06 – Yash Bhatia

K J Somaiya Institute of Management Studies and Research

Vidyavihar, Mumbai – 400 077

Masters in Computer Applications
Trimester III (2020 – 21)

This is to certify that Mr. Yash Deepak Bhatia Roll No. 06 of MCA has completed his IoT Journal as per the Syllabus for the Academic year 2020 - 21. The Journal has also been evaluated by the concerned faculty of SIMSR.

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Signature of the Faculty-Incharge

_ _ _ _ _ _ _ _ _ _ _ _ _

Signature of the Course Coordinator - MCA

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Date: 26/09/2021

Signature of the External Examiner

Index

Practical 1

Practical Name: KEIL Installation, User Interface tour, Project creation, debugging and simulation

1)What is µVision IDE ?

µVision is a window-based software development platform that combines a robust and modern editor with a project manager and make facility tool. It integrates all the tools needed to develop embedded applications including a C/C++ compiler, macro assembler, linker/locator, and a HEX file generator.

µVision helps expedite the development process of embedded applications by providing the following:

- Full-featured source code editor.
- Device Database® for configuring the development tool.
- Project Manager for creating and maintaining your projects.
- Integrated Make Utility functionality for assembling, compiling, and linking your embedded applications.
- Dialogs for all development environment settings.
- True integrated source-level and assembler-level Debugger with high-speed CPU and peripheral Simulator.
- Advanced GDI interface for software debugging on target hardware and for connecting to a Keil® ULINK® Debug Adapter.
- Flash programming utility for downloading the application program into Flash ROM.
- Links to manuals, on-line help, device datasheets, and user guides.

The µVision IDE and Debugger is the central part of the Keil development toolchain and has numerous features that help the programmer to develop embedded applications quickly and successfully. The Keil tools are easy to use, and are guaranteed to help you achieve your design goals in a timely manner.

µVision offers a Build Mode for creating applications and a Debug Mode for debugging applications. Applications can be debugged with the integrated µVision Simulator or directly on hardware, for example with ULINK Debug and Trace Adapters. Developers can also use other AGDI adapters or external third-party tools to analyse applications.

Features and Advantages

Feature	Advantage
Project Manager, Make Utility, Debugger, modern Editor	Have been combined into a **single user interface** accelerating the application development. While editing, debugger features can be configured. While debugging, changes can be made to the source code.
µVision **Simulator**	Write, test, and debug applications with the µVision Simulator. It allows you to investigate different hardware configurations even before the hardware is available. The µVision Simulator models most on-chip peripherals.
Simulator and **Target Debugger**	Both debugging interfaces have been implemented to have the same look & feel, and shorten the learning curve considerably.
System Viewer	Displays information about peripheral registers and allows you to manually change property values at run-time.
Code Coverage	Provides statistical data about the execution of the application. Safety-critical systems can be tested and validated thoroughly. Execution analysis reports can be viewed and printed for certification requirements.
Logic Analyzer	Displays changes of values on a time graph. Study the signal and variable changes and view their dependency or correlation.
Device Database	Allows you to configure the development environment automatically based on the microcontroller in use. Developers are provided with default settings that reduce the time needed to configure the tool.
Template editor	Create common text sequences or header blocks. Use templates to insert standard text, header descriptions, and generic code blocks into the program structure.
Source Browser	Use the browser for navigating quickly among coded procedures. Save time during development. Use this functionality in addition to the Find functions.
Configuration Wizard	Provides a graphical interface for to maintain device and start-up code settings. Instead of scrolling through the start-up file, use this advanced GUI-like feature.
Third-party tools	µVision integrates additional tools such as version control systems or CASE tools. All configuration details are saved in the µVision project file.
Debugging and **Flash programming**	The ULINK Debug and Trace Adapters are delivered with pre-configured Flash programming algorithms, which can be modified and adapted to particular needs.
Multi-Project Manager	Allows you to combine µVision projects, which logically depend on each other, into one single Multiple-Project. This increase the overview, consistency, and transparency of the embedded system application design.

2) What is the Build mode and Debug mode?

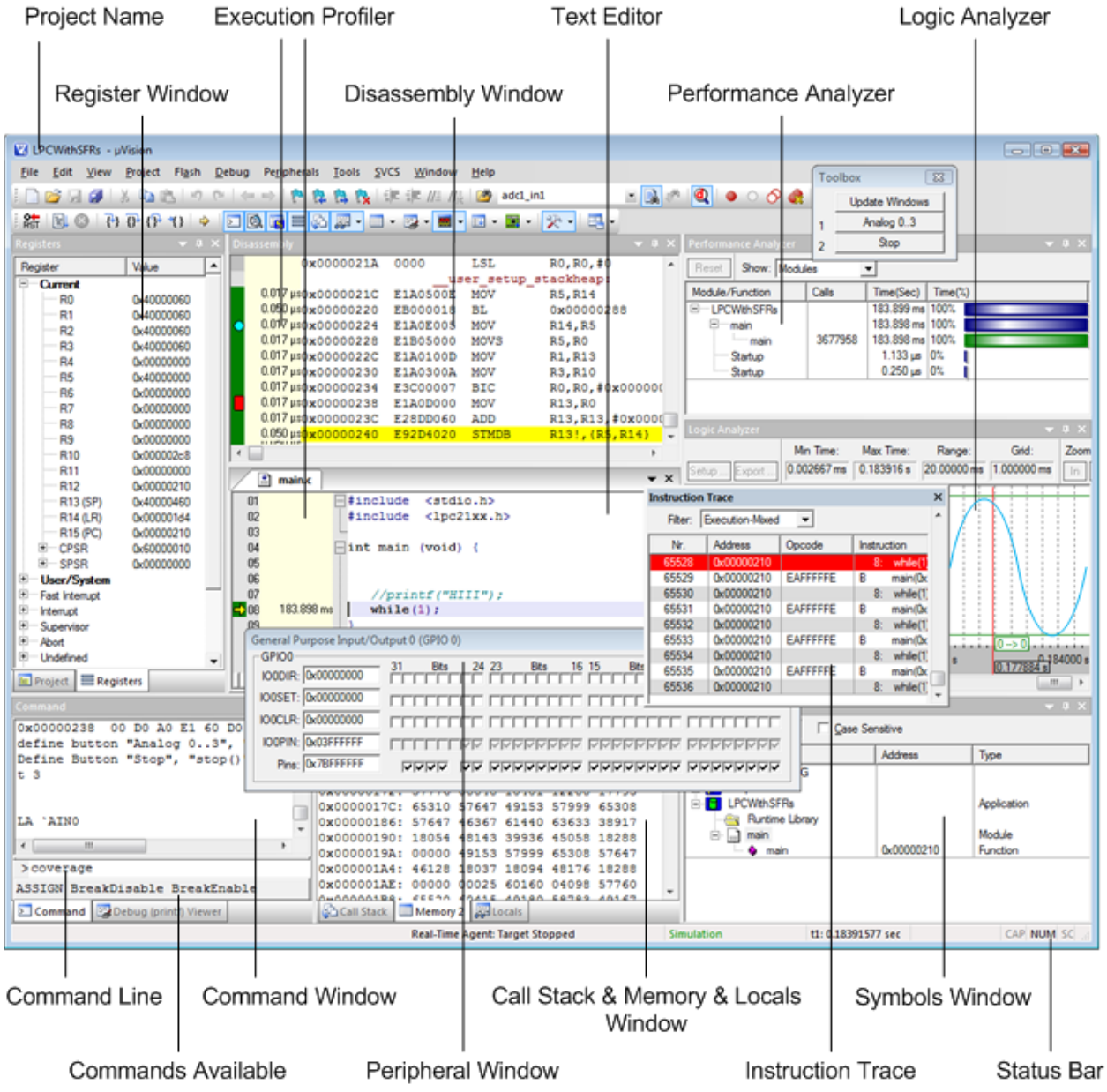
Ans) µVision offers a **Build Mode** for creating applications and a **Debug Mode** for debugging applications.

3) Explain the µVision GUI ?

Ans) The µVision GUI provides menus for selecting commands and toolbars with command buttons. The Status Bar, at the bottom of the window, displays information and messages about the current µVision command. Windows can be relocated and docked to another physical screen. The window layout is saved for each project automatically and restored the next time the project is used. You can restore the default layout using the menu Window – Reset View to Defaults.

µVision has two operating modes, the Build Mode for creating applications and the Debug Mode for analyzing applications, which offers additional Windows and Dialogs.

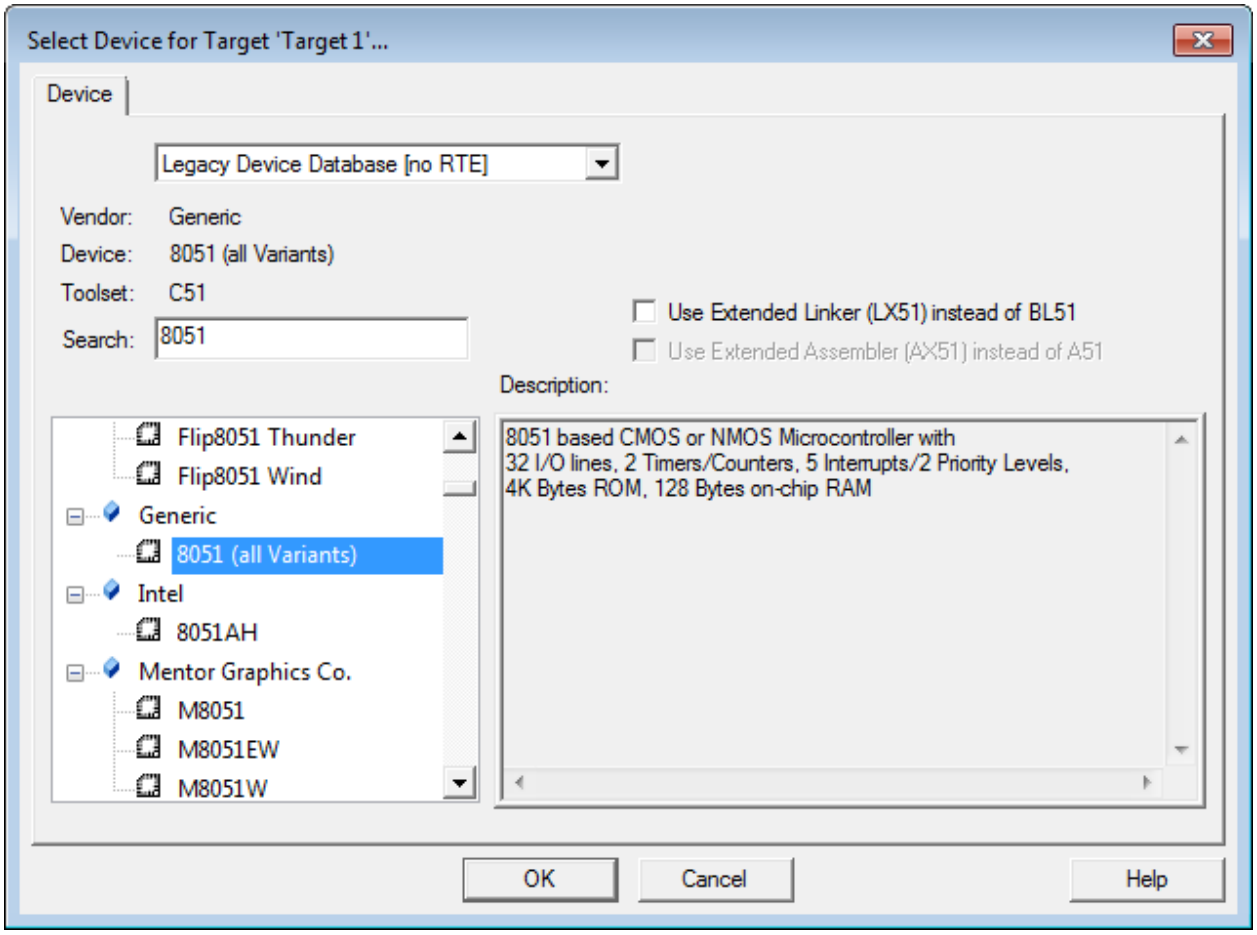
The picture shows some of the key µVision windows in Debug Mode.



4) How to create a project

Ans) The menu Project - New µVision Project creates a new project. Select an empty folder and enter the project name, for example *Project1*. The file extension is **.UVPROJ* for MDK version 4, or **.UVPROJX* for later versions. It is good practice to use a separate folder for each project.

Next, the dialog **Select Device for Target** opens.



Select the device database. Default is **Software Packs**. You can have various local device databases which show up in the drop-down box.

Select the device for your application. This selection defines essential tool settings such as compiler controls, the memory layout for the linker, and the Flash programming algorithms

Copy and Add the Startup Code

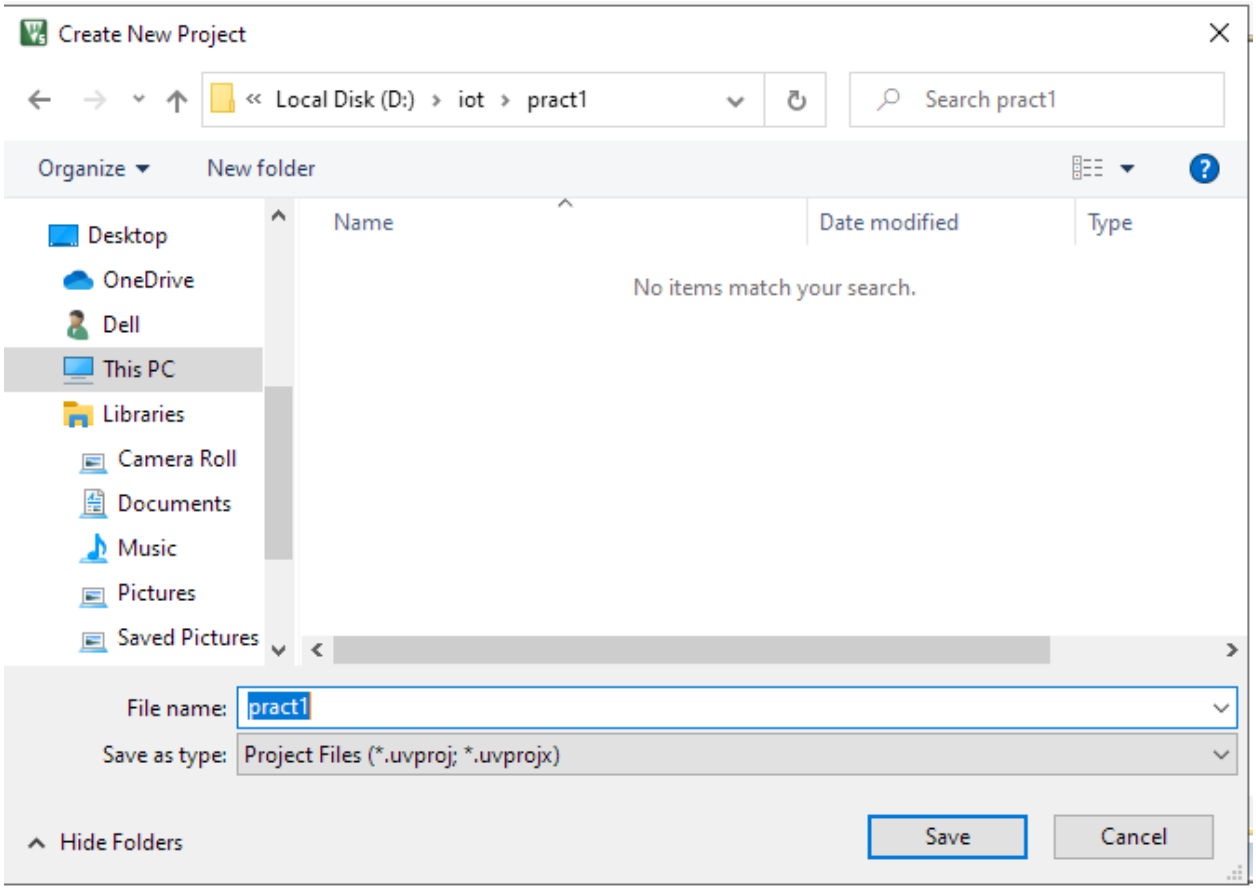
Embedded programs require some kind of microcontroller initialization code that has to match the configuration of the hardware design. The startup code delivered with µVision configures the microcontroller device and initializes the compiler run-time system.For most devices, copy the startup code to the project. This is **required for almost all projects** (**exceptions**: library projects and add-on projects).Therefore, answer this question with **YES**.

