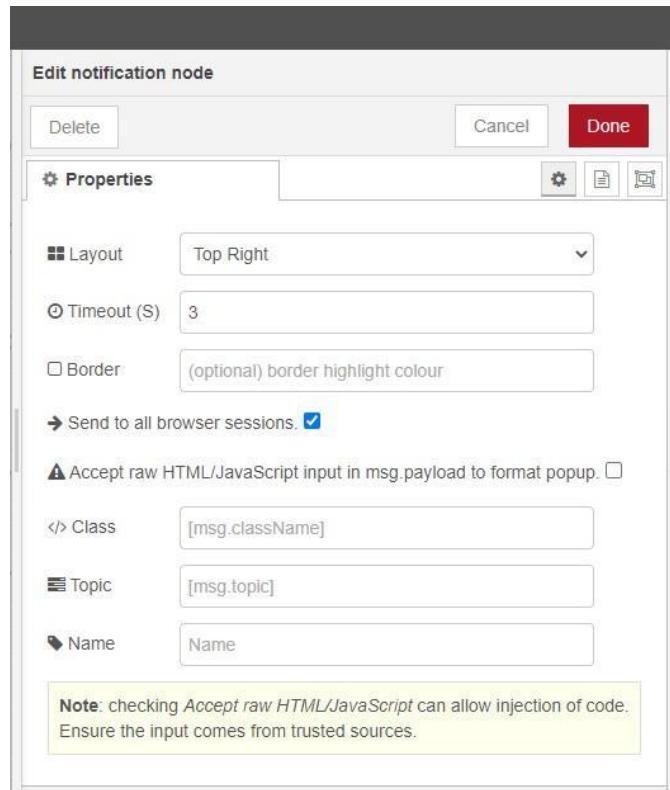
- In switch node add condition: >60



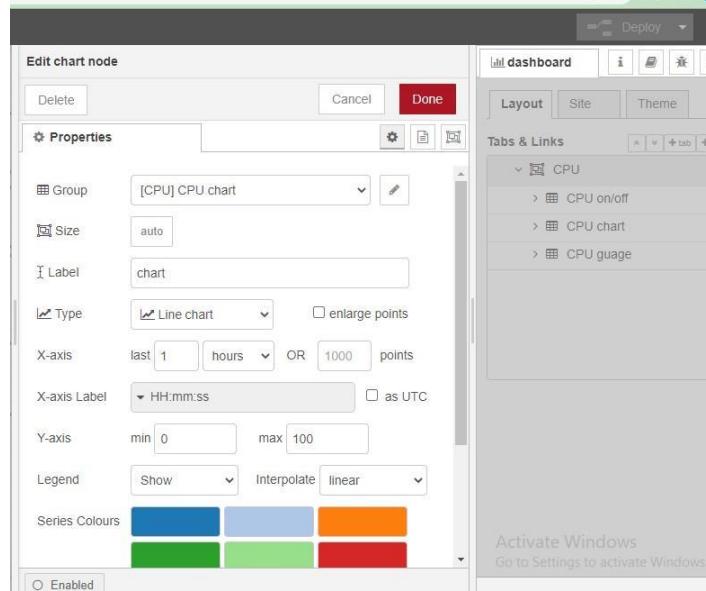
- In template node:



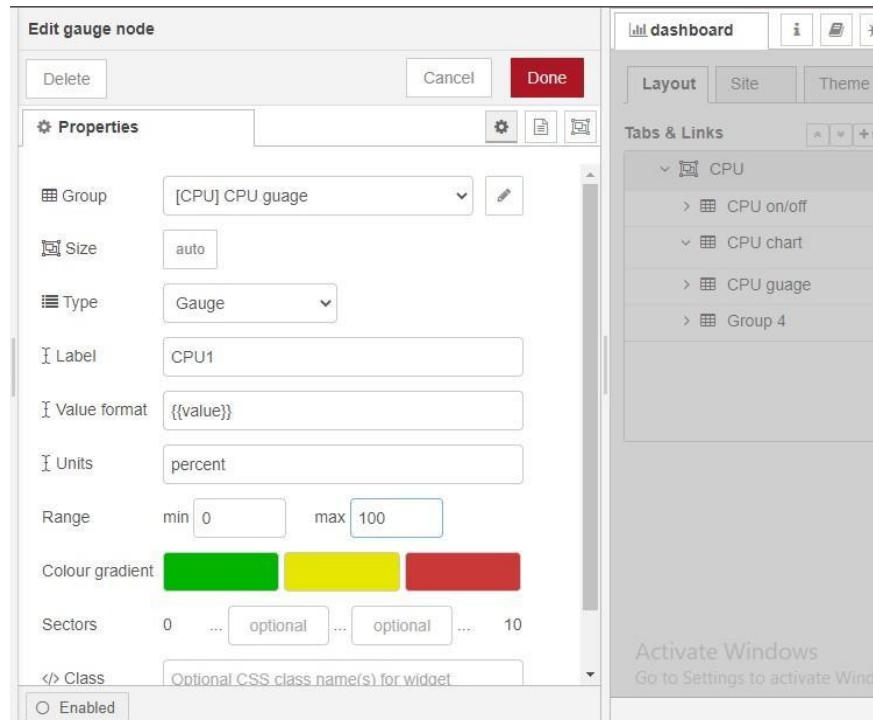
- In notification node:



- Edit chart node as:



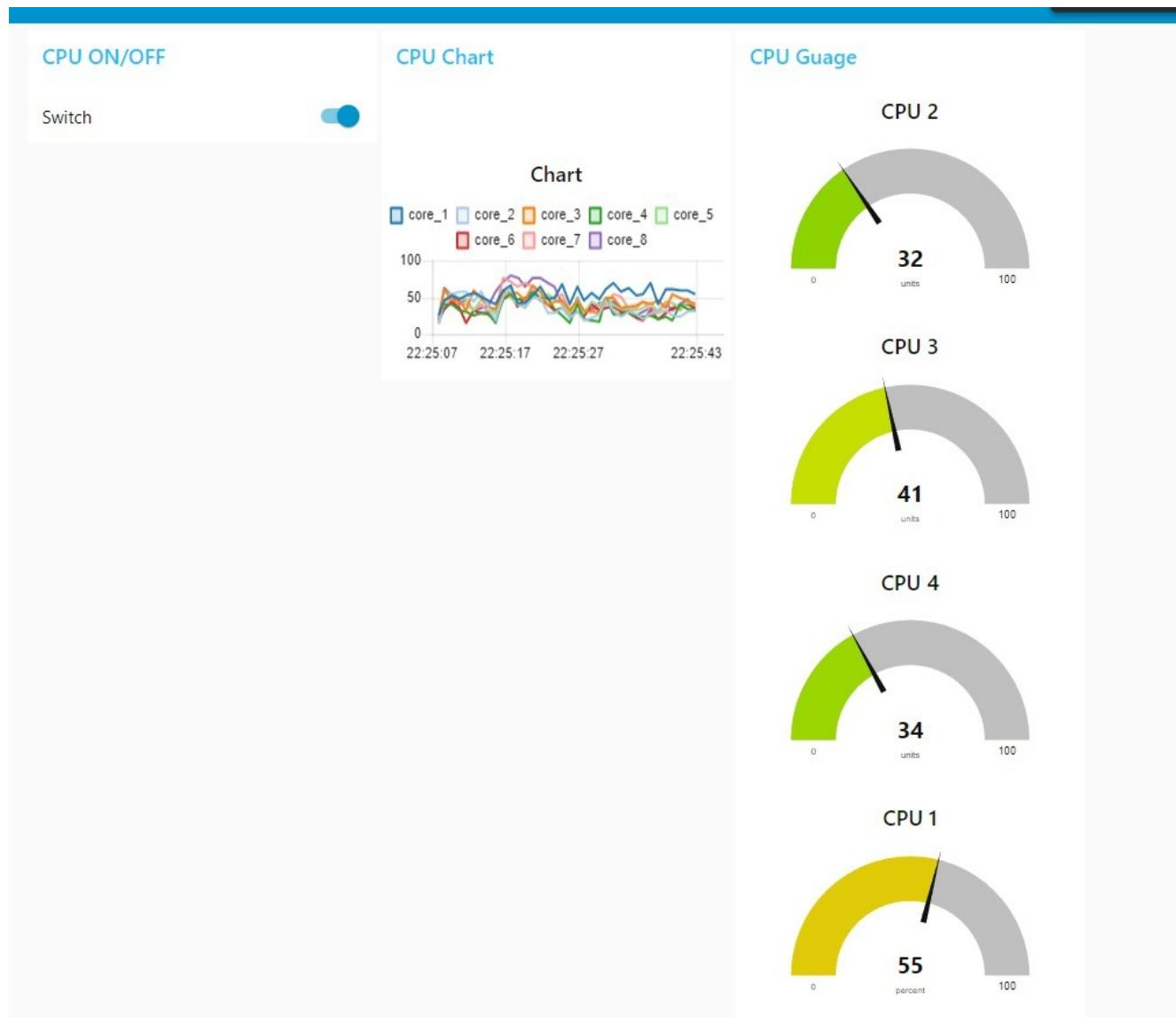
- Edit gauge node as below:



### Output:

In the dashboard we will have a switch, a chart, and 4 CPUs.

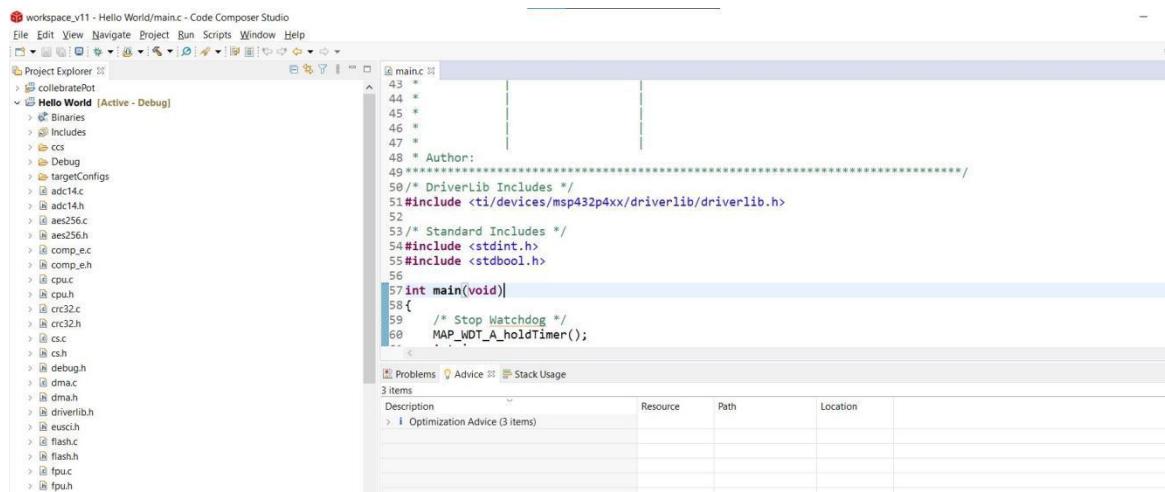
1. To start the dashboard we need to turn on the switch.
2. Chart will show the information i.e voltage about the cores.
3. Gauges represent CPUs in the dashboard. When the power consumption is low the colour will be yellow and it will increase gradiently upto red when power consumption goes to high.
4. On high power consumption we will get the alert message at top left corner.



## Practical No. 07

## Practical Name: WiFi Setup

- Create project by: Projects => import CCS project => Open project from driverlib  
Rename the project



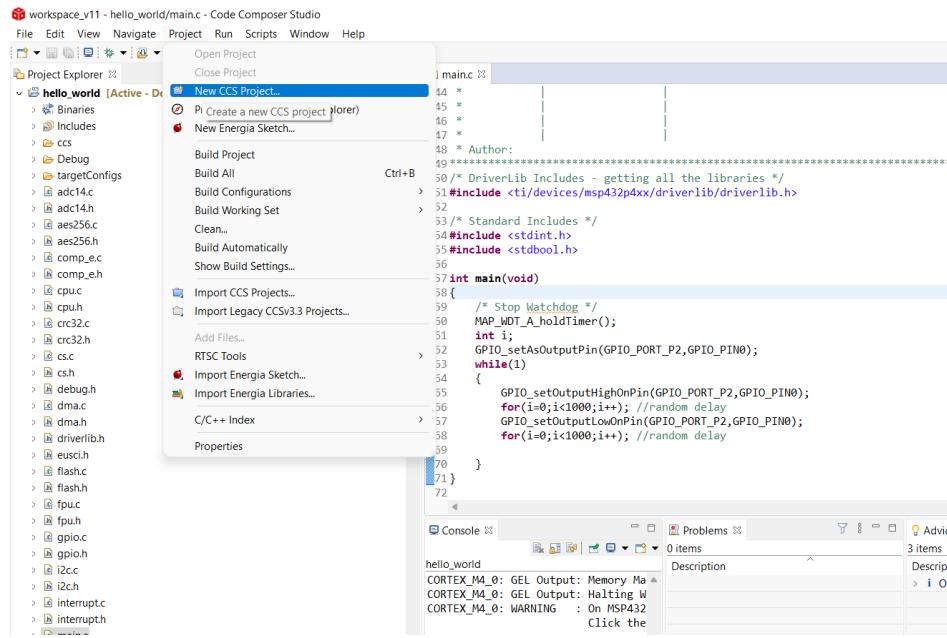
```
#include <ti/devices/msp432p4xx/driverlib/driverlib.h>

/* Standard Includes */
#include <stdint.h>
#include <stdbool.h>

int main(void)
{
    /* Stop Watchdog */
    MAP_WDT_A_holdTimer();
    int i;
    GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0);
    while(1)
    {
        GPIO_setOutputHighOnPin(GPIO_PORT_P2,GPIO_PIN0);
        for(i=0;i<10000;i++); //random delay
        GPIO_setOutputLowOnPin(GPIO_PORT_P2,GPIO_PIN0);
        for(i=0;i<10000;i++); //random delay
    }
}
```

- Now, include libraries in energia and import energia sketch in CCS by:

Projects => import energia sketch => and write following code for wifi setup.



