



SOMAIYA
VIDYAVIHAR UNIVERSITY

K J Somaiya Institute of Management

K J Somaiya Institute of Management

Vidyavihar, Mumbai – 400 077

Master of Computer Applications

Trimester 4 (2021 – 23)

This is to certify that Mr. Vaibhav Kumar Roll No. 19 of MCA, has completed his Internet of Things Journal as per the syllabus for the academic year 2021–2023. The Journal has also been evaluated by the concerned faculty of KJ Somaiya Institute of Management.

Signature of the Faculty-In charge

Signature of the Course Coordinator

Date: 09 / 08 / 2022

Signature of the External Examiner

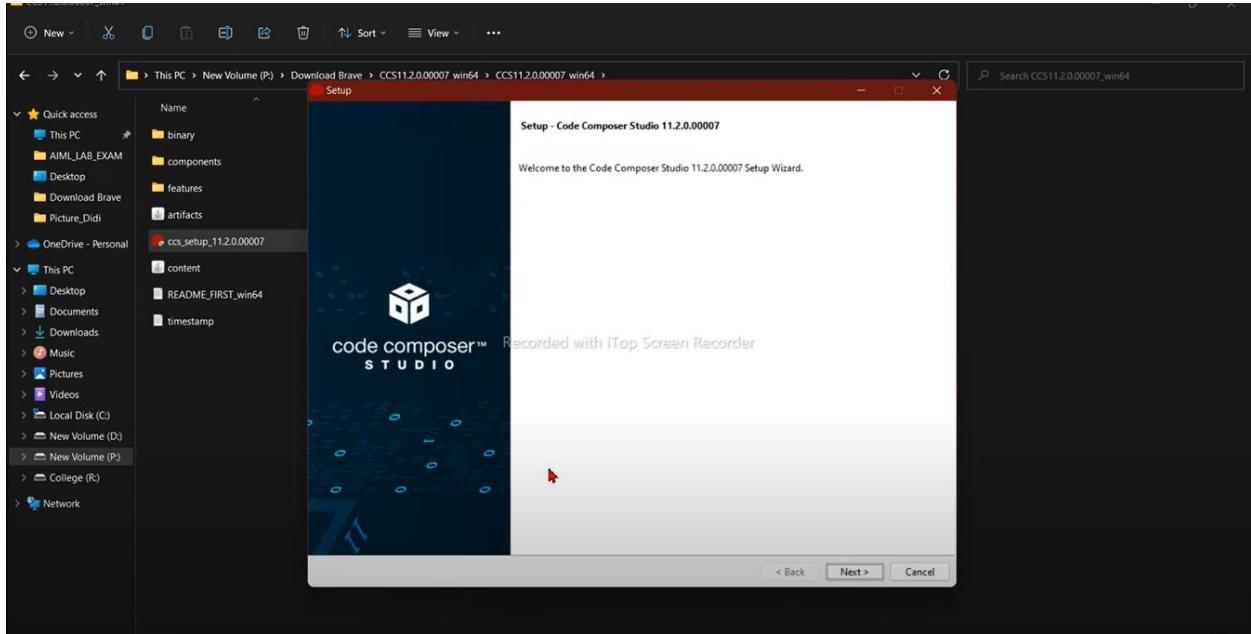
INDEX

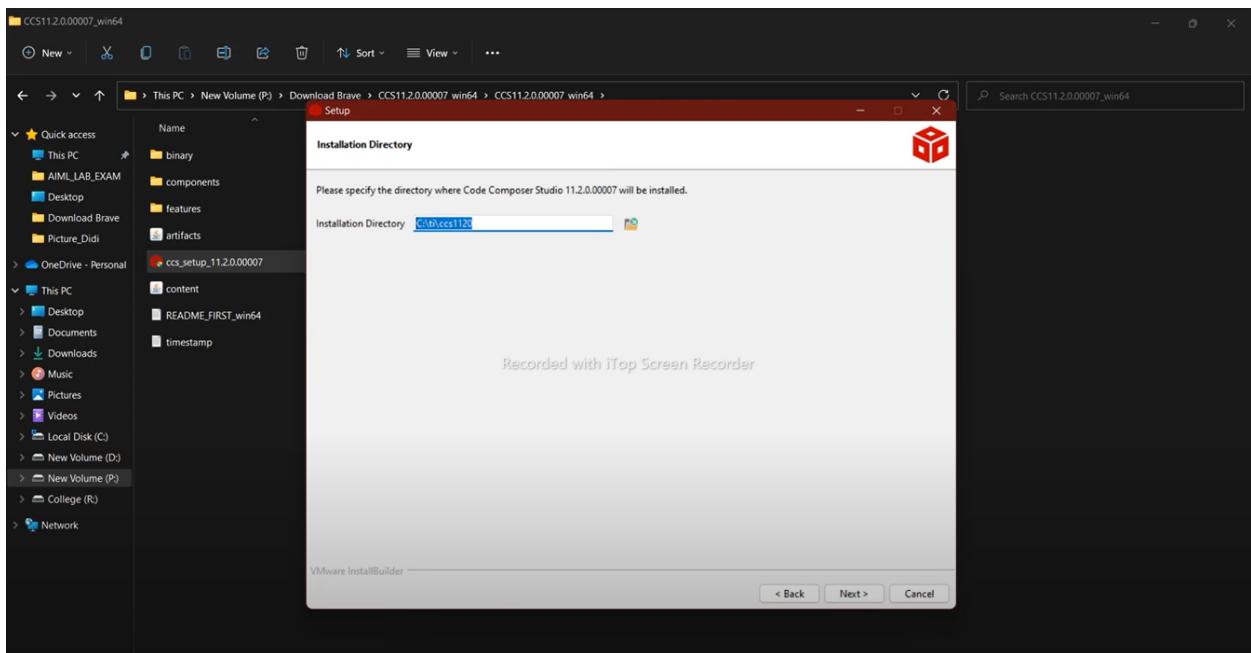
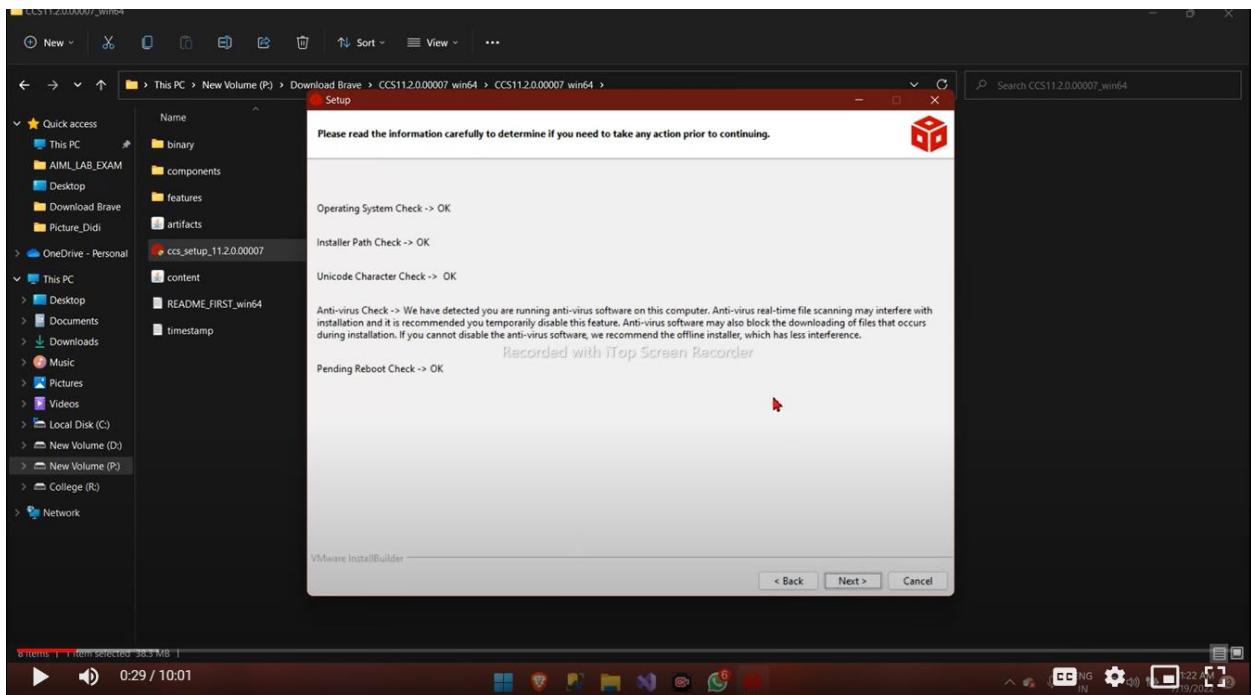
SrNo	Practical	Page No
1	CCS Installation, User Interface tour, Project creation, Compiling and debugging with MSP432	1-18
2	Hello world of Embedded Systems	19-27
3	Installing Node.js, Node-Red and Node Red interface.	28-35
4	Introducing the inject, function, debug and switch nodes	36-38
5	Random number generation with selection of color. Introducing the HTTP node.	39-42
6	CPU utilization flow.	43-44
7	WiFi Setup	45-50
8	Analog to Digital Converter	51-57
9	Sending Raw Data to the Cloud	58-70
10	Connecting MSP432 to the Cloud	71-82

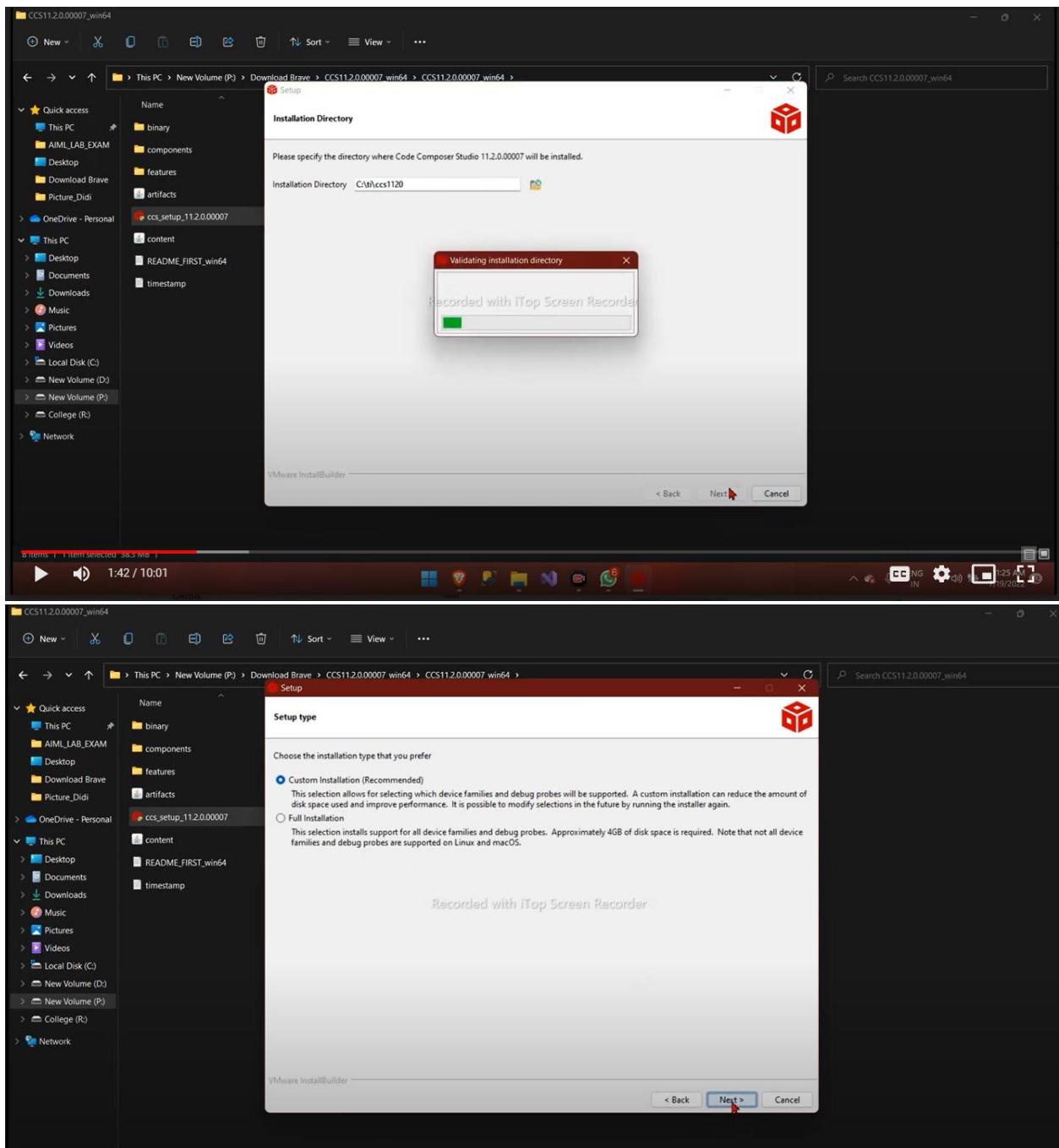
Practical 1

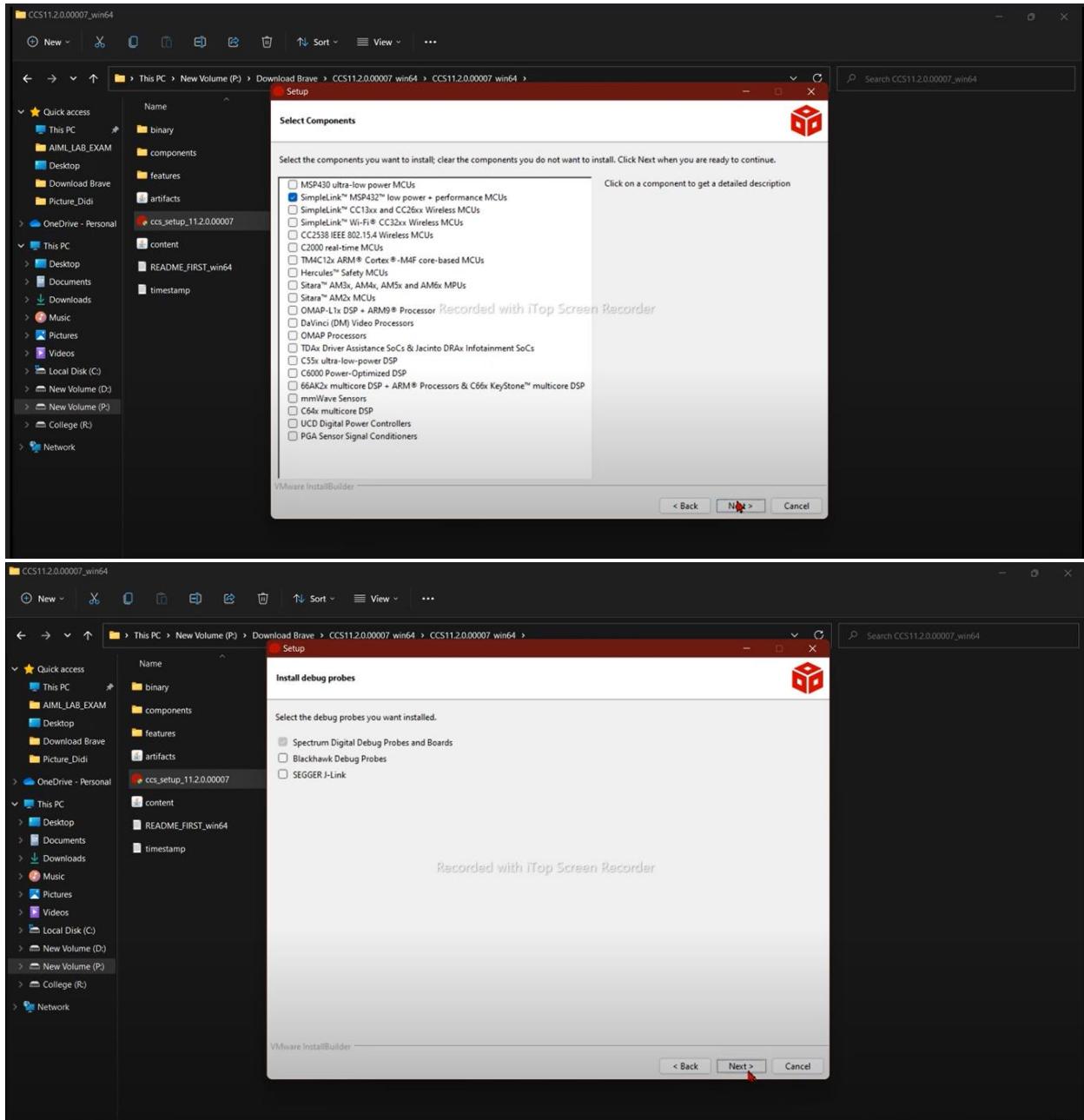
Practical Name: CCS Installation, User Interface tour, Project creation, Compiling and debugging with MSP432

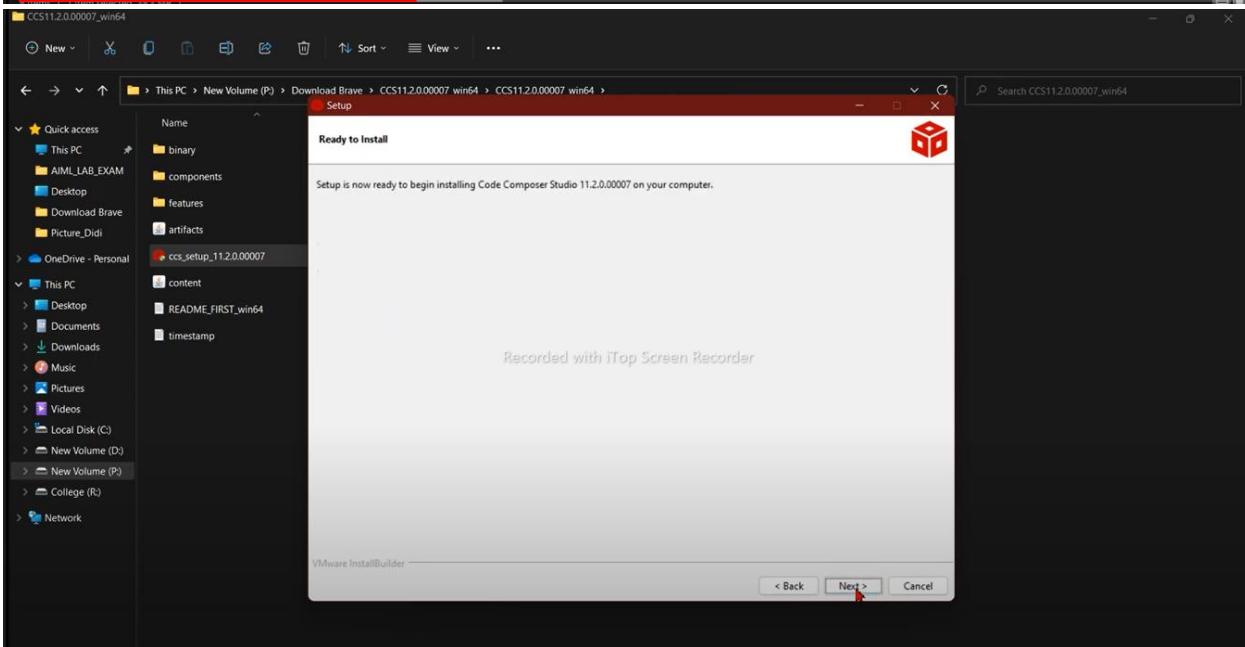
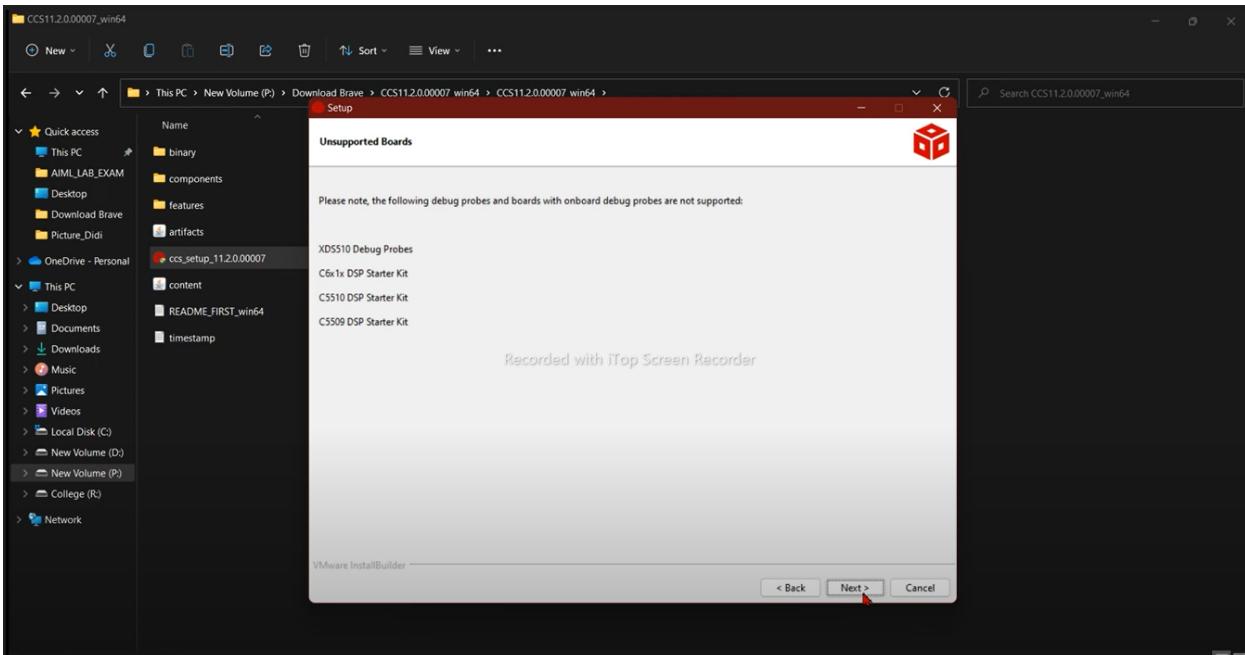
CCS INSTALLATION:

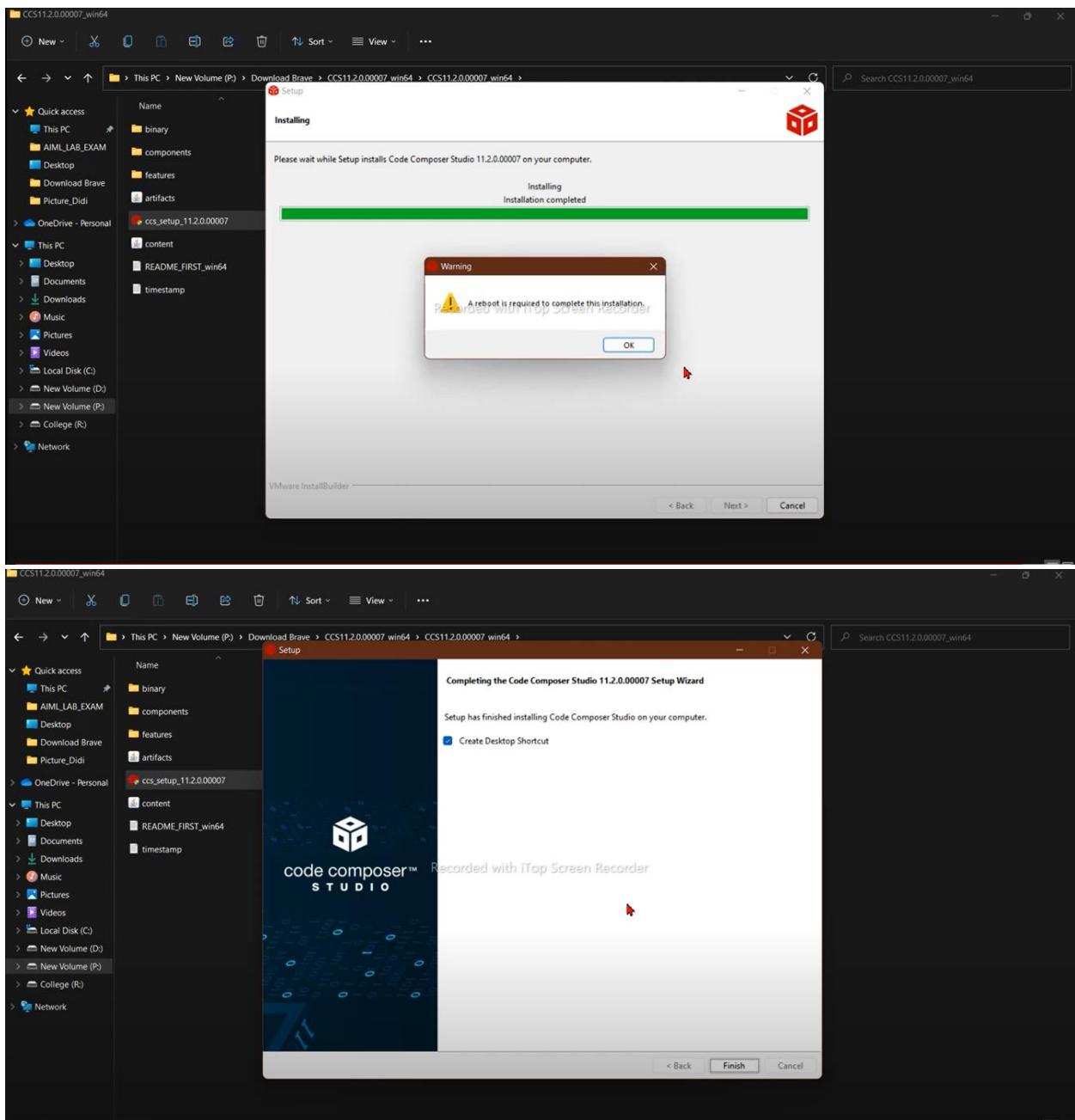






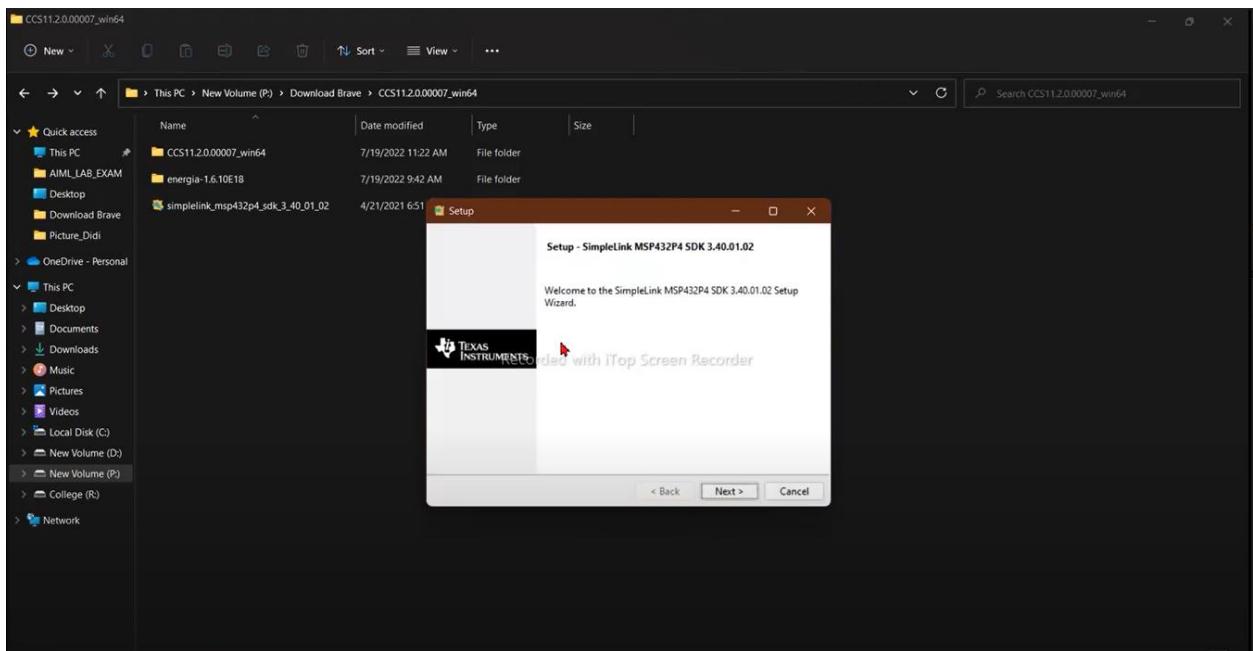


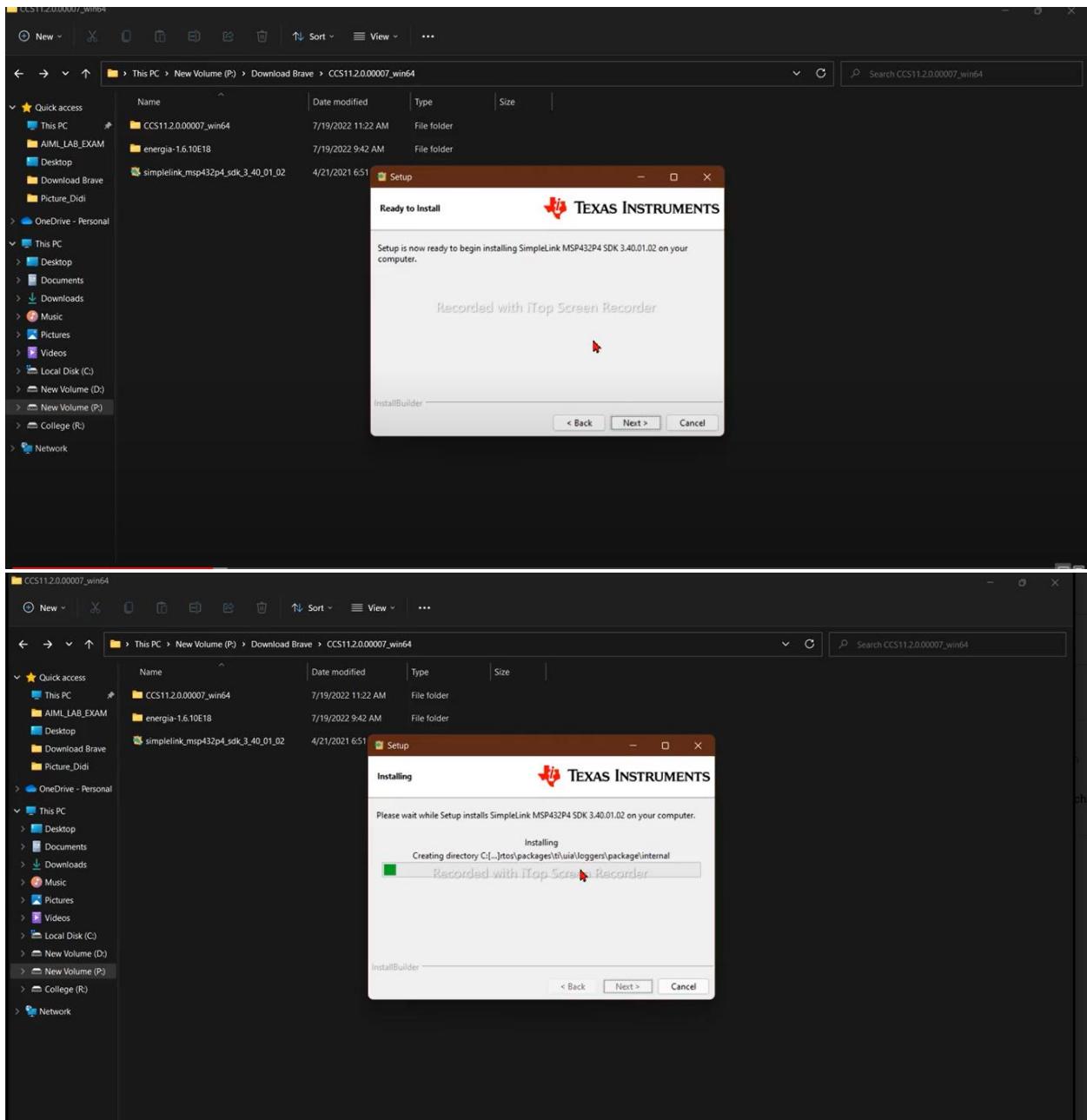


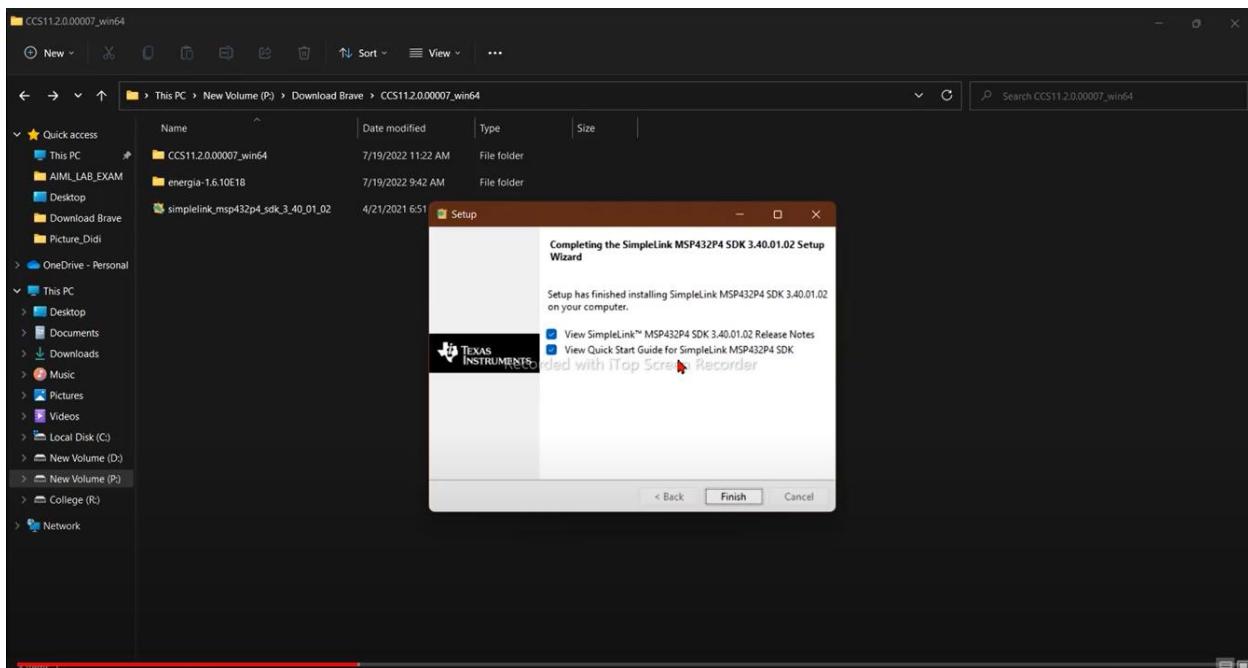




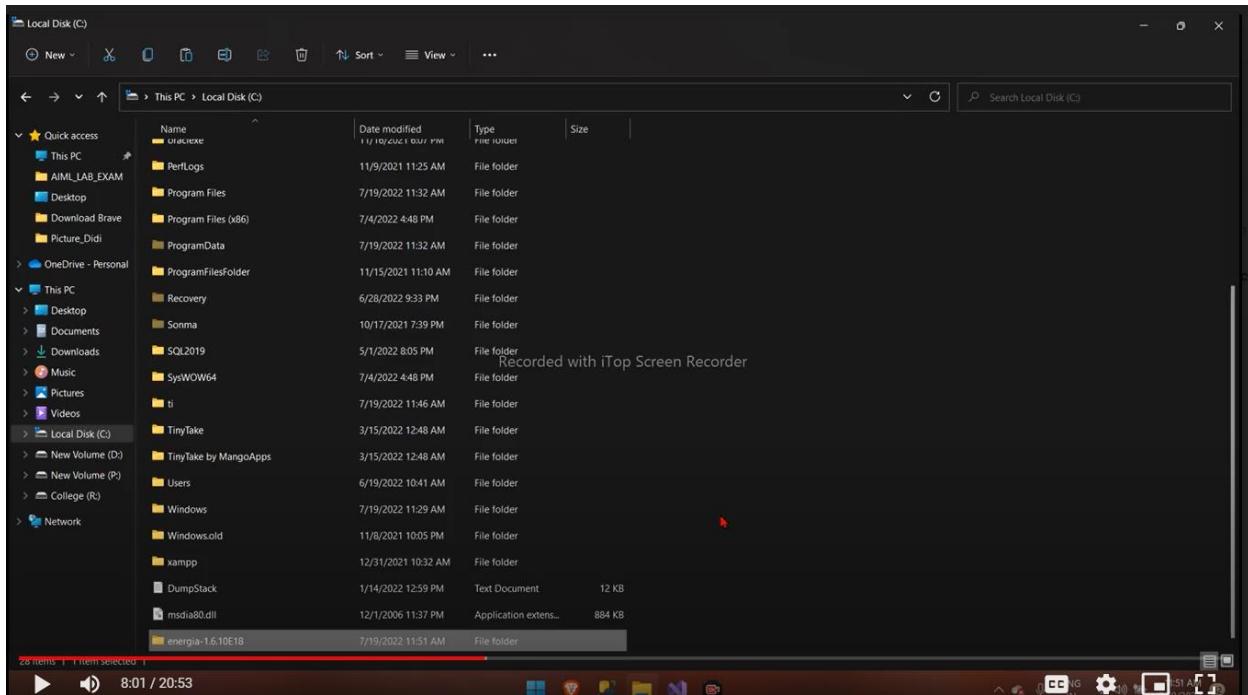
MSP432 SDK INSTALLATION

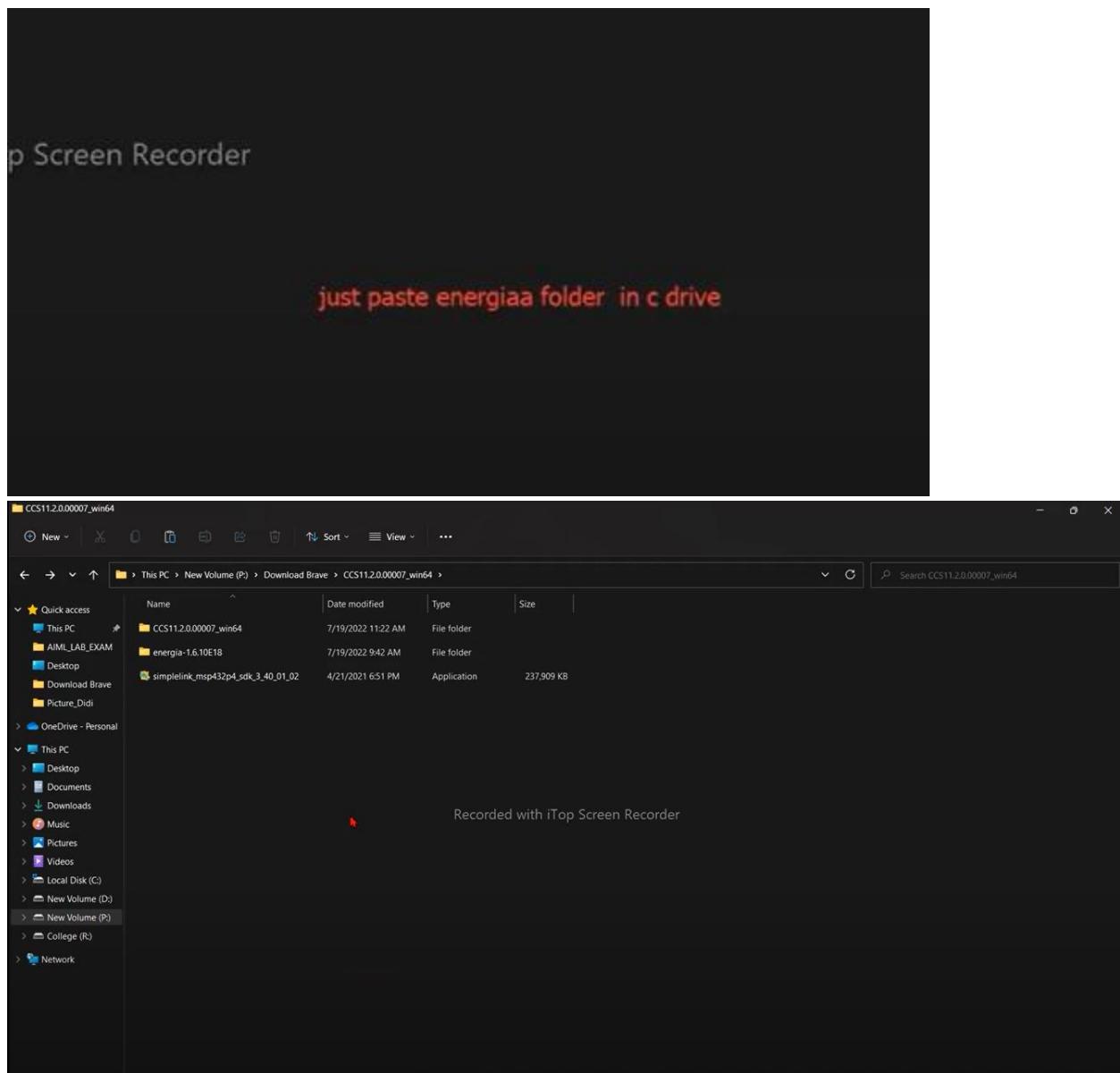


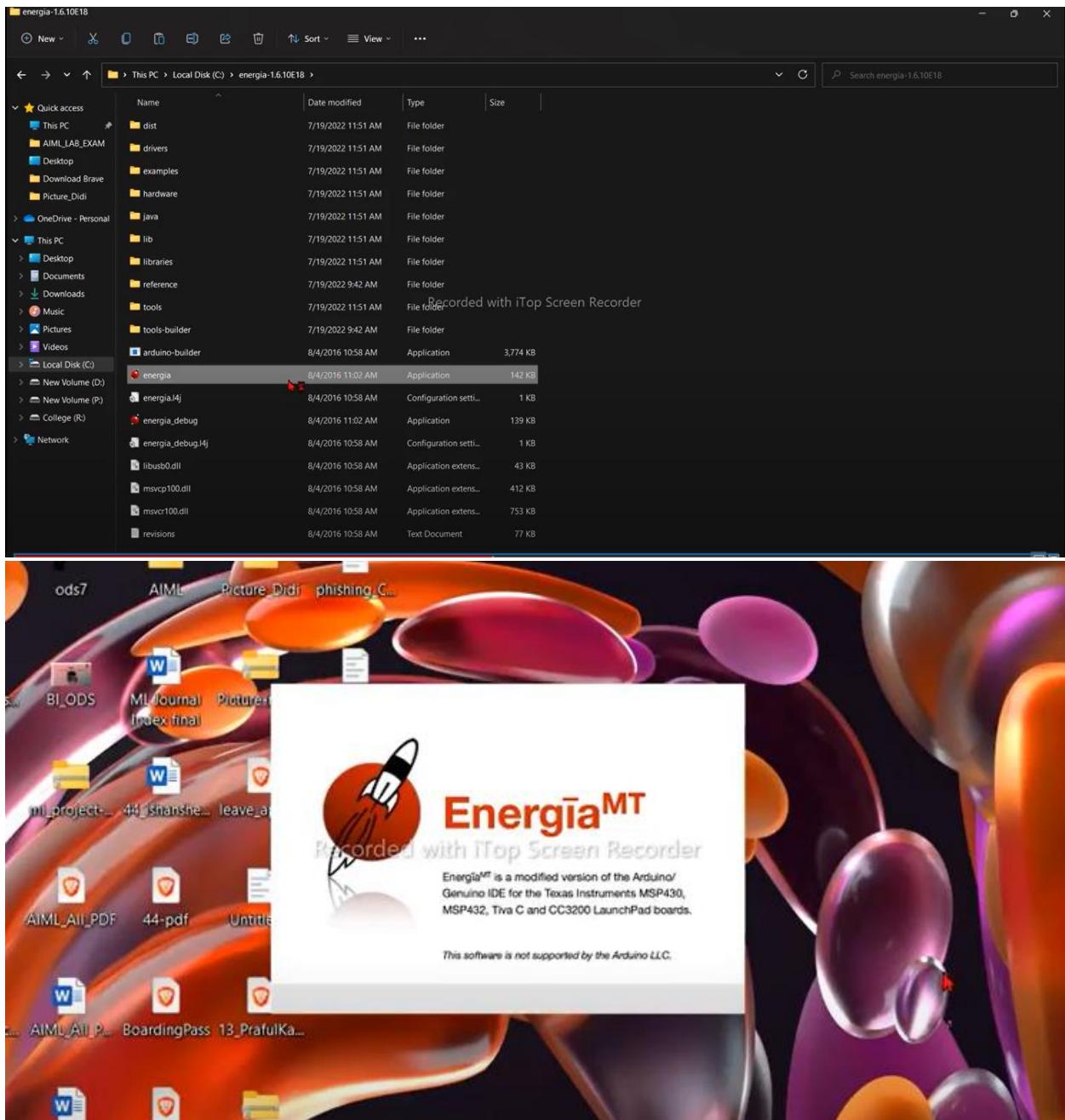


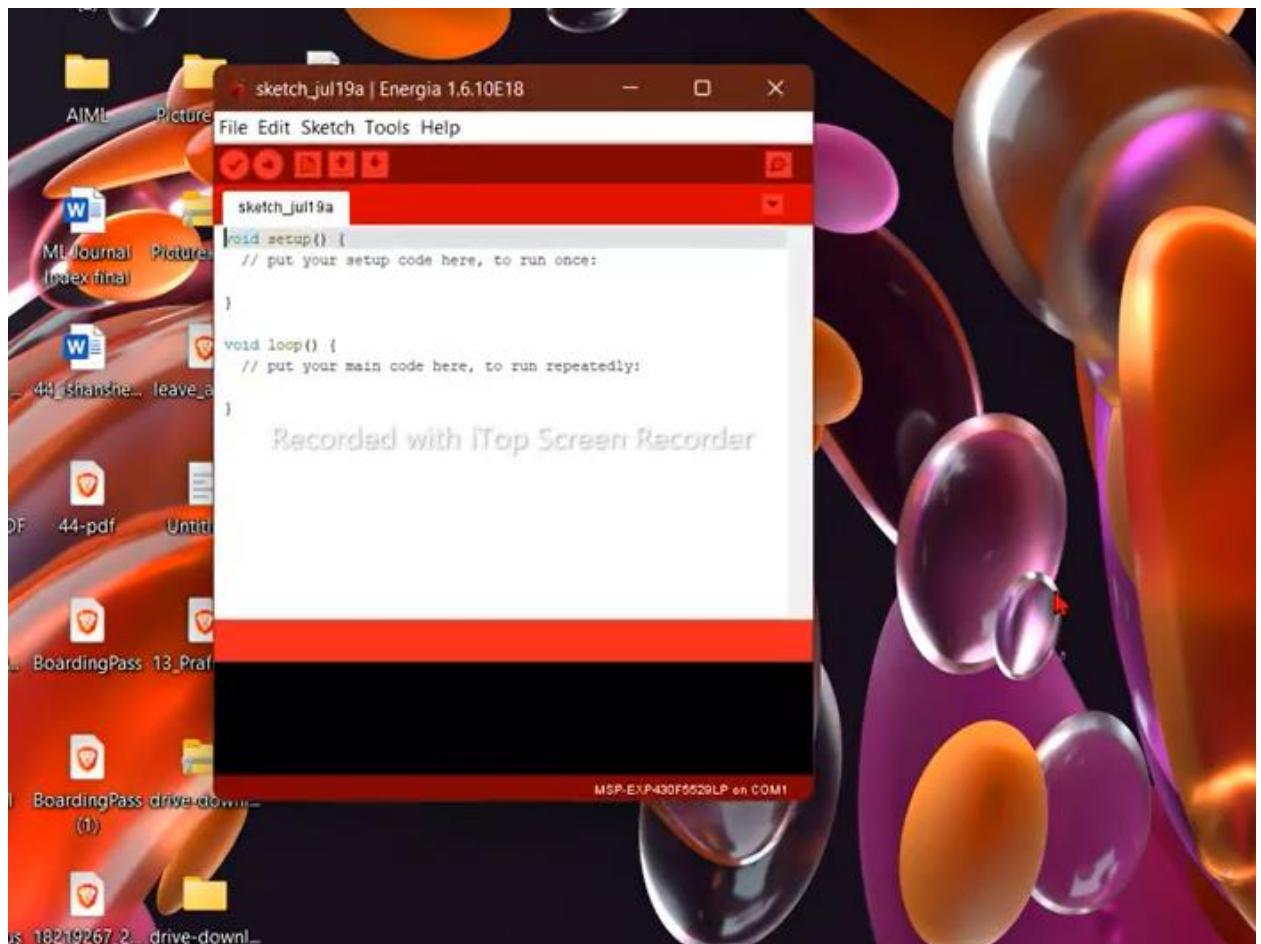


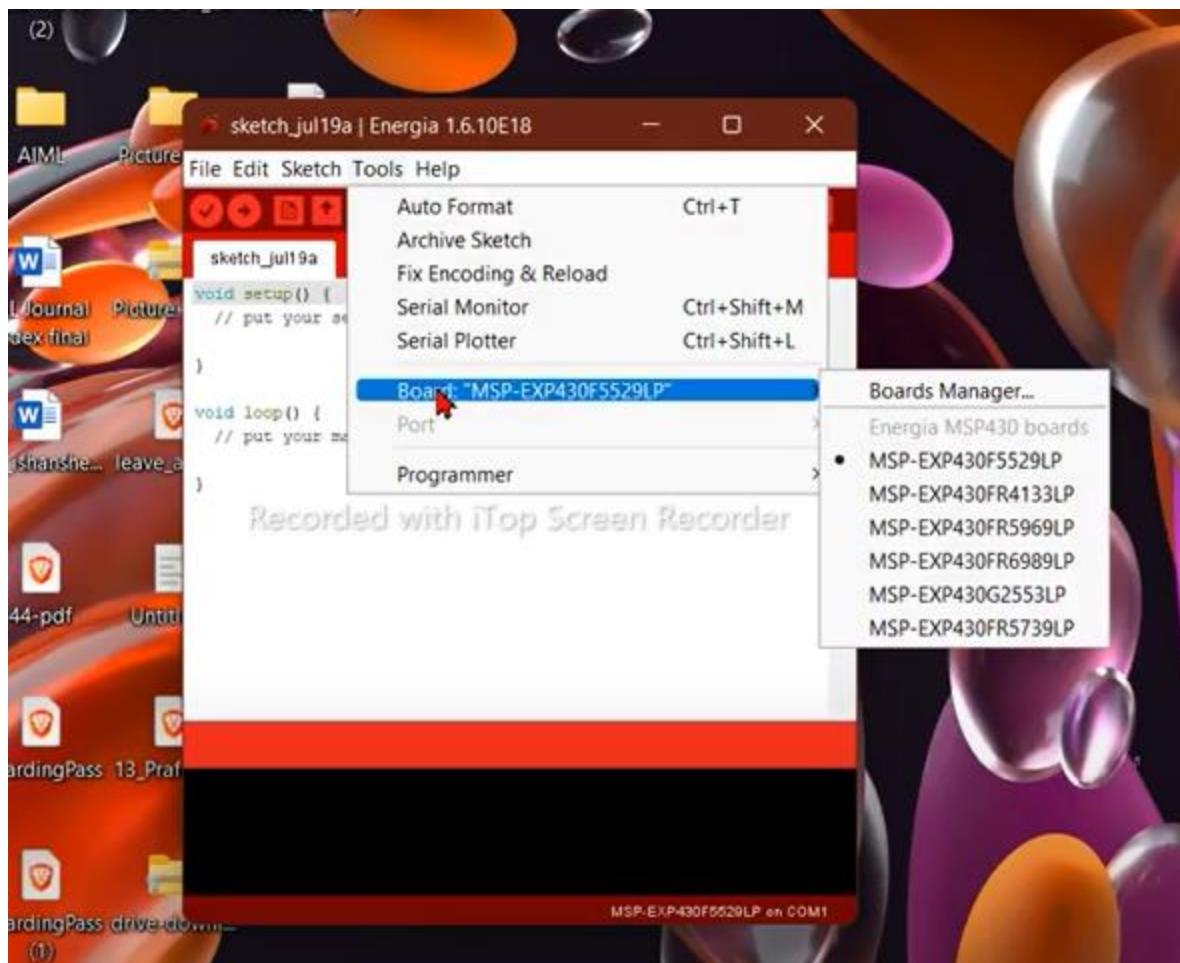
ENERGIA INSTALLATION WITH BOARD MANAGER DETAILS:

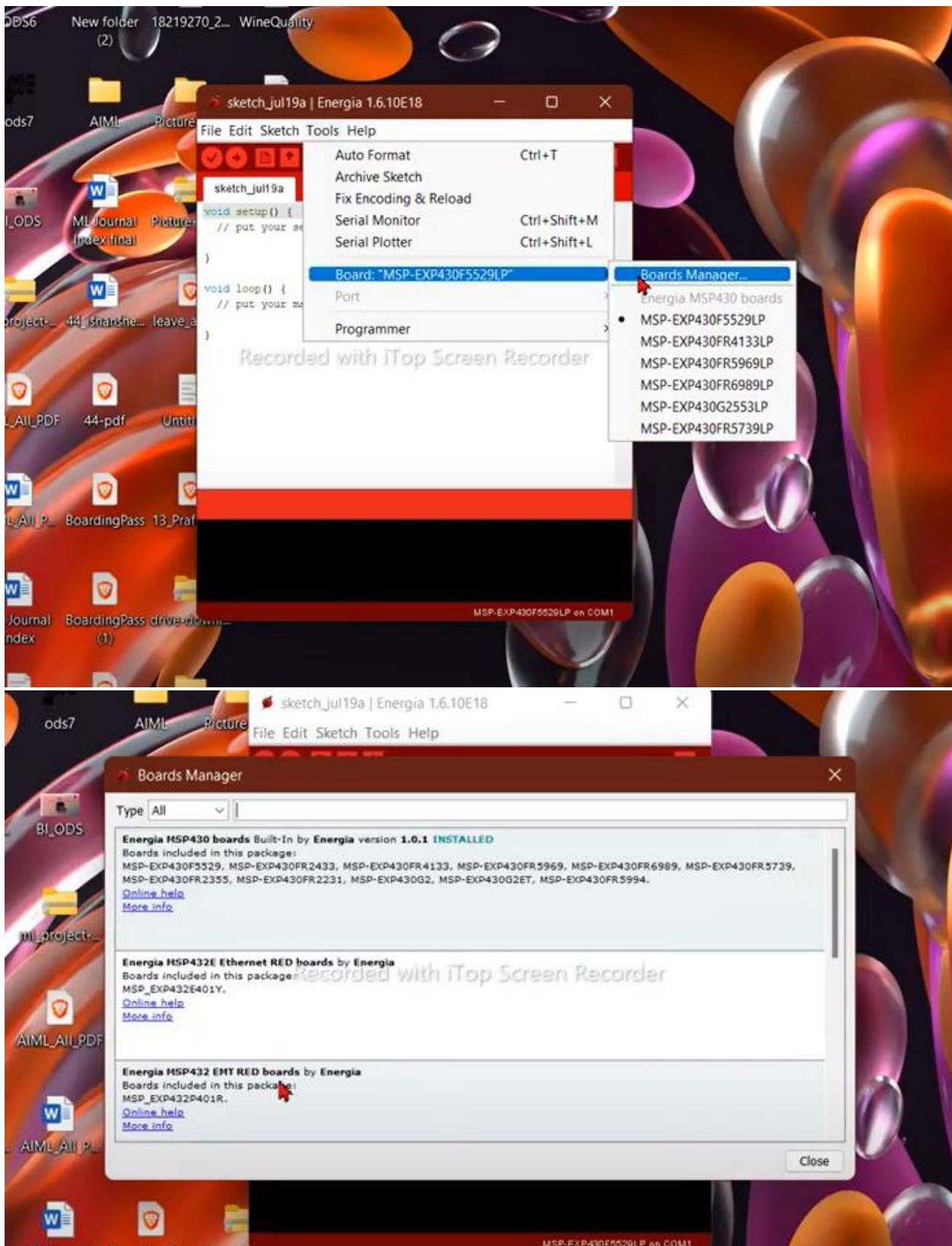


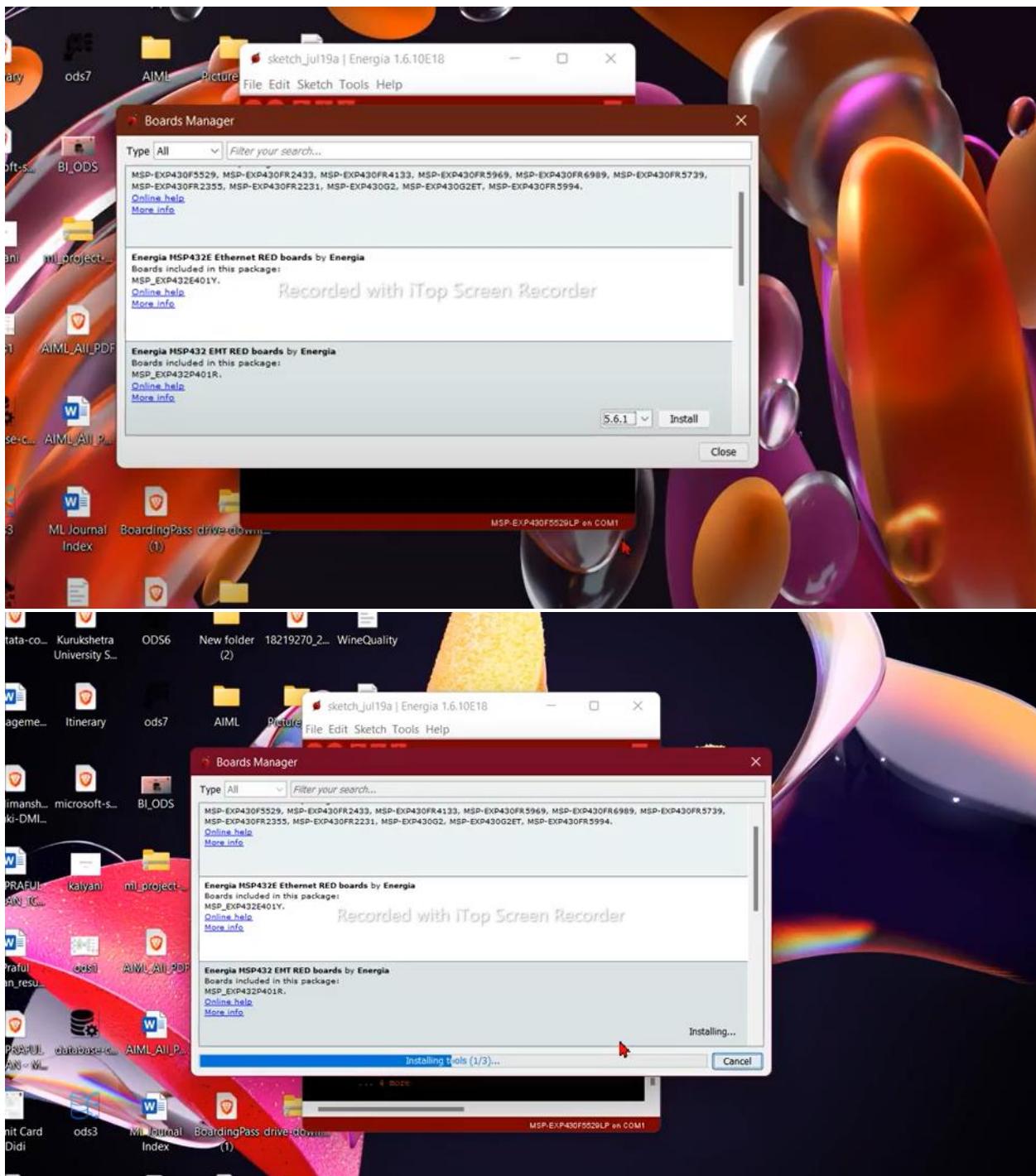












The screenshot shows the Arduino IDE interface. The title bar reads "sketch_jul19a | Energia 1.6.10E18". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for new file, open file, save file, cut, copy, paste, select all, find, and search. The main code editor window contains the following code:

```
sketch_jul19a
void setup() {
  // put your setup code here, to run once:
}

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

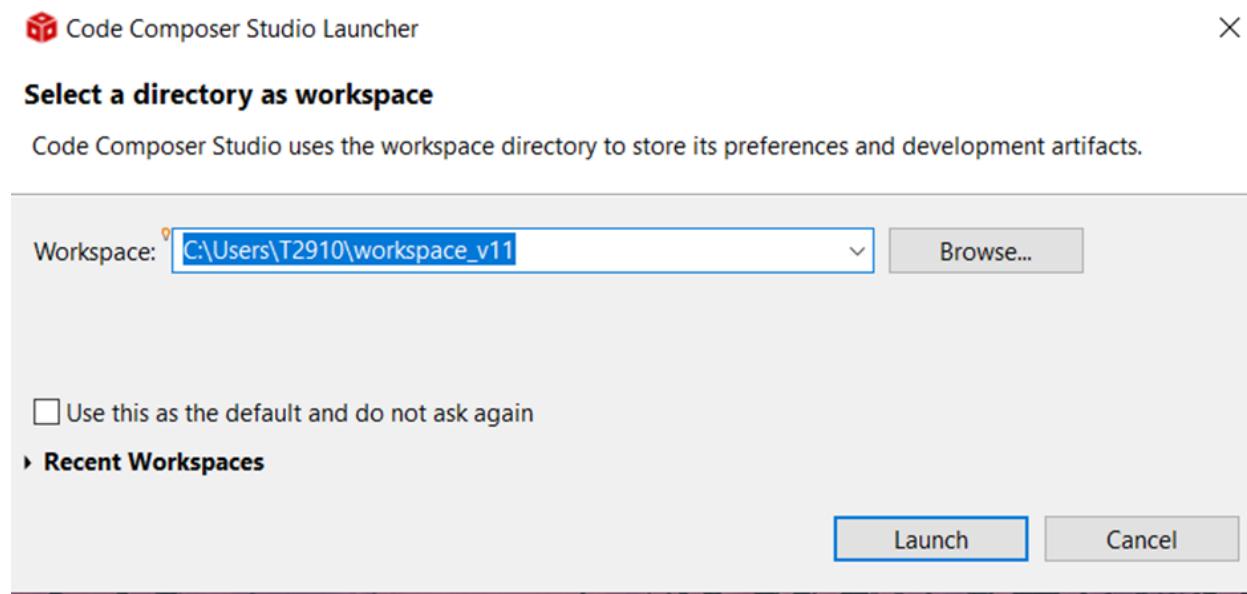
A cursor is visible in the code editor. The bottom status bar displays the path "C:\Users\Public\Documents\Arduino\sketch_jul19a" and the message "... 4 more".

Practical 2

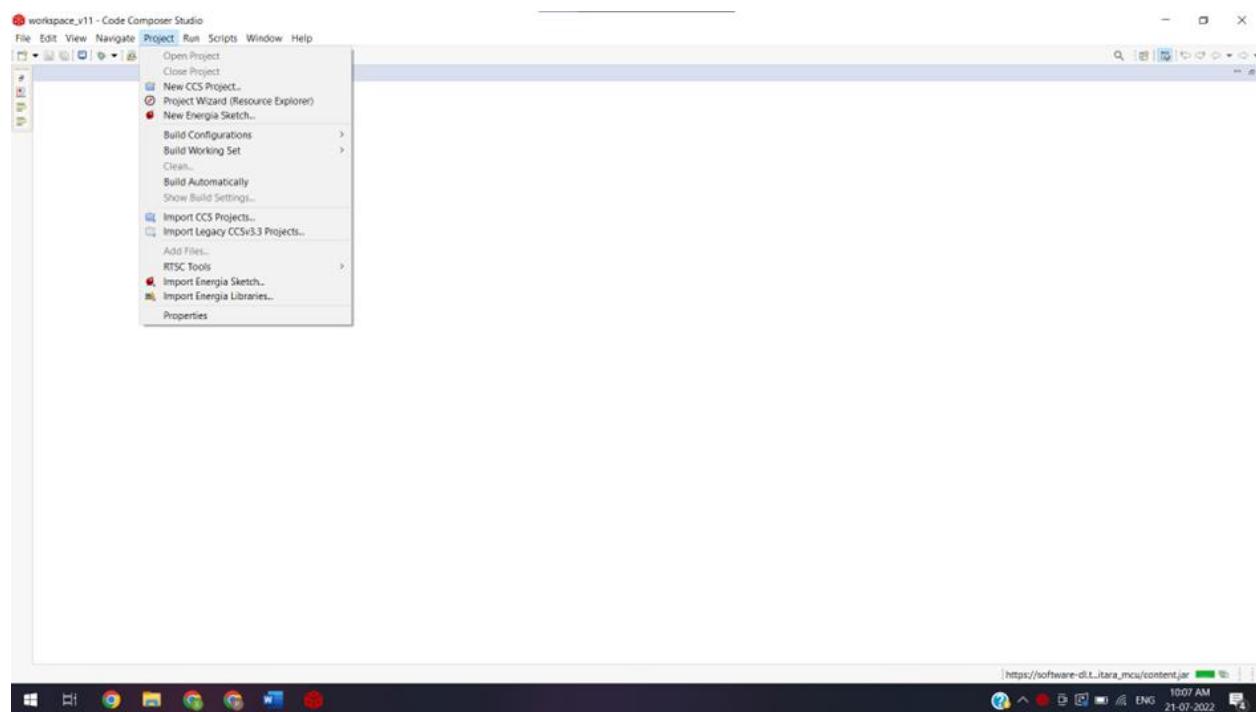
Practical Name: Hello world of Embedded Systems

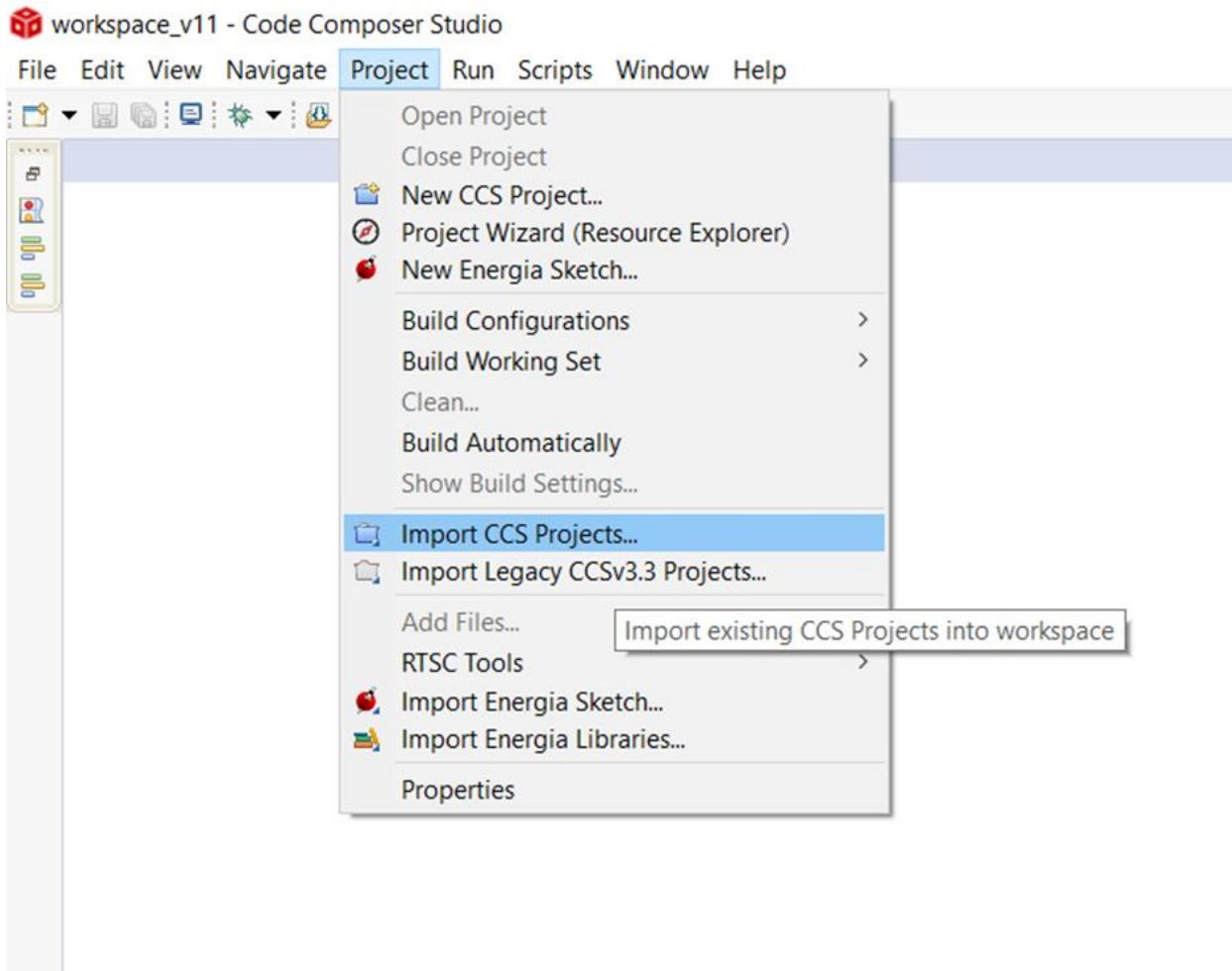
Open Code Composer Studio Launcher

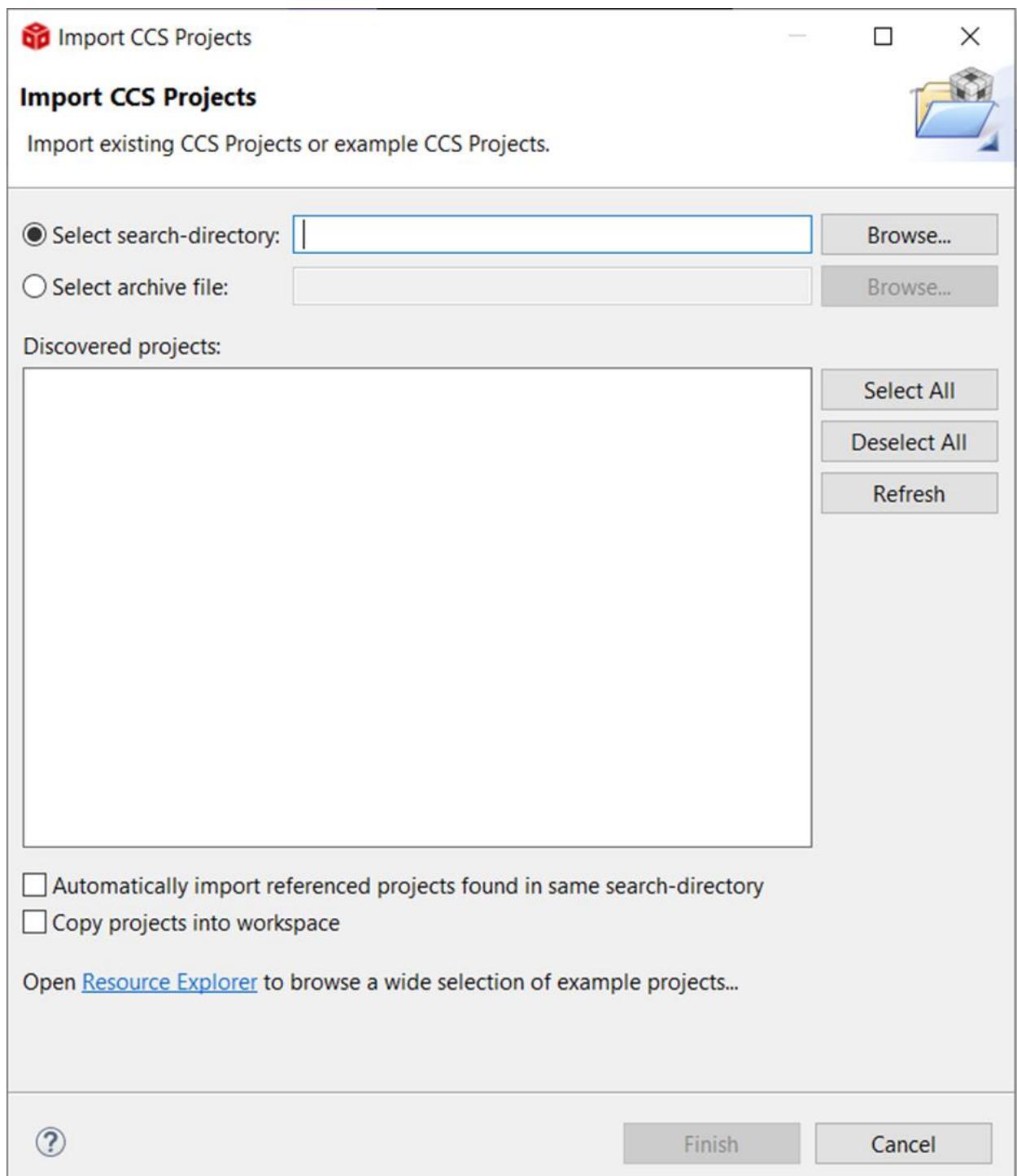
Keep the workspace path by default

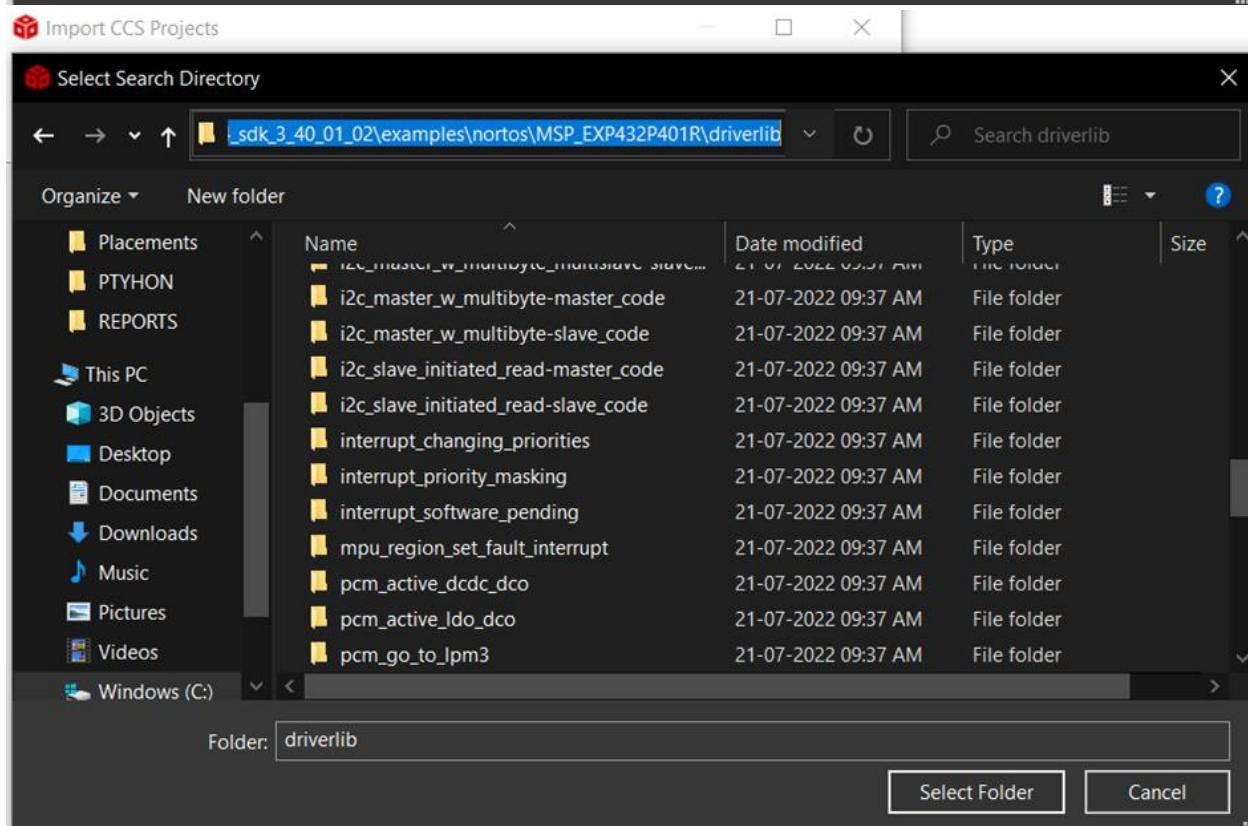
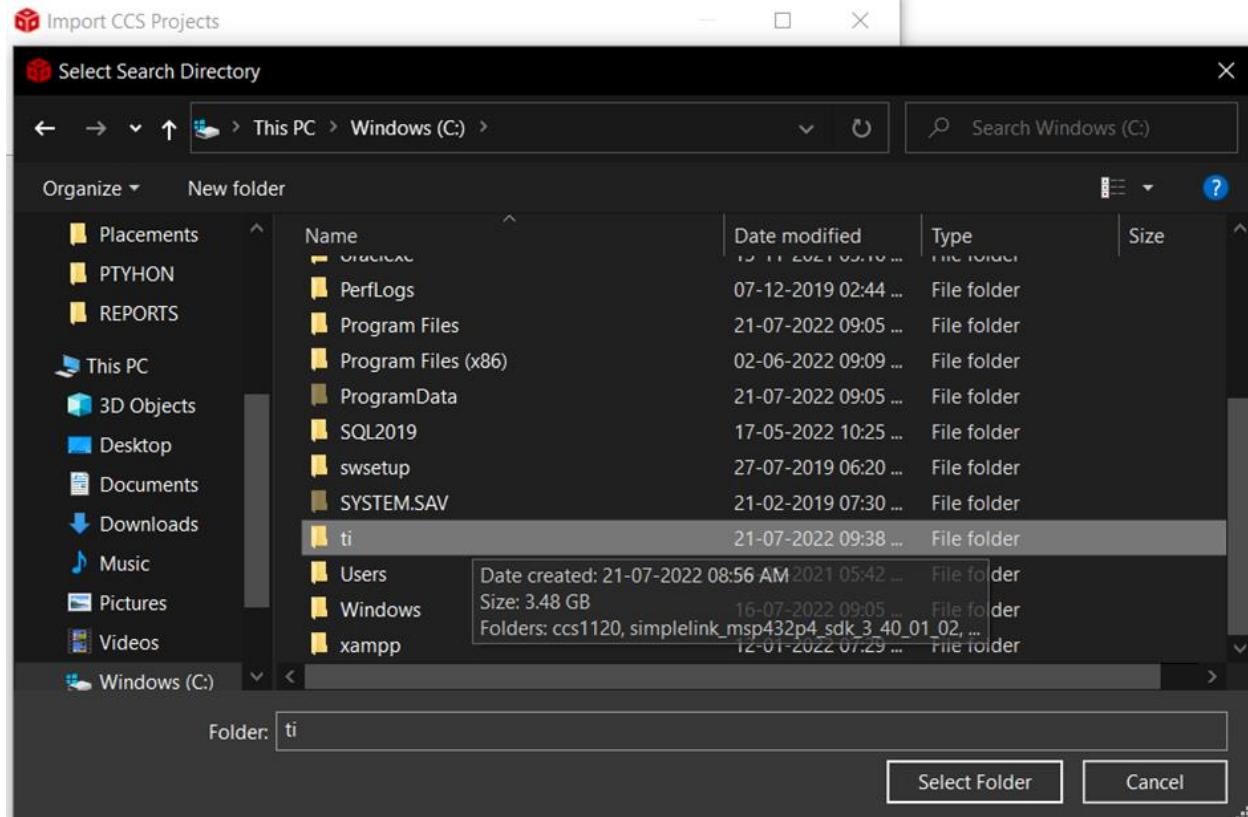


Select the Project Menu->Import CCS Project

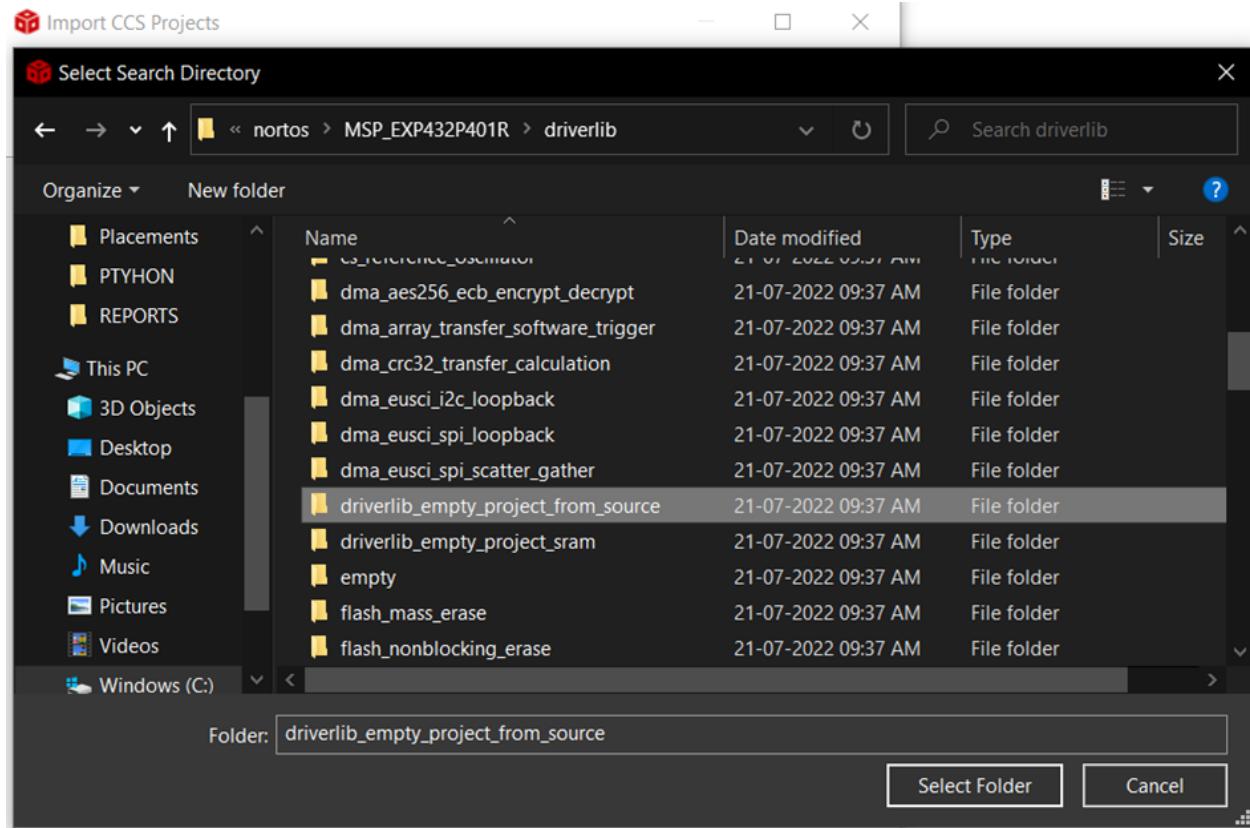


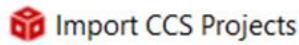






Path:C:\ti\simplelink_msp432p4_sdk_3_40_01_02\examples\nortos\MSP_EXP432P401R\driverlib





Import CCS Projects

Import existing CCS Projects or example CCS Projects.



Select search-directory:

[Browse...](#)

Select archive file:

[Browse...](#)

Discovered projects:

- | | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | driverlib_empty_project_from_source_MSP_EXP432P401R_nortos_ccs [c] |
| <input type="checkbox"/> | driverlib_empty_project_from_source_MSP_EXP432P401R_nortos_gcc [c] |

[Select All](#)

[Deselect All](#)

[Refresh](#)



Automatically import referenced projects found in same search-directory

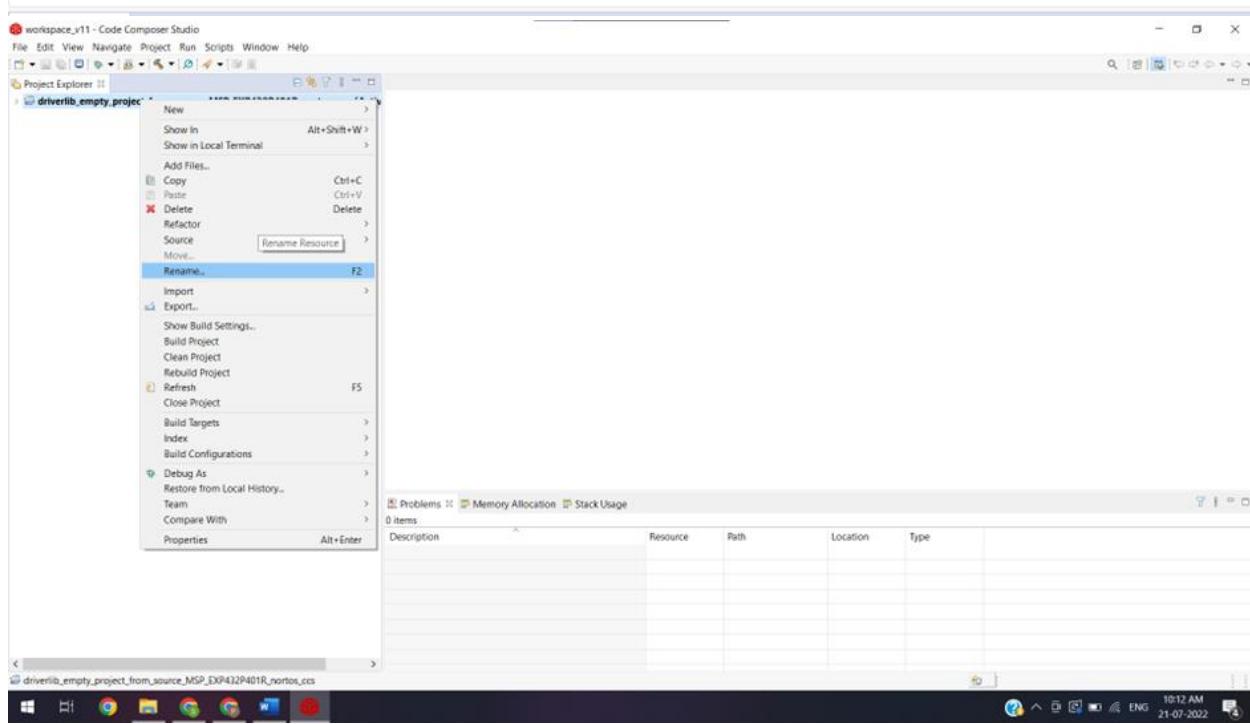
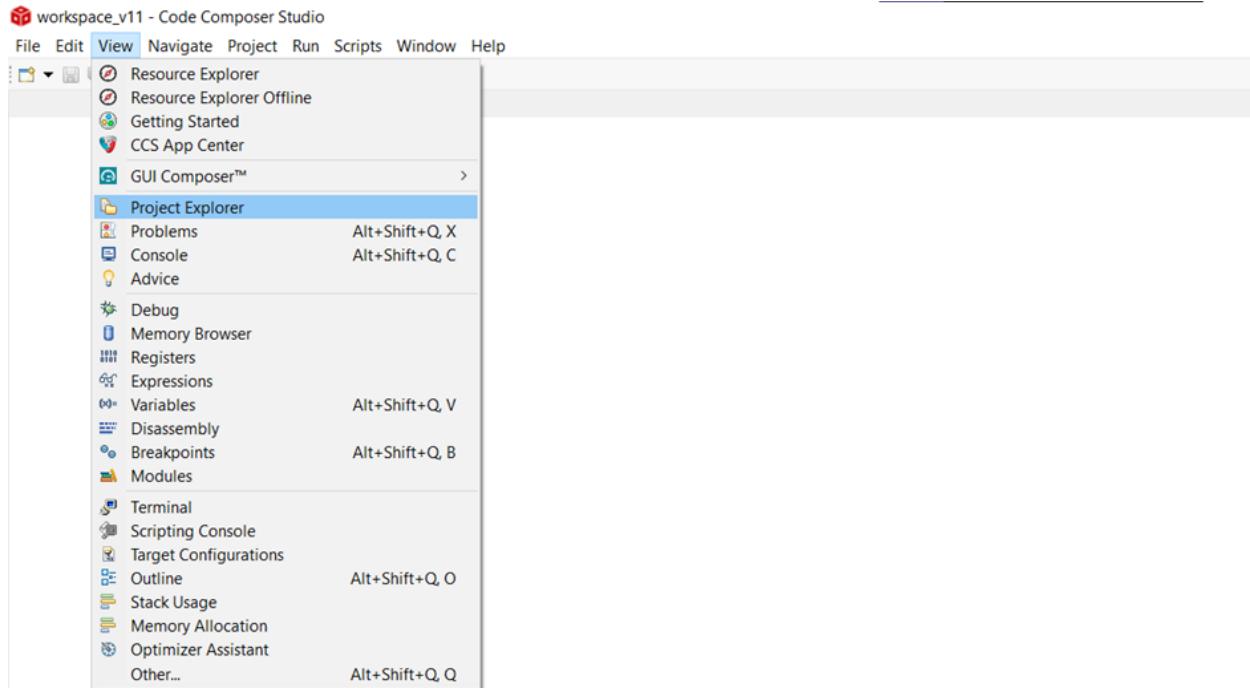
Copy projects into workspace

Open [Resource Explorer](#) to browse a wide selection of example projects...

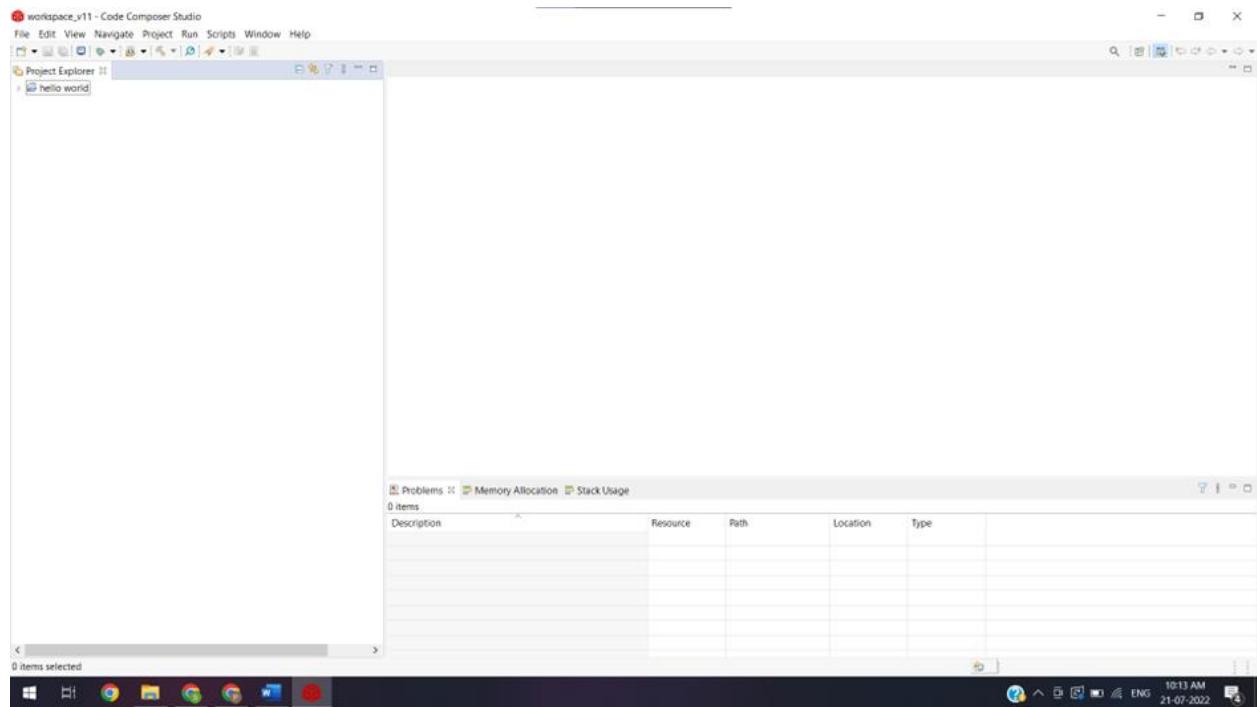
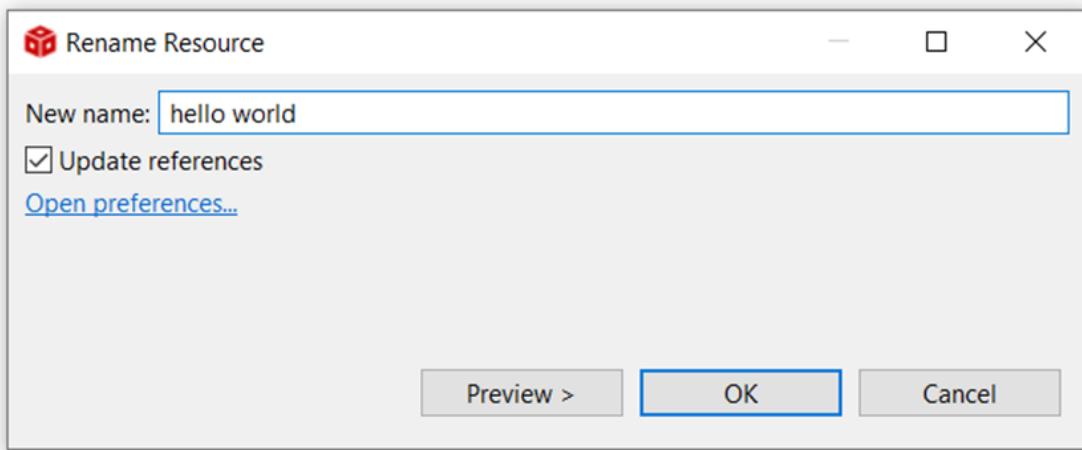


[Finish](#)

[Cancel](#)



Give Project Name HELLO WORLD



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

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

int main(void)
{
    /* Stop Watchdog */
```

```
MAP_WDT_A_holdTimer(); //watch dog timer , this timer starts when kit is given power
// if it stops it will cause kit to reset
int i;
GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0);