Practical 2

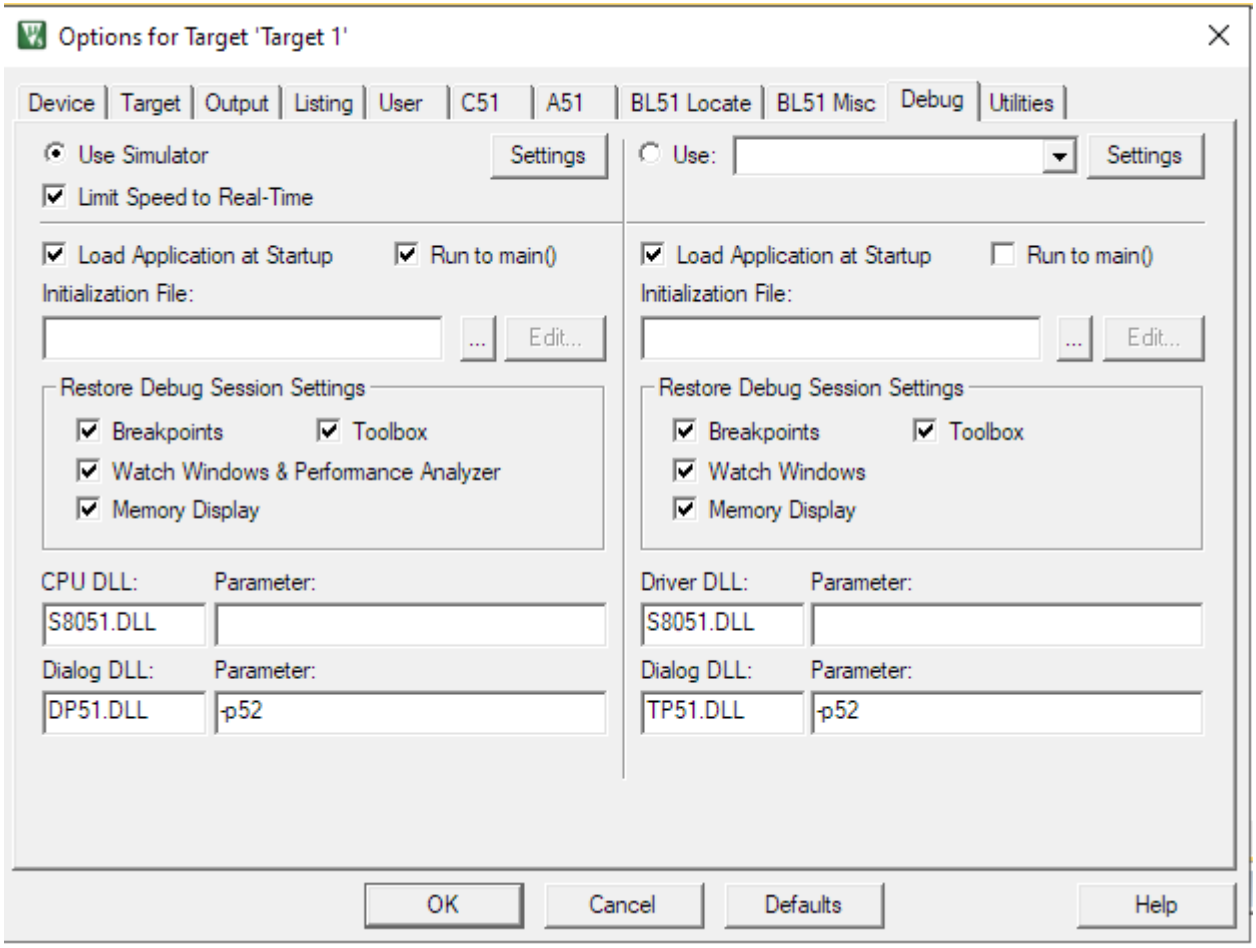
Practical Name: Hello world of Embedded Systems

1)Explain the project creation process from the very beginning with the proper dialog boxes that pop up at every stage during project creation.

Ans) Step1- Create a new project by going to Project tab and select new vision Project.



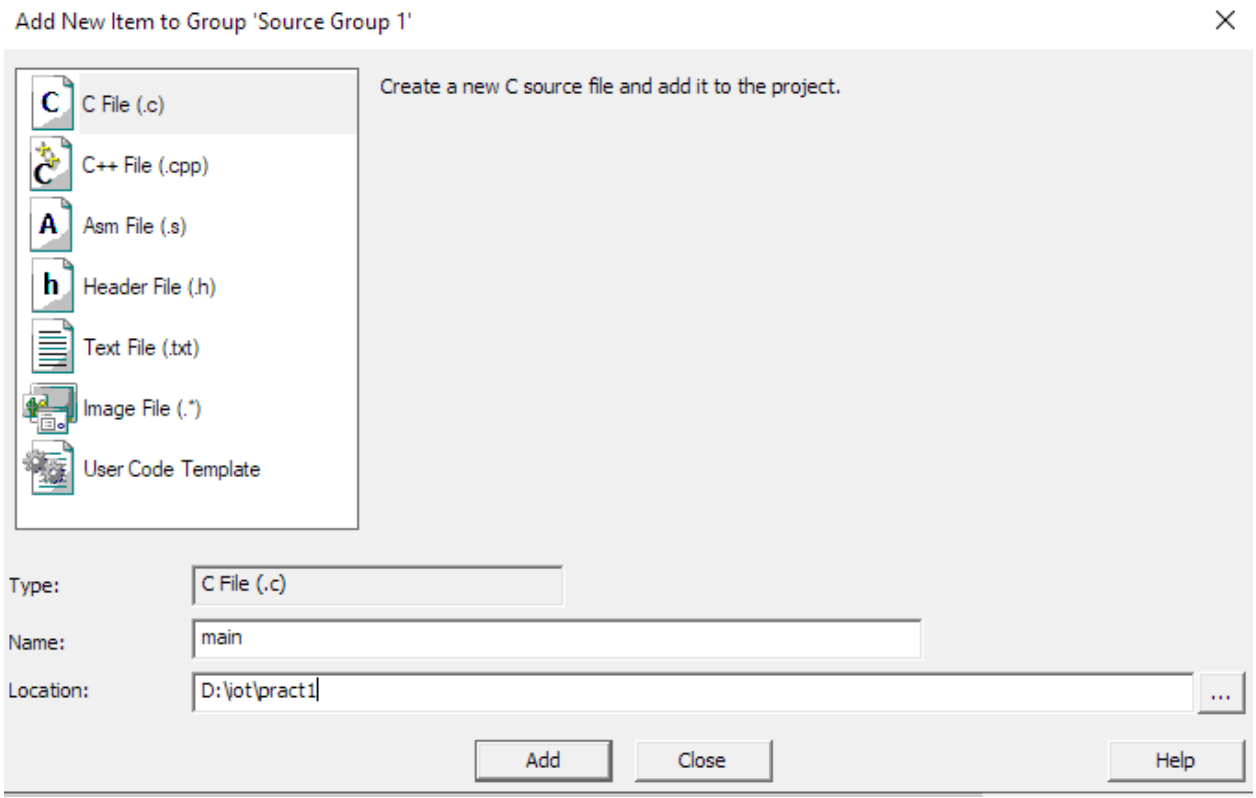
Step2-> After creating the project. Right click on Target and select options for target. Then go to debug tab and checked the ‘Limit speed to real time’ option.



2) Add a main.c file – Put the screen shots for the same.

Step3: Add main.c file by right clicking on source group1 and then click on add new item.

Step4: Select the C file option and provide the name for it.



3) Put the full program for blinking LED.

Code:

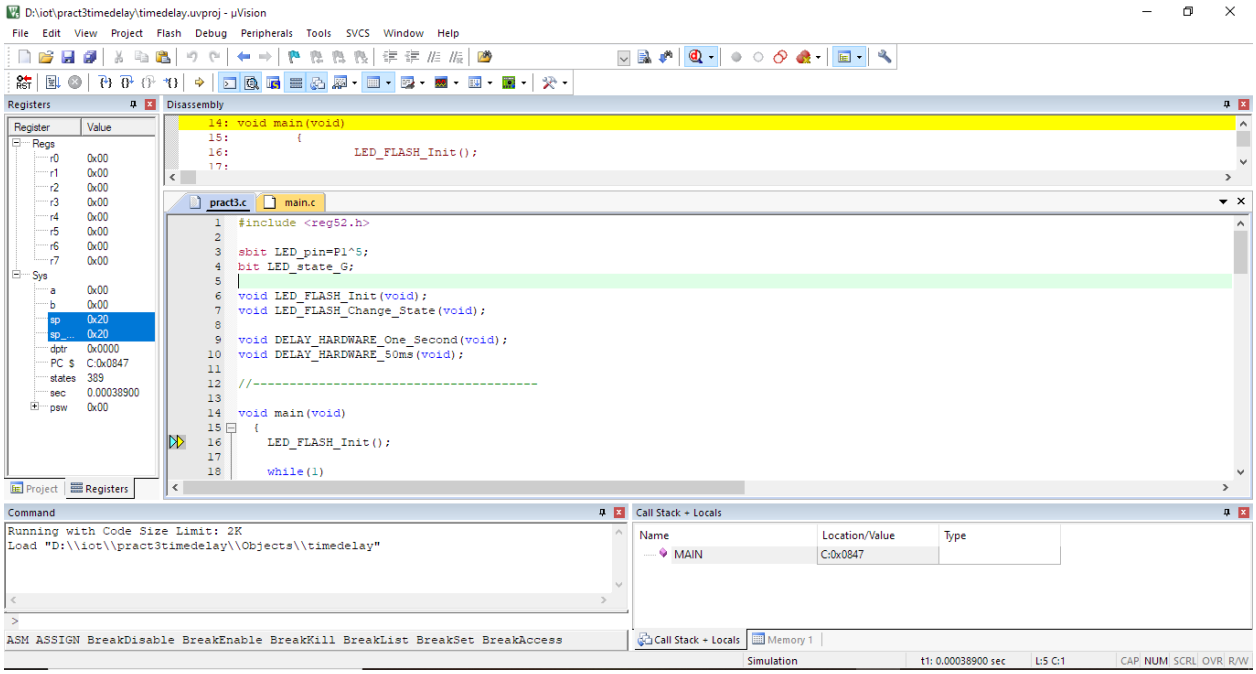
```
#include <reg52.h>

//LED is to be connected to this pin

sbit LED_pin=P1^5;

//stores the LED state

bit LED_state_G;

//Function prototypes

void LED_FLASH_Init(void);

void LED_FLASH_Change_State(void);

void DELAY_LOOP_Wait(const unsigned int);


/*-----------------------------*/

void main(void)

{

        LED_FLASH_Init();

        while(1){

                //Change the LED state(off to on or vice versa)

                LED_FLASH_Change_State();


                //Delay for approx 1000ms

                DELAY_LOOP_Wait(1000);

        }

}

/*----------------------

LED_FLASH_Init()*/
```

```
//Prepare for LED_change_state --see below
void LED_FLASH_Init(void)
{
        LED_state_G=0;
}
//----------------
/*
LED_FLASH_Change_State()
Changes the state of an LED
*/

void LED_FLASH_Change_State(void){
//Change the LED from off to on
                if(LED_state_G==1)
                {
                        LED_state_G=0;
                        LED_pin=0;
                }

                else{
                        LED_state_G=1;
                        LED_pin=1;
                }
                }
```

```
/*----------------

Delay_loop_wait()

Delay duration varies with parameter.

                Parameter is, 'roughly',the delay, in milliseconds,

                on 12MHz 8051(12 osc cycles)

                you need to adjust the timing for your application  */

void DELAY_LOOP_Wait(const unsigned int DELAY){

        unsigned int x,y;

        for(x=0;x<=DELAY;x++){

                for(y=0;y<=120;y++);

                }

                }
```

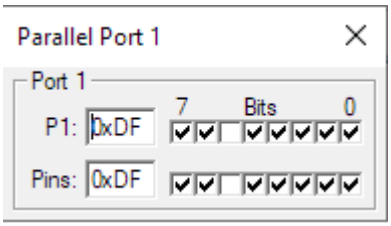
4) Explain the full process of entering the Debug mode and then starting the simulation. Put all the pop-up's that arise in this process and explain each popup.

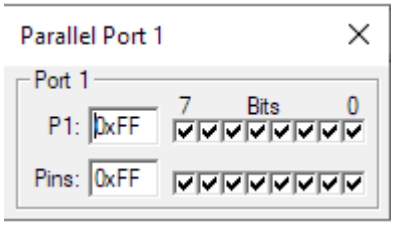
Ans) Go to Debug tab and select 'Start/Stop Debugging'. After that the below screen will change to below settings:



5) Put the screen shot of setting up the IO Port in Debug mode.