### WiFi\_test.ino

```
#include<WiFi.h>
#include<SPI.h>

IPAddress shieldIP, subnetMask, gatewayIP;
uint8_t rssi;
uint8_t networkId;
byte macAddr[6]; //mac address is 6 bytes
byte encryptionType;

char ssid[]="rp";//my cell phone

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    Serial.print("Connecting to WiFi..");
    while(WiFi.begin(ssid)!=WL_CONNECTED)          //for      cell      phone      with      password:-
while(WiFi.begin(ssid,password)!=WL_CONNECTED)

{
    Serial.print(".");
    delay(1);
}
```

```

Serial.println("");
Serial.print("WiFi connected , Fetching WiFi shield's IP Address:");
while(WiFi.localIP()==INADDR_NONE ) {
    Serial.print(".");
    delay(1);
}
shieldIP = WiFi.localIP();
Serial.println(shieldIP);

Serial.print("Access point name :");
Serial.println(ssid);

Serial.print("Signal strength: ");
rssi=WiFi.RSSI();
Serial.println(rssi);

uint8_t networkId = WiFi.scanNetworks();
Serial.print("Number of access points in range:");
Serial.println(networkId);
for(int i=1;i<=networkId;i++){
    Serial.print("Name of Access Points and encryption type:");
    Serial.print(WiFi.SSID(i));
    Serial.print(",");
    encryptionType = WiFi.encryptionType(i);
    //
    Serial.println(encryptionType,DEC);
}

subnetMask = WiFi.subnetMask();
Serial.print("Subnet Mask:");
Serial.println(subnetMask);

gatewayIP = WiFi.gatewayIP();
Serial.print("Gateway IP address:");
Serial.println(gatewayIP);

WiFi.macAddress(macAddr);
Serial.print("Mac Address of shield:");
for(int i =0;i<6;i++){
    Serial.print(macAddr[i],HEX);
    if(i<=4)
        Serial.print(":");
    else
        Serial.println();
}

void loop() {
// put your main code here, to run repeatedly:
}

```

- Build the project:

The screenshot shows the Code Composer Studio interface. The main window displays the `main.c` file with code for connecting to a WiFi network. Below the code editor is the `Console` tab, which shows build logs. The logs indicate successful compilation with no errors or warnings.

```

1 #include<WiFi.h>
2 #include<SPI.h>
3
4
5 //Define SSID "Redmi0A"
6 //Define PASSWORD "12345678"
7 IPAddress shieldIP,subnetMask,gatewayIP;
8 uint8_t rssi;
9 uint8_t networkId;
10 byte macAddr[6]; //mac address is 6 bytes
11 byte encryptionType;
12
13 char ssid[]="Redmi"; //my cell phone
14
15
16 void setup() {
17 // put your setup code here, to run once:
18 Serial.begin(115200);
19 Serial.print("Connecting to WiFi..");
20 while(WiFi.begin(ssid)!=WL_CONNECTED) //for cell phone with password:- while(WiFi.begin(ssid,password)!=WL_CONNECTED)
21 {
22     Serial.print(".");
23     delay(1);
24 }
25 Serial.println("");
26 Serial.print("WiFi connected , Fetching WiFi shield's IP Address:");
27 while(WiFi.localIP()==INADDR_NONE ) {
28     Serial.print(".");
29 }

```

**Console**

```

wifi_test: 0 errors, 0 warning, 0 others
0 items
Description Description Resource Path
CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
CORTEX_M4_0: GEL Output: Halting Watchdog Timer
CORTEX_M4_0: WARNING : On MSP432P401R hitting a breakpoint cannot be detected by the debugger when the device is in low power mode.
Click the

```

- After building without errors, Debug the project.

The screenshot shows the Code Composer Studio interface during a debug session. The top navigation bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, Help, and a Debug icon. The left sidebar shows the project structure under `wifi_test`. The bottom pane displays the `Variables`, `Expressions`, and `Registers` windows, all currently empty. The code editor at the bottom shows the same `main.c` file as the previous screenshot.

```

File Edit View Project Tools Run Scripts Window Help
Debug
wifi_test [Code Composer Studio - Device Debugging]
Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Running)

Variables Expressions Registers
Name Type Value Location

```

```

1 #include<WiFi.h>
2 #include<SPI.h>
3
4
5 //Define SSID "Redmi0A"
6 //Define PASSWORD "12345678"
7 IPAddress shieldIP,subnetMask,gatewayIP;
8 uint8_t rssi;
9 uint8_t networkId;
10 byte macAddr[6]; //mac address is 6 bytes
11 byte encryptionType;
12
13 char ssid[]="Redmi"; //my cell phone
14
15
16 void setup() {
17 // put your setup code here, to run once:
18 Serial.begin(115200);
19 Serial.print("Connecting to WiFi..");
20 while(WiFi.begin(ssid)!=WL_CONNECTED) //for cell phone with password:- while(WiFi.begin(ssid,password)!=WL_CONNECTED)
21 {
22     Serial.print(".");
23     delay(1);
24 }
25 Serial.println("");
26

```

**Console**

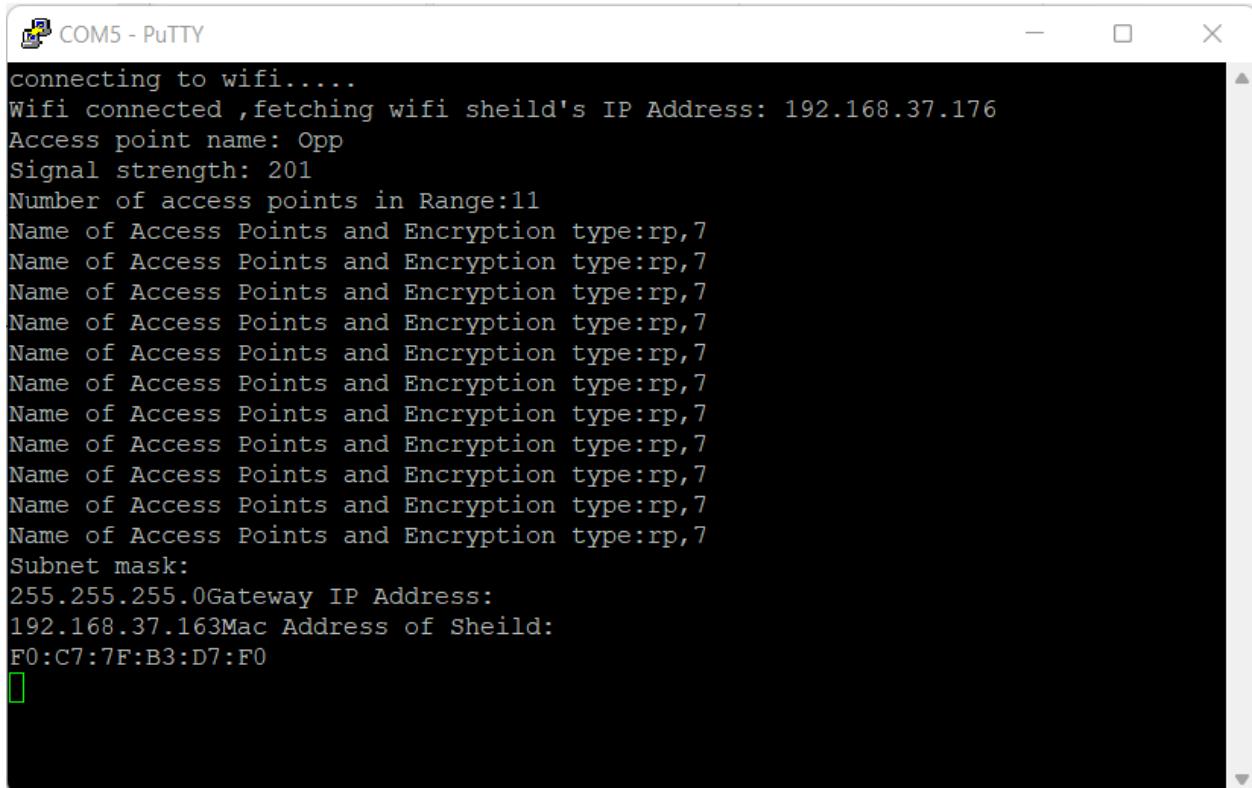
```

wifi_test
CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
CORTEX_M4_0: GEL Output: Halting Watchdog Timer
CORTEX_M4_0: WARNING : On MSP432P401R hitting a breakpoint cannot be detected by the debugger when the device is in low power mode.
Click the pause button during debug to check if the device is held at the breakpoint.

```

- Open putty and do configuration and open the Terminal:  
now, Resume the project then we will be connected to the wifi:

We can see Access point name, Signal strength, Number of access points in range, Subnet Mask, Gateway IP Address, Mac Address of Shield.



The screenshot shows a PuTTY terminal window with the title "COM5 - PuTTY". The terminal output displays the following information:

```
connecting to wifi.....  
Wifi connected ,fetching wifi sheild's IP Address: 192.168.37.176  
Access point name: Opp  
Signal strength: 201  
Number of access points in Range:11  
Name of Access Points and Encryption type:rp,7  
Subnet mask:  
255.255.255.0Gateway IP Address:  
192.168.37.163Mac Address of Sheild:  
F0:C7:7F:B3:D7:F0
```

# Practical No. 08

## **Practical Name: Analog To Digital Converter**

This practical is all about converting the analog signal into digital signal.

The conversion is based on voltage provided by the potentiometer.

If the voltage is too high i.e 3.3 the color of LED will be white.

at low voltage it will be red coloured as soon as we increase the voltage the colour will be green and then blue.

Step 1: Create a project in Energia

Add following code and save:

```
#include <SLFS.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiServer.h>
#include <WiFiUdp.h>

void setup() {
}

void loop() {
```

Now, import energia project in CCS:

Write the following code and save the project.

Code:

```
#include <ti/devices/msp432p4xx/driverlib/driverlib.h>
#include <SLFS.h>
#include <WiFi.h>
```

```
#include <WiFiClient.h>
#include <WiFiServer.h>
#include <WiFiUdp.h>

IPAddress shieldIP,subnetMask,gatewayIP;
uint8_t rssi;
uint8_t networkId;
byte macAddr[6];
byte encryptionType;

char ssid[]="rp";

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    Serial.print("Connecting to Wifi...");
    while(WiFi.begin(ssid)!=WL_CONNECTED) // cell phone
    {
        Serial.print(".");
        delay(1); // delay in ms
    }
    Serial.println("");
    Serial.println("WiFi connected , Fetching WiFi shield's IP Address:");
    while(WiFi.localIP()==INADDR_NONE){
        Serial.print(".");
        delay(1);
    }
    shieldIP = WiFi.localIP();
    Serial.println(shieldIP);
    Serial.print("Access Point Name : ");
}
```

```

Serial.println(ssid);
Serial.print("Signal Strength");
rssI=WiFi.RSSI();
Serial.println(rssi); // RSSI - Received Signal Strength Indication.
uint8_t networkId = WiFi.scanNetworks();
Serial.print("Number of access points in Range :");
Serial.println(networkId);
for(int i=1; i<=networkId; i++){
    Serial.print("Name of Access Points and Encryption type :");
    Serial.print(WiFi.SSID(i));
    Serial.print(",");
    encryptionType = WiFi.encryptionType(i);
    // Serial.println(encryptionType,HEX);
    Serial.println(encryptionType,DEC);
}

```

```

subnetMask = WiFi.subnetMask();
Serial.print("Subnet Mask");
Serial.println(subnetMask);
gatewayIP = WiFi.gatewayIP();
Serial.print("Gateway IP Address:");
Serial.println(gatewayIP);

```

```

WiFi.macAddress(macAddr);
Serial.print("Mac Address of Shield :");
for(int i = 0; i<6; i++){
    Serial.print(macAddr[i],HEX);
    if(i<=4)
        Serial.print(':');
    else

```

```

    Serial.println();
}

GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0|GPIO_PIN1|GPIO_PIN2);

GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P4,GPIO_PIN7,GPIO_TERTIARY_MODULE_FUNCTION);

ADC14_initModule(ADC_CLOCKSOURCE_MCLK,ADC_PREDIVIDER_1,ADC_PREDIVIDER_1,ADC_NOROUTE);

    ADC14_configureSingleSampleMode(ADC_MEM6, true);

ADC14_configureConversionMemory(ADC_MEM6,ADC_VREFPOS_AVCC_VREFNEG_VS
S,ADC_INPUT_A6,false);

    ADC14_setSampleHoldTime(ADC_PULSE_WIDTH_32,ADC_PULSE_WIDTH_4);

    ADC14_setResolution(ADC_12BIT);

    ADC14_enableSampleTimer(ADC_AUTOMATIC_ITERATION);

    ADC14_enableModule();

    ADC14_enableConversion();

    ADC14_toggleConversionTrigger();

}

void loop() {
    // put your main code here, to run repeatedly:
    int result,regressionData1;
    float regressionData;

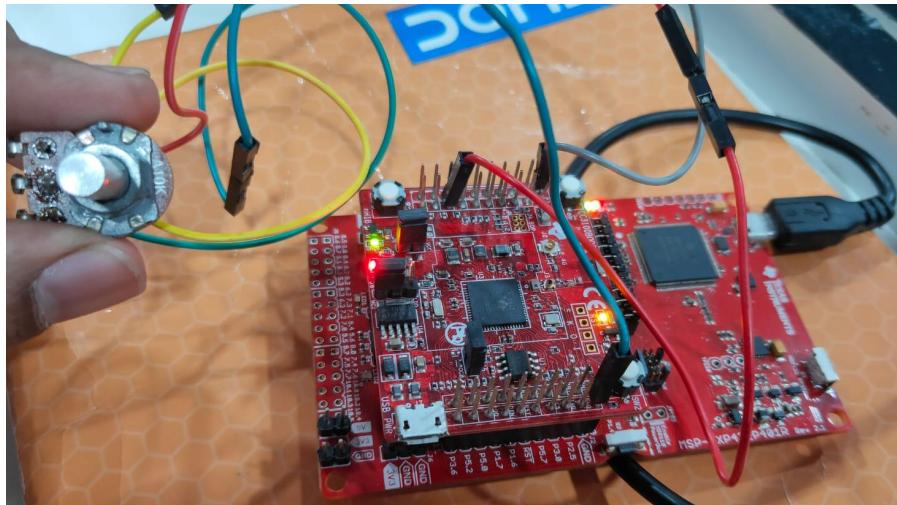
    while(!ADC14_isBusy());
    result = ADC14_getResult(ADC_MEM6);
    P2OUT = result>>8;
    Serial.println(result);
    delay(500);
    ADC14_toggleConversionTrigger();
}