// port is combination of 8 pins 0-7 , means 8 bits , 8 bits forms a port
// this MC has 10 ports
// this func tells mc to make PORT2, pin 0 as OUTPUT pin
// GPIO means general purpose input output

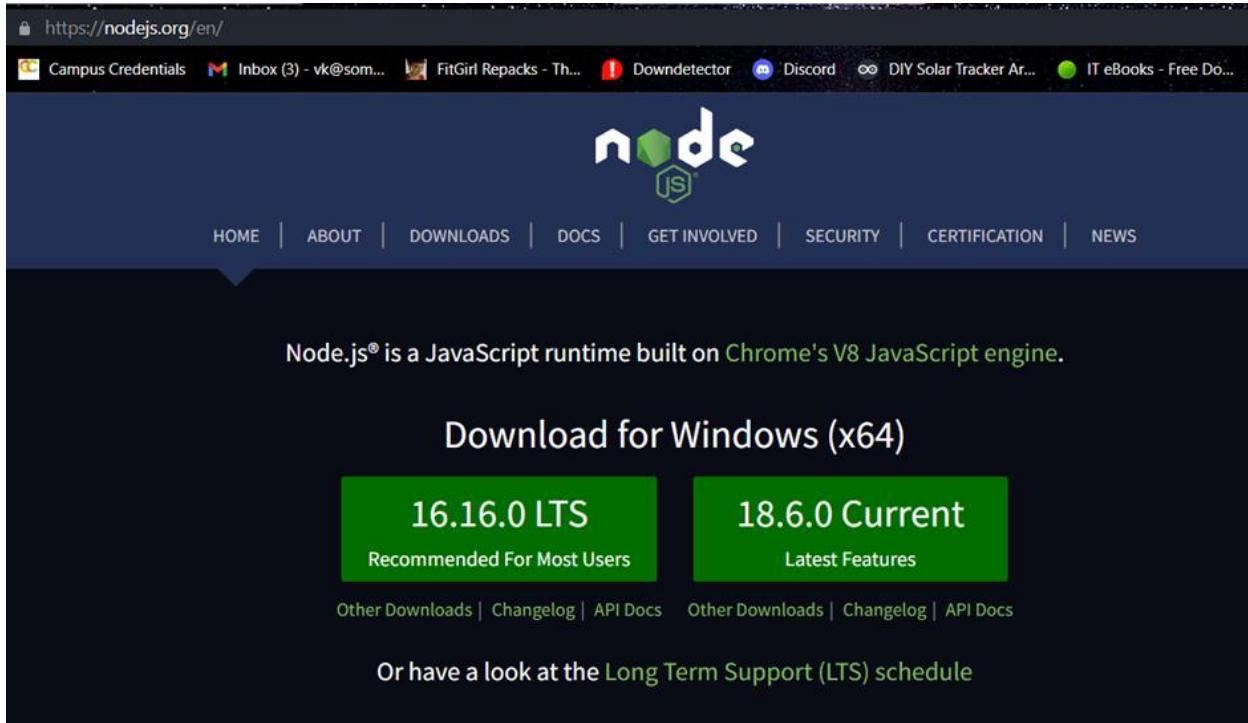
while(1) // super loop
{
    GPIO_setOutputHighOnPin(GPIO_PORT_P2,GPIO_PIN0); //setting the pin as HIGH
    for(i=0;i<10000;i++); // random delay
    GPIO_setOutputLowOnPin(GPIO_PORT_P2,GPIO_PIN0); //setting the pin as LOW
    for(i=0;i<100000;i++); //random delay
}
}
```

Practical 3

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

Visit the URL: <https://nodejs.org/en/>

Download the nodejs for Windows(x64)



Open the Command Prompt

Enter the following commands to install Node.js

Node --version

Install g --unsafe-perm node-red

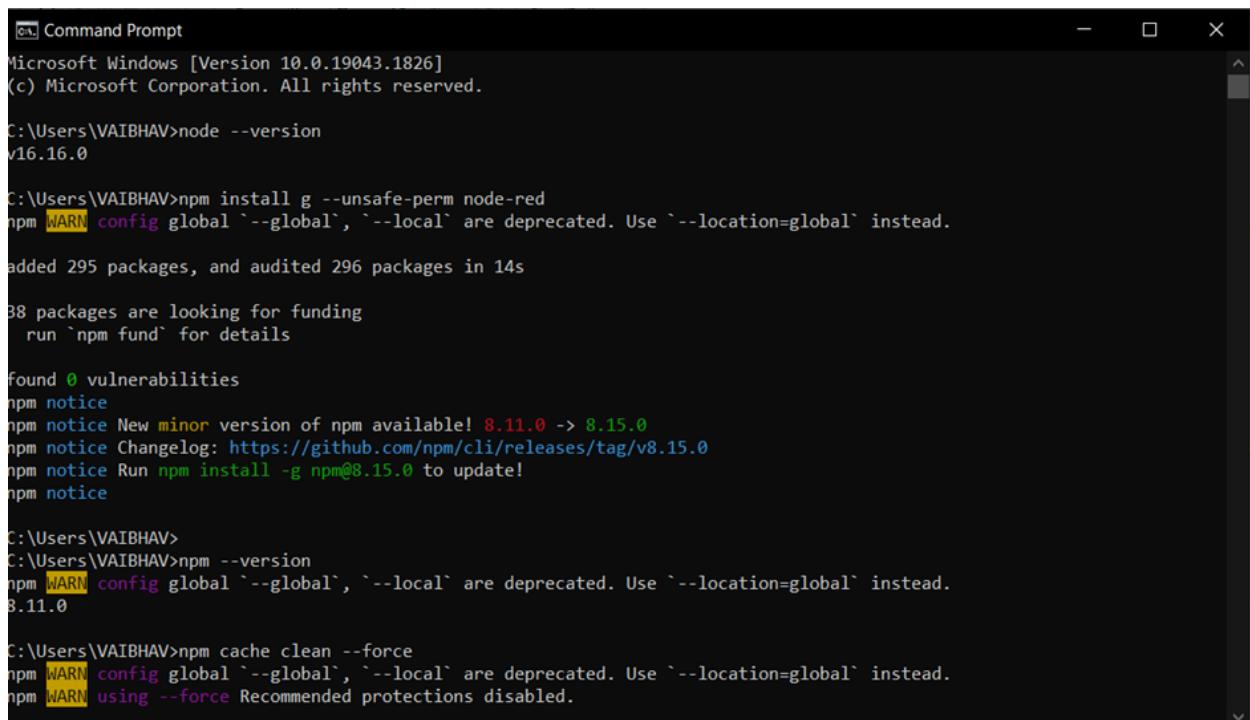
npm --version

npm cache clean --force

node -v

node -red

Then open the browser and start node red at the port 1880 and put the screen shot of the same with the URL used in the web browser.



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\VAIBHAV>node --version
v16.16.0

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

added 295 packages, and audited 296 packages in 14s

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\VAIBHAV>
C:\Users\VAIBHAV>npm --version
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
8.11.0

C:\Users\VAIBHAV>npm cache clean --force
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
npm WARN using --force Recommended protections disabled.
```

```

node-red
Microsoft Windows [Version 10.0.19043.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\VAIBHAV>node -v
v16.16.0

C:\Users\VAIBHAV>node-red
26 Jul 09:02:07 - [info]

Welcome to Node-RED
=====
26 Jul 09:02:07 - [info] Node-RED version: v3.0.1
26 Jul 09:02:07 - [info] Node.js version: v16.16.0
26 Jul 09:02:07 - [info] Windows_NT 10.0.19043 x64 LE
26 Jul 09:02:09 - [info] Loading palette nodes
26 Jul 09:02:12 - [info] Settings file : C:\Users\VAIBHAV\.node-red\settings.js
26 Jul 09:02:12 - [info] Context store : 'default' [module=memory]
26 Jul 09:02:12 - [info] User directory : \Users\VAIBHAV\.node-red
26 Jul 09:02:12 - [warn] Projects disabled : editorTheme.projects.enabled=false
26 Jul 09:02:12 - [info] Flows file : \Users\VAIBHAV\.node-red\flows.json
26 Jul 09:02:12 - [info] Server now running at http://127.0.0.1:1880/
26 Jul 09:02:12 - [warn]

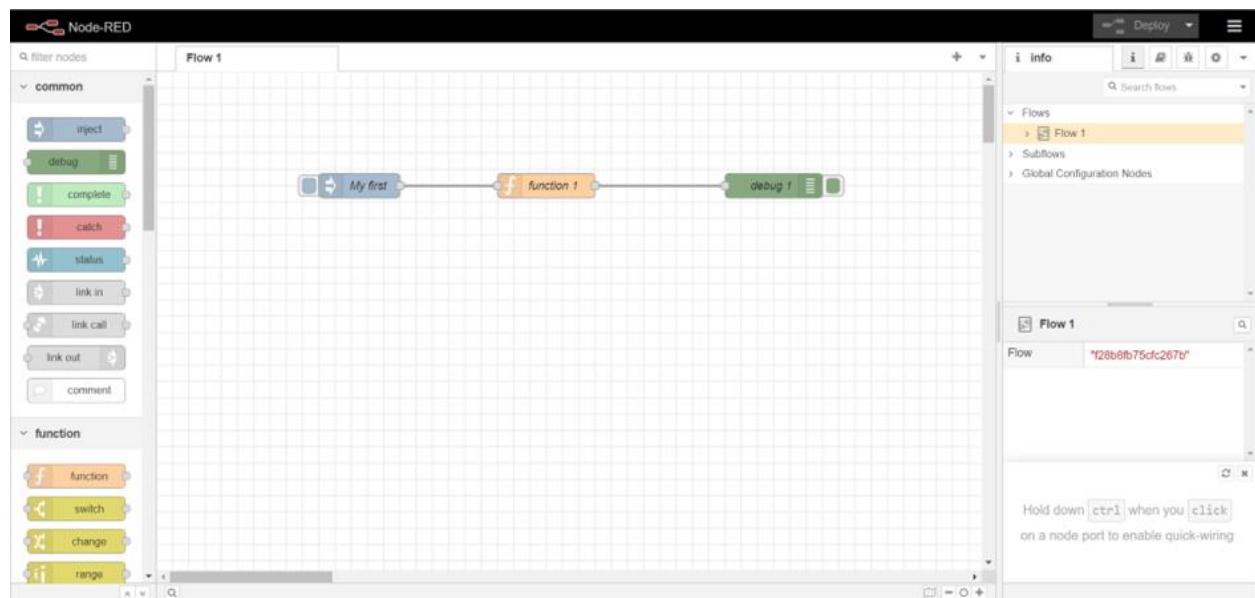
-----
Your flow credentials file is encrypted using a system-generated key.