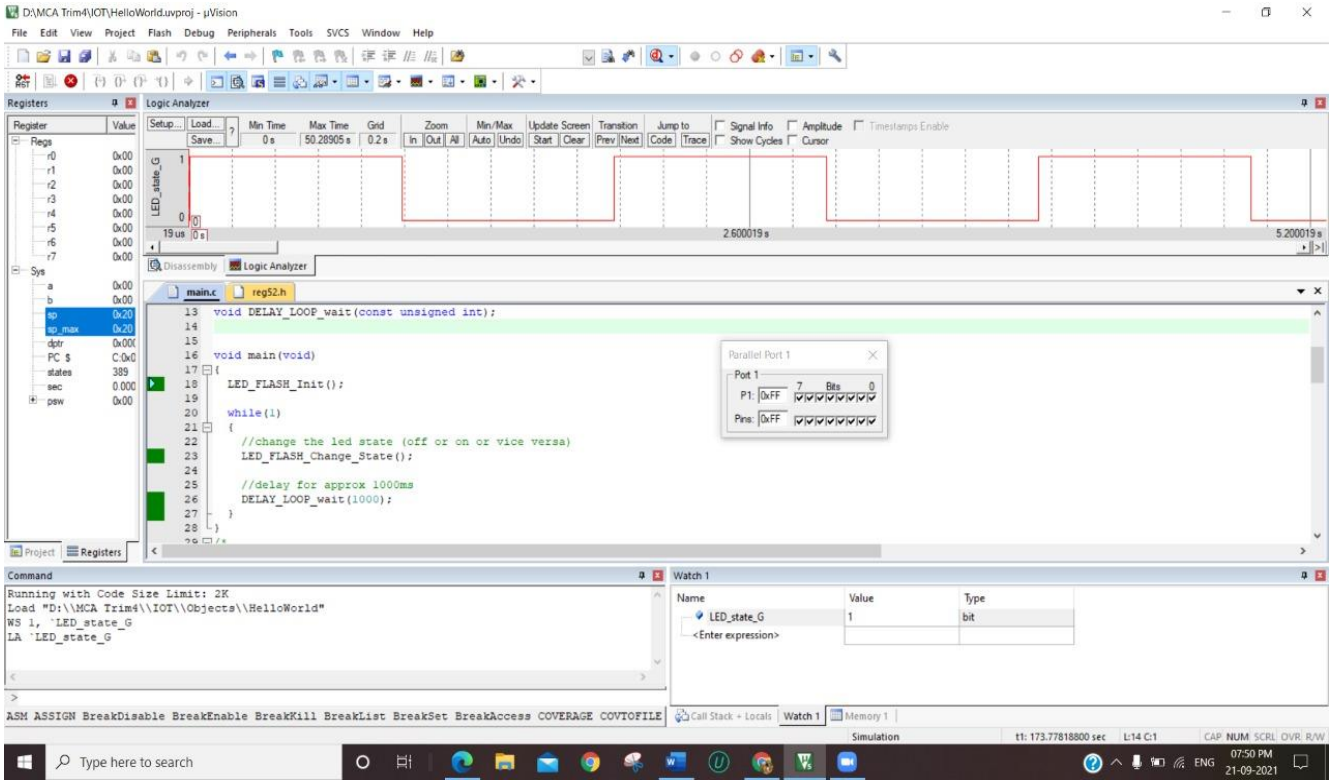
Ans) After starting the debugging mode select 'Peripherals' and select I/O ports option. Select pin1 from the options.





6) Put screen shots of the program running in simulation with the Logic analyzer screen shot showing the waveform of blinking LED with the timing of the blink.

Ans)



7) Explain all the functions used in the program

Ans)

LED FLASH Change State()

Change the state of an LED (or pluses a buzzer , etc) on a specified port pin

must call at twice the required flash rateL thus, for 1 Hz

flash (on for 0.5 seconds, off for 0.5 seconds),

this function must be called twice a second

LED_FLASH_Init()

Prepare for LED_change_state

Delay_loop_wait()

Delay duration varies with parameter.

Parameter is, 'roughly',the delay, in milliseconds,

on 12MHz 8051(12 osc cycles)

you need to adjust the timing for your application

8) Conclude with the timing accuracy of the blink with the explanation of the Delay function and how the timing can be varied by tweaking the delay values.

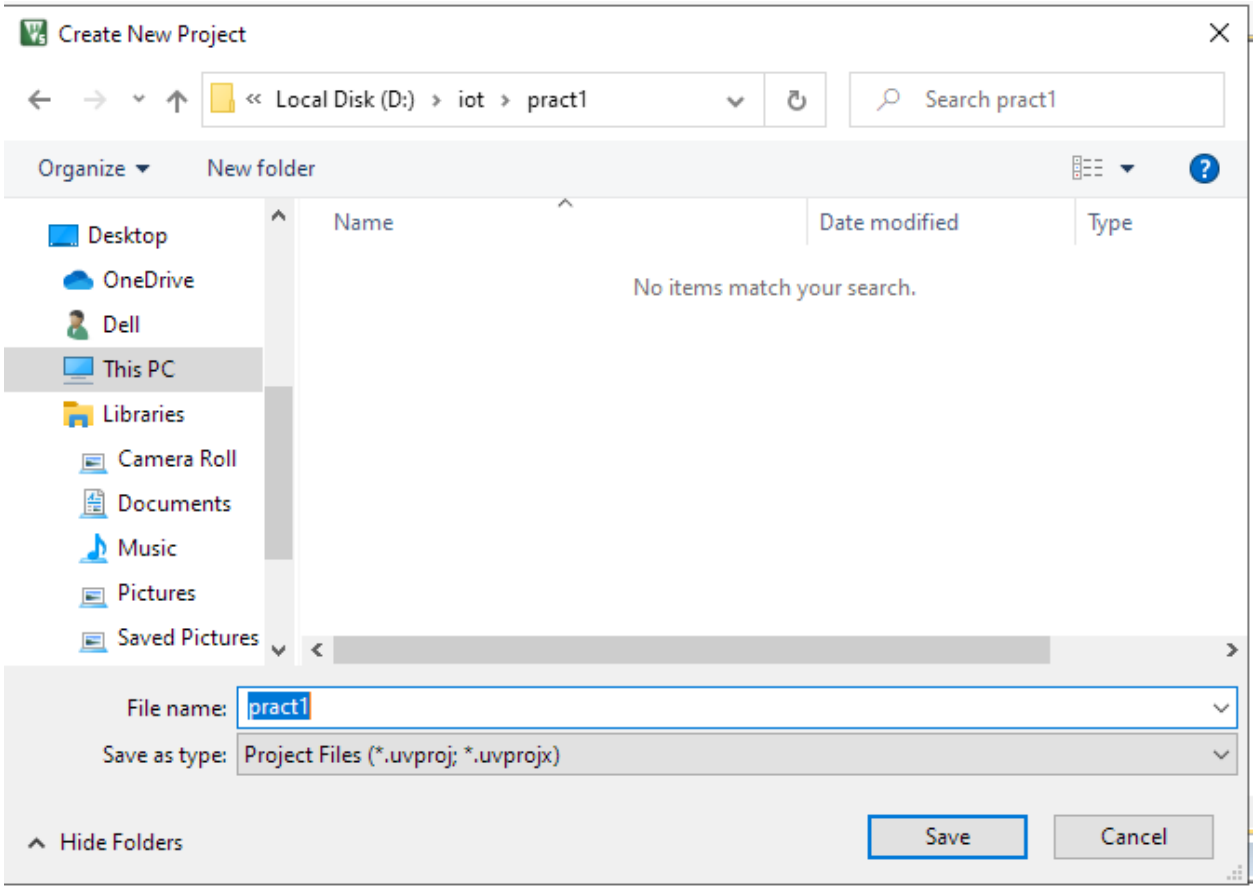
Ans) In delay function, we use 120 as parameter inside the for loop. The timings in delay changes the state of LED accordingly.

Practical 3

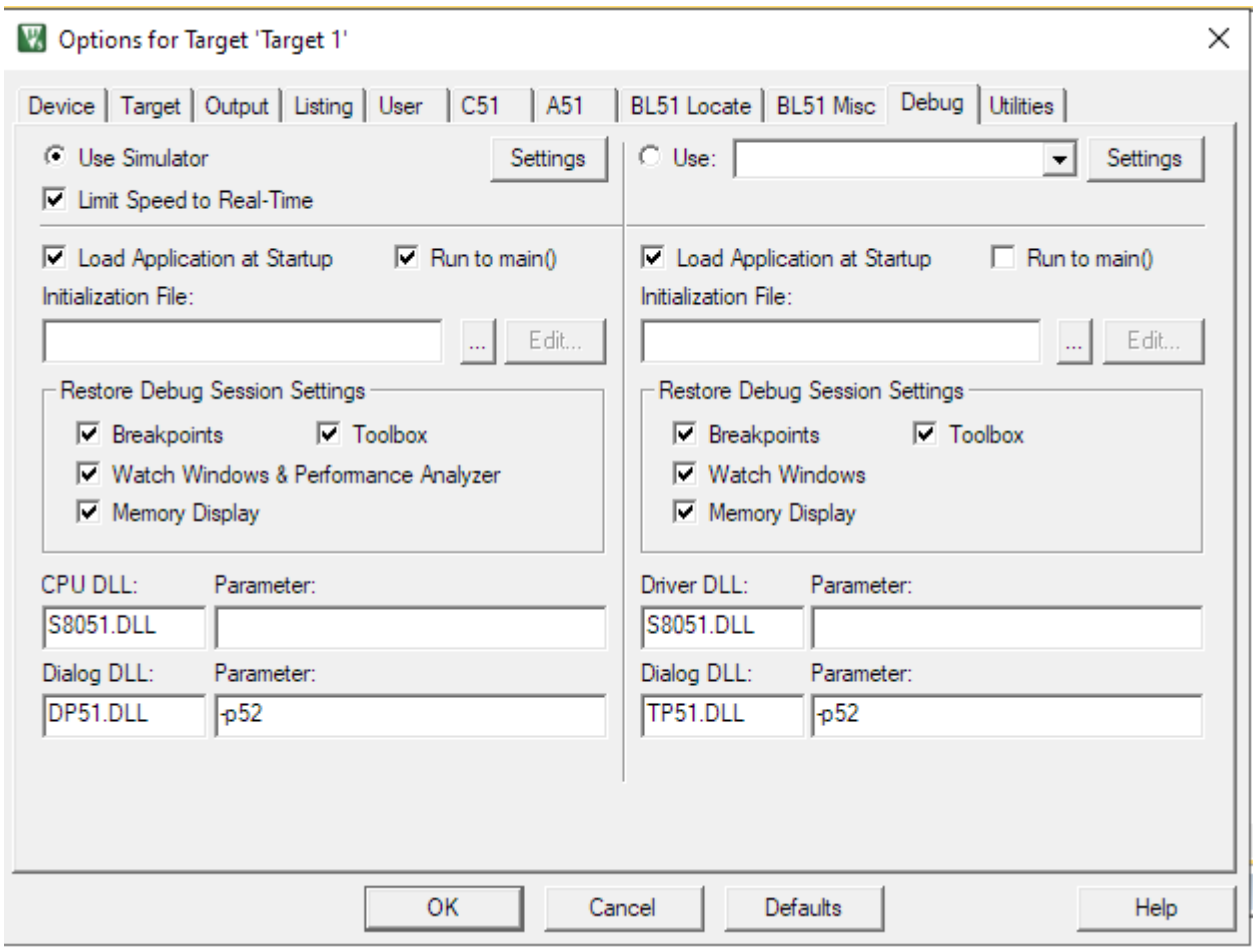
Practical Name: Switch Interface

1)Explain the project creation process from the very beginning with the proper dialog boxes that pop up at every stage during project creation.

Ans) Step1- Create a new project by going to Project tab and select new vision Project.



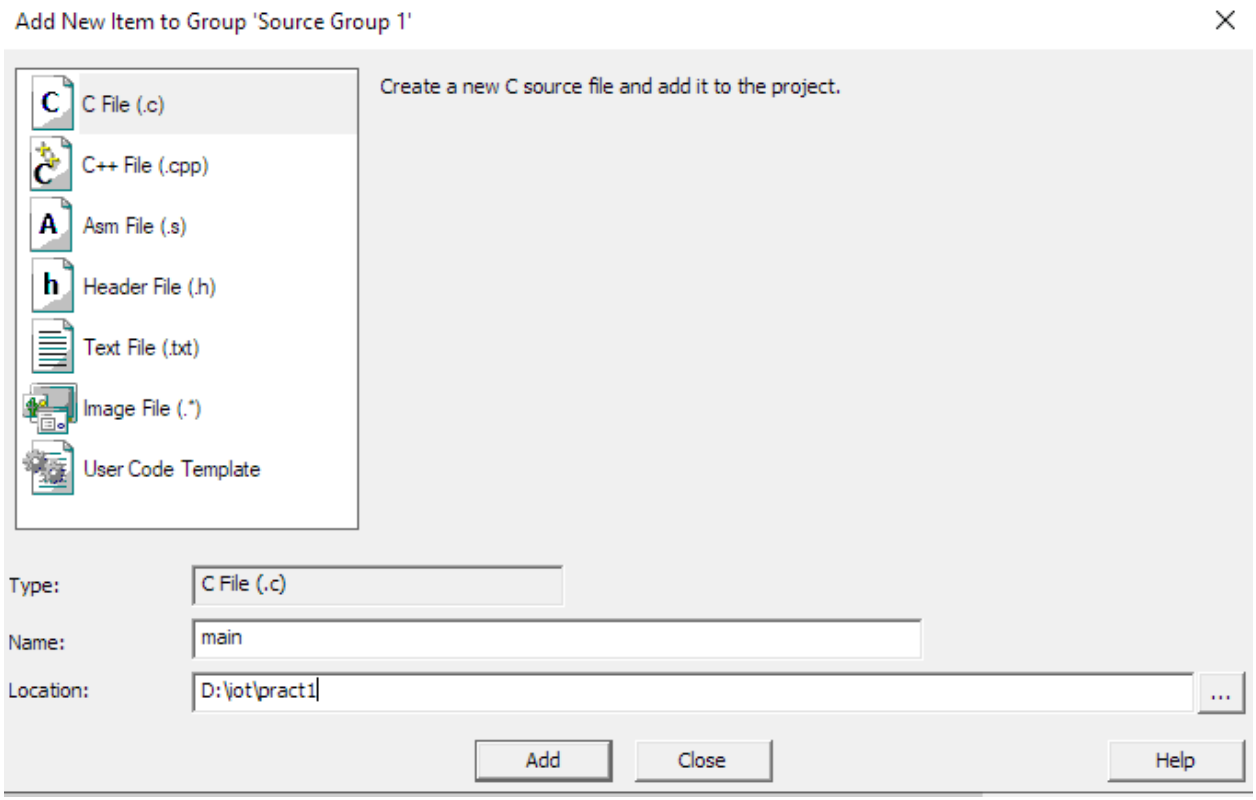
Step2-> After creating the project. Right click on Target and select options for target. Then go to debug tab and checked the ‘Limit speed to real time’ option.



2) Add a main.c file – Put the screen shots for the same.

Step3: Add main.c file by right clicking on source group1 and then click on add new item.

Step4: Select the C file option and provide the name for it.