```

{

Output:

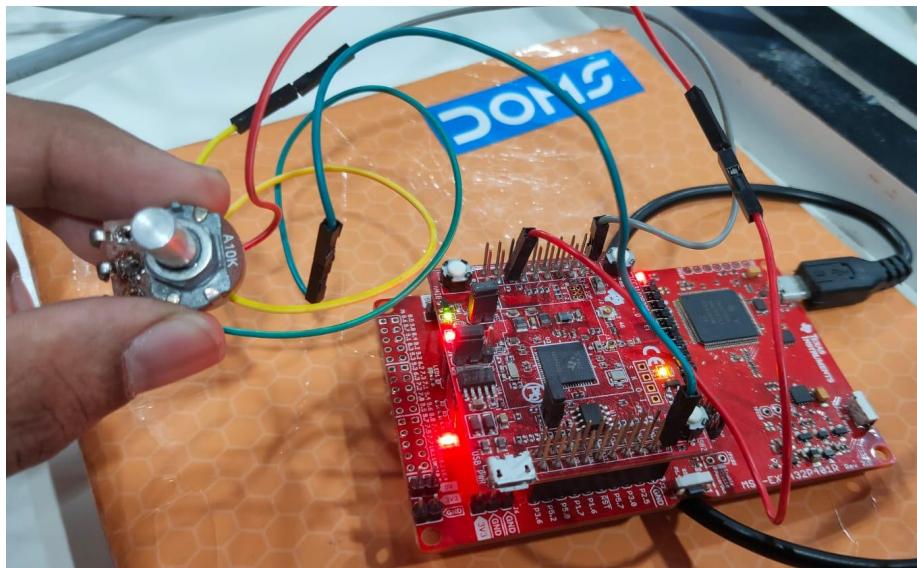
Before debugging the project:



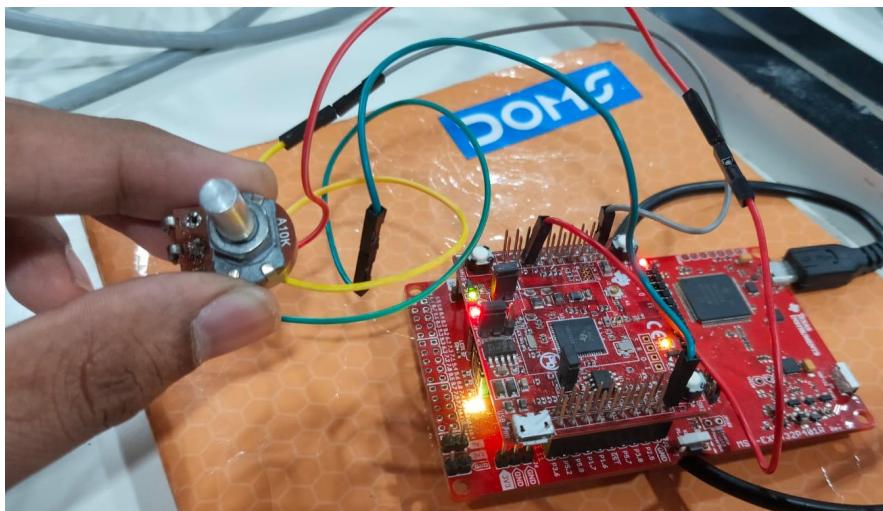
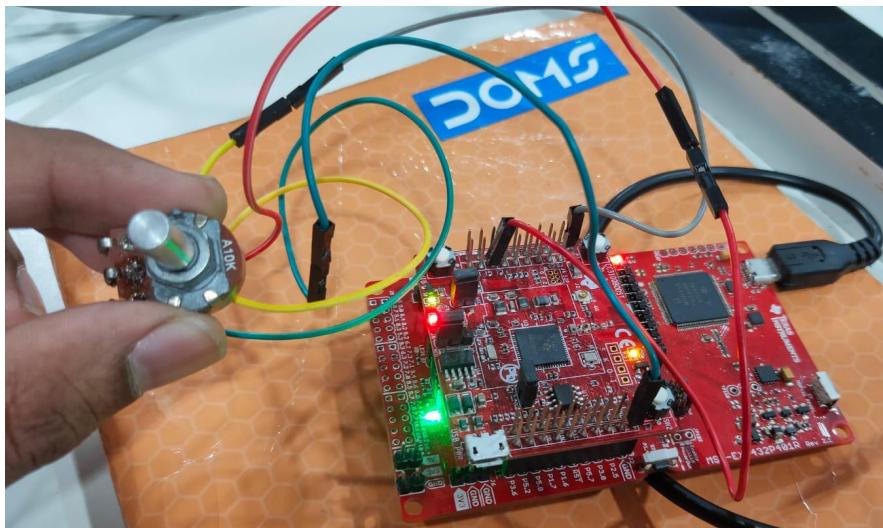
After debugging:

Now, the output or colour of the LED will depend upon the voltage provided by the potentiometer.

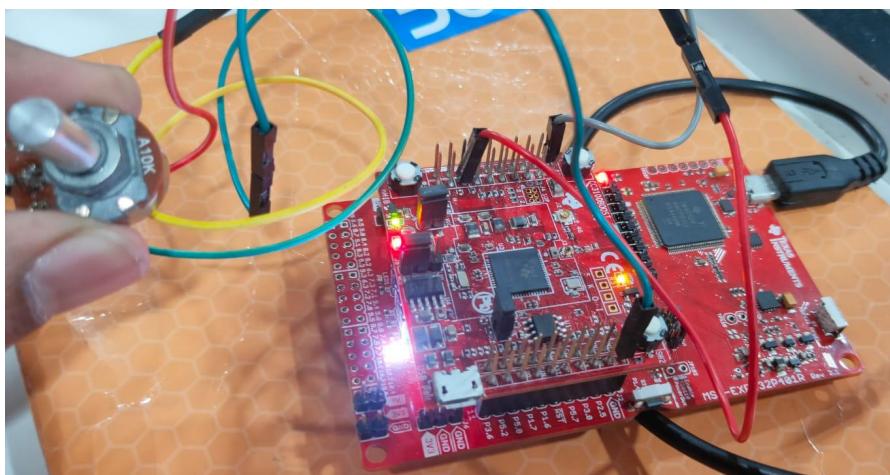
When the voltage provided by potentiometer is low the color of LED will be red.



When the voltage provided by potentiometer increases the color of LED will change accordingly.



When the voltage provided by potentiometer is too high the colour of LED will be white.



# Practical No. 09

## Practical Name: Sending Raw data to the cloud

In this practical we will send our node red data to the cloud and we will perform further operations on the cloud itself.

We will send our data to the cloud. While sending the data to the cloud a protocol is used that is MQTT.

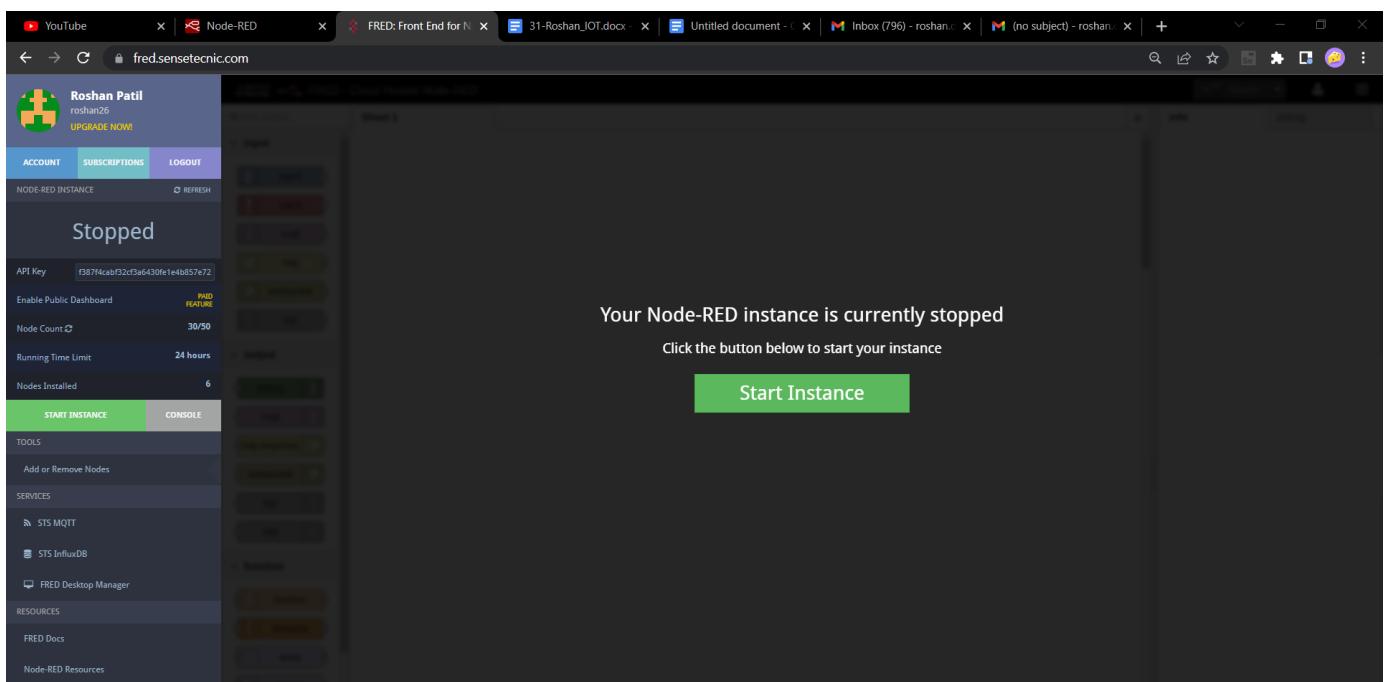
MQTT stands for Message Queuing Telemetry Transport. It works as a broker between publisher and the subscriber.

Publisher publishes the message and the subscriber can see the message by subscribing to it.

To start with the practical first open <https://fred.sensetecnic.com/> in the browser.

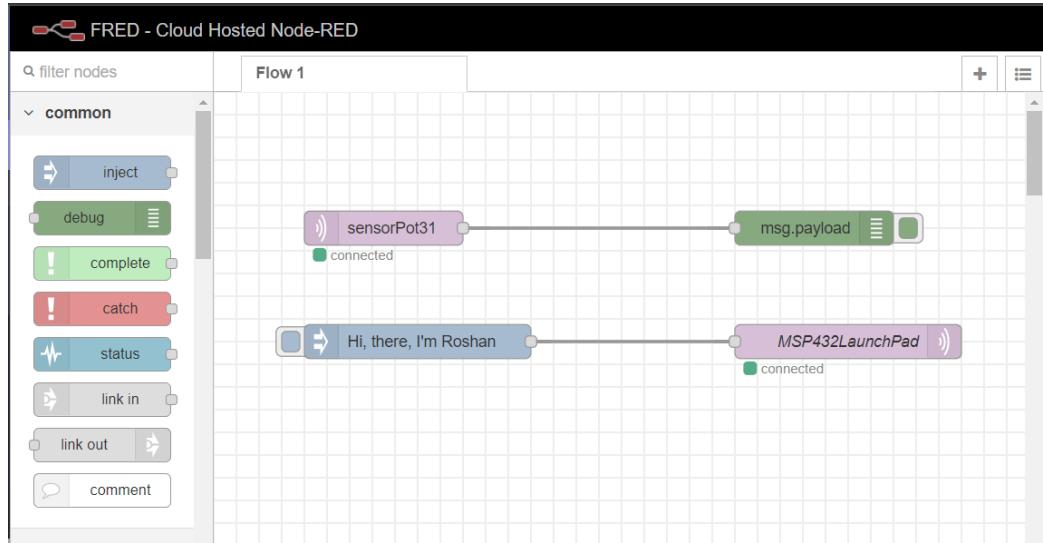
Sign in into it by providing details, and select the fred option.

Now, start the instance:

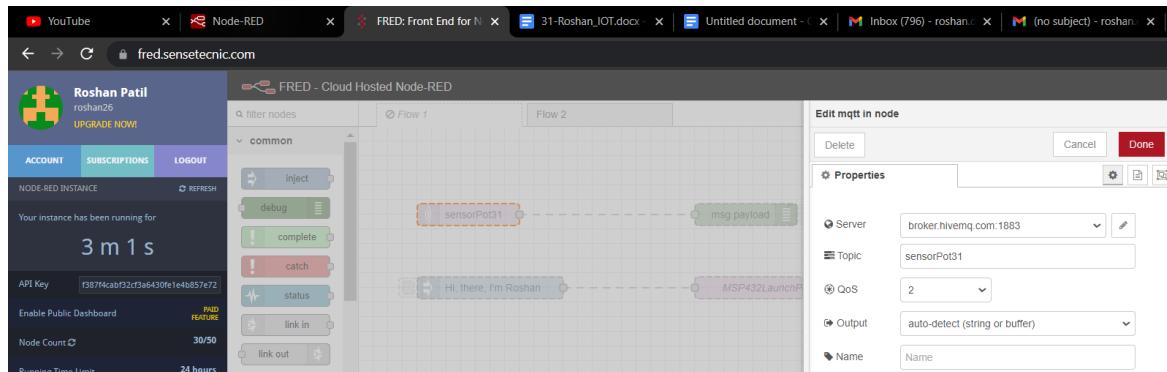


As soon as you start the instance you will be redirected to the node red console where you need to create the flow.