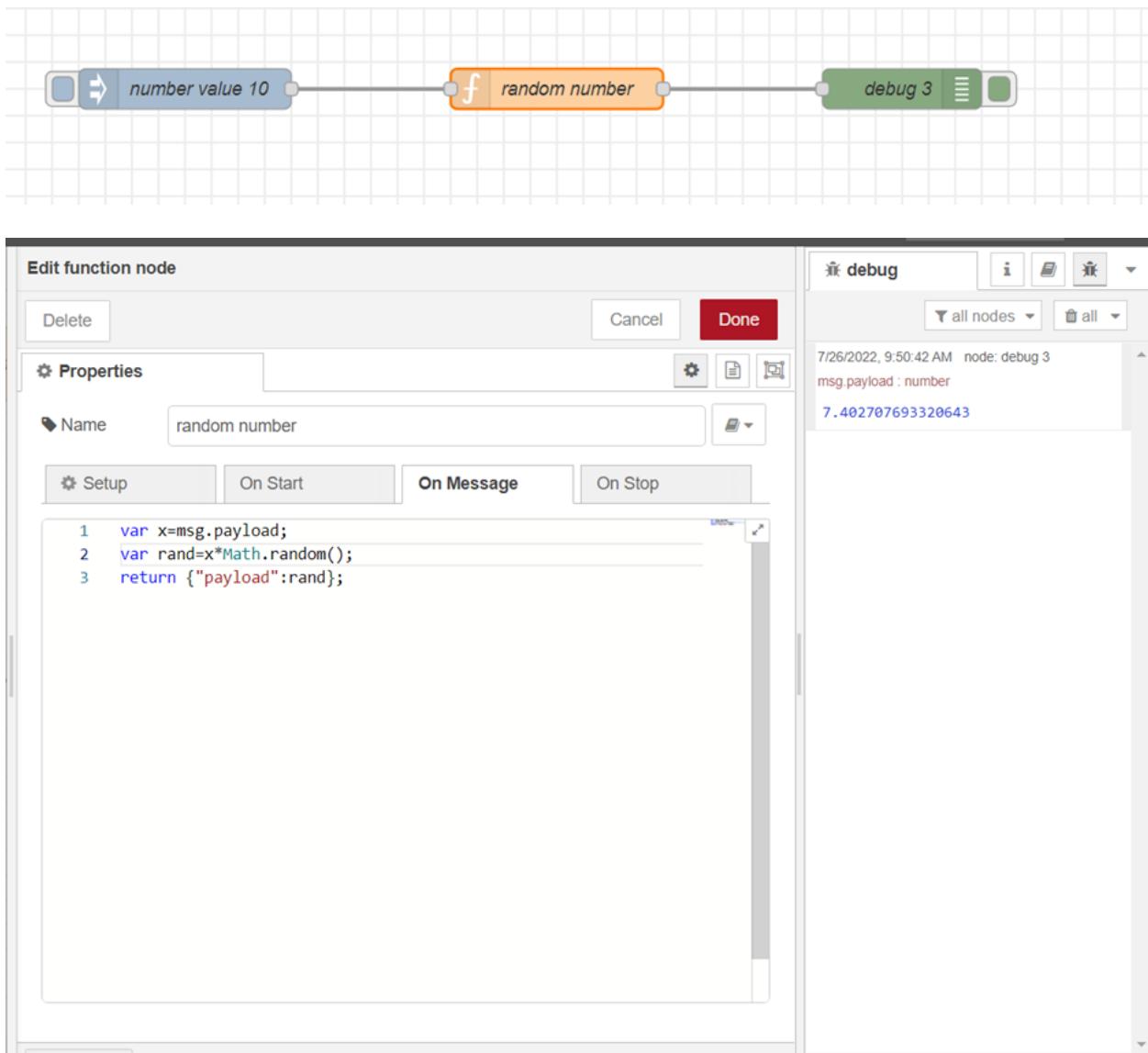
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

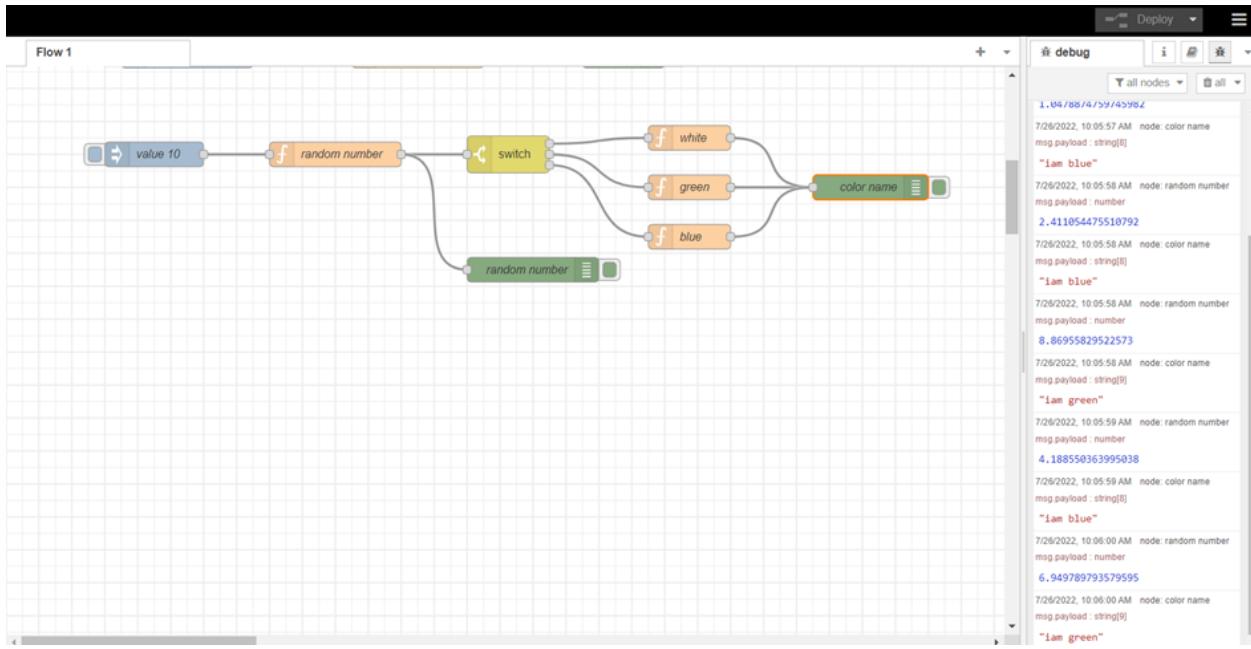
```

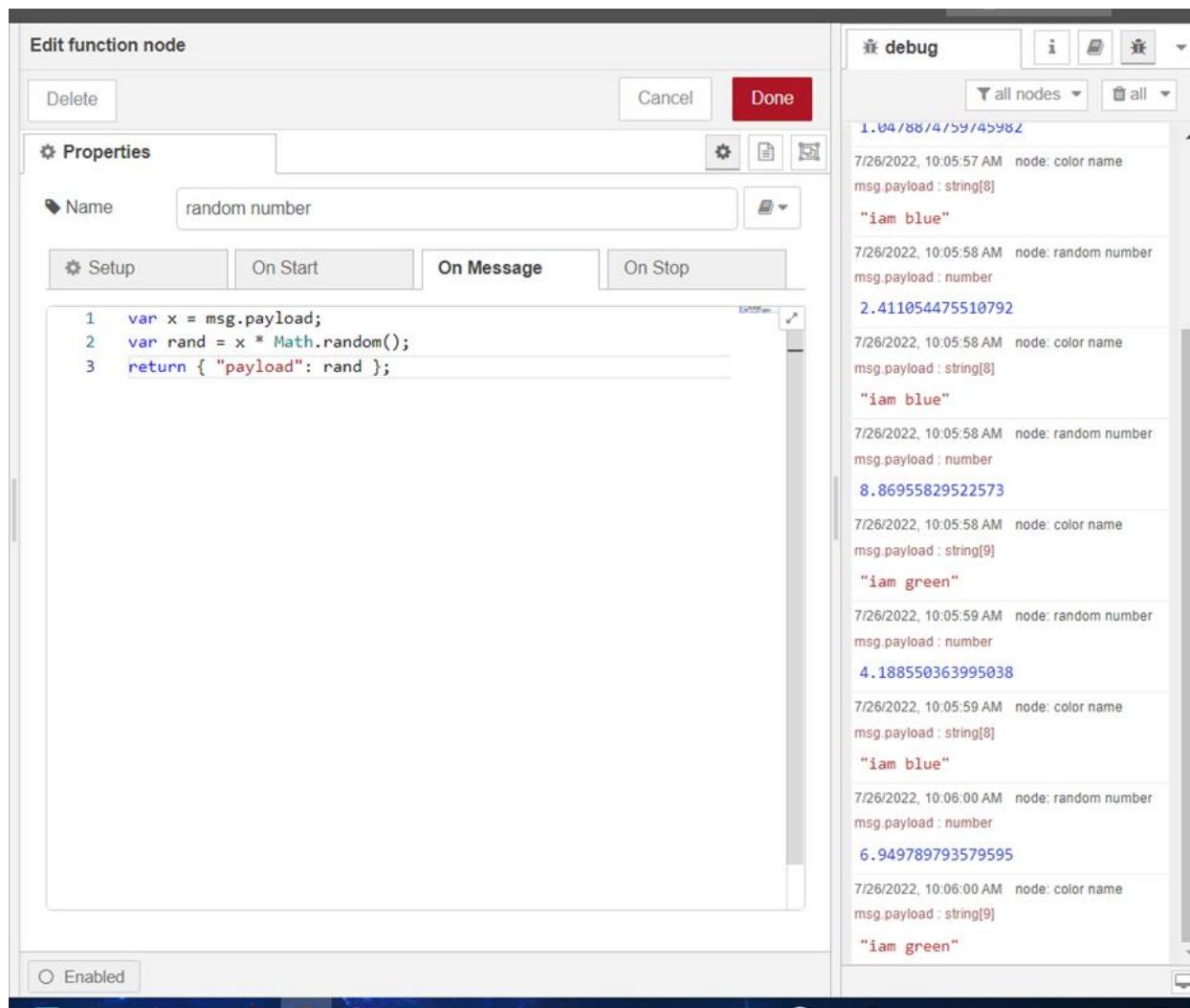
Then open the browser and start node red at the port 1880

<https://127.0.0.1:1880/>





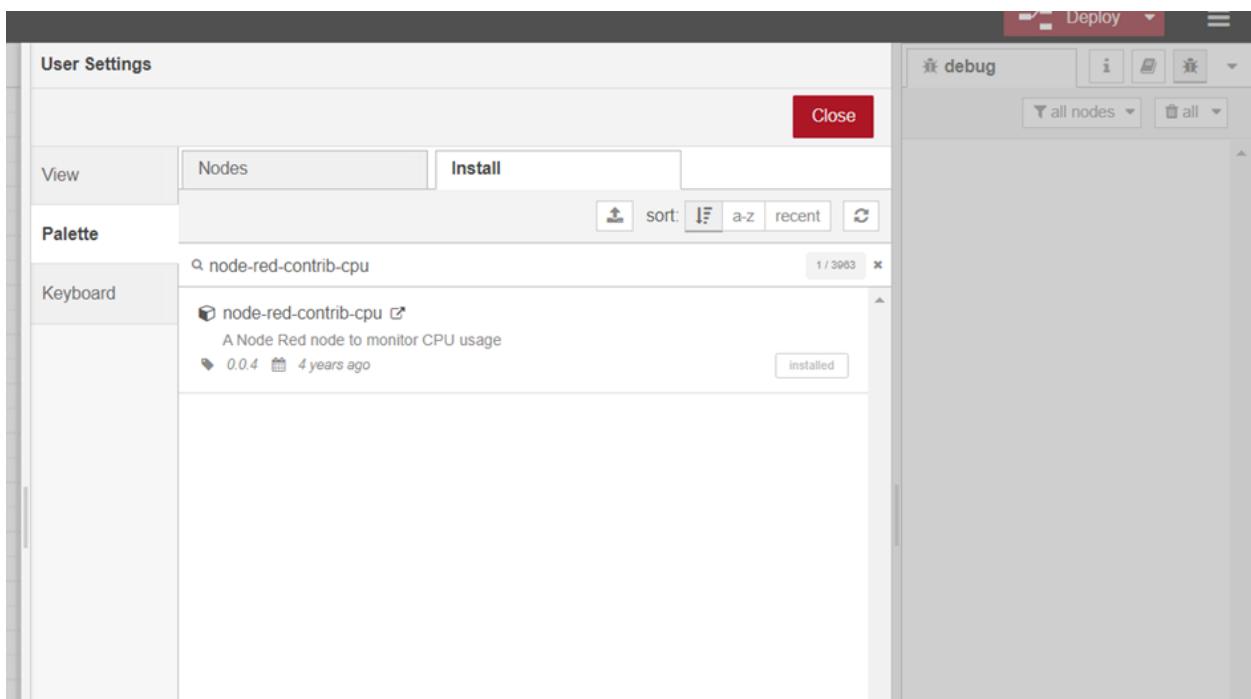
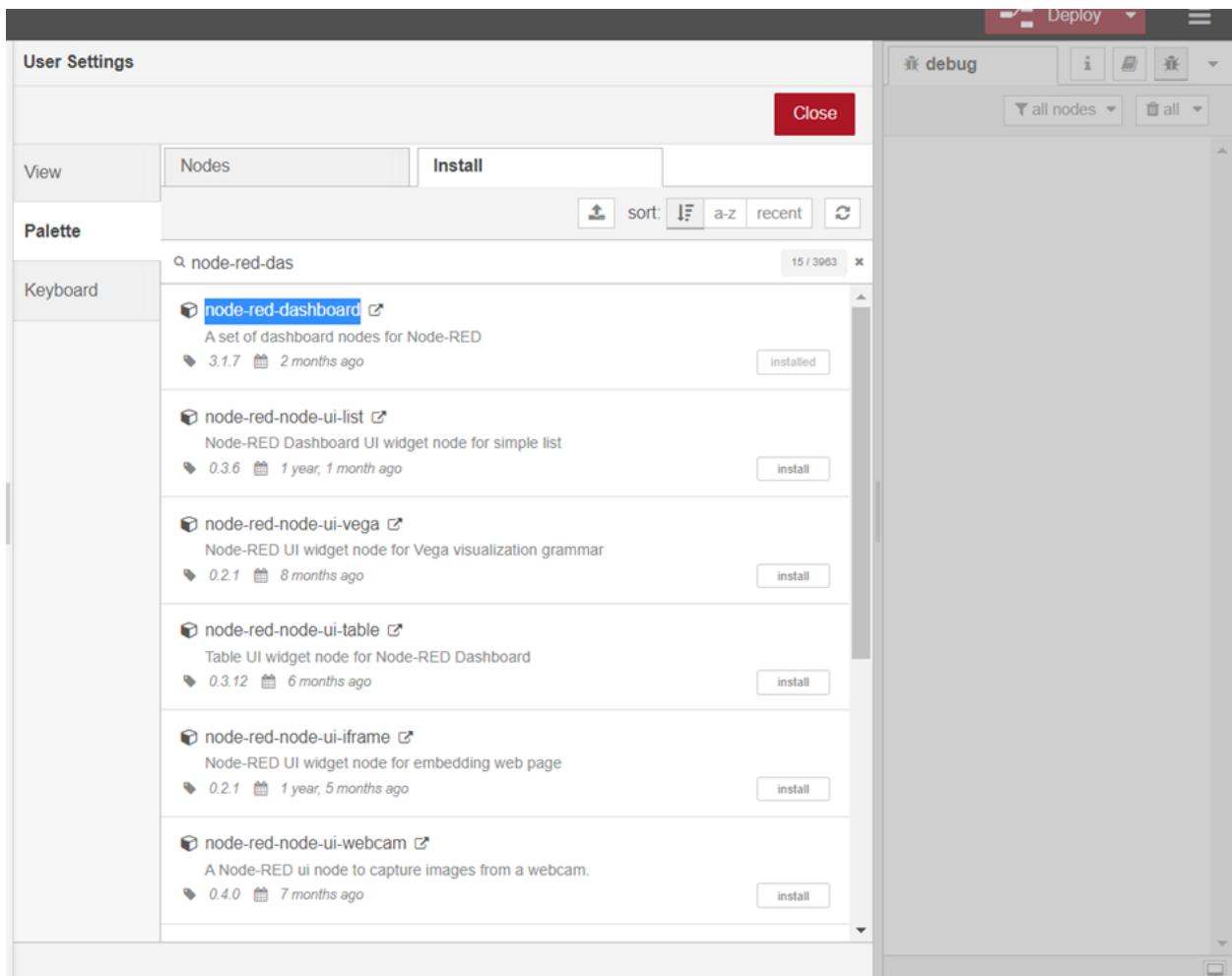


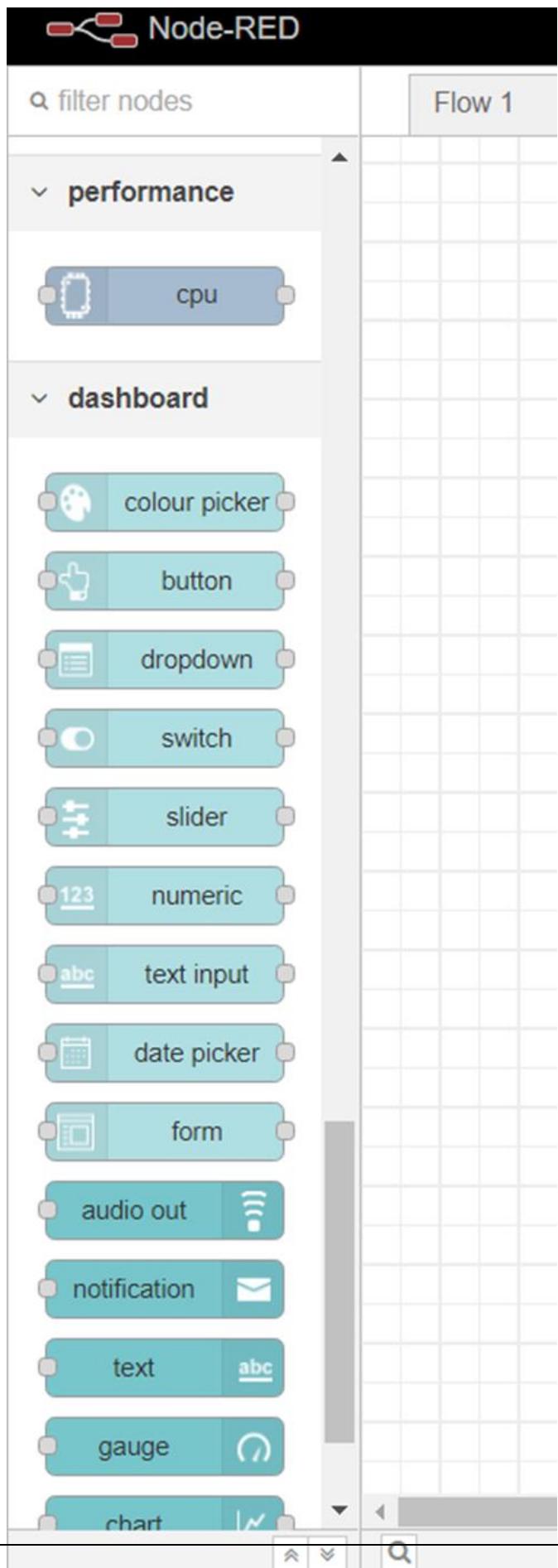


Install pallet

node-red-contrib-cpu

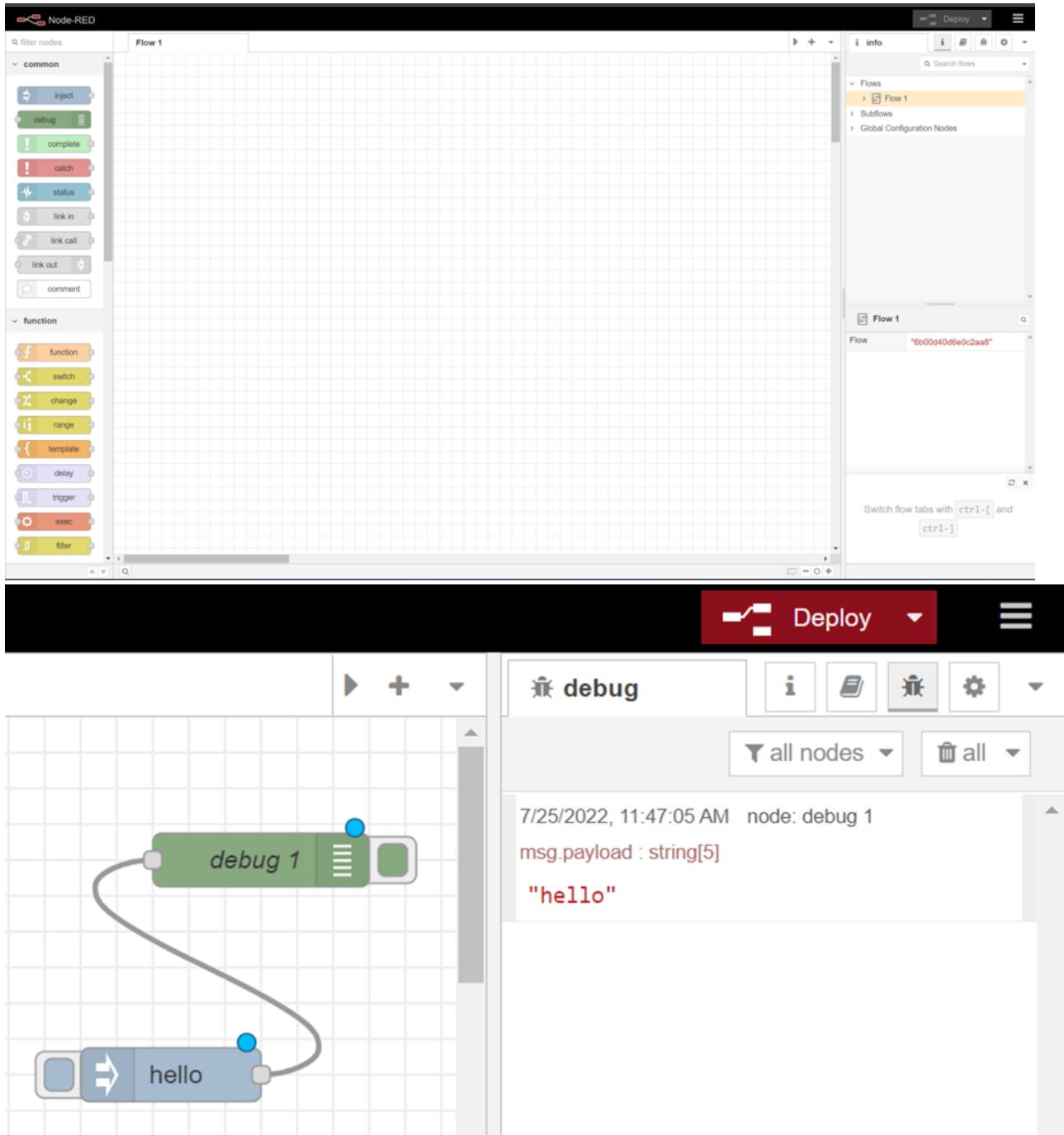
Node-red-dashboard





Practical 4

Practical Name: Introducing the inject, function, debug and switch nodes.

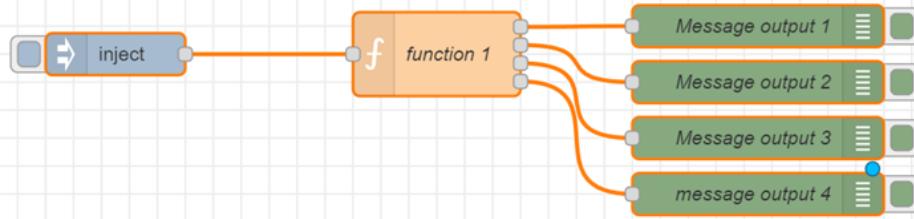
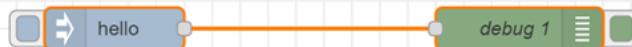


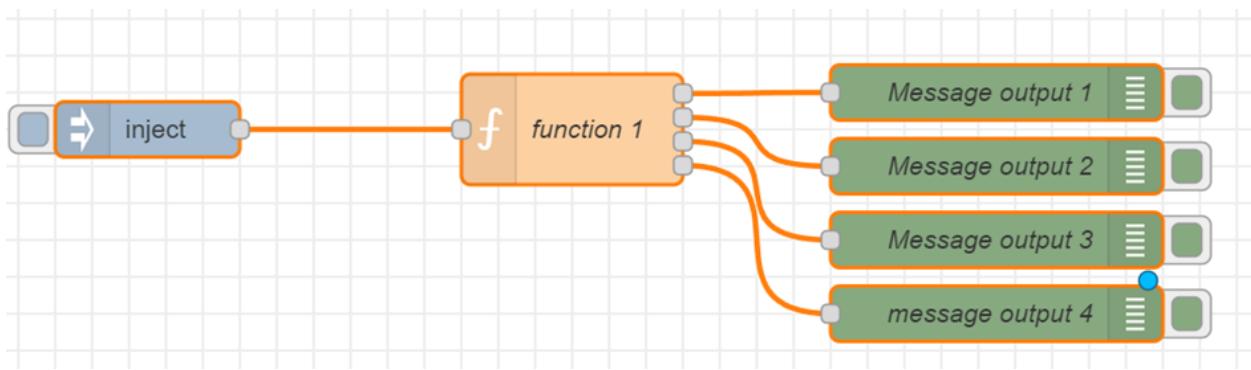


7/25/2022, 11:57:34 AM node: Return string

function : (error)

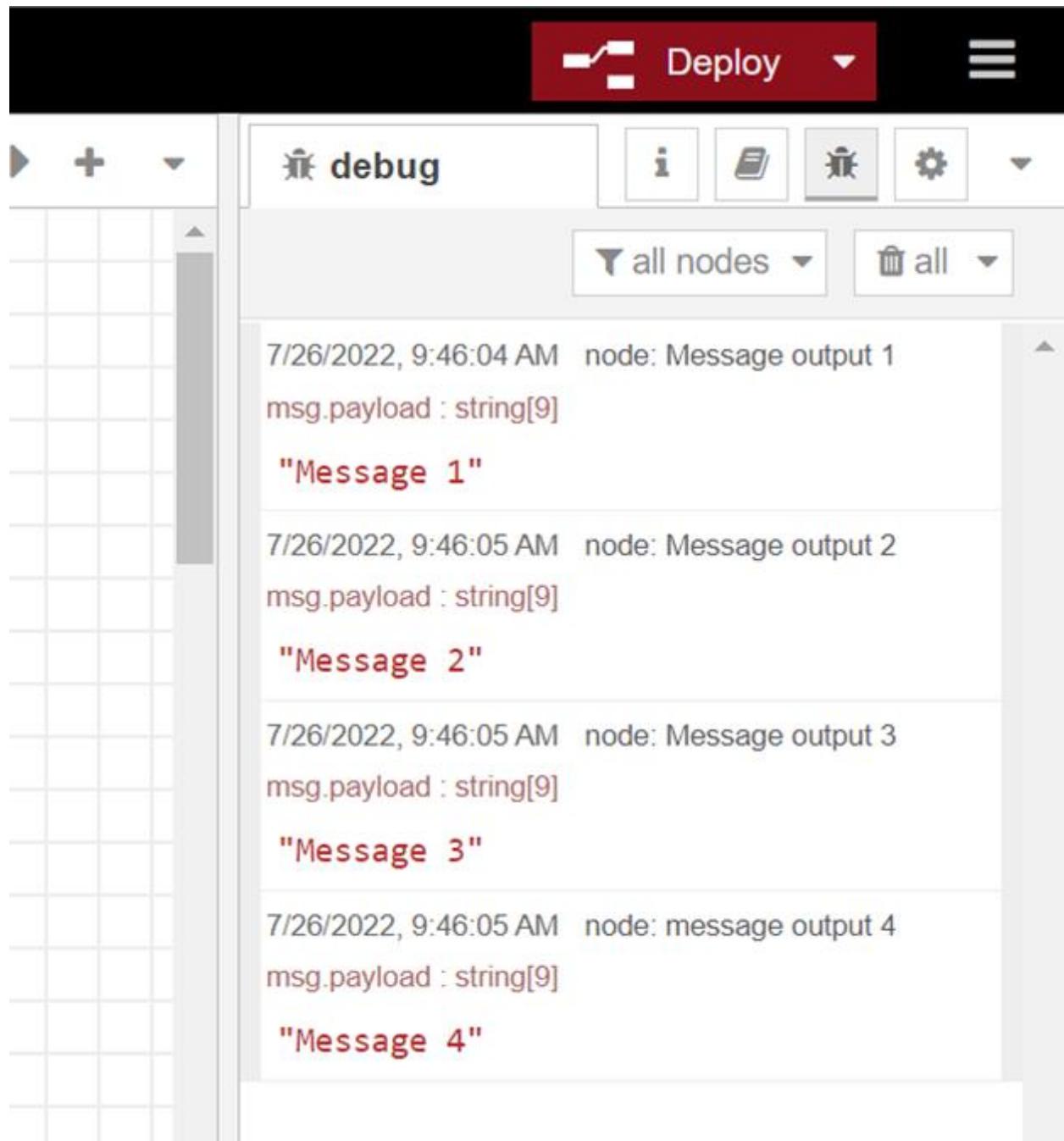
"Function tried to send a message of type string"

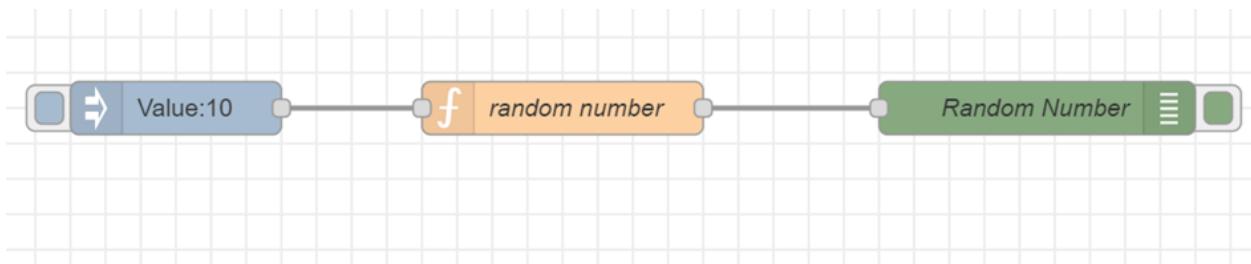




Practical 5

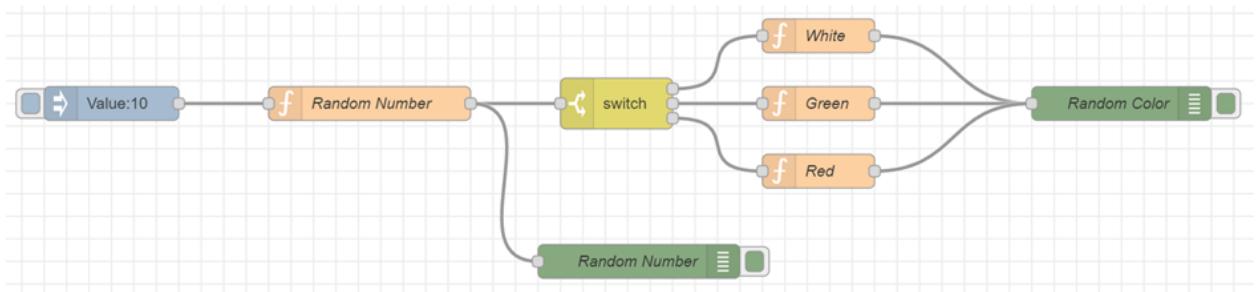
Practical Name: Random number generator with selection of color. Introducing the HTTP node.





The screenshot shows a Node-RED interface with a dark theme. At the top right, there is a 'Deploy' button with a dropdown arrow and a three-line menu icon. Below the header, a toolbar includes icons for play, add, remove, and a gear. The main area is titled 'debug' and contains a list of log entries. Each entry consists of a timestamp, node name, message type, and payload. The payloads are displayed in blue. The log entries are:

- 7/26/2022, 9:52:52 AM node: Random Number msg.payload : number **6.481394324012046**
- 7/26/2022, 9:52:52 AM node: Random Number msg.payload : number **6.17444169791616**
- 7/26/2022, 9:52:53 AM node: Random Number msg.payload : number **2.1077490506627816**
- 7/26/2022, 9:52:53 AM node: Random Number msg.payload : number **2.427645426943099**
- 7/26/2022, 9:52:53 AM node: Random Number msg.payload : number **3.615596701267838**
- 7/26/2022, 9:52:53 AM node: Random Number msg.payload : number **3.9840257509691512**
- 7/26/2022, 9:52:53 AM node: Random Number msg.payload : number **0.30005126662078885**



7/26/2022, 10:10:24 AM node: Random Color

msg.payload : string[8]

"I am Red"

7/26/2022, 10:10:27 AM node: Random Number

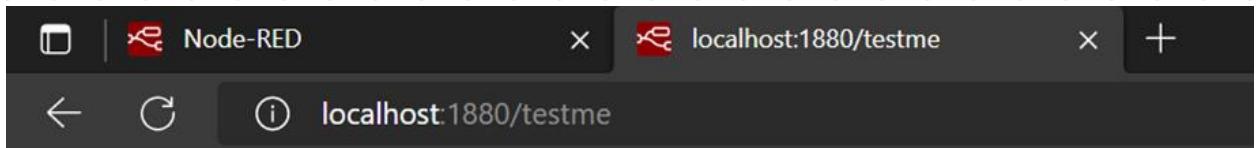
msg.payload : number

2.553326889911689

7/26/2022, 10:10:27 AM node: Random Color

msg.payload : string[10]

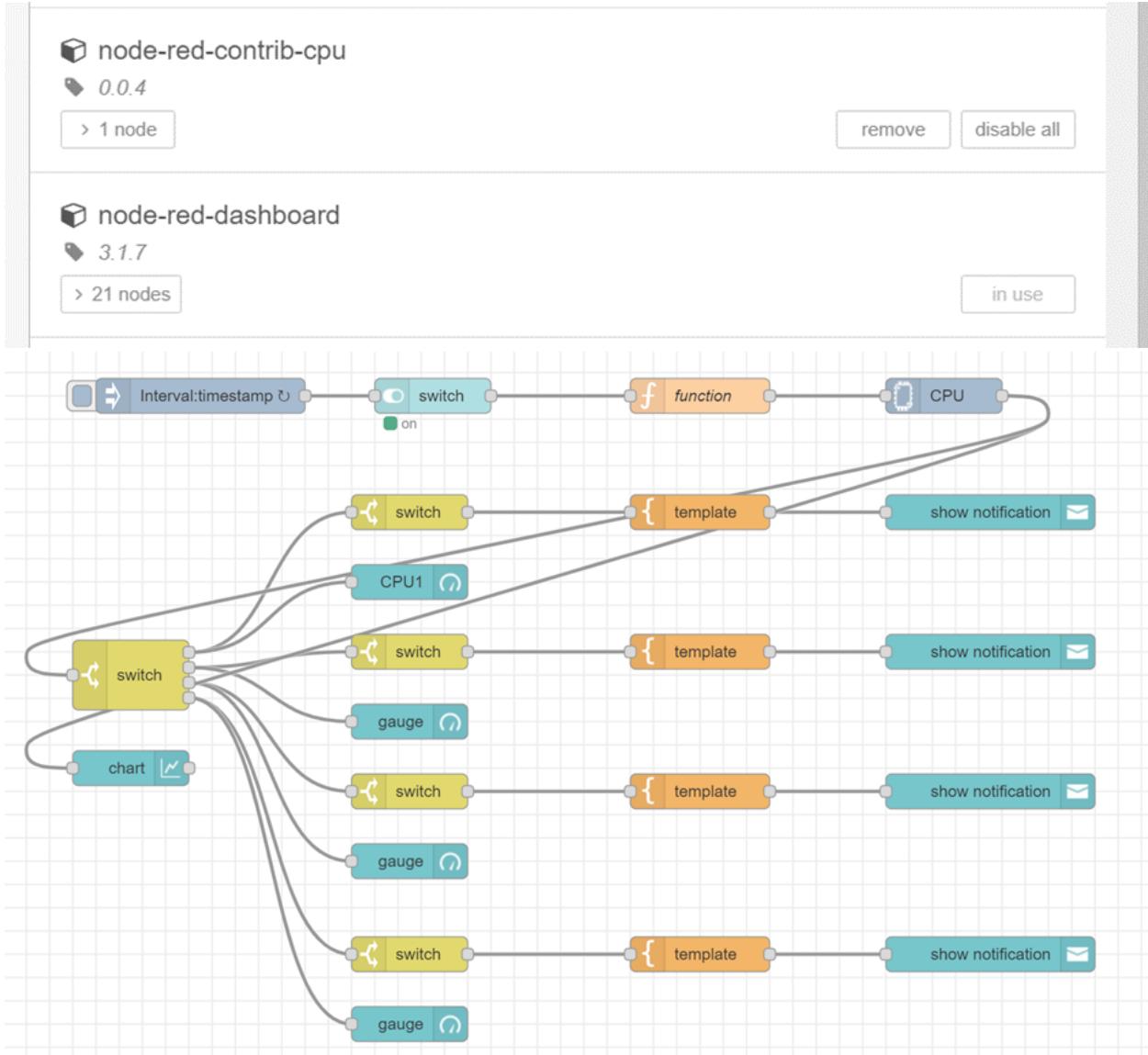
"I am Green"

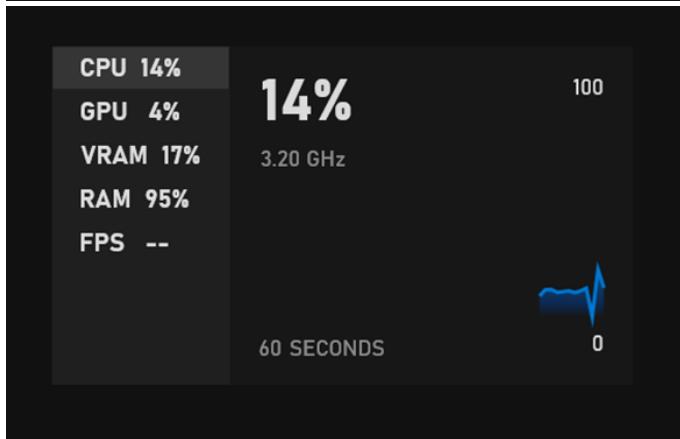
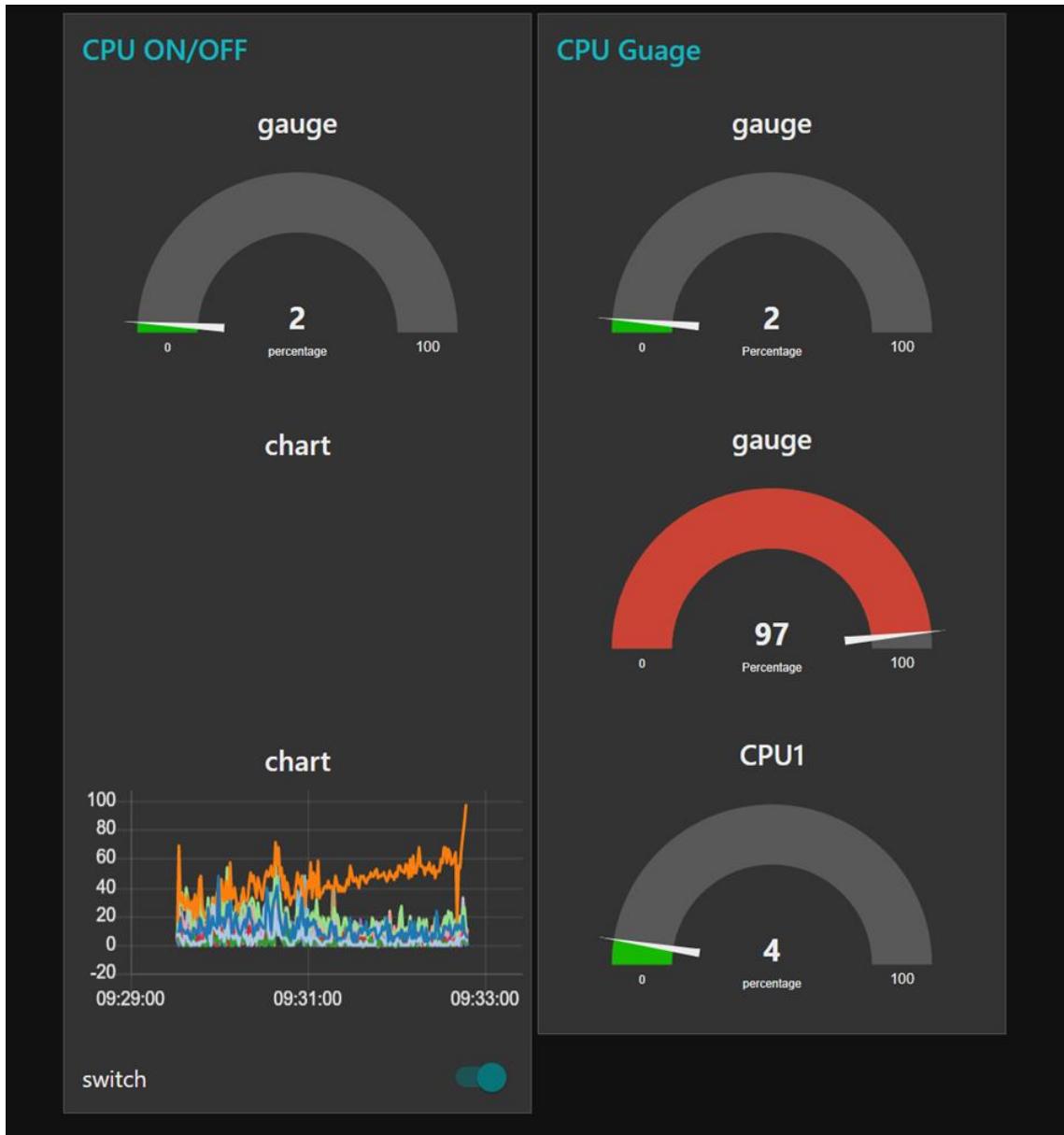


Hello World

Practical 6

Practical Name: CPU utilization flow.

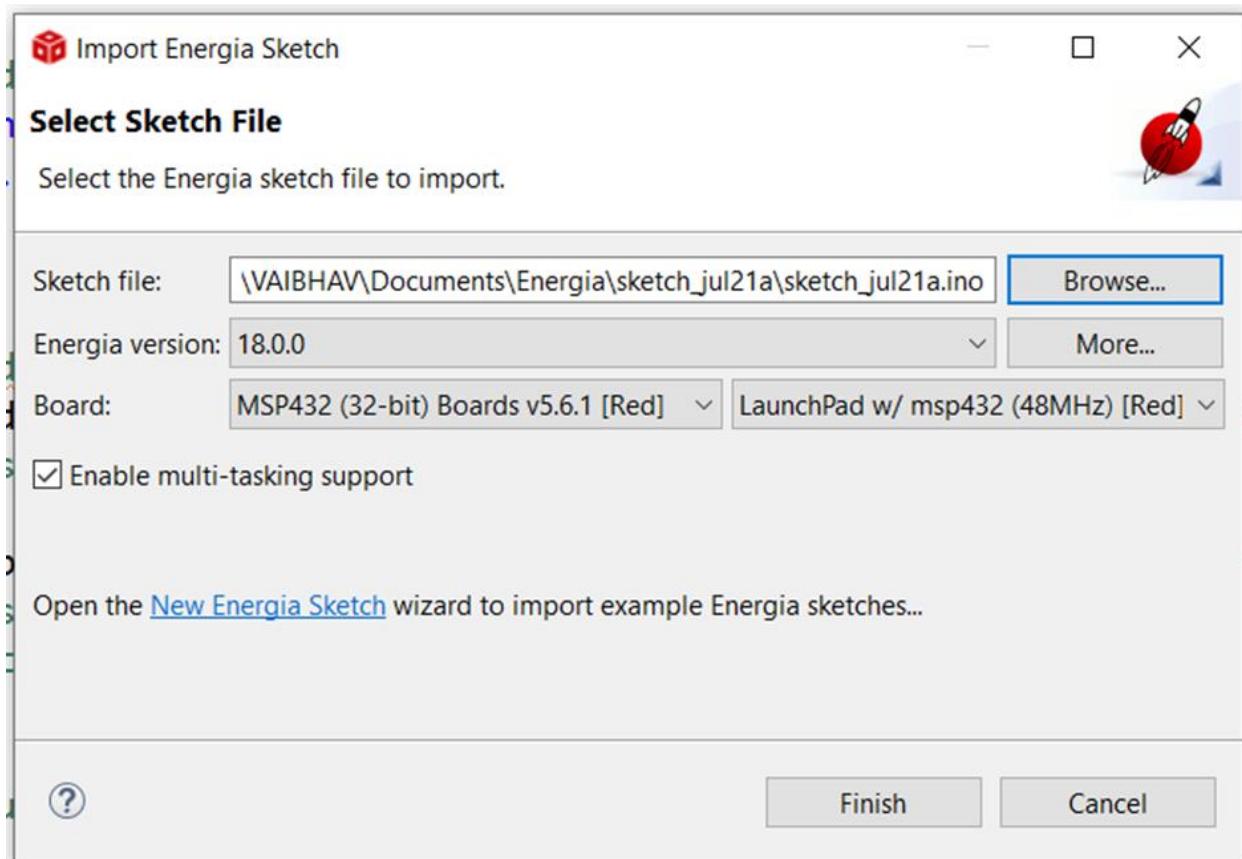




Practical 7

Practical Name: WiFi Setup

Create a project



Write the code in Energia Sketch and save the file

sketch_jul21a.ino