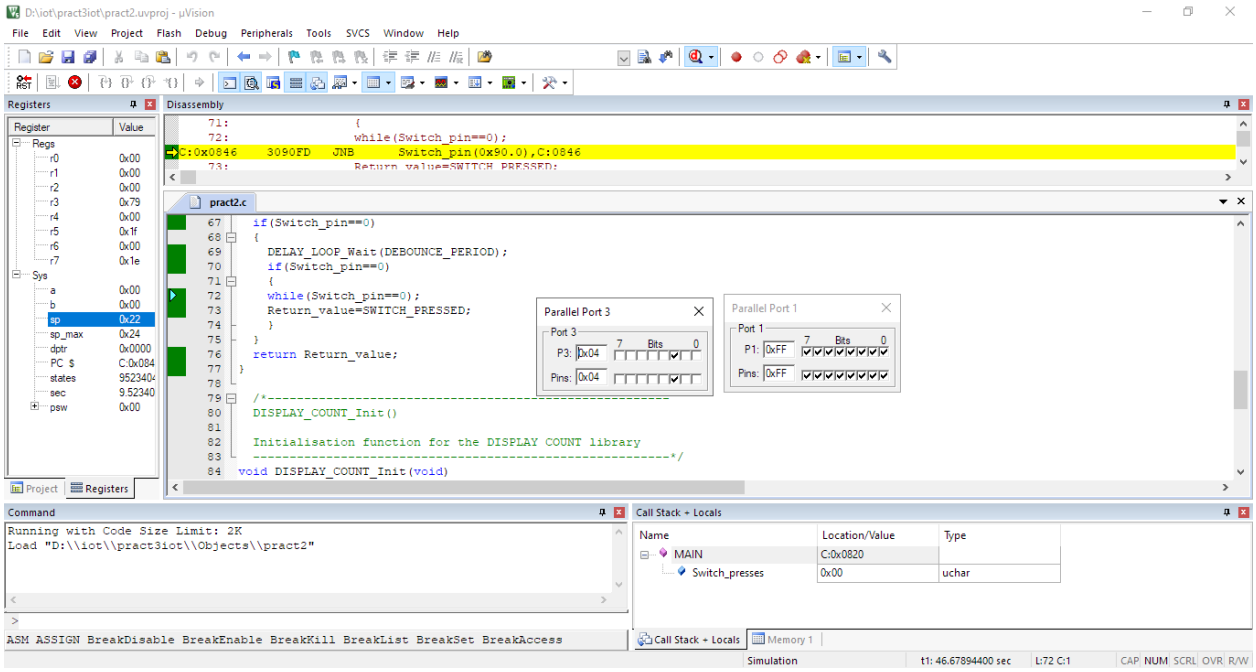
3) Program.

```
#include <reg52.h>

//connect switch to this pin

sbit Switch_pin = P1^0;

//Display count (binary) on this port

#define Count_port P3

//Return values from Switch_Get_Input()

#define SWITCH_NOT_PRESSED (bit) 0

#define SWITCH_PRESSED (bit) 1

//Function Prototype

void SWITCH_Init(void);

bit SWITCH_Get_Input(const unsigned char DEBOUNCE_PERIOD);

void DISPLAY_COUNT_Init(void);

void DISPLAY_COUNT_Update(const unsigned char);

void DELAY_LOOP_Wait(const unsigned int DELAY_MS);

void main(void)

{unsigned char Switch_presses=0;

SWITCH_Init();

        DISPLAY_COUNT_Init();

while(1)

        {

                if(SWITCH_Get_Input(30) == SWITCH_PRESSED)

                {

                        Switch_presses++;
```

```
                }
                DISPLAY_COUNT_Update(Switch_presses);
        }
}

        /*--------------------------------------

        SWITCH_Init()
        Initialissation function for the switch library.

        --------------------------------------*/

void SWITCH_Init(void)
{
        Switch_pin = 1; //Use this pin for input
}/*
        SWITCH_Get_Input()

        Reads and debounces a mechanical switch as follows:
                */
bit SWITCH_Get_Input(const unsigned char DEBOUNCE_PERIOD)
{
        bit Return_value = SWITCH_NOT_PRESSED;
        if(Switch_pin==0)
```

```
        {
                DELAY_LOOP_Wait(DEBOUNCE_PERIOD);
                if(Switch_pin==0)
                {
                while(Switch_pin==0);
                Return_value=SWITCH_PRESSED;
                }
        }
        return Return_value;
}

        /*-------------------------------------------------------
        DISPLAY_COUNT_Init()

        Initialisation function for the DISPLAY COUNT library
        -------------------------------------------------------*/
void DISPLAY_COUNT_Init(void)
{
        Count_port = 0x00;
}

void DISPLAY_COUNT_Update(const unsigned char COUNT)
{
        Count_port = COUNT;
```

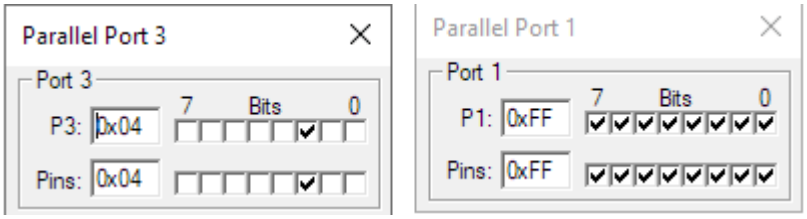
```
}

        /*-------------------------------------------------------

        DISPLAY_LOOP_Wait()


        Delay duration


        -------------------------------------------------------*/


void DELAY_LOOP_Wait(const unsigned int DELAY_MS)

{

        unsigned int x,y;

        for(x=0;x<=DELAY_MS;x++)

        {

                for(y=0;y<=120;y++);

        }

}
```


D:\iot\pract3iot\pract2.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register
Value
Regs
r0 0x00
r1 0x00
r2 0x00
r3 0x79
r4 0x00
r5 0x1f
r6 0x00
r7 0x1e
Sys
a 0x00
b 0x00
sp 0x22
sp_max 0x24
dptr 0x0000
PC $ C:0x084
states 952340
sec 9.52340
psw 0x00
Project
Registers
Disassembly
71: {
72: while(Switch_pin==0);
C:0x0846 3090FD JNB Switch_pin(0x90.0),C:0846
73: Return_value=SWITCH_PRESSED;
pract2.c
67 if(Switch_pin==0)
68 {
69 DELAY_LOOP_Wait(DEBOUNCE_PERIOD);
70 if(Switch_pin==0)
71 {
72 while(Switch_pin==0);
73 Return_value=SWITCH_PRESSED;
74 }
75 }
76 return Return_value;
77 }
78
79 /*---
80 DISPLAY_COUNT_Init()
81
82 Initialisation function for the DISPLAY COUNT library
83 ---*/
84 void DISPLAY_COUNT_Init(void)
Parallel Port 3
Port 3
P3: 0x04
Pins: 0x04
7 Bits 0
Parallel Port 1
Port 1
P1: 0xFF
Pins: 0xFF
7 Bits 0
Command
Running with Code Size Limit: 2K
Load "D:\\iot\\pract3iot\\Objects\\pract2"
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Call Stack + Locals
Name
Location/Value
Type
MAIN
C:0x0820
Switch_presses
0x00
uchar
Call Stack + Locals
Memory 1
Simulation
t1: 46.67894400 sec
L:72 C:1
CAP NUM SCRL OVR R/W

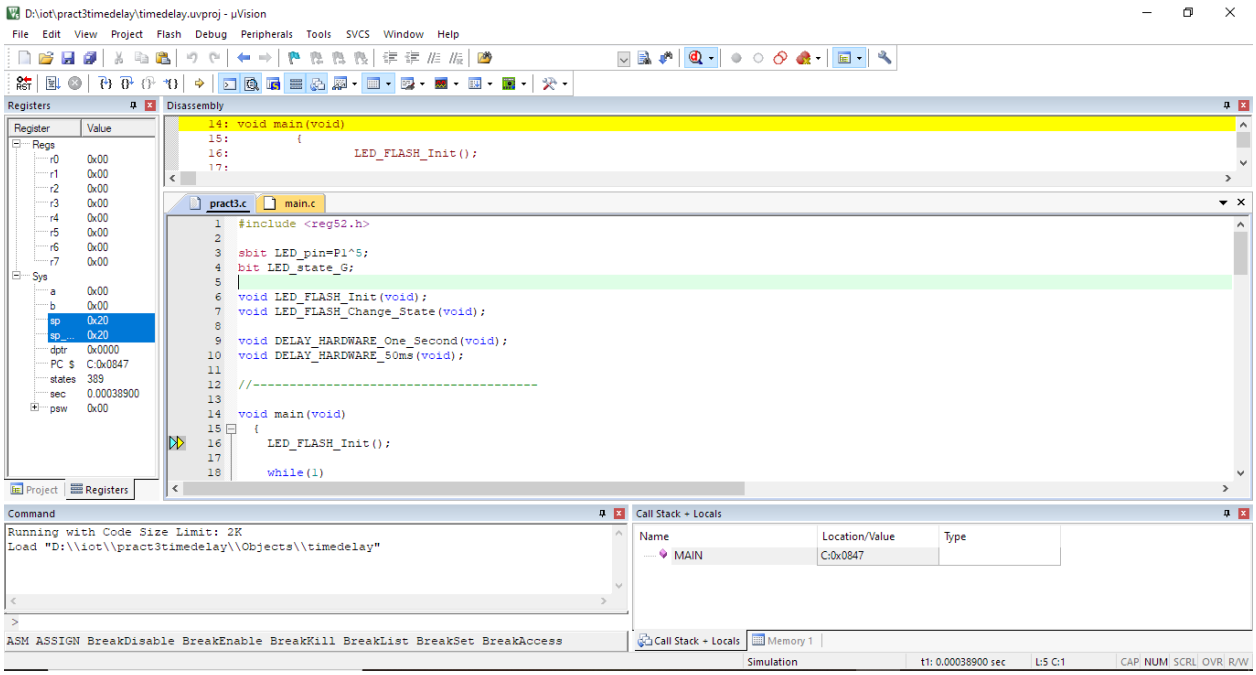
4) Explain the full process of entering the Debug mode and then starting the simulation. Put all the pop-up's that arise in this process and explain each popup.

Ans) Go to Debug tab and select 'Start/Stop Debugging'. After that the below screen will change to below settings:

5) Put the screen shot of setting up the IO Port in Debug mode.



6) Put screen shots of the program running in simulation.



7) Explain the concept of Switch debouncing and how it is achieved in this program

Ans) Switch debouncing is one of those things you generally have to live with when playing with switches and digital circuits. If you want to input a manual switch signal into a digital circuit you'll need to debounce the signal so a single press doesn't appear like multiple presses. Here, SWITCH_Get_Input function helps in Switch debouncing.

8) Explain all the functions used in the program

Ans) SWITCH_Init()

Initialissation function for the switch library.

SWITCH_Get_Input()

Reads and debounces a mechanical switch

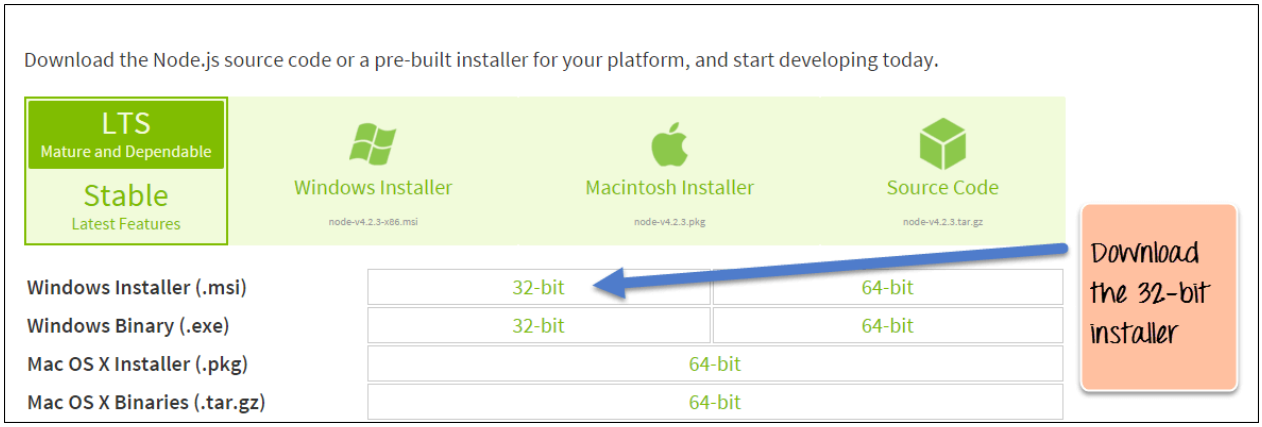
DISPLAY_COUNT_Init()

Initialisation function for the DISPLAY COUNT library

Practical 4: Installing Node.js ,Node-Red and Node Red interface

Step 1) Download Node.js Installer for Windows

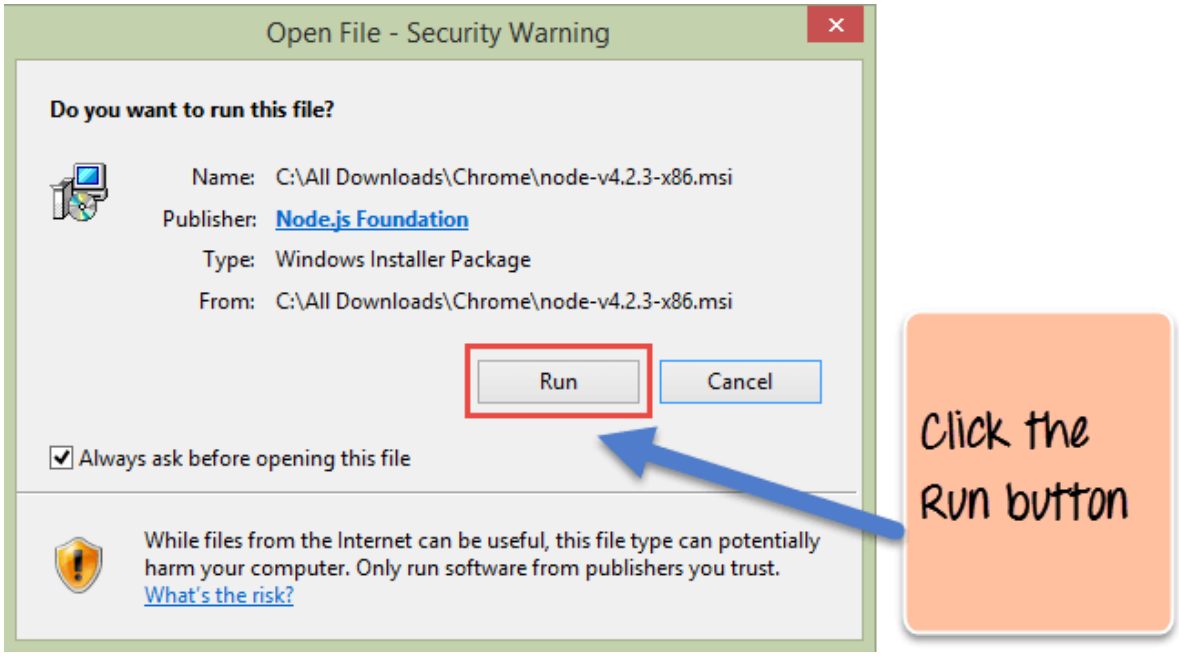
Go to the site https://nodejs.org/en/download/ and download the necessary binary files.