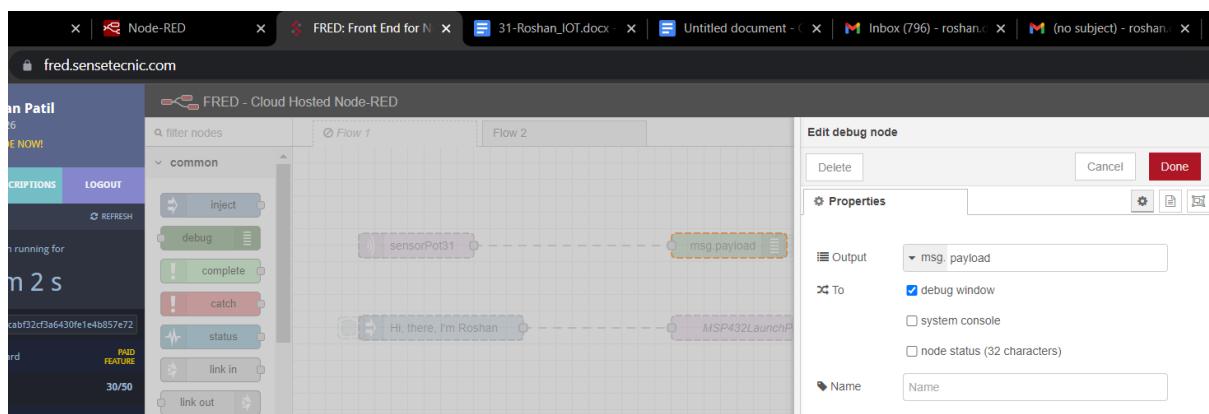
- Create a flow with a mqtt in node, debug node, inject node and mqtt out node,



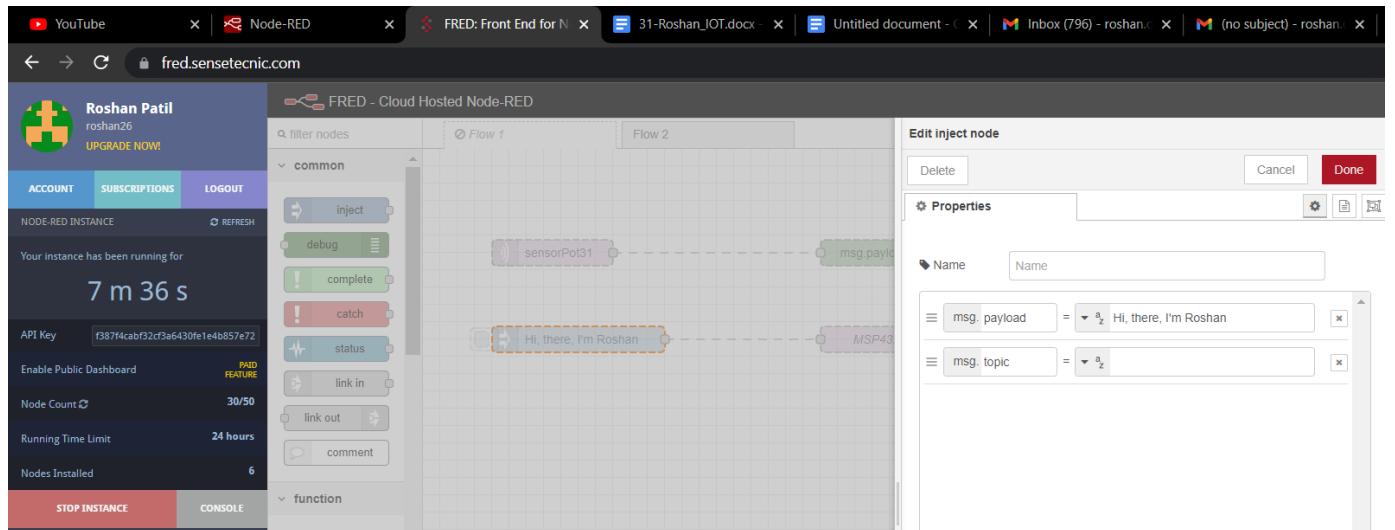
- In mqtt-in node give topic name which will work as a publisher, the name of the topic should be unique.



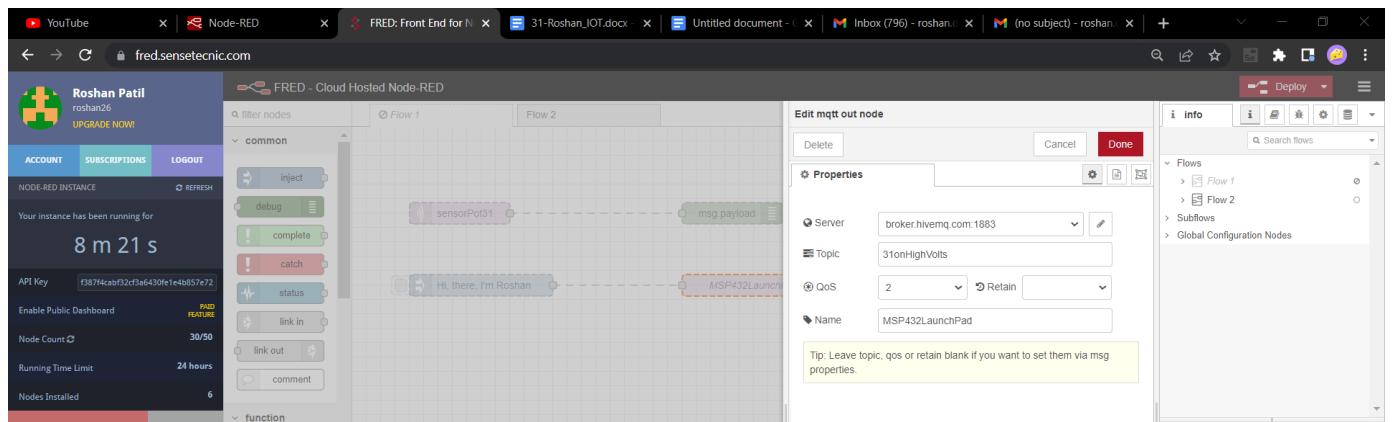
- In debug node:



- In inject node give msg payload:



- In mqtt-out node give topic name which will work as a subscriber. The name should be unique



- Now, import an energia project containing all dependent libraries to the CCS and write below code in the CCS project.

```
#include <ti/devices/msp432p4xx/driverlib/driverlib.h>
#include <WiFi.h>
#include <SPI.h>
#include <PubSubClient.h>
#include <aJSON.h>
```

```

##define SSID "octanauts"
##define PASSWORD "aditya30"

IPAddress shieldIP,subnetMask,gatewayIP; //IPAddress is a datatype , all others are variables of
type IPAddress

uint8_t rssi;//unsigned interger of 8 bits

uint8_t networkId;

byte macAddr[6];// mac address is 6 bytes // physical address of device is mac add

byte encryptionType;

char ssid[]="rp";// my cell phone

WiFiClient wclient;

char server[] = "broker.hivemq.com";

PubSubClient client(server,1883,callback,wclient);

char* jsonPayload;

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);//Serial is class and begin is method all present in SPI library

//putty is a terminal emulator

//it will connect the port and device at a given speed

Serial.print("connecting to WIFI..");

while(WiFi.begin(ssid)!=WL_CONNECTED)//for cell phone //WL_CONNECTED

//while(WiFi.begin(ssid,password)!=WL_CONNECTED)

{

Serial.print(".");

delay(1);

}

Serial.println("");

Serial.print("Wifi Connected , Fetching Wifi Sheild's IP address :");

```

```

while(WiFi.localIP()==INADDR_NONE){
    Serial.print(".");
    delay(1);
}

shieldIP = WiFi.localIP(); //WIFI Method , local IP takes IP address and stores in var in
IPADDRESS DATATYPE

Serial.println(shieldIP);
Serial.print("Access Point name:");
Serial.println(ssid);
Serial.print("Signal Strength");//
rssI=WiFi.RSSI(); //it fetches the signal strength
Serial.println(rssI); //RSSI-Received signal strength indication
uint8_t networkId= WiFi.scanNetworks();
Serial.print("number of access points in range:");
Serial.println(networkId);
for(int i=1;i<=networkId;i++){
    Serial.print("Name of Access points and encryption type:");
    Serial.print(WiFi.SSID(i));
    Serial.print(",");
    encryptionType=WiFi.encryptionType(i);
    //Serial.println(encryptionType,HEX);
    Serial.println(encryptionType,DEC);
}
subnetMask = WiFi.subnetMask();
Serial.print("Subnet Mask:");
Serial.println(subnetMask);
gatewayIP=WiFi.gatewayIP();
Serial.print("Gateway IP Address:");
Serial.println(gatewayIP);

```

```

WiFi.macAddress(macAddr);

Serial.print("Mac Address of Sheild:");

for(int i=0;i<6;i++){

    Serial.print(macAddr[i],HEX);

    if(i<=4)

        Serial.print(':');

    else

        Serial.println();

}

GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0|GPIO_PIN1|GPIO_PIN2);

GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P4,GPIO_PIN7,GPIO_TERTIARY_MODULE_FUNCTION);

ADC14_initModule(ADC_CLOCKSOURCE_MCLK,ADC_PREDIVIDER_1,ADC_PREDIVIDER_1,ADC_NOROUTE);

ADC14_configureSingleSampleMode(ADC_MEM6,true);

ADC14_configureConversionMemory(ADC_MEM6,ADC_VREFPOS_AVCC_VREFNEG_VS, ADC_INPUT_A6, false);

ADC14_setSampleHoldTime(ADC_PULSE_WIDTH_64, ADC_PULSE_WIDTH_64);

ADC14_enableSampleTimer(ADC_AUTOMATIC_ITERATION);

ADC14_enableModule();

ADC14_enableConversion();

ADC14_toggleConversionTrigger();

}

void loop() {

    // put your main code here, to run repeatedly:

    if(!client.connected())

    {

        Serial.println("Disconnected:Reconnecting");

        Serial.println("");

        if(!client.connect("31Msp432"))

```

```

{
    Serial.println("connection failed");

}

else{
    Serial.println("Connection success");
    if(client.subscribe("31onHighVolts"))
    {
        Serial.println("subscription successfull");
    }
}

int result,regressedData1;

float regressedData;

while(!ADC14_isBusy());

result = ADC14_getResult(ADC_MEM6);

P2OUT = result>>8;

Serial.println(result);

ADC14_toggleConversionTrigger();

regressedData = ((result*0.853015741)+4.9932982)/1000;

Serial.println(regressedData);

regressedData1 = ((result*0.853015741)+4.9932982)/1000;

String str = (String)regressedData1;

int str_len = str.length() + 2;

char char_array[str_len];

str.toCharArray(char_array, str_len);

char pot_reading[str_len];

if(regressedData<1)

{

```

```

pot_reading[0] = '0';
pot_reading[1] = ':';
for(int i = 2; i<=str_len;i++){
    pot_reading[i]=char_array[i-2];
}
else
{
    pot_reading[0] = char_array[0];
    pot_reading[1] = ':';
    for(int i = 2;j<=str_len;i++)
    {
        pot_reading[i] = char_array[i-1];
    }
}

aJsonObject *root = aJson.createObject();
aJsonObject *d = aJson.createObject();
aJson.addItemToObject(root,"d",d);
aJson.addStringToObject(d, "Roshan","MSP432");
aJson.addStringToObject(d, "potValue", pot_reading);
jsonPayload = aJson.print(root)+"\0";
aJson.deleteItem(d);
aJson.deleteItem(root);
client.publish("sensorPot31",jsonPayload);
client.poll();
delay(1000);
}

void callback(char* inTopic, byte* payload, unsigned int length)
{

```

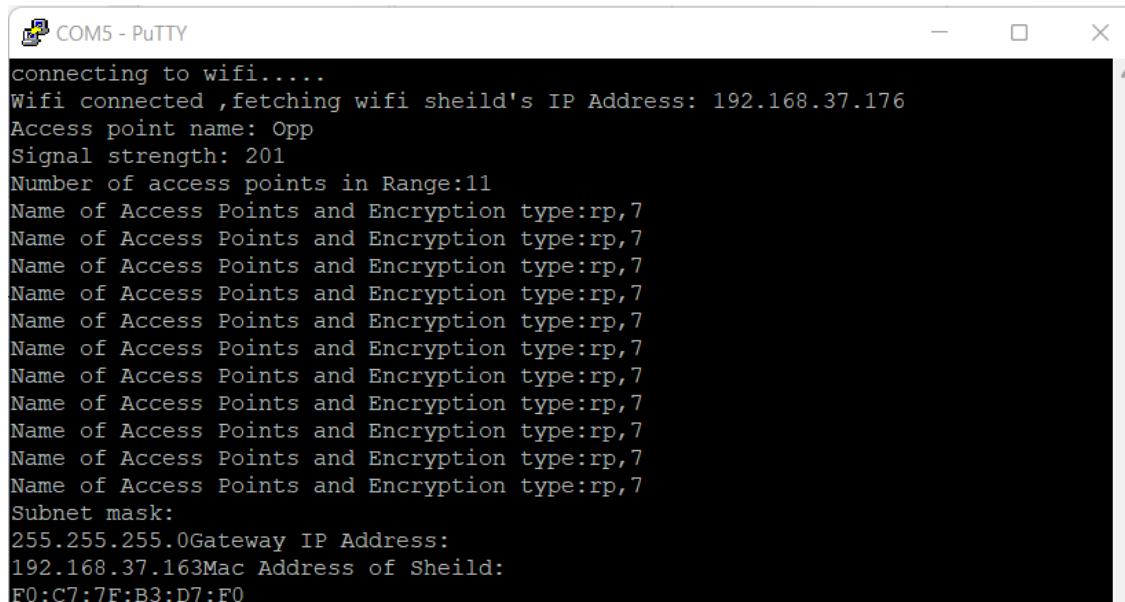
```
Serial.println("Message arrived on topic...");  
Serial.print(inTopic);  
Serial.print(". Message: ");  
String arrivedMessage;  
for(int i=0; i<length;i++)  
{  
    Serial.print((char)(payload[i]));  
    arrivedMessage+=(char)payload[i];  
}  
Serial.println();  
}
```

Now, build the project and debug the project.

Open putty and give com number(COM5) and speed(115200) in Serial and open.