```
#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:

}

main.css

#include <WiFi.h>

#include <SPI.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[]="vivoY73";// my cell phone

char ssid[]="VK";

/*
char ssid[]="vK";
char password[]="";
*/

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 connnect 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 it is not
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();

}

void loop() {

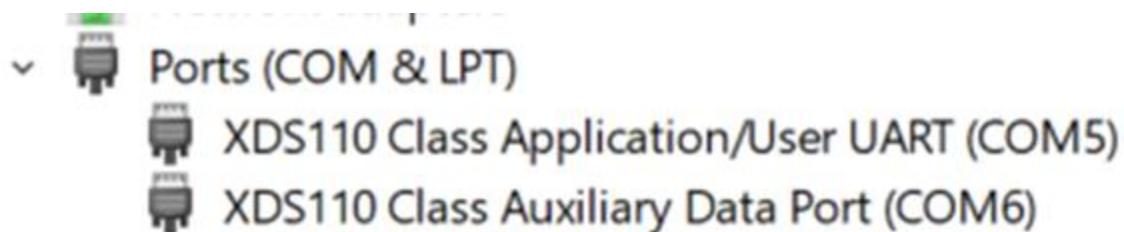
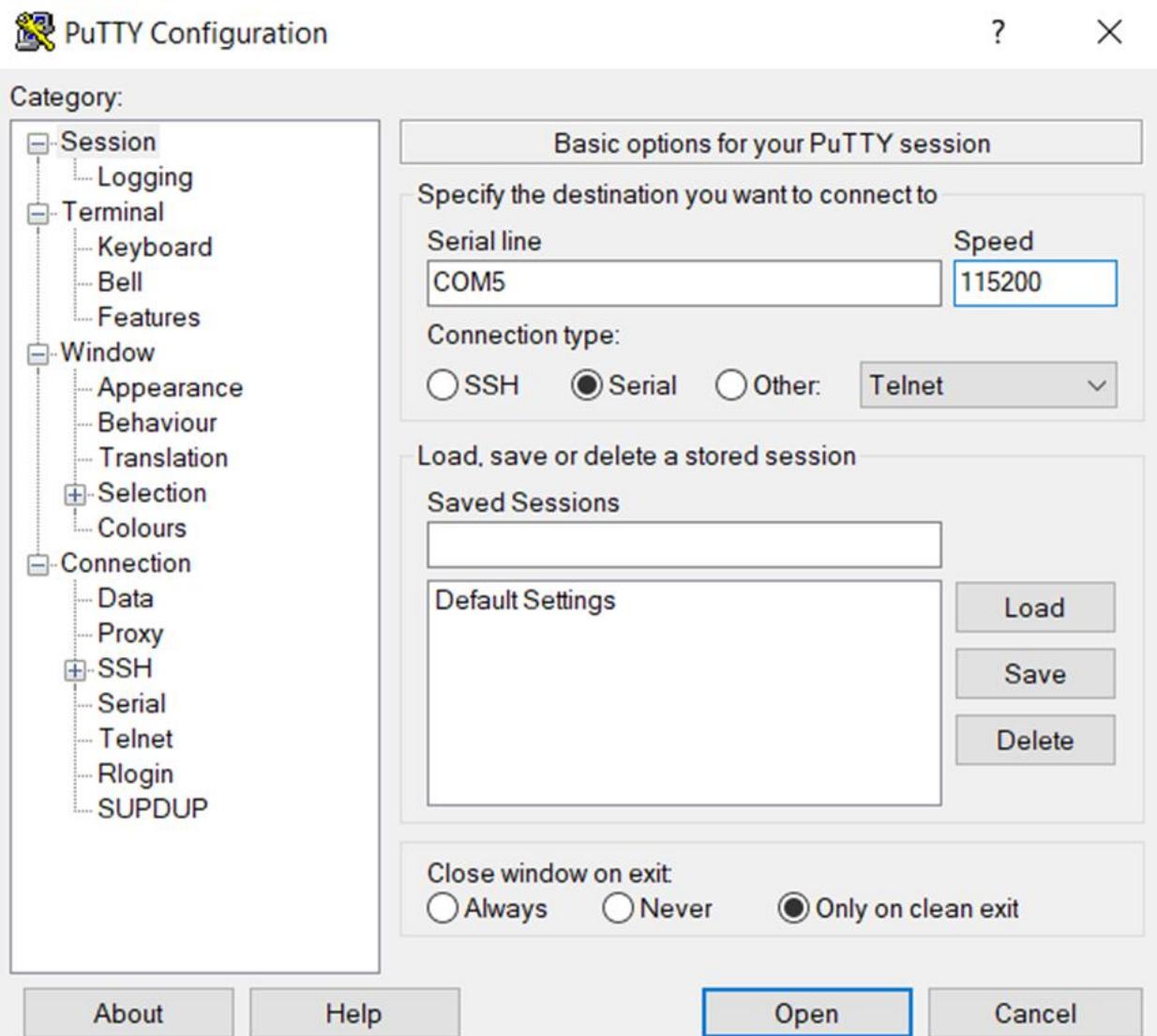
    // put your main code here, to run repeatedly:

}

```

Putty terminal is need to connect the port and device at given speed.

Setting up the putty configuration:



Output:

COM5 - PuTTY

```
.....connecting to WIFI...
Wifi Connected , Fetching Wifi Sheild's IP address : .....192.168.40.125
Access Point name:123456789
Signal Strength199
number of access points in range:12
Name of Access points and encryption type:DIRECT-LXDESKTOP-P1Q9VR6ms02,2
Name of Access points and encryption type:SOMAIYA-WIFI,2
Name of Access points and encryption type:roshan,7
Name of Access points and encryption type:Parmeet,7
Name of Access points and encryption type:SOMAIYA-GUEST,7
Name of Access points and encryption type:SOMAIYA-WIFI,2
Name of Access points and encryption type:Redmi,7
Name of Access points and encryption type:123,7
Name of Access points and encryption type:123456789,7
Name of Access points and encryption type:Redmi Note 9 Pro Max,7
Name of Access points and encryption type:MOTO G10 POWER,7
Name of Access points and encryption type:(,,0
Subnet Mask:255.255.255.0
Gateway IP Address:192.168.40.168
Mac Address of Sheild:A8:1B:6A:99:BF:F8
[redacted]
```

COM5 - PuTTY