Step 2) Run the installation

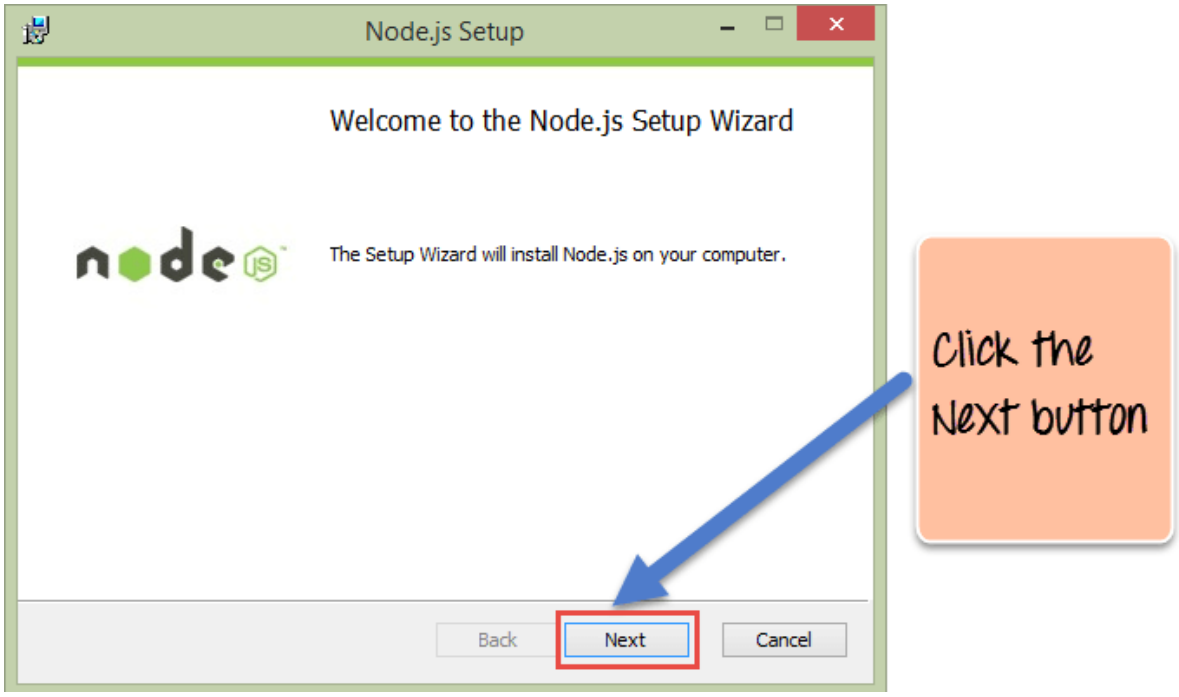
Double click on the downloaded .msi file to start the installation.

Click the Run button on the first screen to begin the installation.

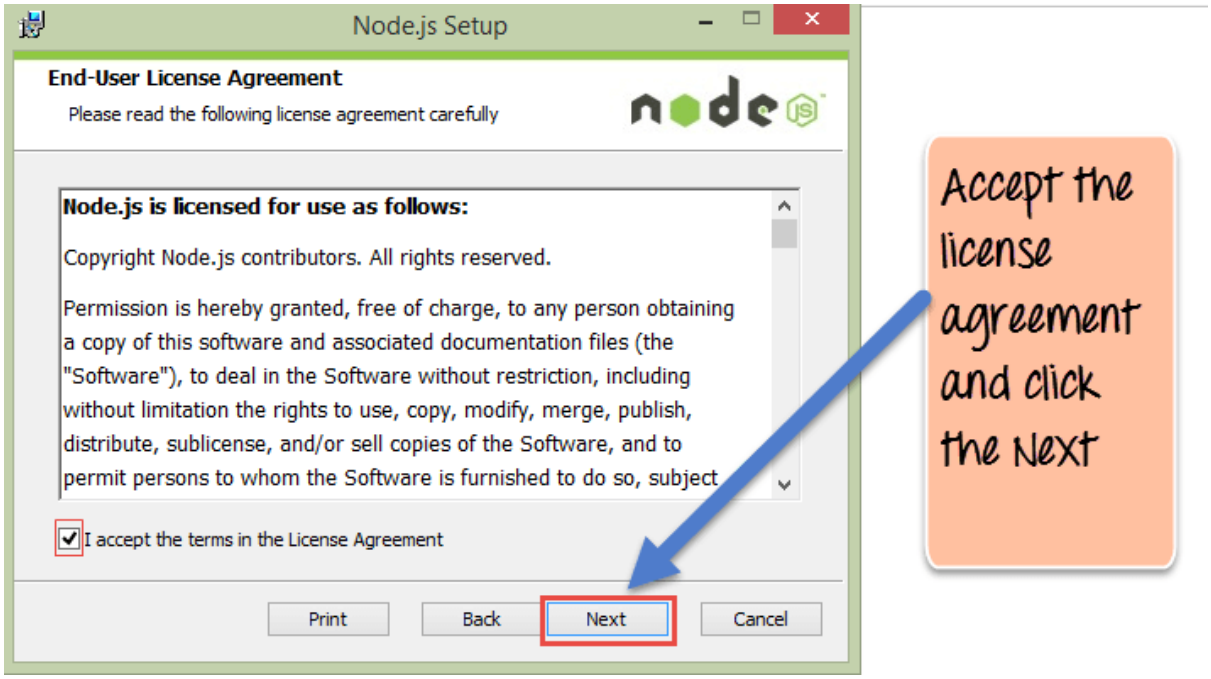


Step 3) Continue with the installation steps

In the next screen, click the "Next" button to continue with the installation



Step 4) Accept the terms and conditions
In the next screen, Accept the license agreement and click on the Next button.

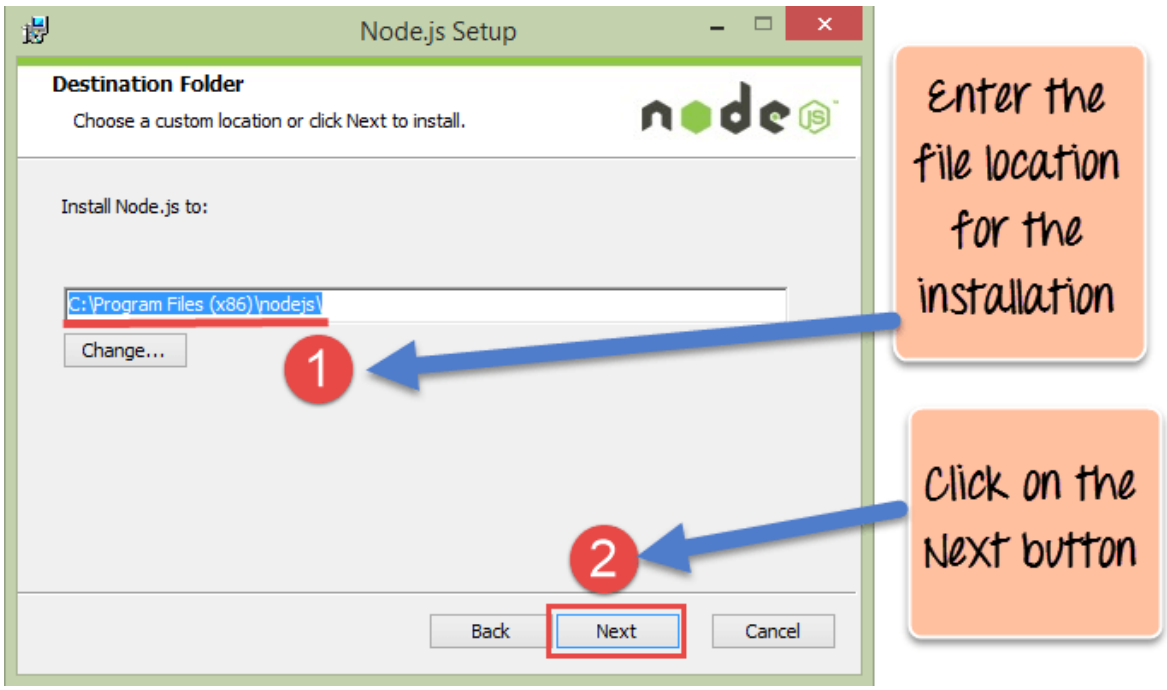


Step 5) Set up the path

In the next screen, choose the location where Node.js needs to be installed and then click on the Next button.

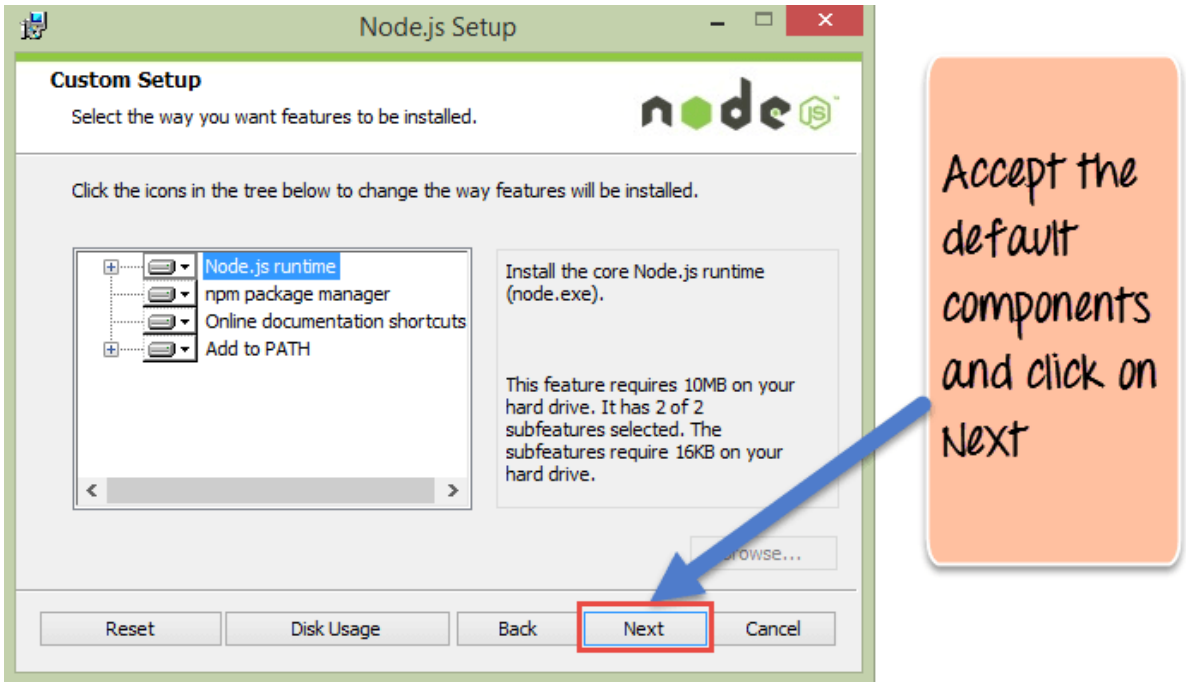
1. First, enter the file location for the installation of Node.js. This is where the files for Node.js will be stored after the installation.

2. Click on the Next button to proceed ahead with the installation.



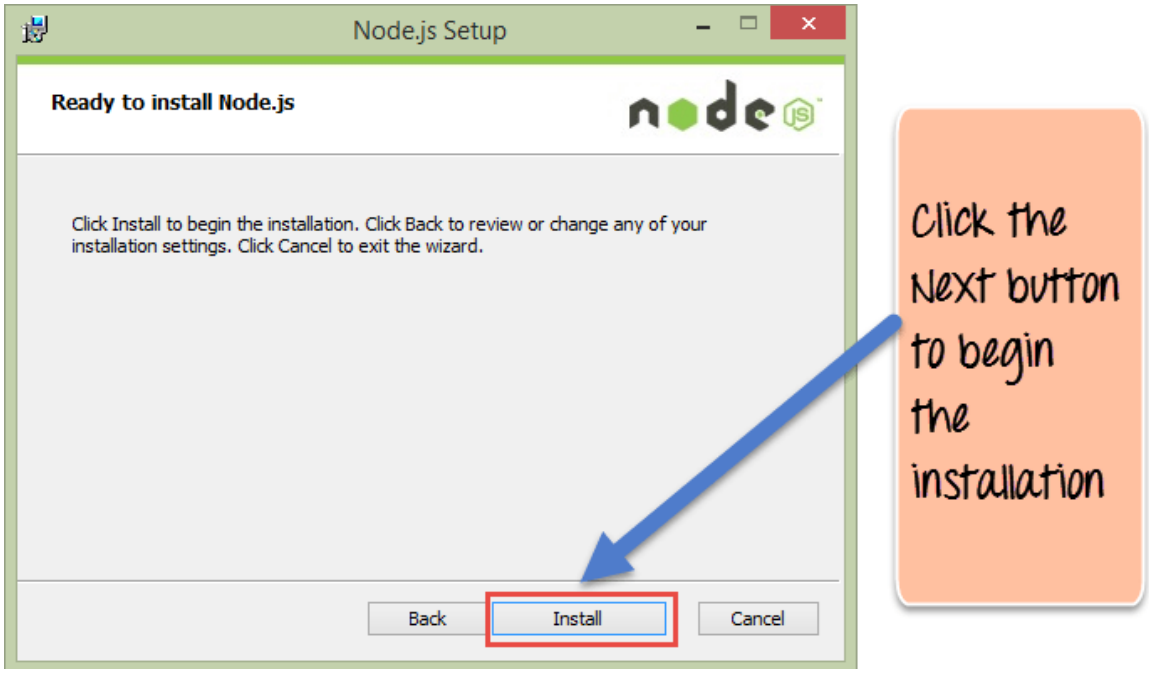
Step 6) Select the default components to be installed

Accept the default components and click on the Next button.



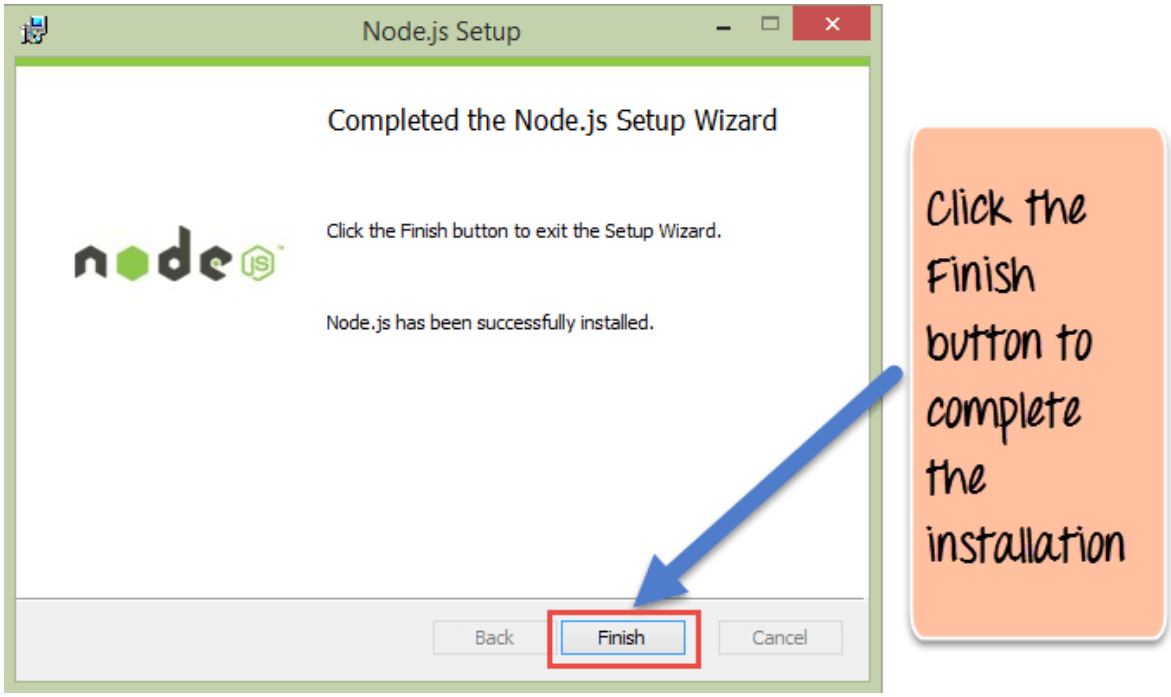
Step 7) Start the installation

In the next screen, click the Install button to start installing Node.js on Windows.