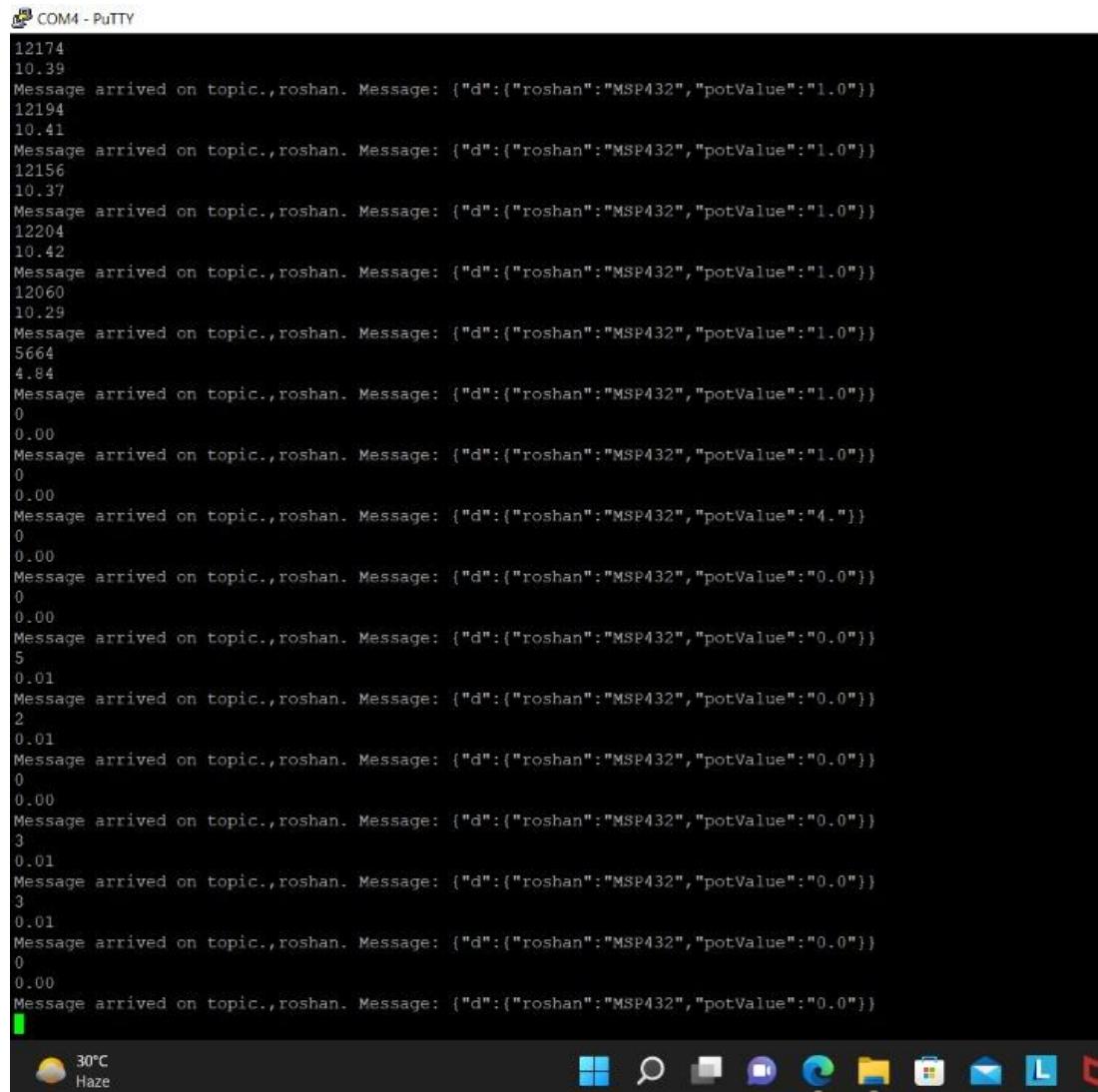
Resume the project and wait for wifi to be connected.

Output:



The screenshot shows a terminal window titled "COM5 - PuTTY". The window displays the following text output:

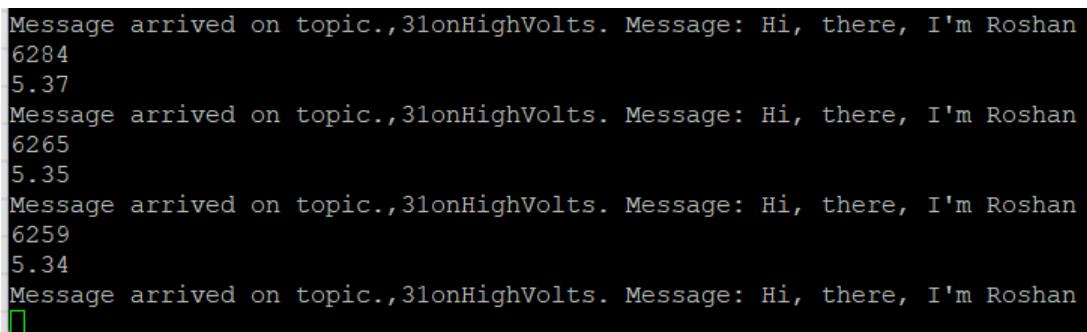
```
connecting to wifi.....  
Wifi connected ,fetching wifi sheild's IP Address: 192.168.37.176  
Access point name: Opp  
Signal strength: 201  
Number of access points in Range:11  
Name of Access Points and Encryption type:rp,7  
Subnet mask:  
255.255.255.0Gateway IP Address:  
192.168.37.163Mac Address of Sheild:  
F0:C7:7F:B3:D7:F0
```



```

COM4 - PutTY
12174
10.39
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"1.0"}}
12194
10.41
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"1.0"}}
12196
10.37
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"1.0"}}
12204
10.42
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"1.0"}}
12060
10.29
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"1.0"}}
5664
4.84
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"1.0"}}
0
0.00
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"1.0"}}
0
0.00
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"4."}}
0
0.00
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}
0
0.00
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}
5
0.01
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}
2
0.01
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}
0
0.00
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}
3
0.01
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}
3
0.01
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}
0
0.00
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}
0
0.00
Message arrived on topic.,roshan. Message: {"d":{"roshan":"MSP432","potValue":"0.0"}}

```



```

Message arrived on topic.,3lonHighVolts. Message: Hi, there, I'm Roshan
6284
5.37
Message arrived on topic.,3lonHighVolts. Message: Hi, there, I'm Roshan
6265
5.35
Message arrived on topic.,3lonHighVolts. Message: Hi, there, I'm Roshan
6259
5.34
Message arrived on topic.,3lonHighVolts. Message: Hi, there, I'm Roshan

```

In fred node-red debug terminal:

```
sensorPot31 : msg.payload : string[42]
{
  "d": {
    "Roshan": "MSP432", "potValue": "1.3"
  }
}

8/2/2022, 9:42:25 AM node: 8812fb86.d37318
sensorPot31 : msg.payload : string[42]
{
  "d": {
    "Roshan": "MSP432", "potValue": "1.3"
  }
}

8/2/2022, 9:42:26 AM node: 8812fb86.d37318
sensorPot31 : msg.payload : string[42]
{
  "d": {
    "Roshan": "MSP432", "potValue": "1.3"
  }
}

8/2/2022, 9:42:26 AM node: 8812fb86.d37318
sensorPot31 : msg.payload : string[42]
{
  "d": {
    "Roshan": "MSP432", "potValue": "1.3"
  }
}

8/2/2022, 9:42:28 AM node: 8812fb86.d37318
sensorPot31 : msg.payload : string[42]
{
  "d": {
    "Roshan": "MSP432", "potValue": "1.3"
  }
}

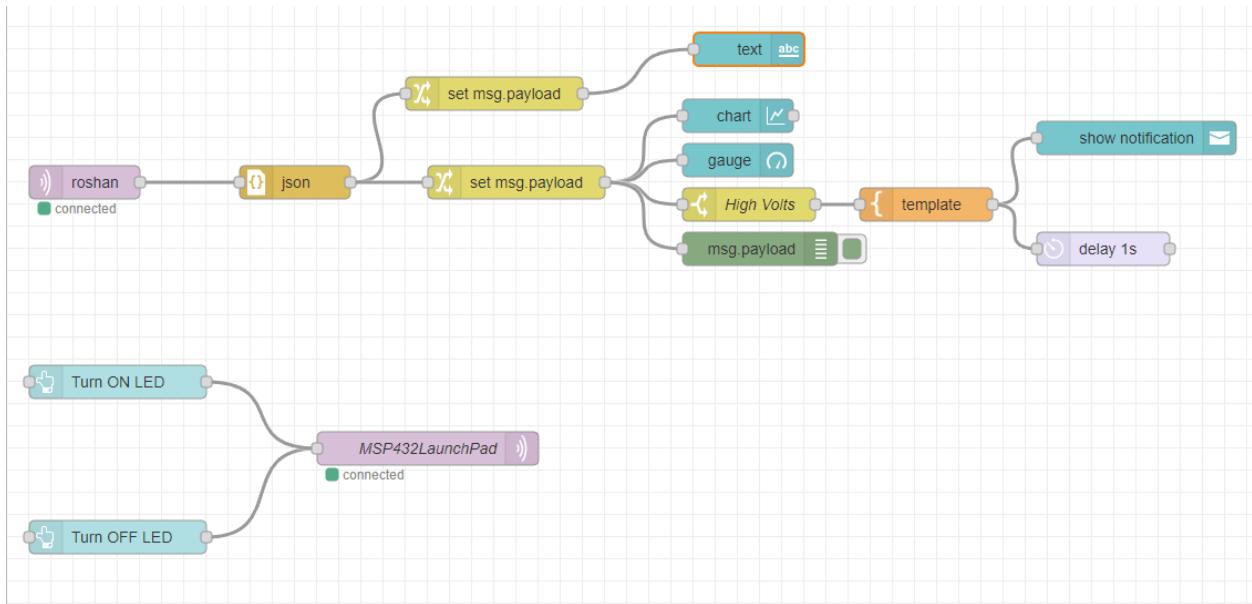
8/2/2022, 9:42:29 AM node: 8812fb86.d37318
sensorPot31 : msg.payload : string[42]
{
  "d": {
    "Roshan": "MSP432", "potValue": "1.3"
  }
}
```

# Practical No. 10

## Practical Name: Connecting MSP432 to the cloud

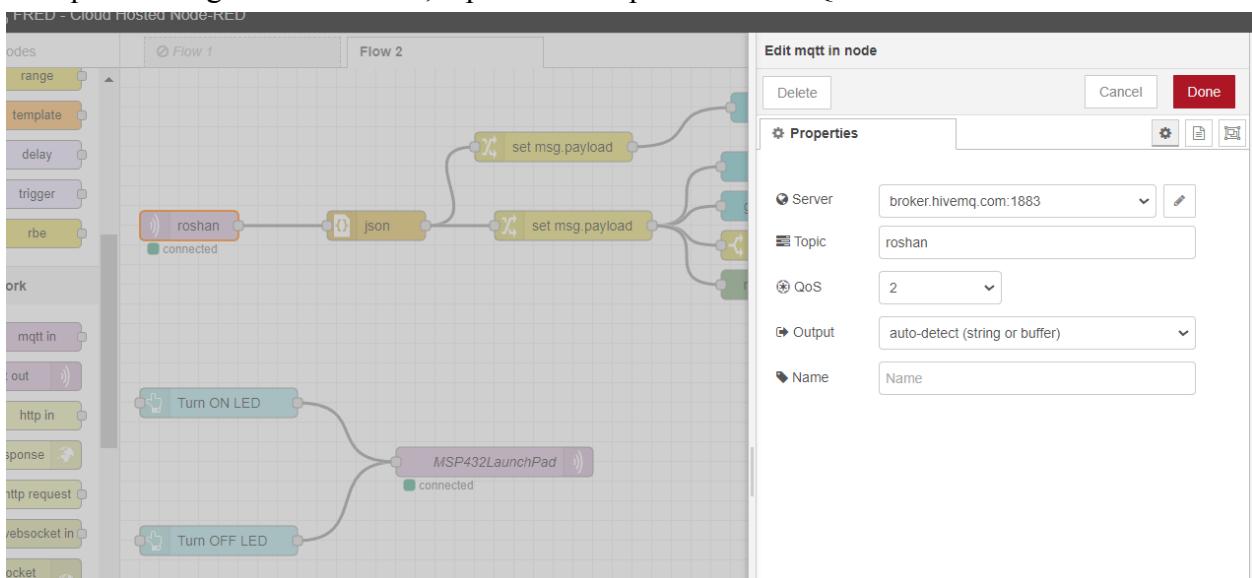
- Step 1: Open <https://fred.sensetecnic.com/> cloud.