```
.....connecting to WIFI...
Wifi Connected , Fetching Wifi Sheild's IP address : .....192.168.171.196
Access Point name:vivoY73
Signal Strength206
number of access points in range:8
Name of Access points and encryption type:vivoY73,7
Name of Access points and encryption type:OPPO F19 Pro+,7
Name of Access points and encryption type:MOTOG10POWER,7
Name of Access points and encryption type:DIRECT-BxLAPTOP-RU5M5INVmsV7,2
Name of Access points and encryption type:DIRECT-JMKARANmsJt,2
Name of Access points and encryption type:mactavish747,7
Name of Access points and encryption type:Bjr3-c2FydGhhay52azEz,7
Name of Access points and encryption type:(,,0
Subnet Mask:255.255.255.0
Gateway IP Address:192.168.171.189
Mac Address of Sheild:A8:1B:6A:99:C8:36
[redacted]
```

Practical 8

Practical Name: Analog To Digital Converter

1)Explain the ADC properties:

Analog to Digital Converter (ADC) is an electronic integrated circuit used to convert the analog signals such as voltages to digital or binary form consisting of 1s and 0s. Most of the ADCs take a voltage input as 0 to 10V, -5V to +5V, etc., and correspondingly produces digital output as some sort of a binary number.

Main features of ADC are:

- The sample rate of an ADC is nothing but how fast an ADC can convert the signal from analog to digital.
- Bit resolution is nothing but how much accuracy can an analog to digital converter can convert the signal from analog to digital

2)Explain the API used for configuring the ADC:

1)GPIO_setAsOutputPin(): Its NOT a digital OR operator,we want to use all 3 pins that's why we used Pipe operator GPIO_setAsPeripheralModuleFunctionInputPin() : potentiometer is connect in port 4 ,pin 7 becoz ADC exists in PIN 7 of port 4 and then digital signal goes to LED

2)ADC14_initModule() : It is called first to initialize the ADC14 module with the desired ADC14 clock, clock predivider and clock divider. ADC Noroute parameter means no internal analog source is user for the conversion.

3)ADC14_configureSingleSampleMode(ADC_MEM6,true) : It is executed with ADC_MEM6 as the target conversion memory register for input channel A6.The true boolean means that this single conversion is a repeat-signal-mode

4)ADC14_configureConversionMemory(ADC_MEM6,ADC_VREFPOS_AVCC_VREFNEG_VSS,ADC_INPUT_A6,false): It is used to set up the conversion memory register as ADC14MEM6 for the input analog channel A6. The ADC references are default reference voltages. The last Boolean value false indicates that this conversion is not a differential conversion instead it is single ended conversion.

5)ADC14_setSampleHoldTime(): It is used to set up the length of sample-and-hold during the AC operations. The first parameter controls the registers ADC14MEM0~ADC14MEM7 and ADC14MEM24~ADC14MEM31. Since A6 belongs to this range, we set the length as 64 ADC14CLK cycles. The second parameter is not used in this project, but it is also set to this value.

6) ADC14_enableSampleTimer() : It is called with the parameter ADC_AUTOMATIC_ITERATION to enable the sample timer to repeat its trigger itself.

7) ADC14_enableConversion_ (): It only enables the conversion, but the conversion cannot be started until it is triggered by a trigger source. Generally, two API functions, ADC14 set_SampleHoldTrigger() and ADC14_toggleConversionTrigger (), can handle this trigger job. However, the latter is more powerful to trigger and start an ADC conversion.

8) An infinitive while (): loop is used starting at line 28 to repeatedly check the conversion results and assign them to the GPIO P2 pins P2.0-P2.3 to display them in a three-color LED.

9) ADC14_isBusy(): It is tested with another conditional while() loop to see whether the ADC conversion is complete. A true is returned if the ADC is busy in conversion. This while() loop will not be released until a false is returned, which means that the ADC is not busy and a conversion has been done.

10) ADC14_getResult (): It is used for the conversion result stored in the ADC14MEM6 Register if a false is returned. Then the result is shift left by eight bits and assigned to the GPIO P2 to display it on the three-color LED.

Code:

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

// combination of energia and CCS functions
// IPAddress shieldIP, subnetMask, gatewayIP;
// IPAddress is a datatype, all others are variables of type IPAddress

uint8_t rssi;// unsigned integer 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[]="vivoY73";// my cell phone
//char ssid[]="VK";
```

```

/*
char ssid[]="vK";
char password[]="";
*/
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 it is not connected
        //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(); //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();
}

//ADC CODE STARTS

GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0|GPIO_PIN1|GPIO_PIN2);

//its NOT a digital OR operator,we want to use all 3 pins thats why we used Pipe operator

GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P4,GPIO_PIN7,GPIO_TERTIARY_MODULE_FUNCTION);

//potentiometer is connect in port 4 ,

//pin 7 becoz ADC exists in PIN 7 of port 4 and then digital signal goes to LED

ADC14_initModule(ADC_CLOCKSOURCE_MCLK,ADC_PREDIVIDER_1,ADC_PREDIVIDER_1,ADC_NOROUTE);

//initModule requires 4 parameters//clock is started

//predivider is divding the clock to make it small

//master clock is 3 MHz which is divider by 2 divders

// ADC Noroute means use ADC to connect to that pin

// more the predivider more the numbers we can get

ADC14_configureSingleSampleMode(ADC_MEM6,true);

//ADC_MEM6 internal memory for storing converted result

ADC14_configureConversionMemory(ADC_MEM6,ADC_VREFPOS_AVCC_VREFNEG_VSS,ADC_INPUT_A6,false);

//2nd parameter is 3.3v i.e the references voltage ,3rd para is use that channel 6 inside MC and store it to the mem 6,false is it will run only once

/*changing the vaules to 64 to 32 and 4*/

ADC14_setSampleHoldTime(ADC_PULSE_WIDTH_32,ADC_PULSE_WIDTH_4);

```

//it will hold the signal for 64 clock cycle , 2 times becoz it require 2 parameter ; our MC is 32bit so it divided into 16-16 bits channels 2nd para is useless as of now

```
ADC14_setResolution(ADC_12BIT);  
ADC14_enableSampleTimer(ADC_AUTOMATIC_ITERATION);//it will run automatically  
ADC14_enableModule();//for activating the module  
ADC14_enableConversion();//start conversion  
ADC14_toggleConversionTrigger();// trigger the conversion again and again
```

// ADC has to be given some reference voltage means range of voltage to work

```
}
```

```
void loop() {  
// put your main code here, to run repeatedly:  
int result,regressedData1;  
float regressedData;
```

while(!ADC14_isBusy());//it will wait till the dADC works to convert the signals****

```
result=ADC14_getResult(ADC_MEM6);  
P2OUT=result>>8;//right shift operator , for using multiple colors by changing bits  
Serial.println(result);  
delay(500);  
ADC14_toggleConversionTrigger();// trigger the conversion again and again  
}
```

34		
35	slope	0.786296
36	intecept	7.695073
37	corelation	0.999397
38		

Slope= slope(TRUTH,FINAL)

Intercept=intercept(TRUTH,FINAL)

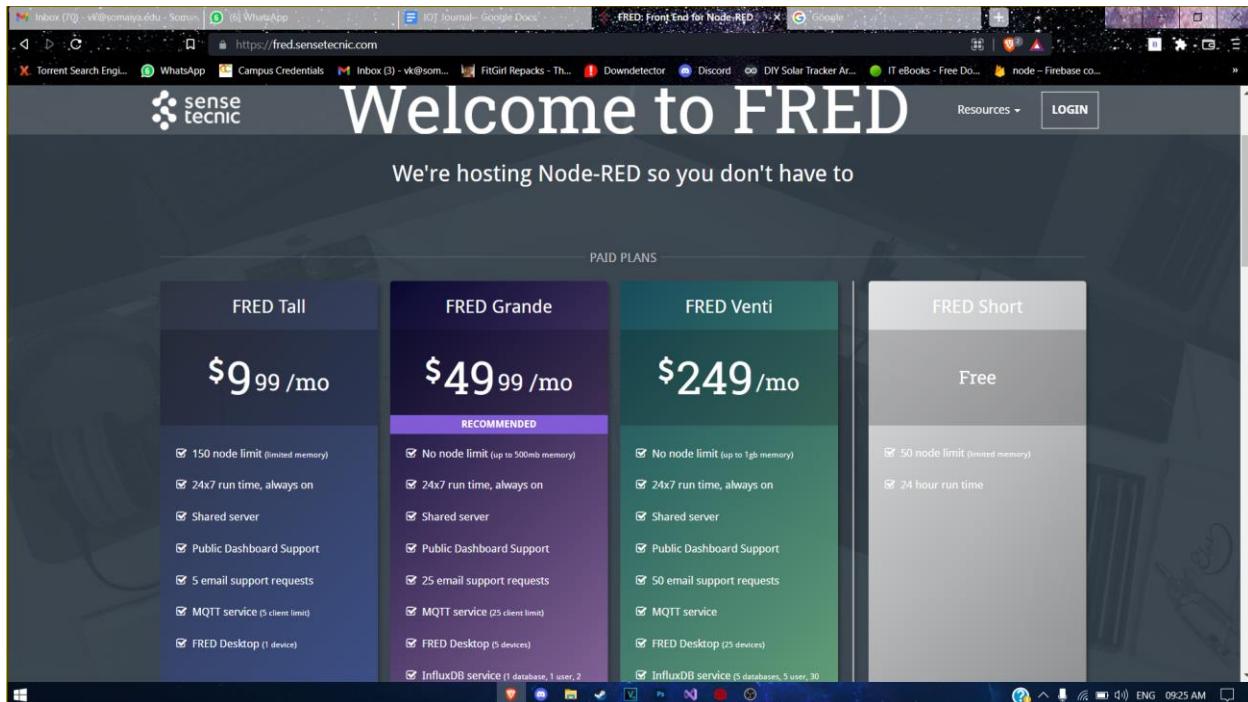
Corelation=correal(TRUTH,FINAL)

Practical 9

Practical Name: Sending Raw data to the cloud

1) Explain the MQTT input and output node.

Using protocol mqtt its is most popular and most used protocol in IOT. It works on publish subscribe model. It is located across the world and many non profit organizations used this protocol. It requires port 1883. The url of server is broker.hivemq.com.

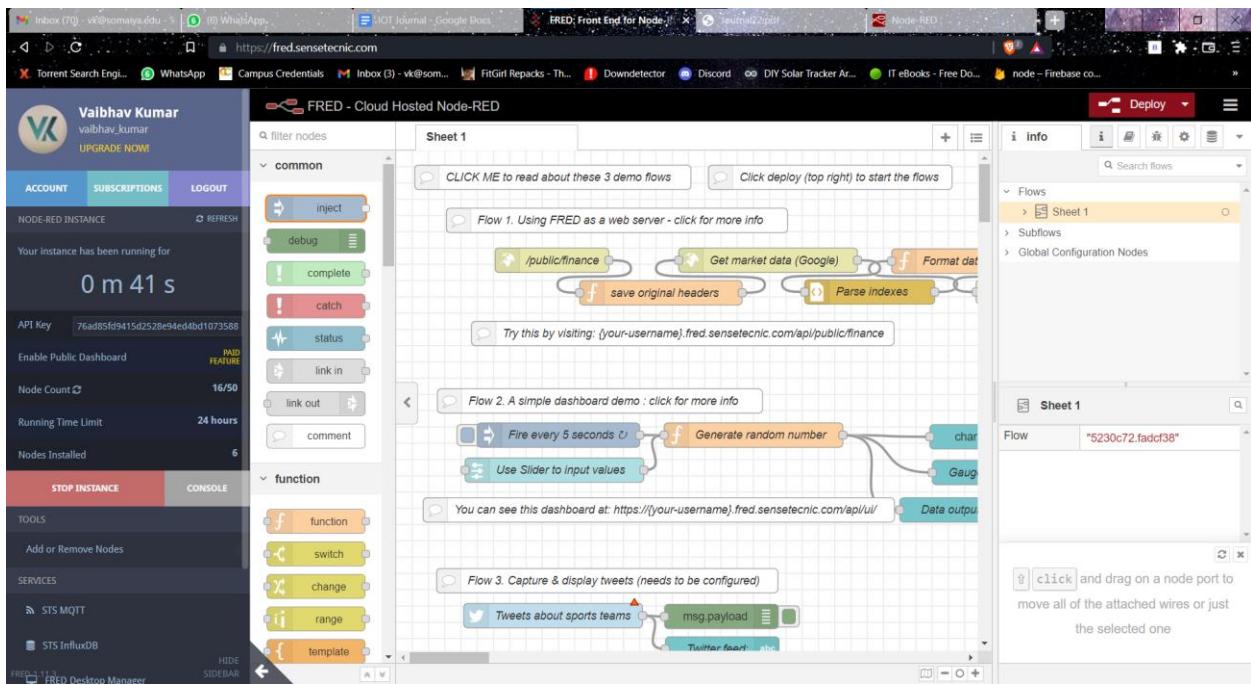


The screenshot shows a web browser window with multiple tabs open. The active tab is 'Sense Tecnic Accounts' at <https://users.sensetechnic.com/app/services>. The page displays three service sections: 'MQTT' (represented by a cloud icon), 'InfluxDB' (represented by a database icon), and 'Manager' (represented by a circular icon with 'FRED' inside). Each section has a brief description and a link to 'Click here to go to [Service] >'. The top navigation bar includes links for 'Services', 'User Profile', 'Account Settings', and 'Manager'. The top right corner shows a user profile for 'Vaibhav Kumar'.

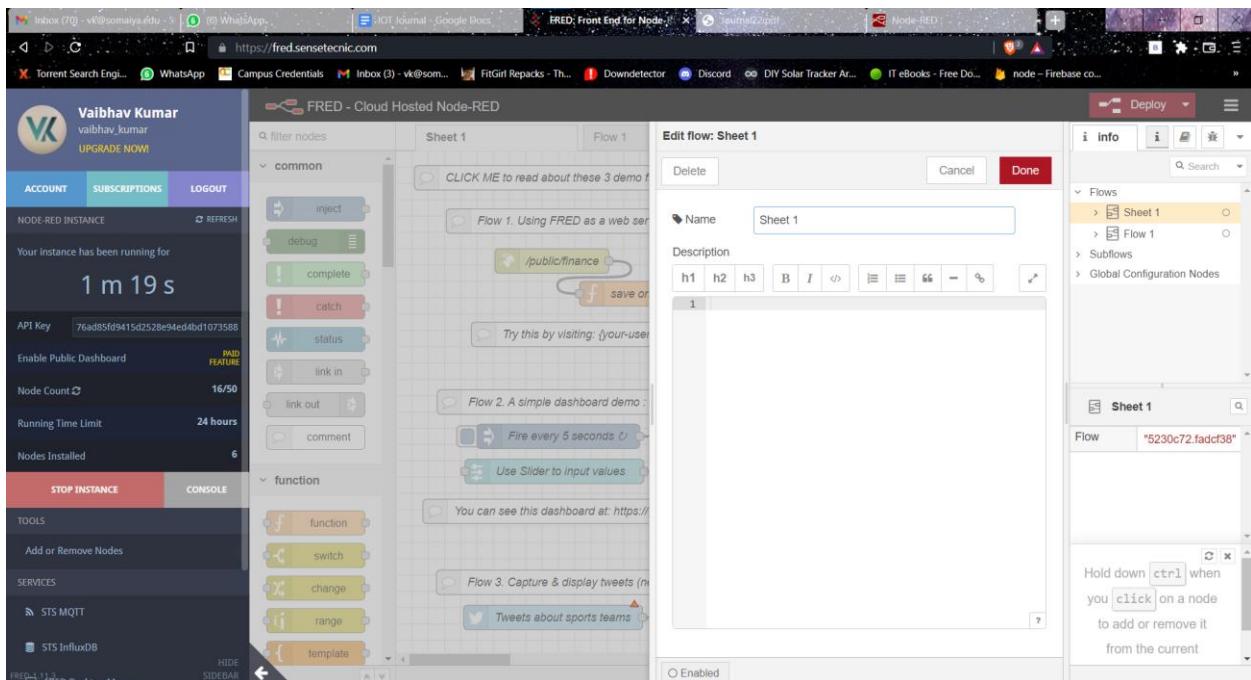
Click on to go to FRED

The screenshot shows the 'FRED: Front End for Node-RED' interface. On the left, a sidebar menu includes 'ACCOUNT' (selected), 'SUBSCRIPTIONS', 'LOGOUT', 'NODE-RED INSTANCE' (status: Stopped), 'API Key' (76ad85fd9415d2528e94ed4bd1073588), 'Enable Public Dashboard' (Paid Feature), 'Node Count' (16/50), 'Running Time Limit' (24 hours), 'Nodes Installed' (6), 'START INSTANCE' (button), 'CONSOLE', 'TOOLS' (Add or Remove Nodes), 'SERVICES' (STS MQTT, STS InfluxDB), and 'FRED Desktop Manager'. The main area displays a message: 'Your Node-RED instance is currently stopped' with a 'Start Instance' button.

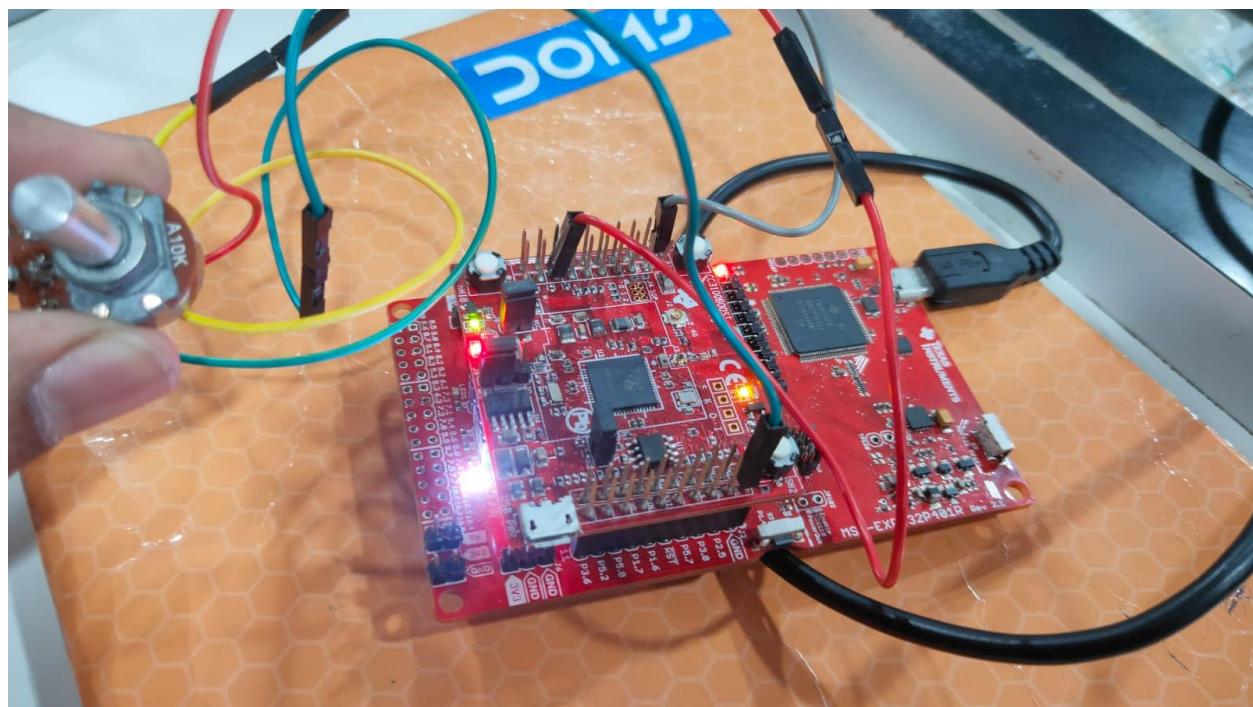
Click on start Instance and then create the empty flow

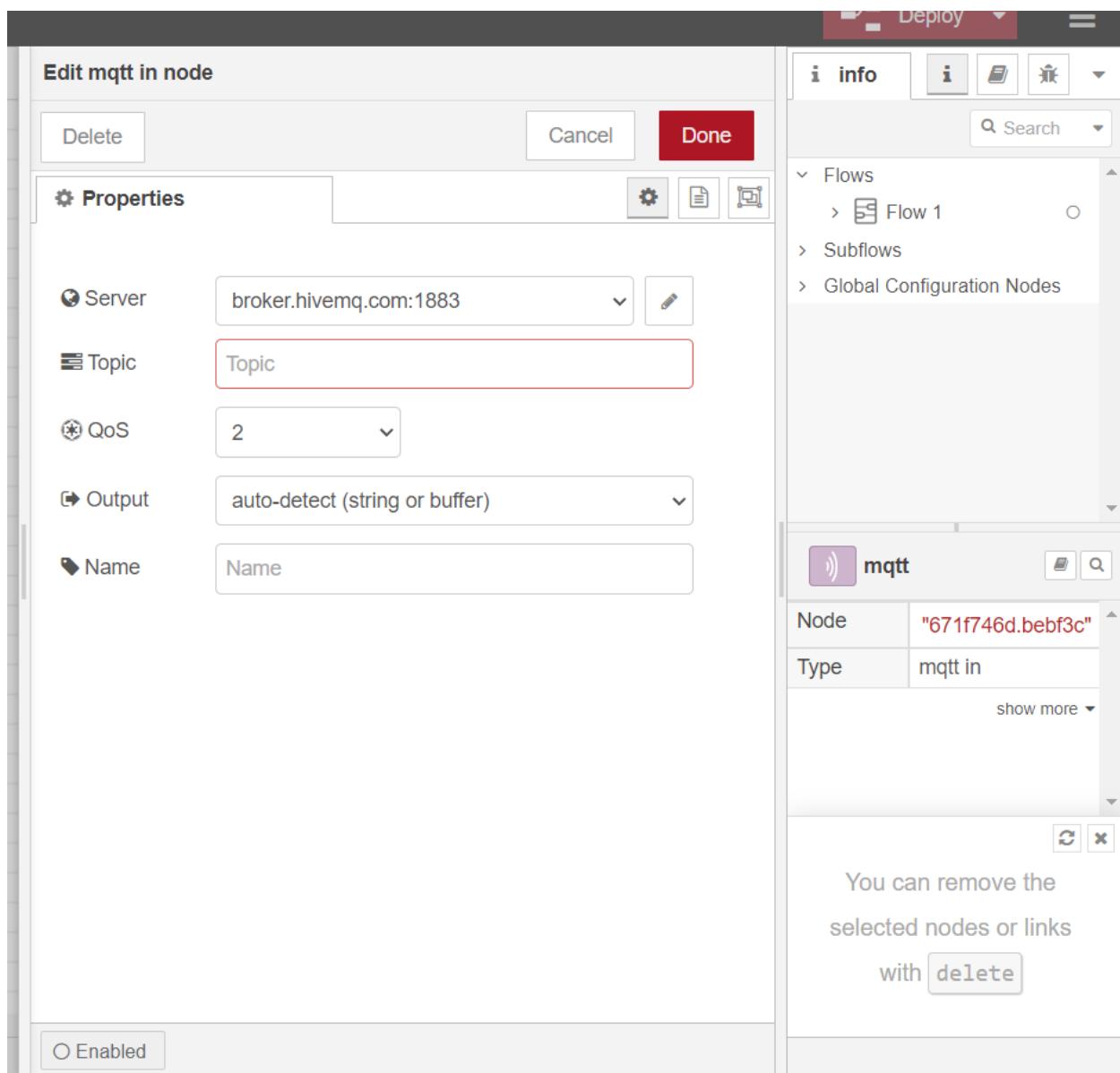


Now delete the sheet 1 by double clicking on sheet and click delete button



The screenshot shows the FRED - Cloud Hosted Node-RED interface. On the left, there's a sidebar with user information (Vaibhav Kumar), account details (NODE-RED INSTANCE, API Key, Node Count, Running Time Limit, Nodes Installed), and links for STOP INSTANCE, CONSOLE, TOOLS, and SERVICES (STS MQTT, STS InfluxDB). The main area is titled "Flow 1" and contains a node palette with categories like common (inject, debug, complete, catch, status, link in, link out, comment) and function (function, switch, change, range, template). A tip message on the right says: "Enable or disable these tips from the option in the settings".





Click on pencil beside the Server

packs - Th... ! Downdetector Discord DIY Solar Tracker Ar... eBooks - Free Do... node - Firebase co...

Edit mqtt in node > Add new mqtt-broker config node

Deploy

Properties

Name: Name

Connection: broker.hivemq.com Port: 1883

Enable secure (SSL/TLS) connection

Client ID: Leave blank for auto generated

Keep alive time (s): 60 Use clean session

Use legacy MQTT 3.1 support

Enabled: 0 nodes use this config On all flows

info

Search

Flows

- Flow 1

Subflows

Global Configuration Nodes

undefined:1883

Node	"1dea38c9.603f97"
Type	mqtt-broker

Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the

63

Edit mqtt in node

Delete Cancel Done

Properties

Server: broker.hivemq.com:1883

Topic: sensorPot

QoS: 2

Output: auto-detect (string or buffer)

Name: Name

Enabled

Flows > Flow 1

Subflows

Global Configuration Nodes

mqtt

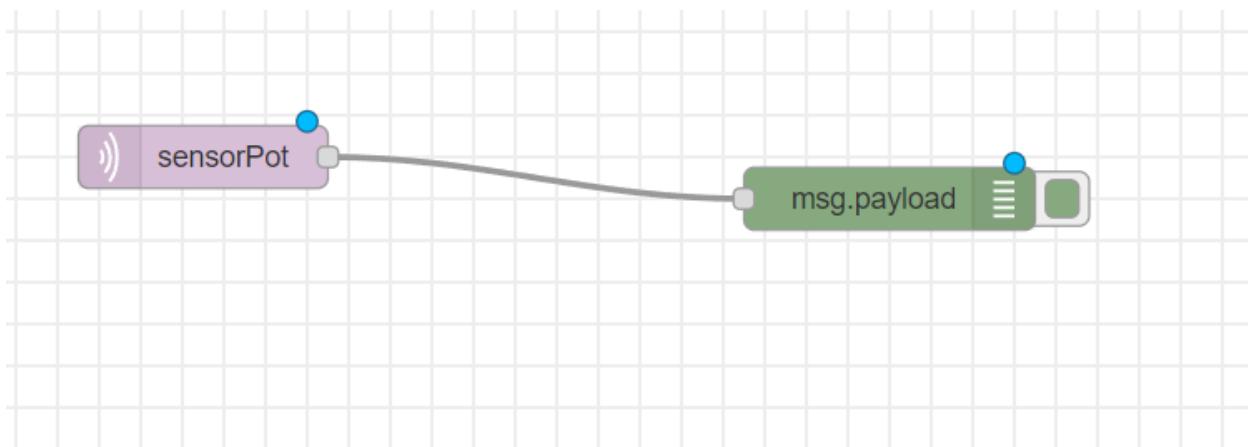
Node: "671f746d.bebf3c"

Type: mqtt in

Search for nodes using ctrl-f

This screenshot shows a configuration dialog for an MQTT node and a search interface for nodes. The configuration dialog has fields for Server (broker.hivemq.com:1883), Topic (sensorPot), QoS (2), Output (auto-detect), and Name (Name). The search interface shows a list of nodes with Node ID "671f746d.bebf3c" and Type "mqtt in". A search bar at the bottom allows users to search for nodes using 'ctrl-f'.