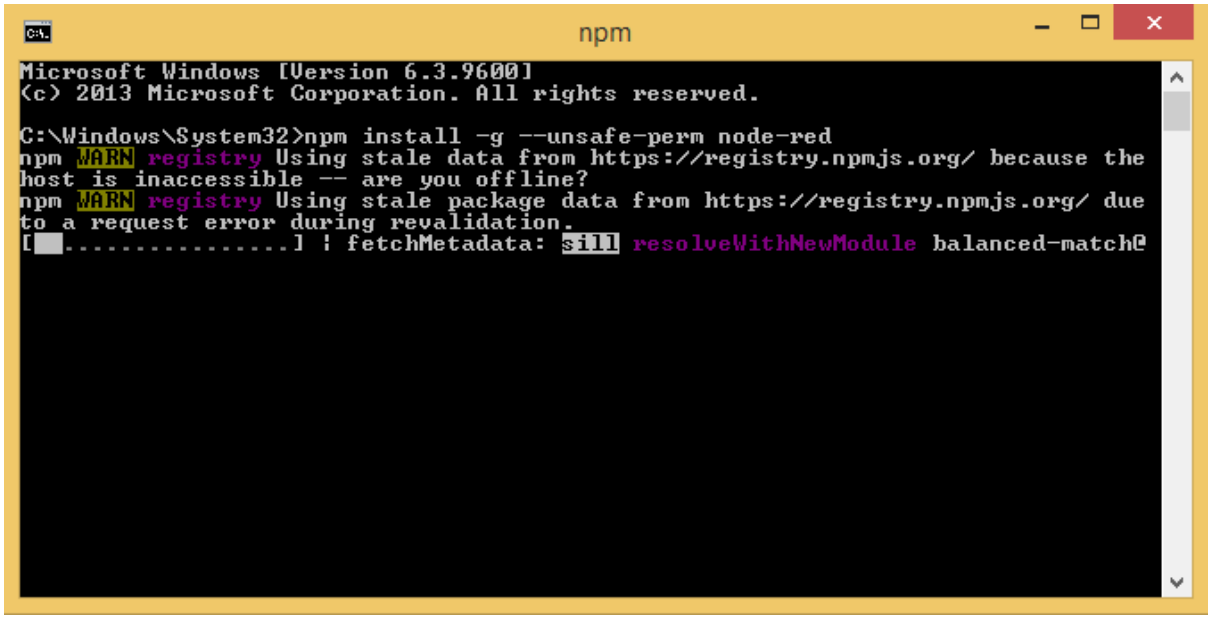


Step 8) Complete the installation

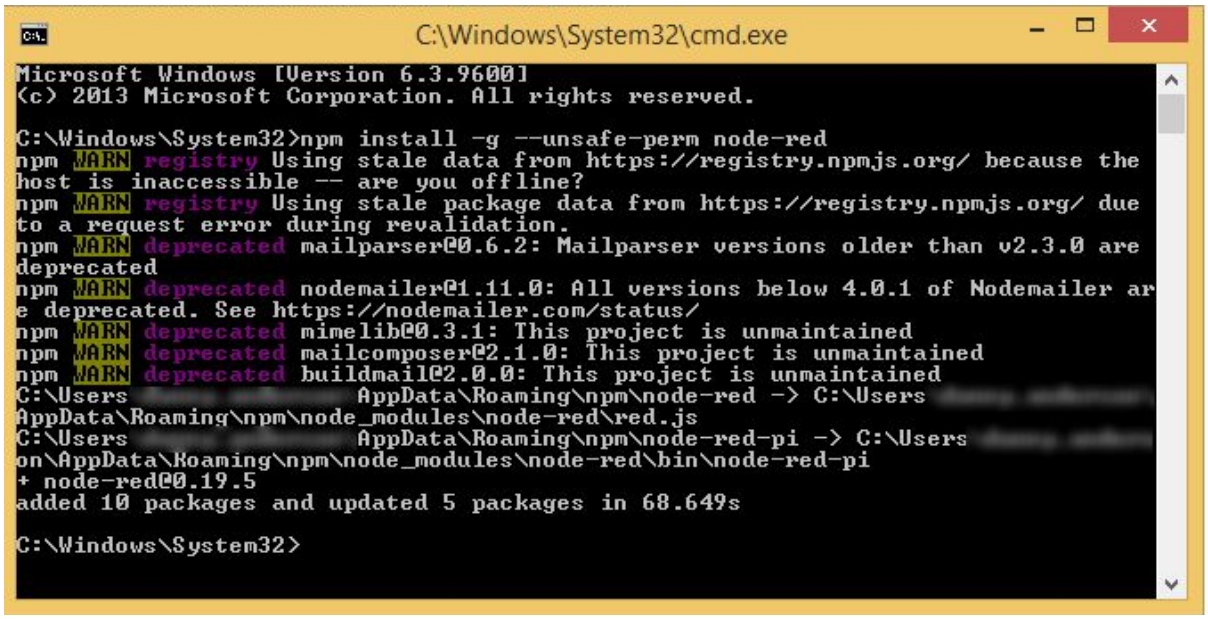
Click the Finish button to complete the installation.



Now we can begin to install Node-RED, navigate to CMD and input the command:



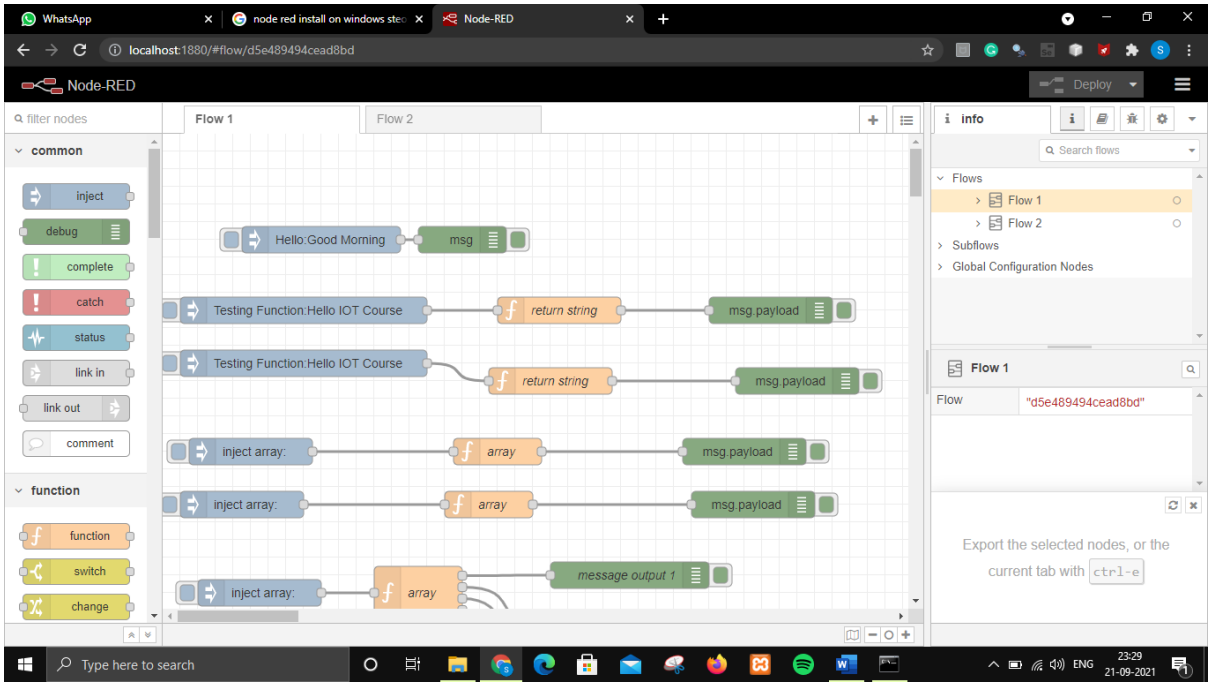
Once complete, the output should look like this:



start the node-red flow using command prompt

```
C:\Users\admin>node-red
21 Sep 19:40:26 - [info]

Welcome to Node-RED
===================

21 Sep 19:40:26 - [info] Node-RED version: v2.0.5
21 Sep 19:40:26 - [info] Node.js  version: v14.17.5
21 Sep 19:40:26 - [info] Windows_NT 10.0.19042 x64 LE
21 Sep 19:40:44 - [info] Loading palette nodes
21 Sep 19:41:34 - [info] Dashboard version 2.30.0 started at /ui
21 Sep 19:41:35 - [info] Settings file  : C:\Users\admin\.node-red\settings.js
21 Sep 19:41:35 - [info] Context store  : 'default' [module=memory]
21 Sep 19:41:35 - [info] User directory : \Users\admin\.node-red
21 Sep 19:41:35 - [warn] Projects disabled : editorTheme.projects.enabled=false
21 Sep 19:41:35 - [info] Flows file     : \Users\admin\.node-red\flows.json
21 Sep 19:41:35 - [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.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

21 Sep 19:41:35 - [info] Server now running at http://127.0.0.1:1880/
21 Sep 19:41:35 - [info] Starting flows
21 Sep 19:41:35 - [info] Started flows
21 Sep 20:15:37 - [info] Stopping flows
21 Sep 20:15:37 - [info] Stopped flows
21 Sep 20:15:37 - [info] Starting flows
21 Sep 20:15:37 - [info] Started flows
21 Sep 20:15:44 - [error] [function:return string] Function tried to send a message of type string
```

DashBoard

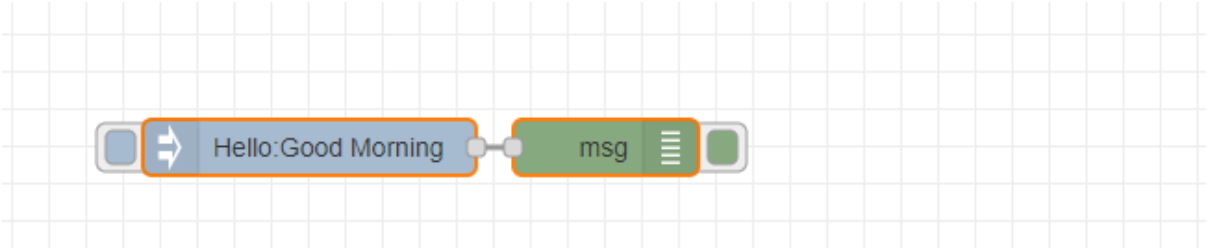


Practical 5 : Introducing the inject, function, debug and switch nodes

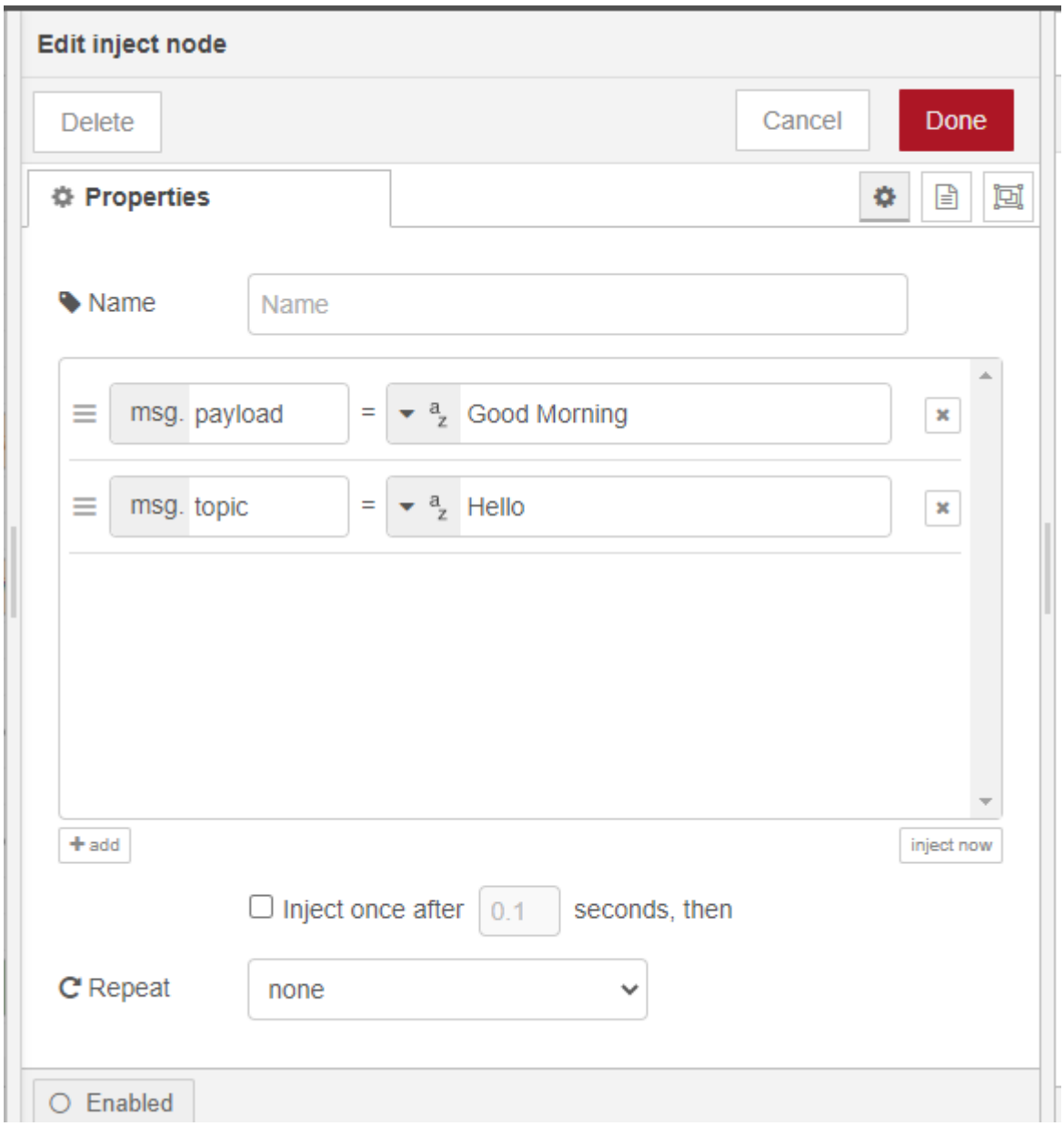
Step 1: start the node-red flow using command prompt

```
C:\Users\admin>node-red
21 Sep 19:40:26 - [info]

Welcome to Node-RED
===================

21 Sep 19:40:26 - [info] Node-RED version: v2.0.5
21 Sep 19:40:26 - [info] Node.js  version: v14.17.5
21 Sep 19:40:26 - [info] Windows_NT 10.0.19042 x64 LE
21 Sep 19:40:44 - [info] Loading palette nodes
21 Sep 19:41:34 - [info] Dashboard version 2.30.0 started at /ui
21 Sep 19:41:35 - [info] Settings file  : C:\Users\admin\.node-red\settings.js
21 Sep 19:41:35 - [info] Context store  : 'default' [module=memory]
21 Sep 19:41:35 - [info] User directory : \Users\admin\.node-red
21 Sep 19:41:35 - [warn] Projects disabled : editorTheme.projects.enabled=false
21 Sep 19:41:35 - [info] Flows file     : \Users\admin\.node-red\flows.json
21 Sep 19:41:35 - [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.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

21 Sep 19:41:35 - [info] Server now running at http://127.0.0.1:1880/
21 Sep 19:41:35 - [info] Starting flows
21 Sep 19:41:35 - [info] Started flows
21 Sep 20:15:37 - [info] Stopping flows
21 Sep 20:15:37 - [info] Stopped flows
21 Sep 20:15:37 - [info] Starting flows
21 Sep 20:15:37 - [info] Started flows
21 Sep 20:15:44 - [error] [function:return string] Function tried to send a message of type string
```