To create dashboards add following flow:

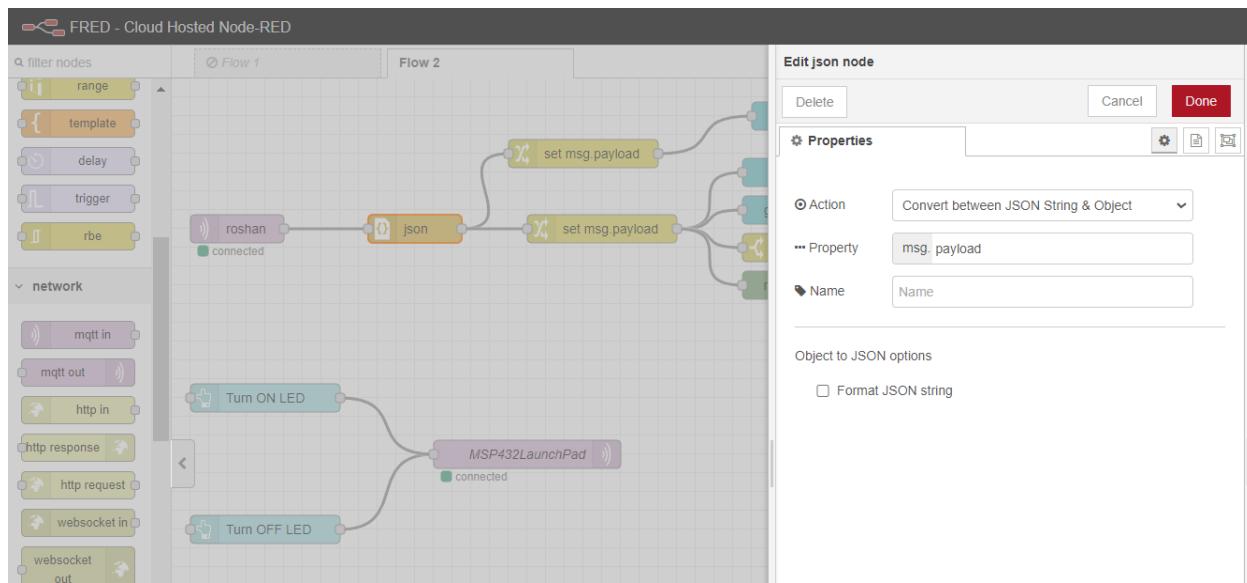


- Configure the nodes as follow:

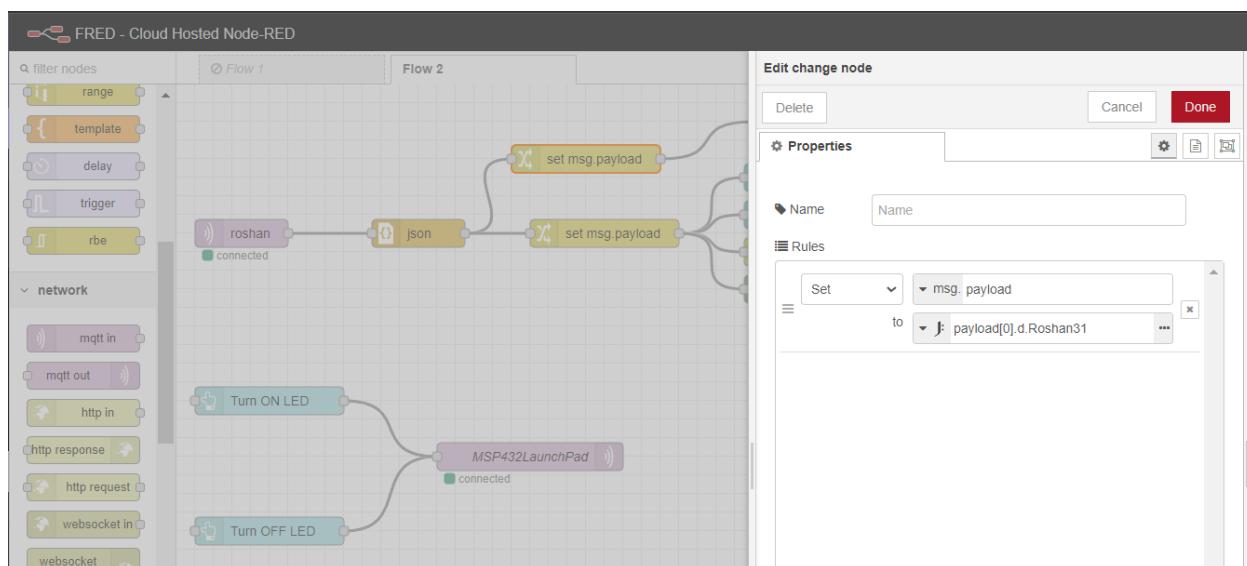
In mqtt in node give server name, topic name i.e publisher and QoS:



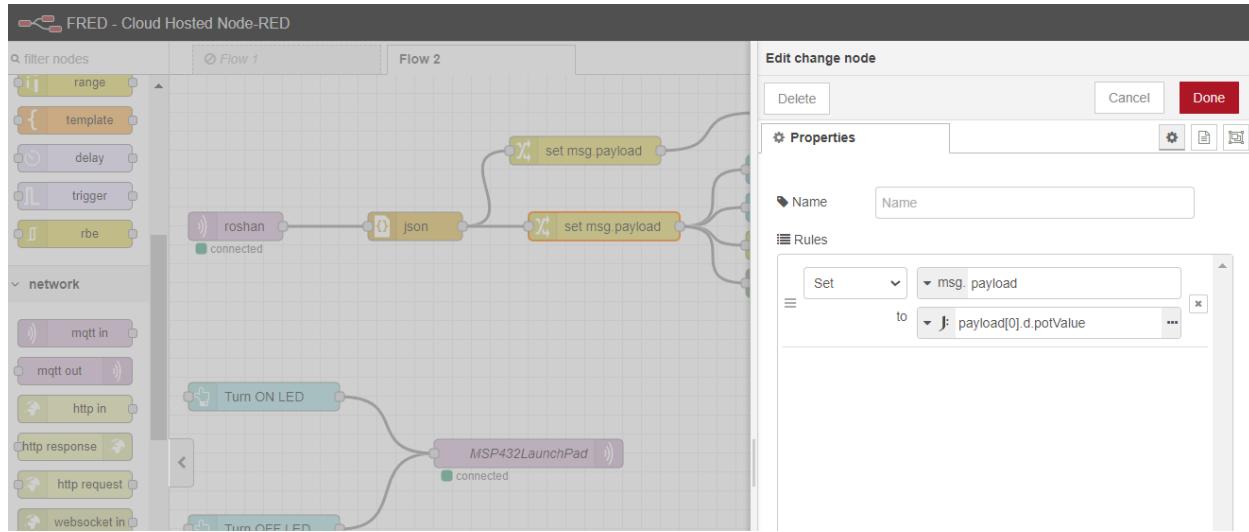
- In json node select action and give a msg payload:



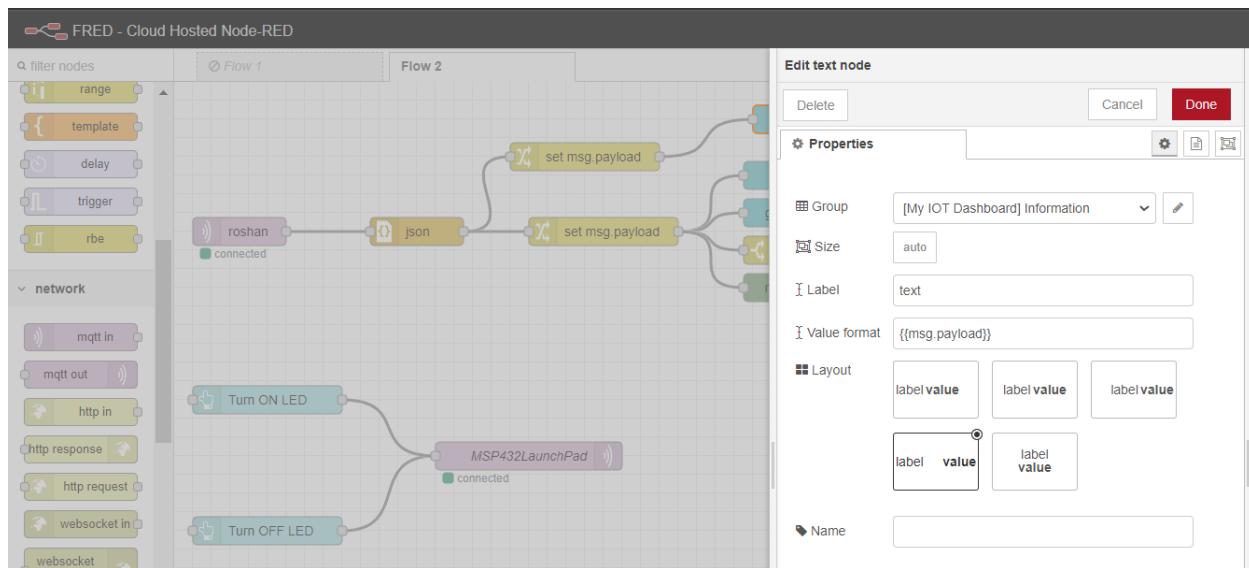
- In change node: select msg payload and write expression as payload[0].d.Roshan31



- In the second change node give msg payload and expression with potValue

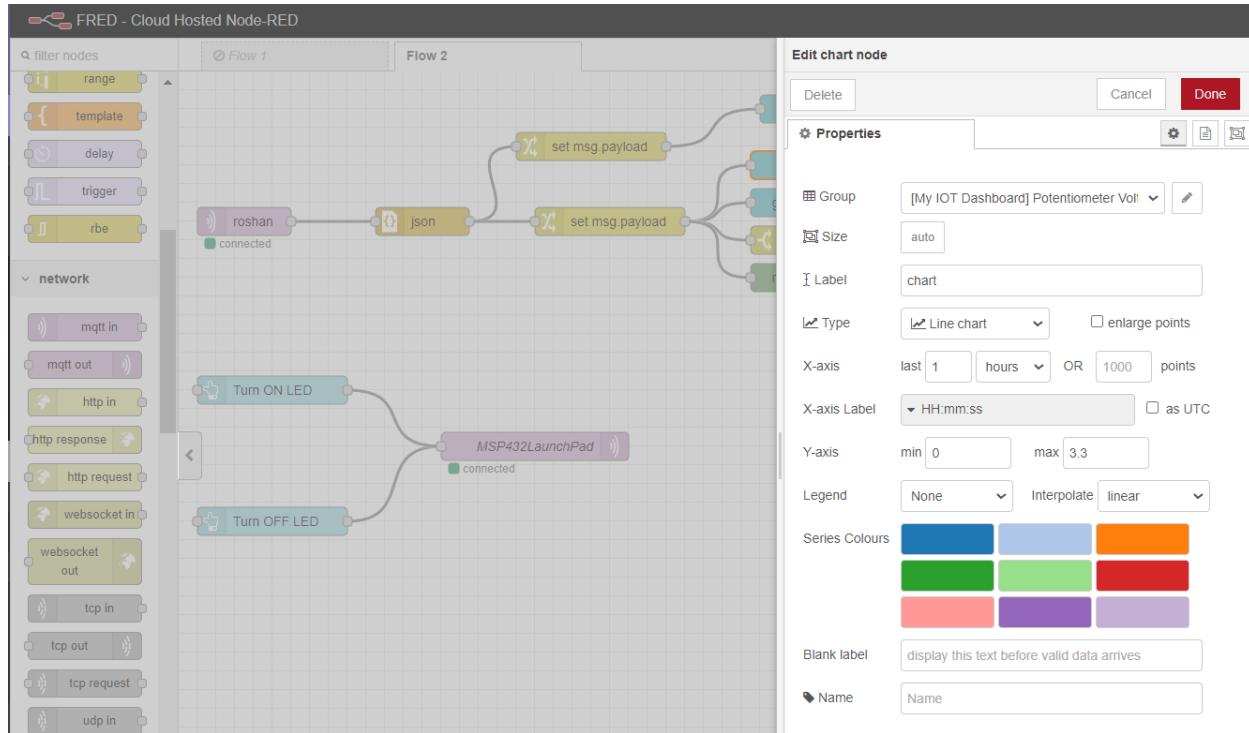


- In the text node add a group i.e. IOT Dashboard Information:

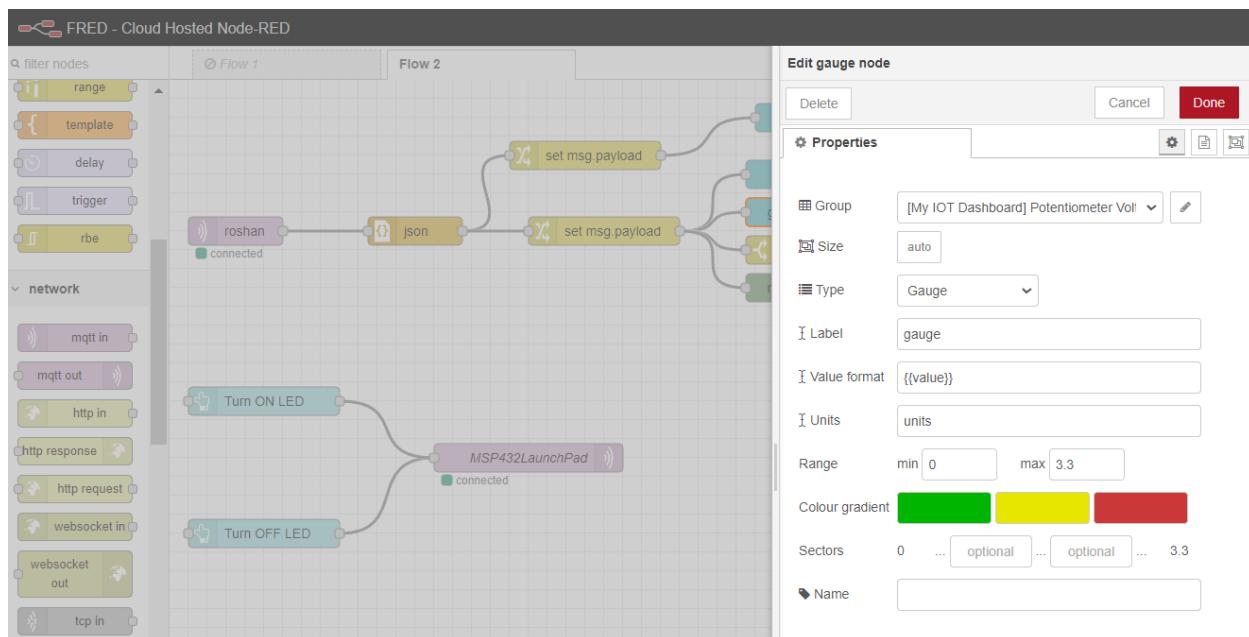


- In chart node:

1. Add group name(IOT Dashboard - Potentiometer Voltage)
2. Add minimum voltage as 0 and maximum voltage 3.3

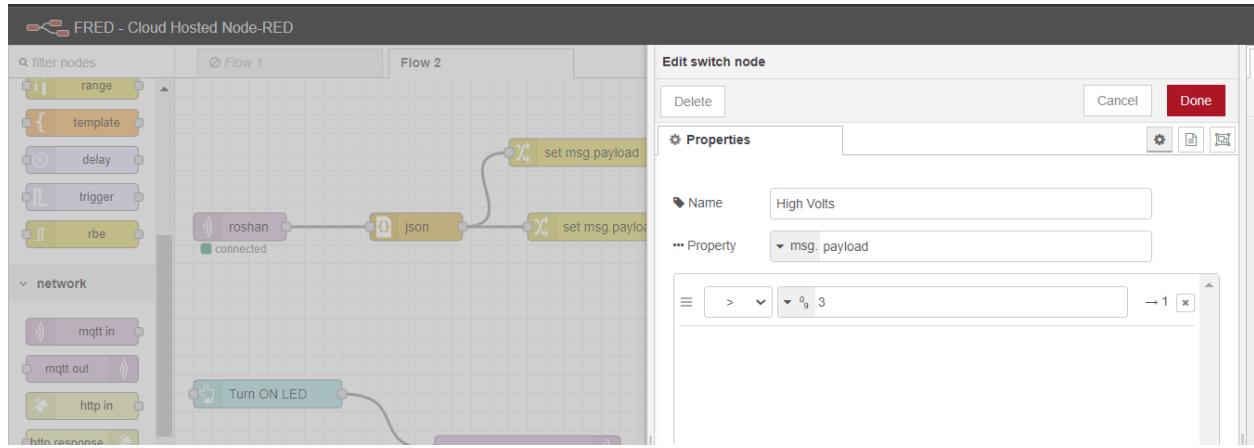


- In guage node add Group name and min voltage 0 and max voltage 3.3:

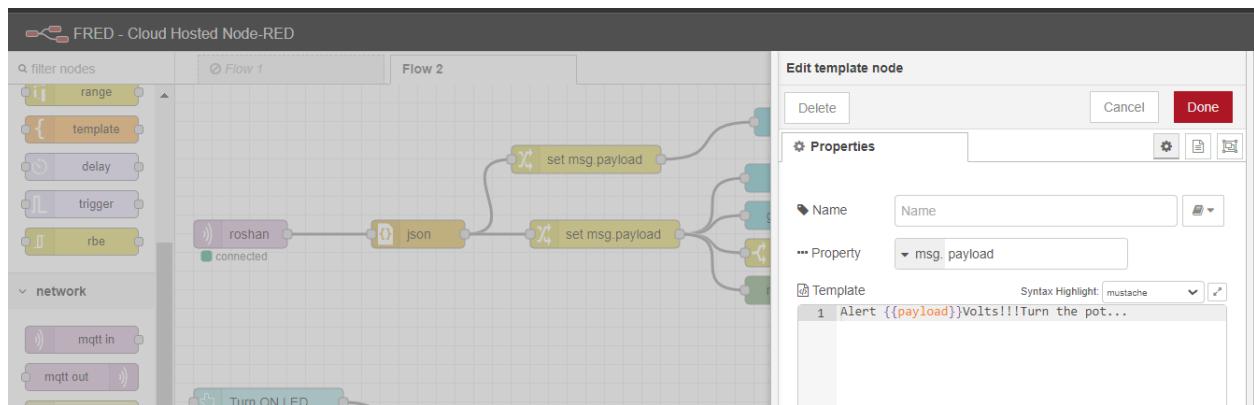


In switch node:

1. Give msg payload
2. Give condition i.e > 3 volts

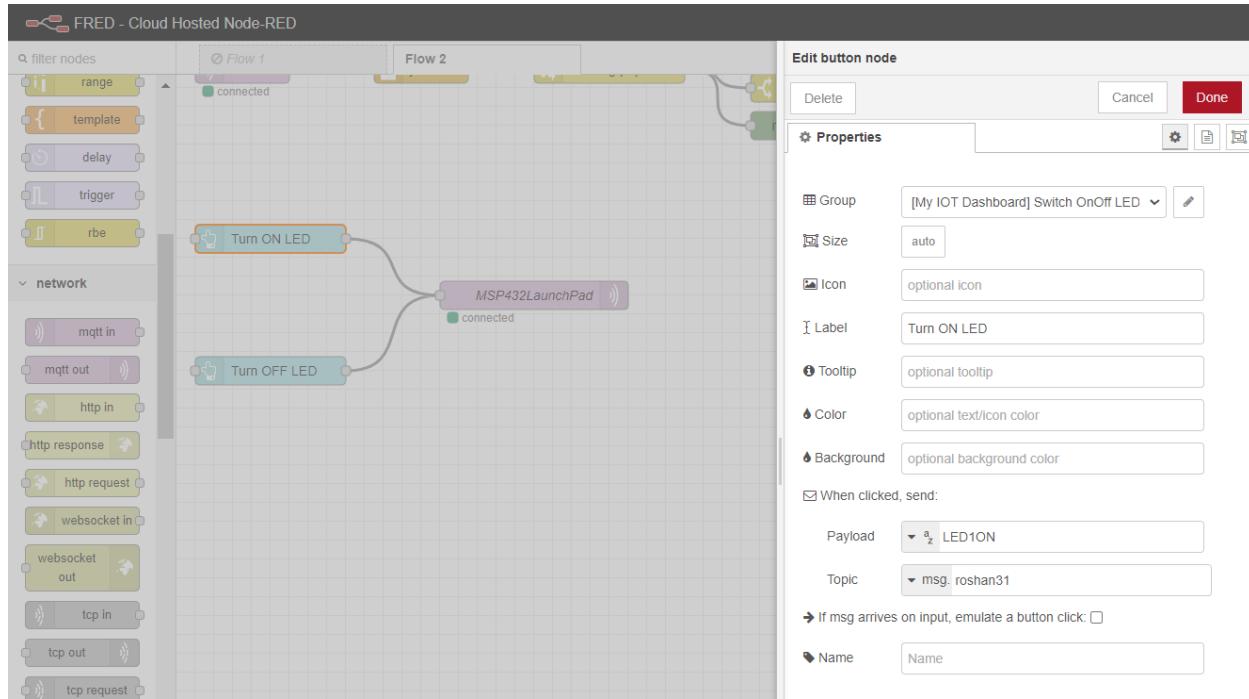


- In the template node add alert as: Alert {{payload}} Volts!!!Turn the pot



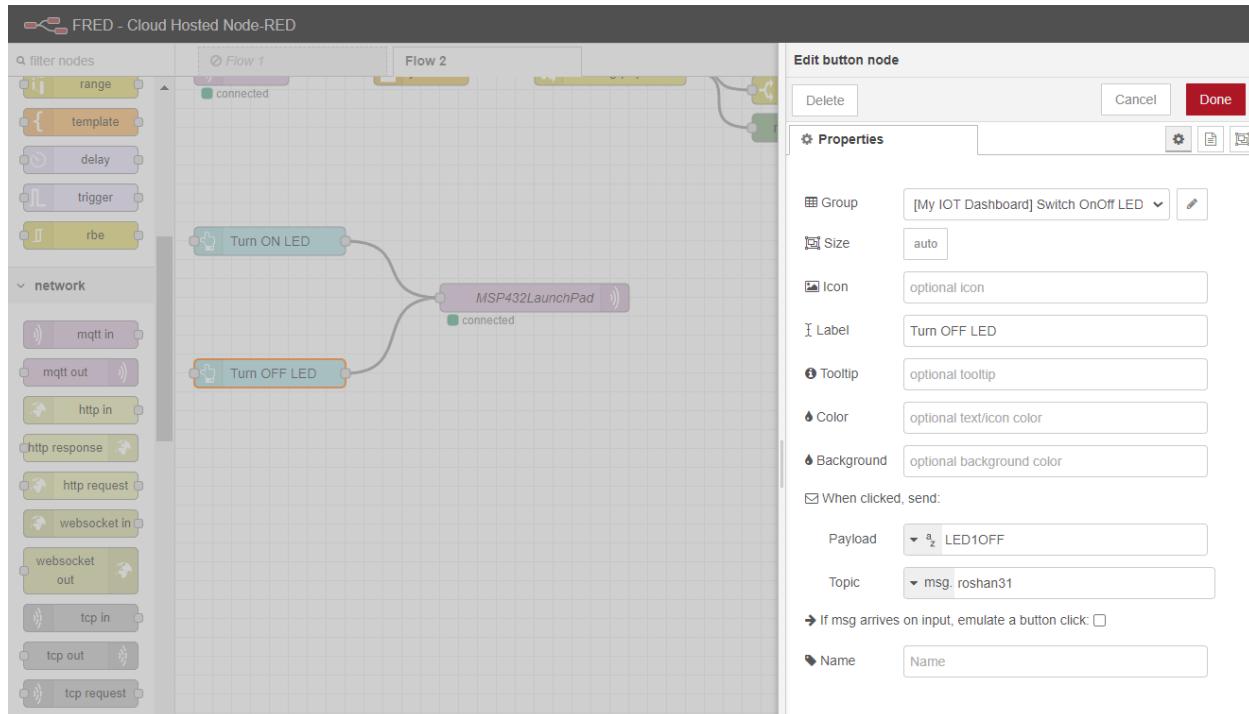
In the first button node(Turn ON LED):

1. Give a group name
2. A label as Turn ON LED
3. Payload as LED1ON
4. And topic name(subscriber) i.e roshan31



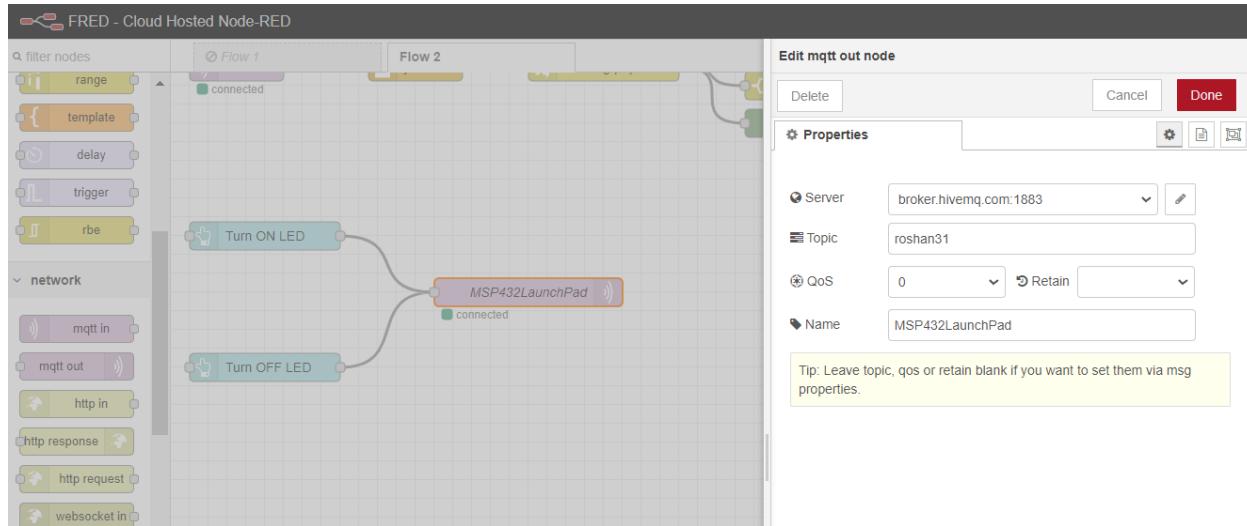
In the first button node(Turn OFF LED):

1. Give a group name
2. A label as Turn OFF LED
3. Payload as LED1OFF
4. And topic name(subscriber) i.e roshan31



In mqtt out node:

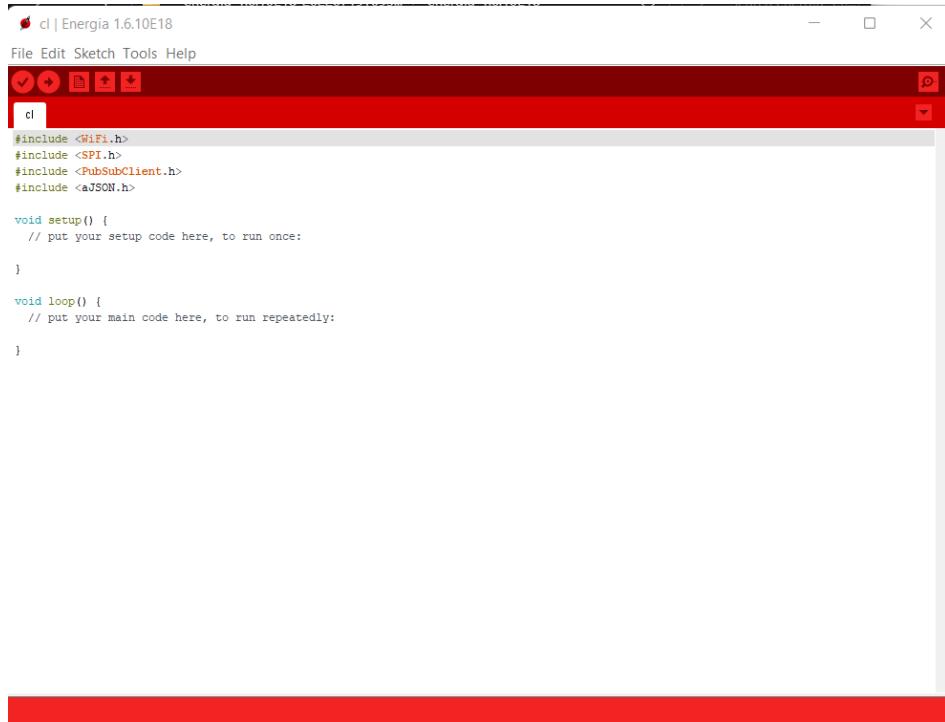
1. Give Server name
2. Topic name i.e Subscriber - roshan31
3. QoS as 0



Now, Open energia and write below code and save the file:

```
#include <WiFi.h>
#include <SPI.h>
#include <PubSubClient.h>
#include <aJSON.h>

void setup() {
}
void loop(){}
```



```
#include <WiFi.h>
#include <SPI.h>
#include <PubSubClient.h>
#include <aJSON.h>

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Now, go to CCS:

1. Create project by: Projects(menu bar) => Import a CCS project => Browse folder ti => Select a project from empty source:
2. Rename the project
3. After creating the project, now import the energia project that we have created earlier. Project => Import Energia project => Browse project and select that we have created.
4. Include this package from CCS main.c #include <ti/devices/msp432p4xx/driverlib/driverlib.h>
5. Write below code in the file:

Code:

```
#include <ti/devices/msp432p4xx/driverlib/driverlib.h>
#include <WiFi.h>
#include <SPI.h> //serial communication
#include <PubSubClient.h> //publisher and //subscribe
#include <aJSON.h>