That sets the first node



Deploy

Edit inject node

Name: Name

msg. payload = `hi , We are vK and Pallav!`

msg. topic = `a_z`

+ add

Inject once after `0.1` seconds, then

Repeat: none

Enabled

Done

info

Search

Flows

- Flow 1
- Subflows
- Global Configuration Nodes

timestamp

Node	"fcfc1a7b.101548"
Type	inject

click and drag on a node port to move all of the attached wires or just the selected one

The screenshot shows the JBoss Fuse Workbench interface. On the left, a dialog box titled "Edit inject node" is open, allowing configuration of an inject node. It includes fields for "Name" (set to "Name"), "msg. payload" (containing the value "hi , We are vK and Pallav!"), "msg. topic" (containing the value "a_z"), and options for "Inject once after" (set to 0.1 seconds) and "Repeat" (set to "none"). A "Done" button is at the top right of the dialog. On the right, the "info" tab of the node palette is selected, showing details for the "timestamp" node, including its node ID ("fcfc1a7b.101548") and type ("inject"). Below this, instructions for moving nodes are displayed: "click and drag on a node port to move all of the attached wires or just the selected one". The top bar has a "Deploy" button and other standard application icons.

Edit mqtt out node > **Edit mqtt-broker node**

[Delete](#)

[Cancel](#)

Update

Properties



Name

Name

Connection

Security

Messages

Server

broker.hivemq.com

Port

1883

Enable secure (SSL/TLS) connection

Client ID

Leave blank for auto generated

Keep alive time (s)

Use clean session

Use legacy MQTT 3.1 support

Deploy

Edit mqtt out node

Delete Cancel Done

Properties

- Server: broker.hivemq.com:1883
- Topic: onHighVolts
- QoS: 0 Retain
- Name: MSP432LaunchPad

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Enabled

info

Search

Flows > Flow 1

Subflows

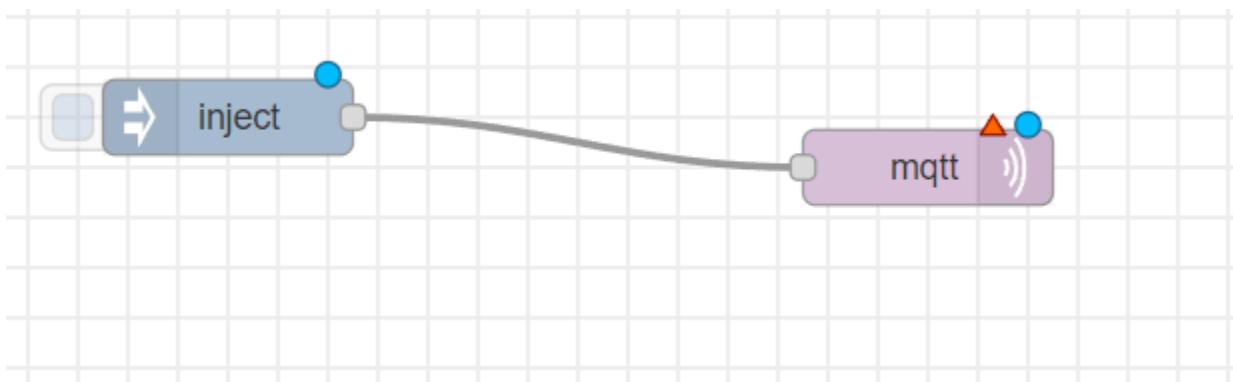
Global Configuration Nodes

mqtt

Node	"56347bca.f1bce4"
Type	mqtt out

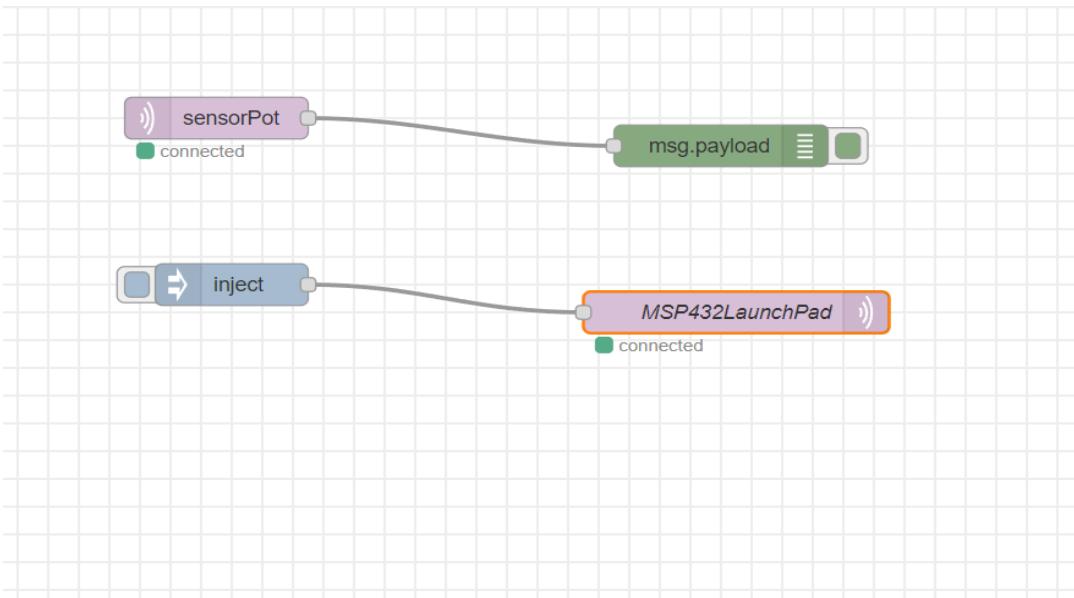
show more ▾

Export the selected nodes, or the current tab with **ctrl-e**



The one publishes on topic and the other one subscriber subscribe on the same topic.

Now deploy.



2) Explain the JSON data frame created in the firmware.

cloudEnergia | Energia 1.6.10E18

File Edit Sketch Tools Help

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

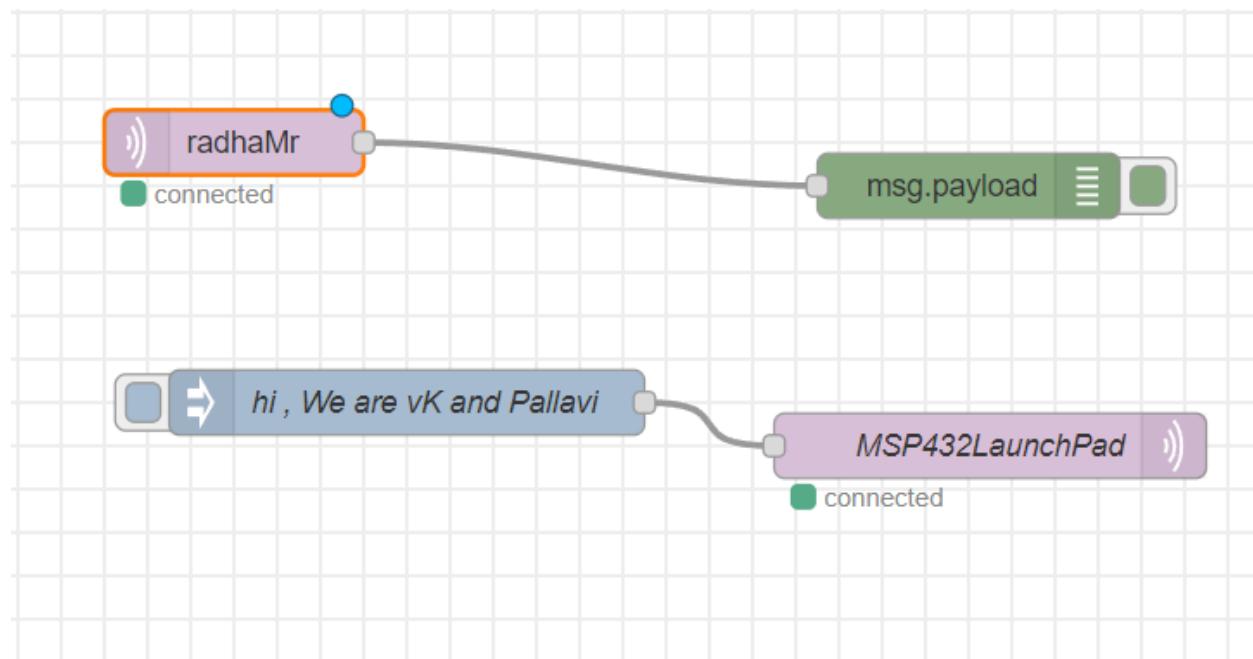
void setup() {
    // put your setup code here, to run once:

}

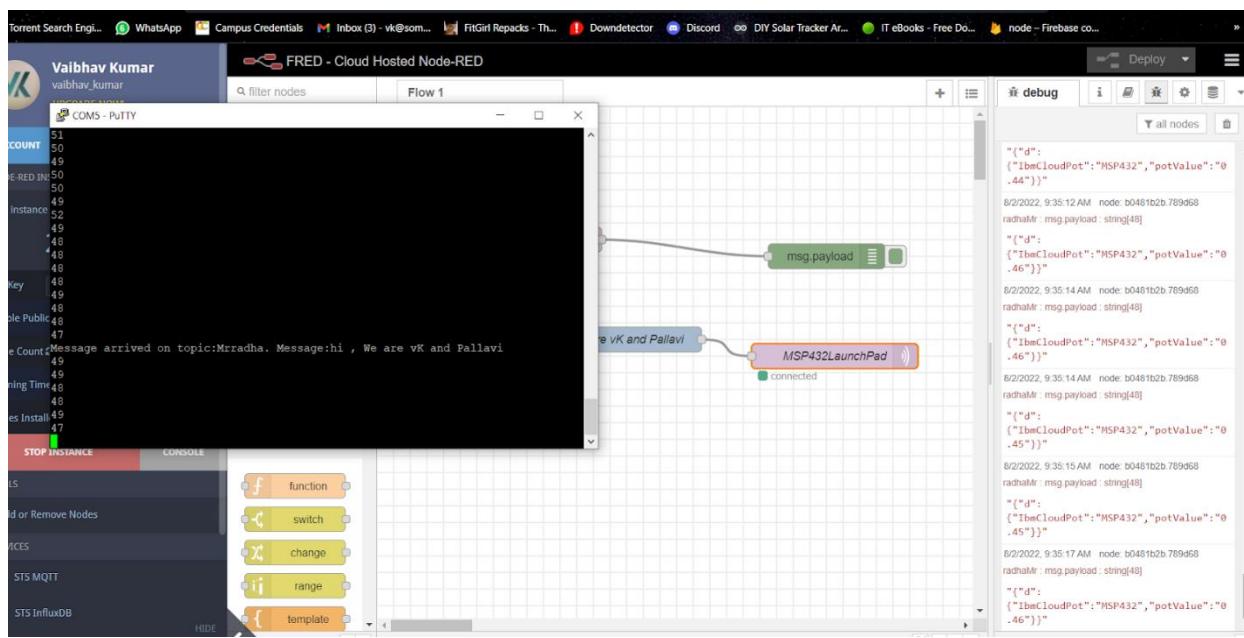
void loop() {
    // put your main code here, to run repeatedly:
}
```

Create the project and first compile it empty without any code.

The screenshot shows the Code Composer Studio interface. The Project Explorer panel lists files like 'jsonIBMCLOUD_P8' (Active - Debug), 'main.c', 'sketch.jul21a.ino', and 'main.c'. The main editor window displays an empty 'main.c' file with a copyright notice and a standard C99 header. The 'Console' tab shows a build log for 'jsonIBMCLOUD_P8' with the command 'iverlib.lib" -llibc.a' and the message 'Finished building target: "jsonIBMCLOUD_P8.out"'. The status bar at the bottom indicates '**** Build Finished ****'.

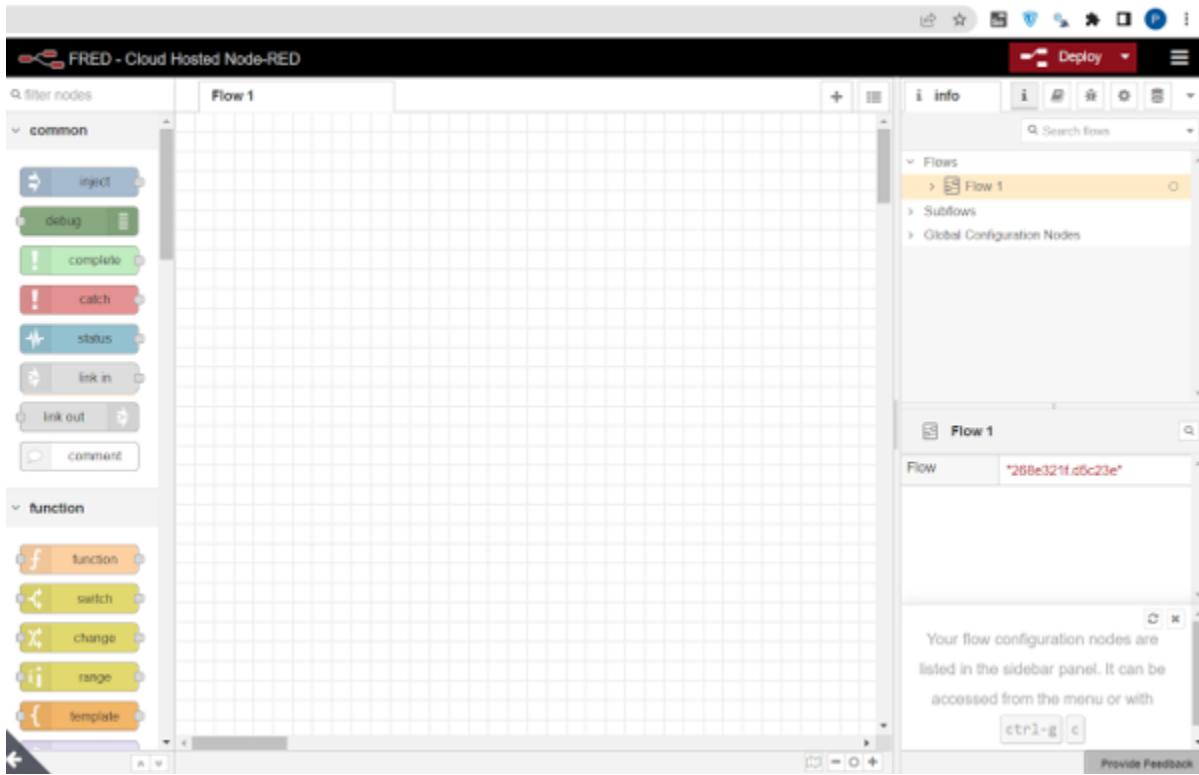
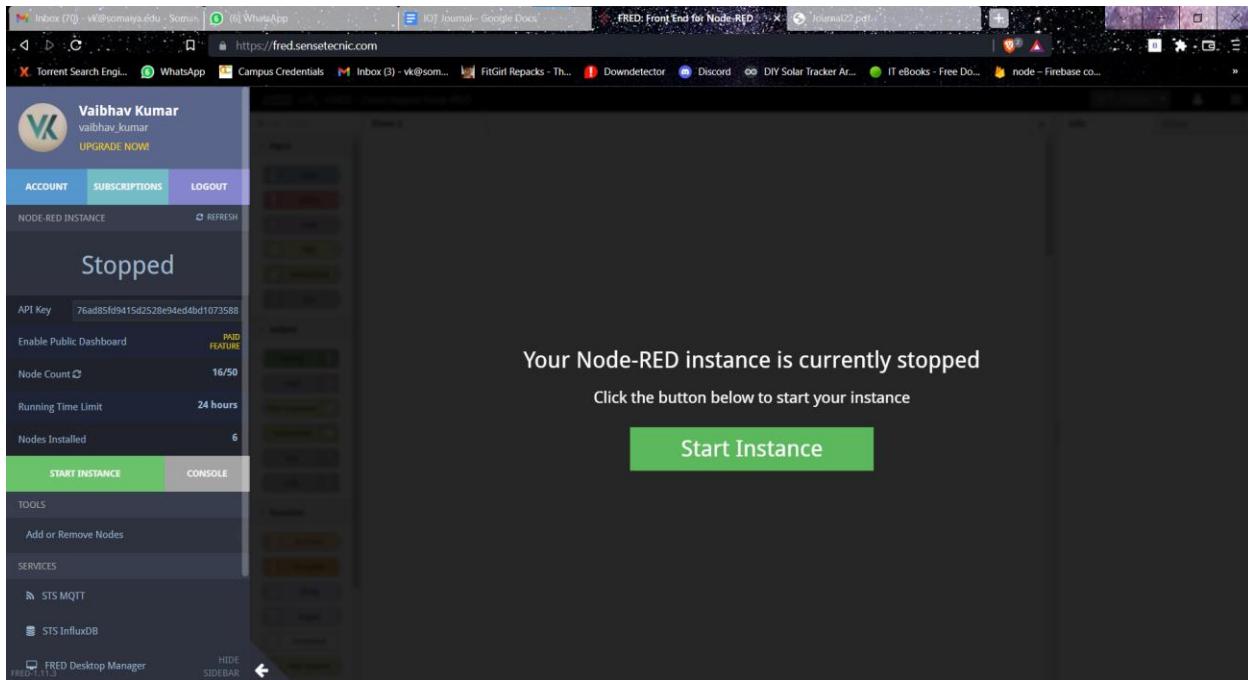


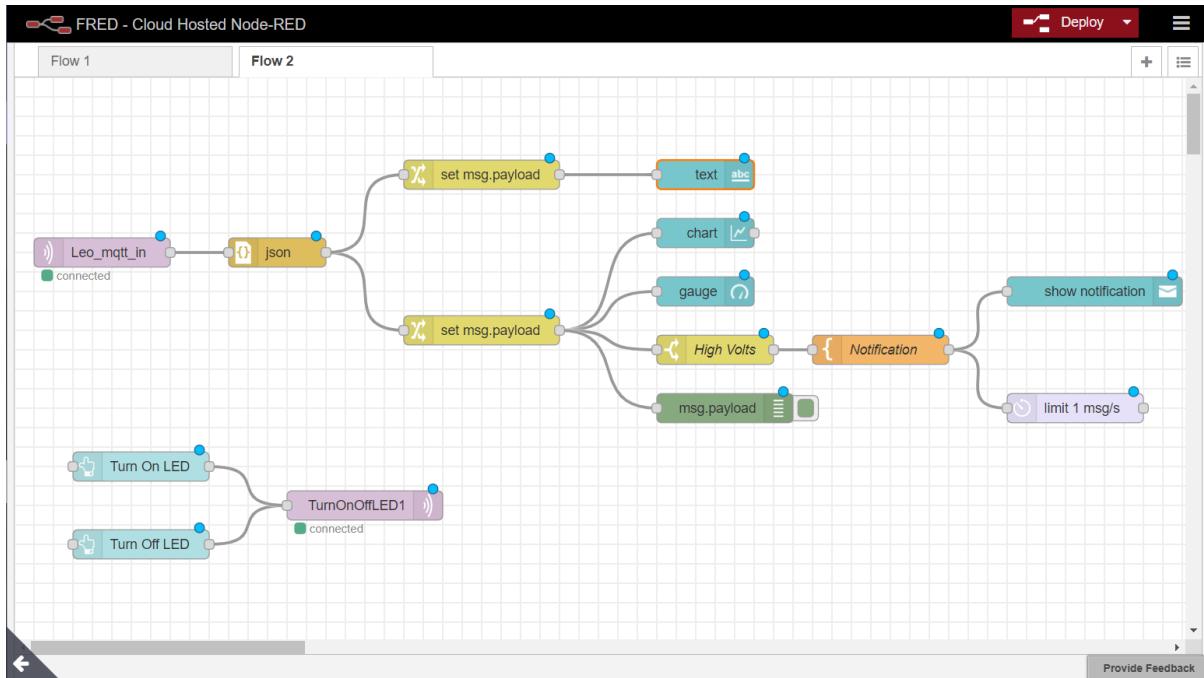
```
COM5 - PuTTY
3720
3747
3748
3745
3744
3752
3749
3747
3759
3747
3747
3760
3748
3759
3752
Message arrived on topic:Mrradha. Message:hi , We are vK and Pallavi
3744
3752
3758
3747
3748
3759
3759
```



Practical 10

Connecting MSP432 to the cloud





Edit mqtt in node

Delete Cancel Done

Properties

Server	broker.hivemq.com:1883	<input type="button" value=""/>
Topic	Leo_mqtt_in	
QoS	0	
Output	auto-detect (string or buffer)	
Name	Name	

Edit json node

Delete Cancel Done

Properties

Action: Convert between JSON String & Object

Property: msg. payload

Name: Name

Object to JSON options

Format JSON string

Edit change node

Delete Cancel Done

Properties

Name: Name

Rules

Set msg. payload to J: payload[0].d.Name

Edit change node

Delete Cancel Done

Properties

Name: Name

Rules

Set msg. payload to J: payload[0].d.potValue

```
graph TD; Set[Set] --> Payload[msg. payload]; Set --> To[to]; To --> J[J: payload[0].d.potValue]
```

Edit text node

Delete Cancel Done

Properties

Group: [My IOT Dashboard] Information

Size: auto

Label:

Value format: {{msg.payload}}

Layout:

label value label value label value

label value label value

Name:

```
graph LR; Node1["label value"] --- Node2["label value"]; Node3["label value"] --- Node4["label value"]
```

Edit chart node

Delete Cancel Done

Properties

Group: [My IOT Dashboard] Potentiometer Voltage Chart

Size: auto

Label: chart

Type: Line chart enlarge points

X-axis: last 1 hours OR 1000 points

X-axis Label: HH:mm:ss as UTC

Y-axis: min 0 max 3.3

Legend: None Interpolate linear

Series Colours:

Blue	Light Blue	Orange
Green	Light Green	Red
Pink	Purple	Lavender

Edit gauge node

Delete Cancel Done

Properties

Group: [My IOT Dashboard] Potentiometer Voltage Gauge

Size: auto

Type: Gauge

Label: gauge

Value format: {{value}}

Units: Volts

Range: min 0 max 3.3

Colour gradient:

Sectors: 0 ... optional ... optional ... 3.3

Name:

Edit switch node

Delete Cancel Done

Properties

Name: High Volts

Property: msg. payload

Condition: `= > a_z 3 → 1`

CSSCODE

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

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

char ssid[]="Galaxy M21F8D8";

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)//for cell phone //WL_CONNECTED while it is
not connected
//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);
}
shieldIP = WiFi.localIP();
```

```

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_P1,GPIO_PIN0);
GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0|GPIO_PIN1|GPIO_PIN2);

GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P4,GPIO_PIN7,GPIO_TERTIAR
Y_MODULE_FUNCTION);//potentiometer is connect in port 4 ,

```

//pin 7 becoz ADC

exists in PIN 7 of port 4 and then digital signal goes to LED

ADC14_initModule(ADC_CLOCKSOURCE_MCLK,ADC_PREDIVIDER_1,ADC_PREDIVIDER_1,ADC_NOROUTE); //initModule requires 4 parameters //clock is started //predivider is dividing the clock to make it small //master clock is 3 MHz which is divider by 2 divders // ADC Noroute means use ADC to connect to that pin // more the predivider more the numbers we can get

ADC14_configureSingleSampleMode(ADC_MEM6,true); //ADC_MEM6 internal memory for storing converted result

ADC14_configureConversionMemory(ADC_MEM6,ADC_VREFPOS_AVCC_VREFNEG_VSS,ADC_INPUT_A6,false); //2nd parameter is 3.3v i.e the references voltage , 3rd para is use that channel 6 inside MC and store it to the mem 6 , false is it will run only once

ADC14_setSampleHoldTime(ADC_PULSE_WIDTH_32,ADC_PULSE_WIDTH_4); //it will hold the signal for 64 clock cycle , 2 times becoz it require 2 parameter ; our MC is 32bit so it divided into 16-16 bits channels 2nd para is useless as of now

ADC14_setResolution(ADC_12BIT);
ADC14_enableSampleTimer(ADC_AUTOMATIC_ITERATION); //it will run automatically

ADC14_enableModule(); //for activating the module
ADC14_enableConversion(); //start conversion
ADC14_toggleConversionTrigger(); // trigger the conversion again and again
}

```
void loop() {
    // put your main code here, to run repeatedly:
    if(!client.connected())
    {
        Serial.println("Disconnected:Reconnecting...");
        if(!client.connect("sagarPraful"))
        {
            Serial.println("connection failed");
        }
        else
        {
            Serial.println("Connection success");
            if(client.subscribe("prafulNalanda"))
            {
                Serial.println("Subscription sucessfull");
            }
        }
    }
}
```

```

int result,regressedData1;
float regressedData;
while(!ADC14_isBusy());
result=ADC14_getResult(ADC_MEM6);
//P2OUT=result>>8;
//Serial.println(result);
ADC14_toggleConversionTrigger();
//result*slope+itercept/1000
regressedData=((result*0.786295966)+7.695073417)/1000;
Serial.println(regressedData);
regressedData1=((result*0.786295966)+7.695073417);
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,"IbmCloudPot","MSP432");
aJson.addStringToObject(d,"potValue",pot_reading);
jsonPayload=aJson.print(root)+'\0';//renders the js object back to a string;

```

```

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