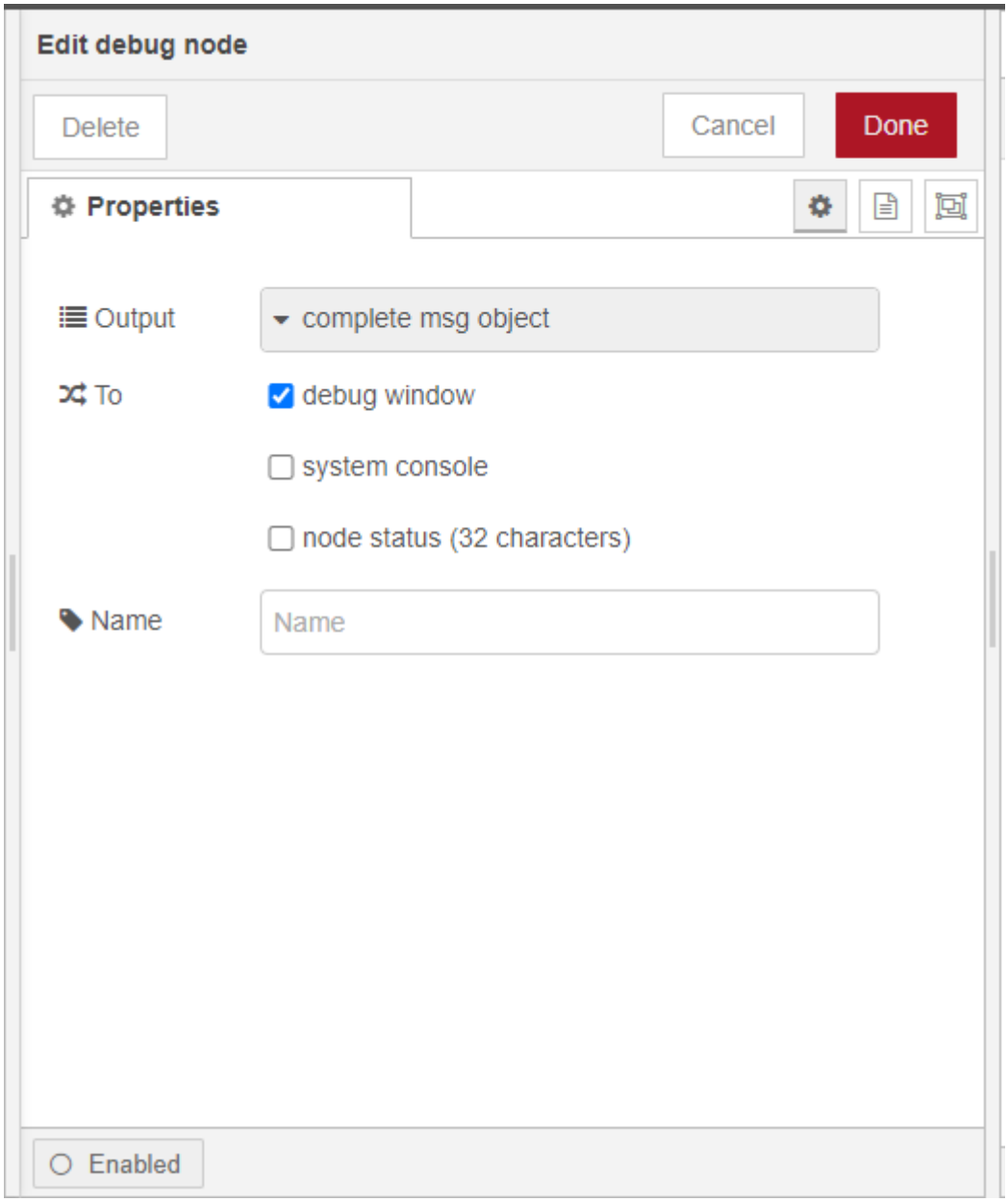
1.Hello Flow SS:-

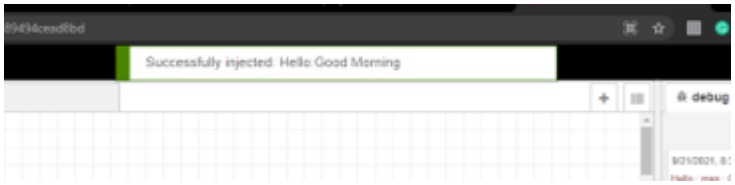


Inject Node:-


Edit inject node
Delete
Cancel
Done
Properties
Name
Name
msg. payload
=
Good Morning
msg. topic
=
Hello
add
inject now
Inject once after
0.1
seconds, then
Repeat
none
Enabled