//WIFI SETUP CODE
IPAddress shieldIP, subnetMask, gatewayIP;
uint8_t rssi;
uint8_t networkId;
byte macAddr[6]; //mac address is physical address of hardware
byte encryptionType;

char ssid[] = "rp"; //My cell name
/*char ssid[]=""
```

```

* char password[]="";
WiFiClient wclient;

char server[]{"broker.hivemq.com"};
PubSubClient client(server,1883,callback,wclient);

char*jsonPayload;
aJsonStream serial_stream(&Serial);
void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    Serial.print("Connecting to wifi...");
    while(WiFi.begin(ssid)!=WL_CONNECTED)
    {
        Serial.print(".");
        delay(1); //delay in ms
    }
    Serial.println("");
    Serial.println("Wifi connected, Fetching wifi shield's IP Address: ");
    while(WiFi.localIP()==INADDR_NONE)
    {
        Serial.print(".");
        delay(1);

    }
    shieldIP = WiFi.localIP();
    Serial.println(shieldIP);

    Serial.print("Access point name: ");
    Serial.println(ssid);

    Serial.print("Signal strength: ");

    rssi = WiFi.RSSI();
    Serial.println(rssi);

    uint8_t networkId = WiFi.scanNetworks();
    Serial.print("Number of access point in range: ");
    Serial.println(networkId);

    for(int i=1;i<=networkId;i++)
    {
        Serial.print("Name of Access Points and encryption type: ");
        Serial.print(WiFi.SSID(i));

```

```

Serial.print(",");
encryptionType = WiFi.encryptionType(i);
Serial.println(encryptionType,HEX);
}

subnetMask = WiFi.subnetMask();
Serial.print("Subnet Mask");
Serial.println(subnetMask);

gatewayIP = WiFi.gatewayIP();
Serial.print("Gateway IP address: ");
Serial.println(gatewayIP);

WiFi.macAddress(macAddr);
Serial.print("Mac Address of shield: ");
for(int i=0; i<6;i++)
{
    Serial.print(macAddr[i],HEX);
    if(i<=4)
        Serial.print(':');
    else
        Serial.println();
}
GPIO_setAsOutputPin(GPIO_PORT_P1,GPIO_PIN0);
GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0|GPIO_PIN1|GPIO_PIN2); //3 pin for 3
color(RED GREEN AND BLUE) COMBINATION

GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P4,GPIO_PIN7,GPIO_TERTIARY_MODULE_FUNCTION); //pin 7 used for convert analog to digital
ADC14_initModule(ADC_CLOCKSOURCE_MCLK, ADC_PREDIVIDER_1, ADC_PREDIVIDER_1,
ADC_NOROUTE); //MCLK-> MASTER CLOCK //SETTING UP THE CLOCK //PREDIVIDER USEED
FOR TO FIND SAMLLEST NUMBER PREDRIVIDER USED MULTIPLE TIME //ADC_NOROTUE->
USED FOR EXTRNAL CONNECTION //INIMODULE NEED 4 PARAMETER SO 2 PREDRIVDER IS
UESD
ADC14_configureSingleSampleMode(ADC_MEM6, true); //ADC_MEM6 INTERNAL MEMORY
WHERE THE RESULT ARE STORED
ADC14_configureConversionMemory(ADC_MEM6, ADC_VREFPOS_AVCC_VREFNEG_VSS,
ADC_INPUT_A6, false); //reference voltage-->ADC_VREFPOS_AVCC_VREFNEG_VSS //use
channel-->ADC_INPUT_A6 //false-->conversation of memory only onces
ADC14_setSampleHoldTime(ADC_PULSE_WIDTH_32, ADC_PULSE_WIDTH_4);
ADC14_setResolution(ADC_12BIT);
ADC14_enableSampleTimer(ADC_AUTOMATIC_ITERATION); //will run automatically
ADC14_enableModule();
ADC14_enableConversion();
ADC14_toggleConversionTrigger();

```

```

}

void loop()
{
    if(!client.connected())
    {
        Serial.println("Disconnected:Reconnecting..");
        if(!client.connect("Roshan"))
        {
            Serial.println("connection failed");
        }
        else
        {
            Serial.println("Connection success");
            if(client.subscribe("roshan31"))

            {
                Serial.println("Subscription successful");
            }
        }
    }

    int result,regressedData1;
    float regressedData;
    while(!ADC14_isBusy());
    result=ADC14_getResult(ADC_MEM6);
    //P2OUT=result>>8;
    //Serial.println(result);
    ADC14_toggleConversionTrigger();

    regressedData=((result*0.78734)+24.9251)/1000;
    Serial.println(regressedData);
    regressedData1=((result*0.78734)+24.9251);
    P2->OUT=result>>8;

    String str=(String)regressedData1;
    int str_len=str.length()+2;

    char char_array[str_len];
    str.toCharArray(char_array,str_len);
    char pot_reading[str_len];

    if(regressedData<1)

```

```

{
    pot_reading[0]='0';
    pot_reading[1]='.';
    for(int i=2;i<=str_len;i++)
    {
        pot_reading[i]=char_array[i-2];
    }
}
else
{
    pot_reading[0]=char_array[0];
    pot_reading[1]='.';
    for(int i=2;i<=str_len;i++)
    {
        pot_reading[i]=char_array[i-1];
    }
}

aJsonObject*root=aJson.createObject();
aJsonObject*d=aJson.createObject();
aJson.addItemToObject(root,"d",d);
aJson.addStringToObject(d,"Roshan31","RoshanIOT");

aJson.addStringToObject(d,"potValue",pot_reading);
jsonPayload=aJson.print(root)+'\0';

aJson.deleteItem(d);
aJson.deleteItem(root);
//publish data to MQTT broker
client.publish("roshan",jsonPayload);

client.poll();
delay(1000);
}

void callback(char*inTopic,byte*payload,unsigned int length)
{
    Serial.print("Message arrived on topic:");
    Serial.print(inTopic);
    Serial.print(".Message:");
    String arrivedMessage;

    for(int i=0;i<length;i++){
        Serial.print((char)(payload[i]));
    }
}

```

```

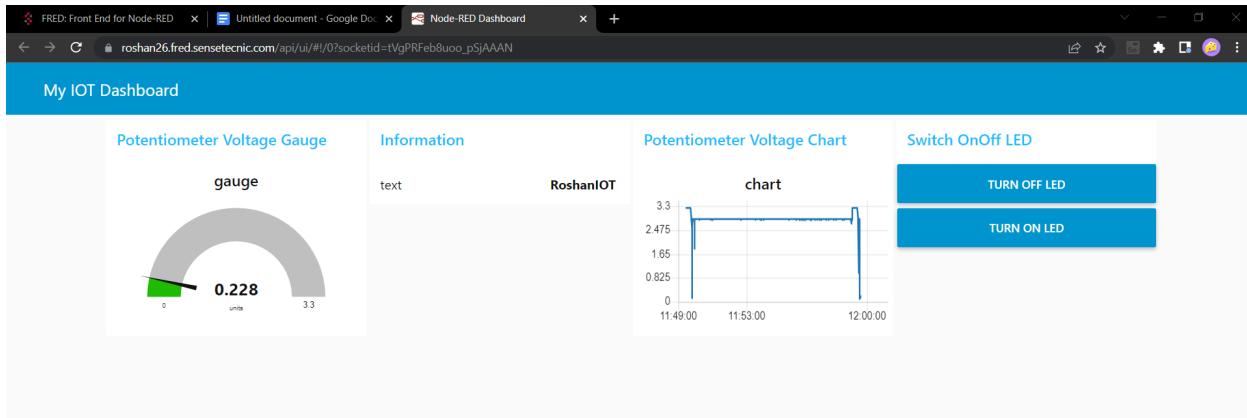
arrivedMessage+=(char)payload[i];
}

Serial.println();
if(arrivedMessage=="LED1ON")
    GPIO_setOutputHighOnPin(GPIO_PORT_P1,GPIO_PIN0);
else
    if(arrivedMessage=="LED1OFF")
        GPIO_setOutputLowOnPin(GPIO_PORT_P1,GPIO_PIN0);
}

```

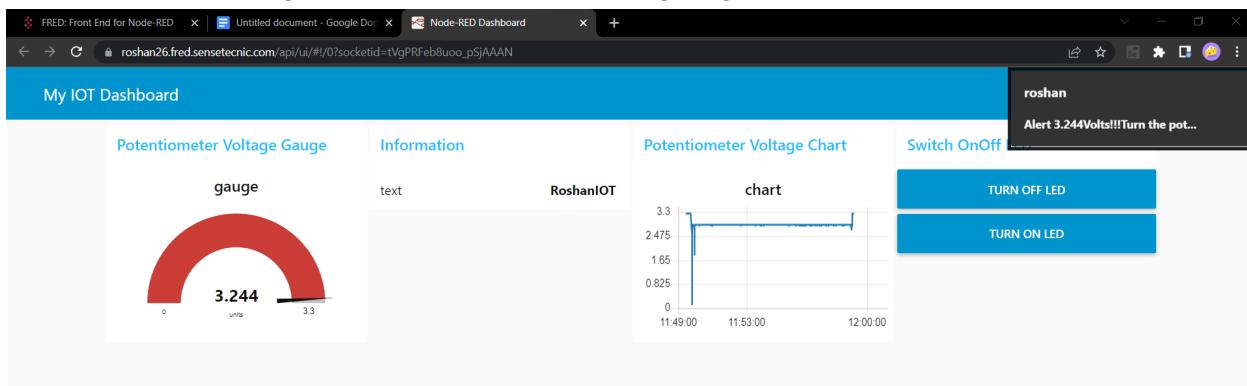
### Output:

When potentiometer is low i.e voltage is lower than 3:

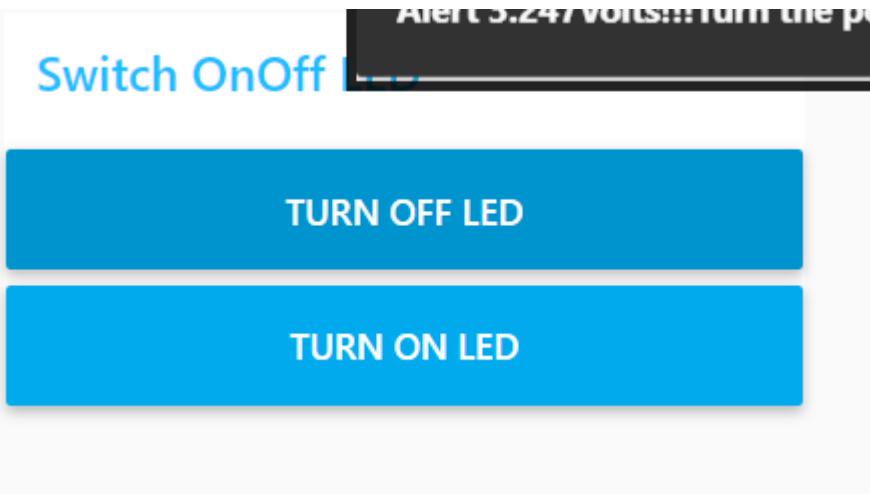


When potentiometer is high:

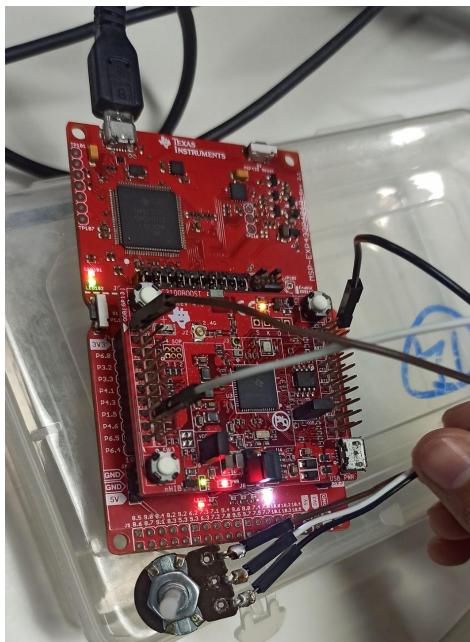
We can see an alert at right corner which indicates voltage is greater than 3:



Buttons which controls the LED on the board:

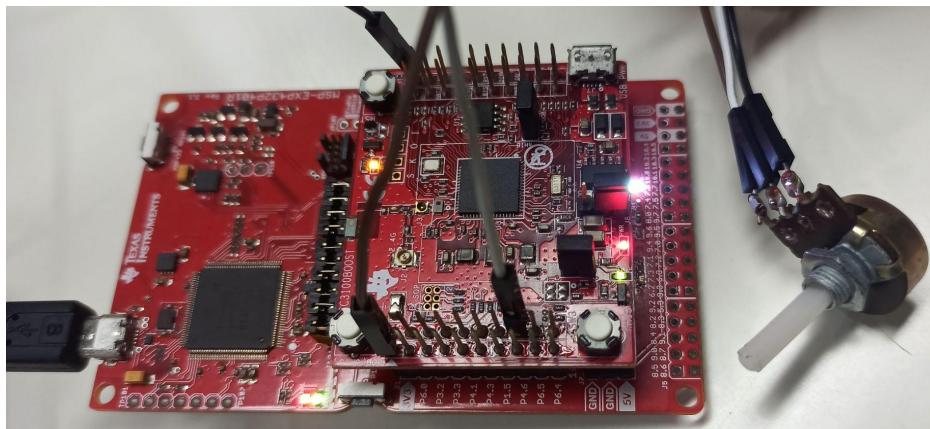


- When TURN ON LED Button is pressed:



We can see the LED (red) has been turned on.

- When TURN OFF LED Button is pressed:



We can see the LED has been turned off.

Configure Putty and We can see potentiometer reading in putty:

COM5 - PuTTY

```
2.85  
2.85  
2.84  
2.85  
2.85  
2.85  
2.86  
2.85  
2.85  
2.85  
2.86  
2.85  
2.85  
2.85  
2.85  
2.85  
2.85  
2.85  
2.84  
2.85  
2.85  
2.85  
2.83  
2.85
```

We can see this in Debug terminal at fred node-red as well:

```
8/3/2022, 11:52:36 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.849"
8/3/2022, 11:52:37 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.853"
8/3/2022, 11:52:38 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.847"
8/3/2022, 11:52:39 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.845"
8/3/2022, 11:52:40 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.834"
8/3/2022, 11:52:41 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.849"
8/3/2022, 11:52:42 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.849"
8/3/2022, 11:52:43 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.853"
8/3/2022, 11:52:44 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.847"
8/3/2022, 11:52:45 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
"2.847"
8/3/2022, 11:52:46 AM node: 9a4e38fe.ba5e38
roshan : msg.payload : string[5]
```