Message Node:-

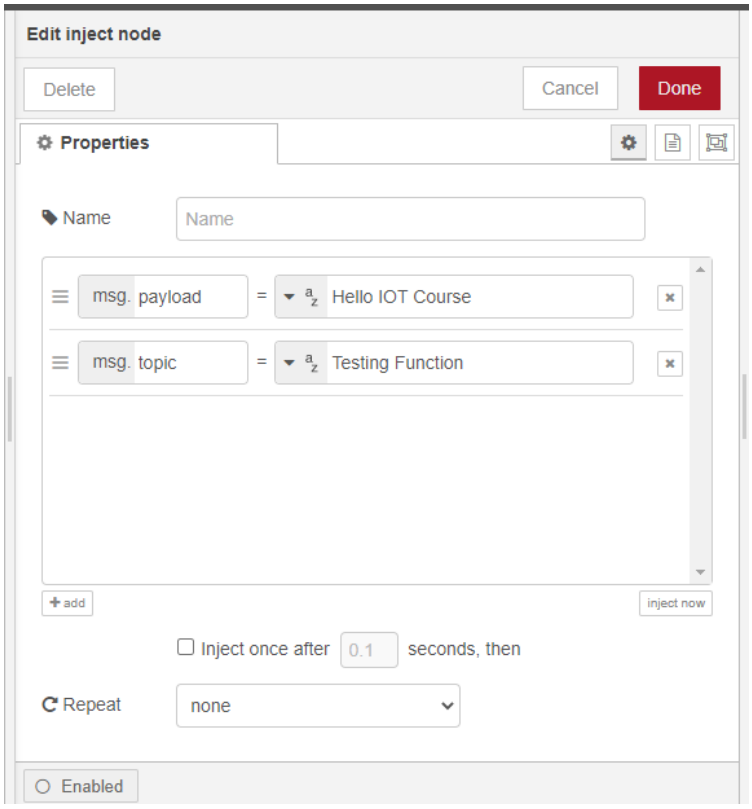




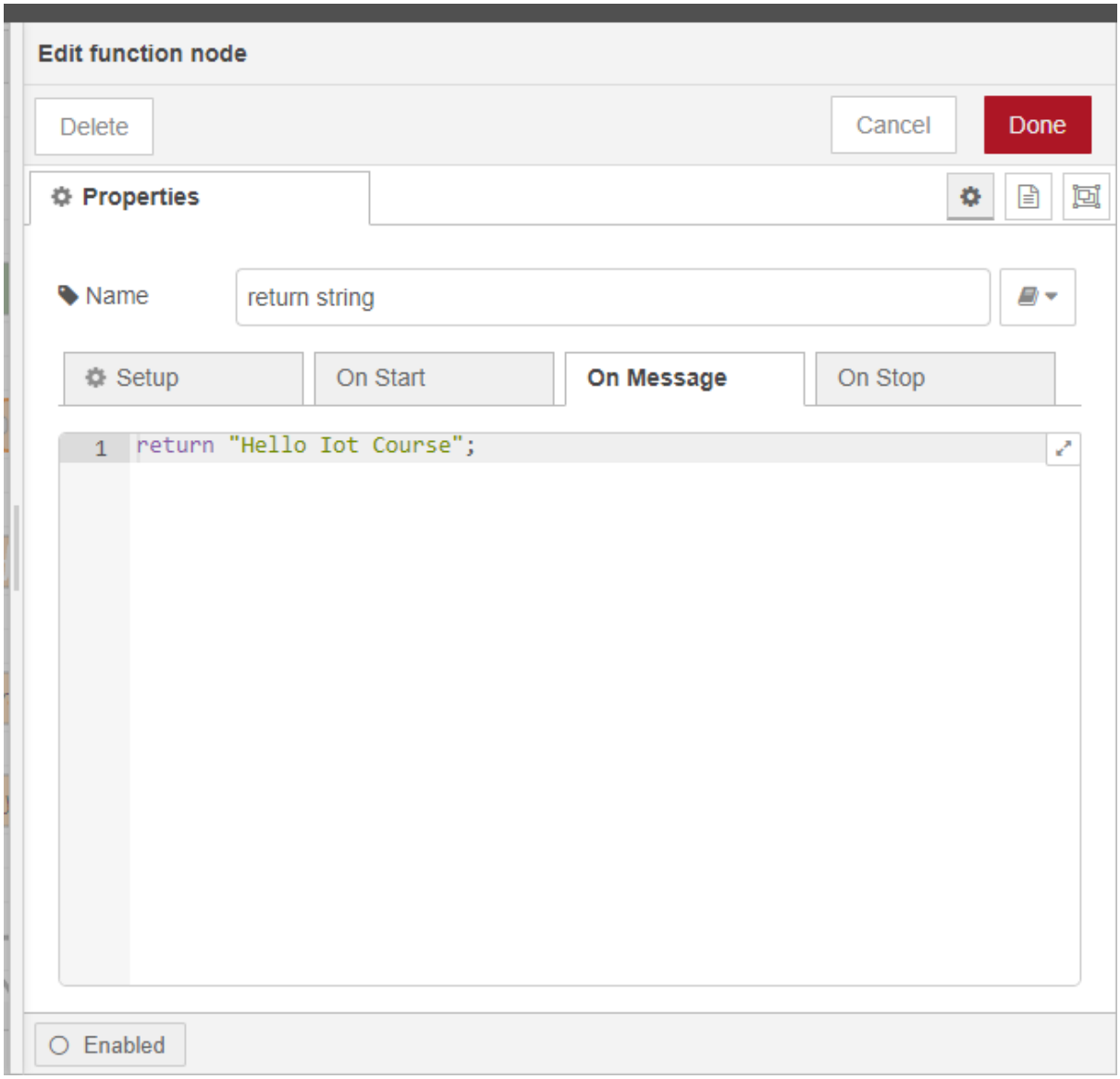
Debug window:-

2. Hello IOT flow with the erroneous function

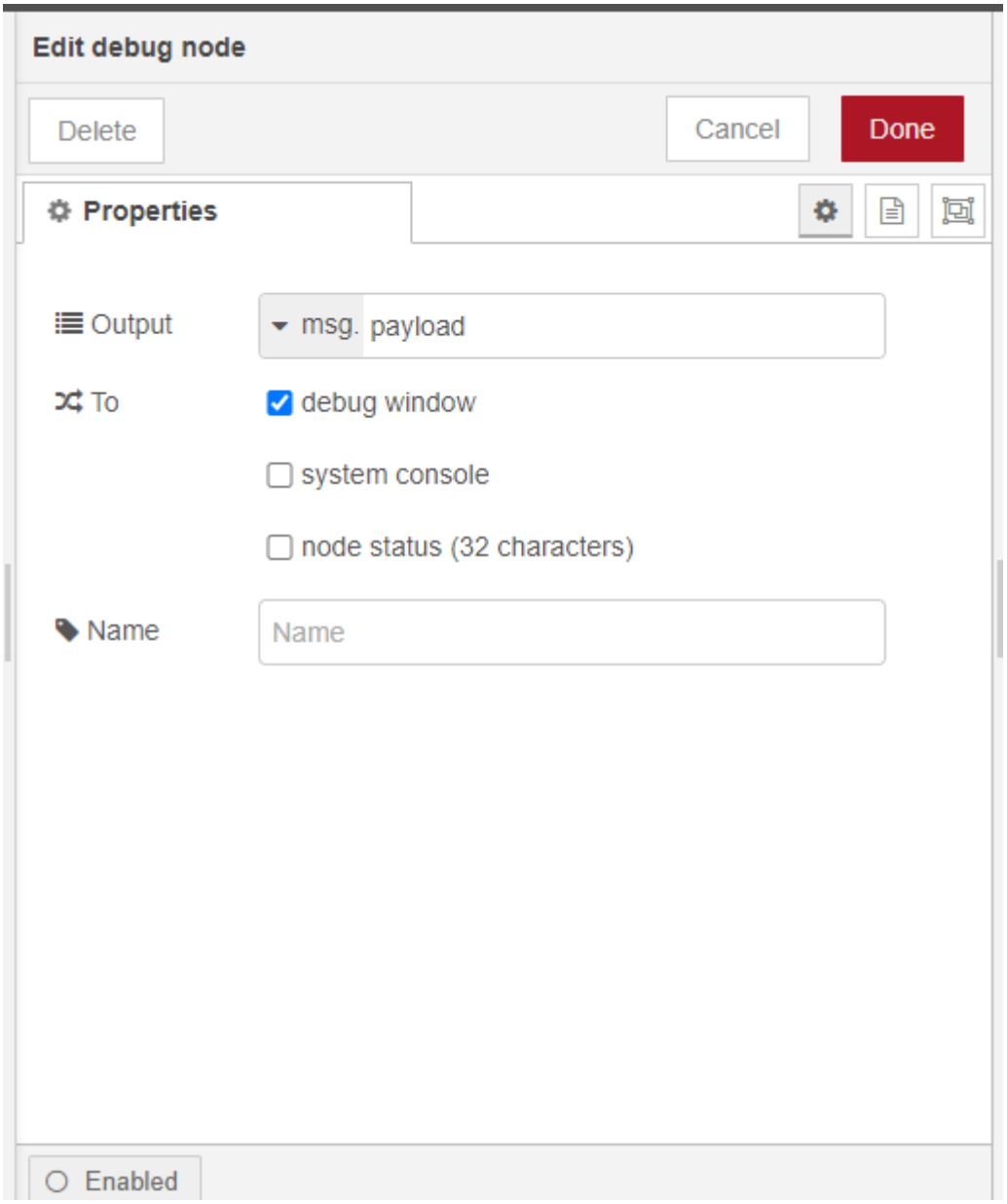
Inject Node:-



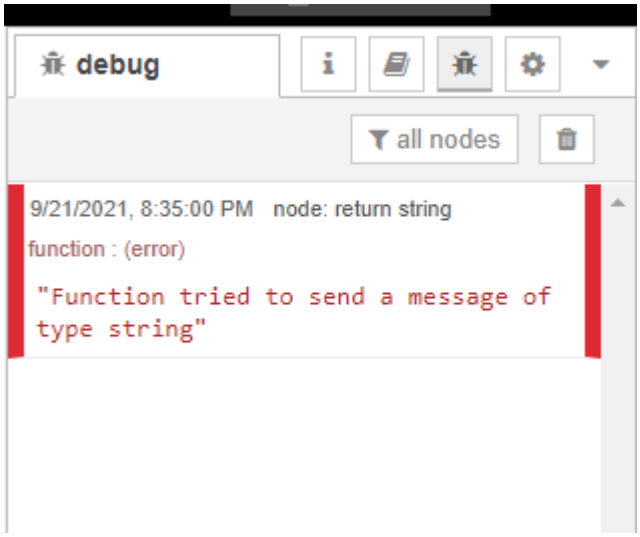
Function Node:-



Debug node:-

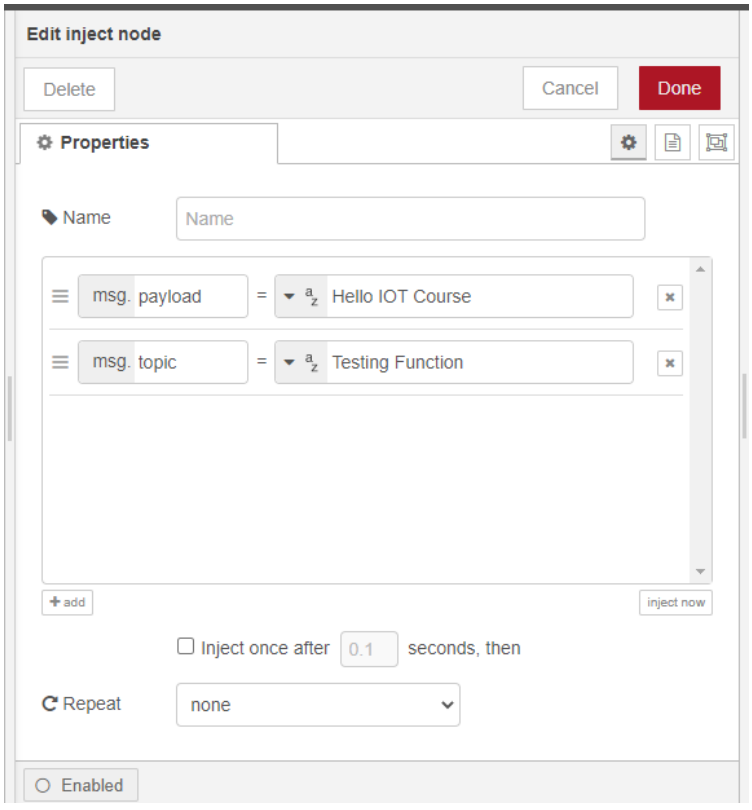

Edit debug node
Delete
Cancel
Done
Properties
Output
msg.
payload
To
debug window
system console
node status (32 characters)
Name
Name
Enabled

Debug window:-

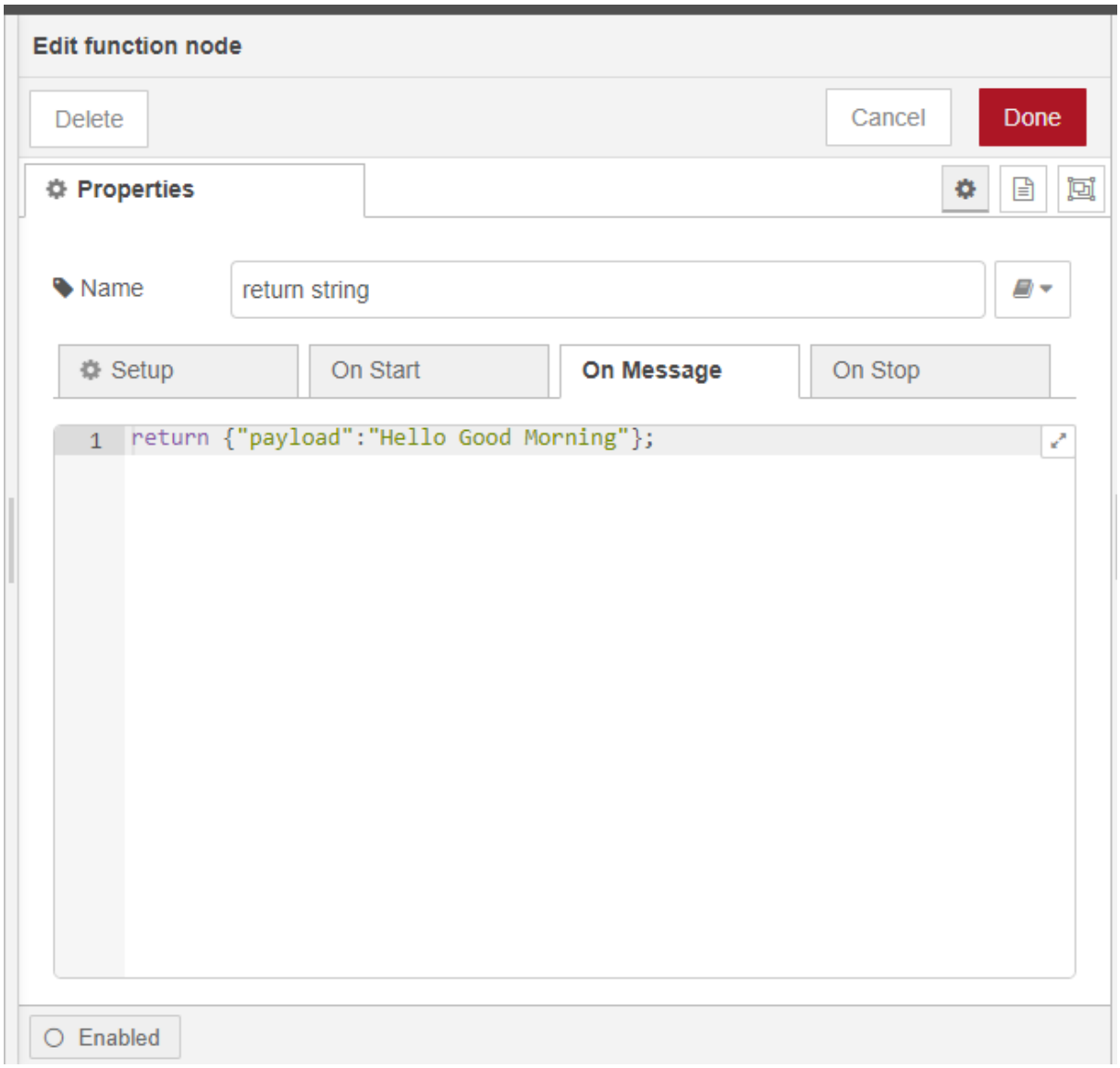


3. Hello IOT flow with the erroneous function node and the corrected function node

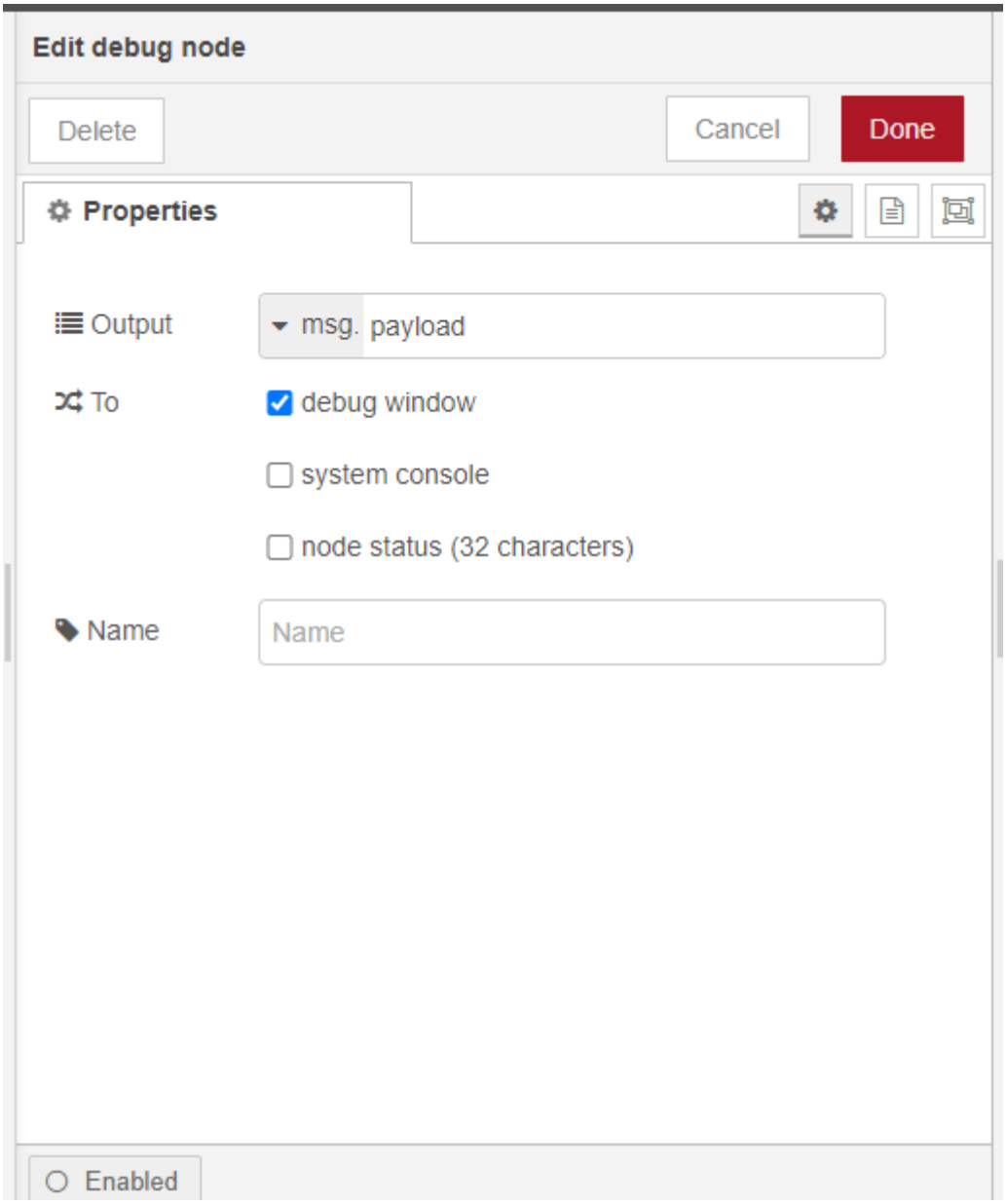
Inject Node:-



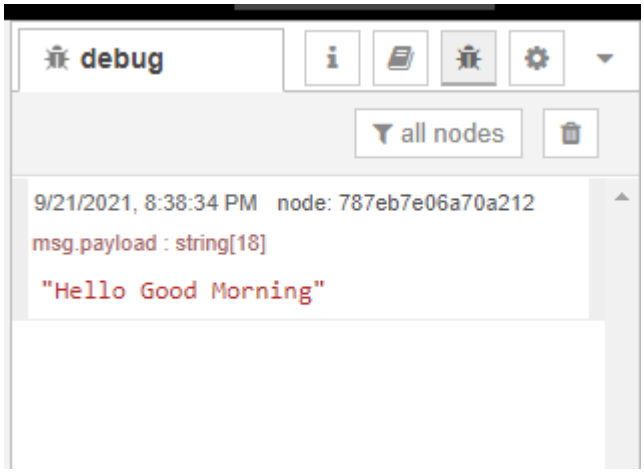
Function Node:-



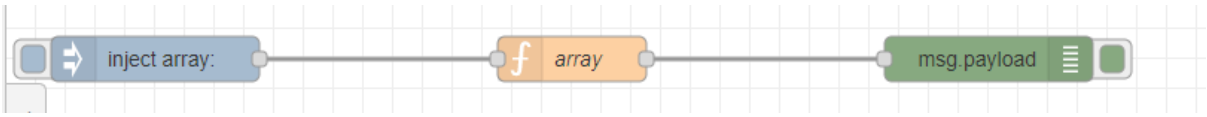
Debug node:-


Edit debug node
Delete
Cancel
Done
Properties
Output
msg.
payload
To
debug window
system console
node status (32 characters)
Name
Name
Enabled

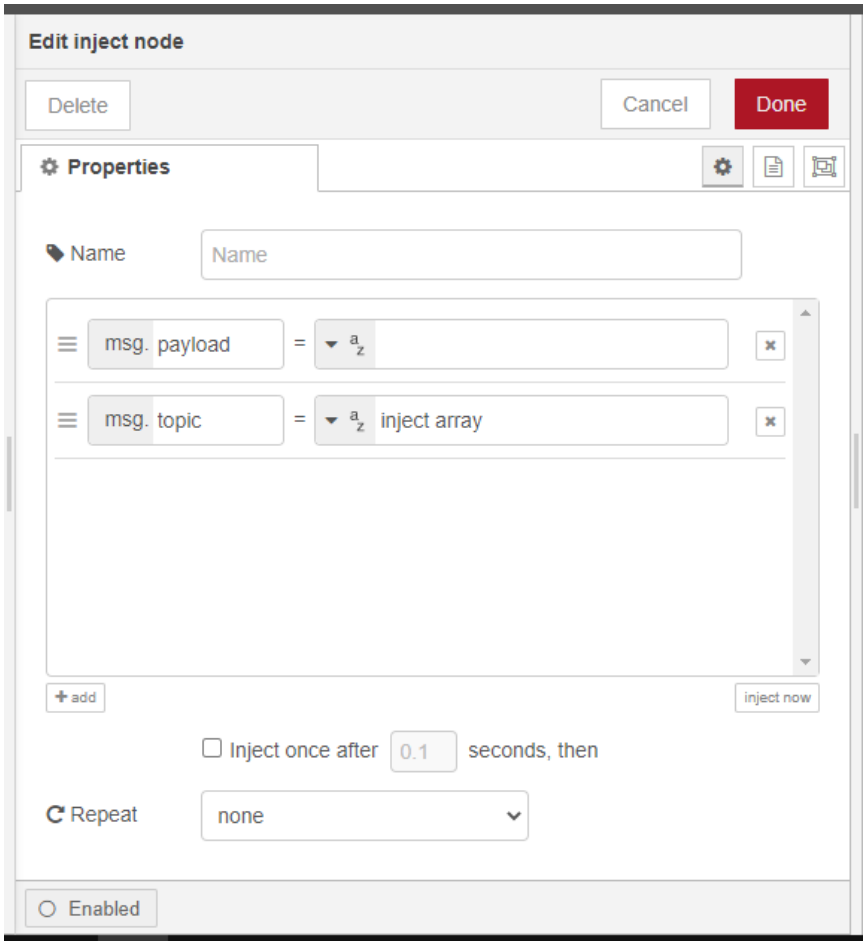
Debug window:-



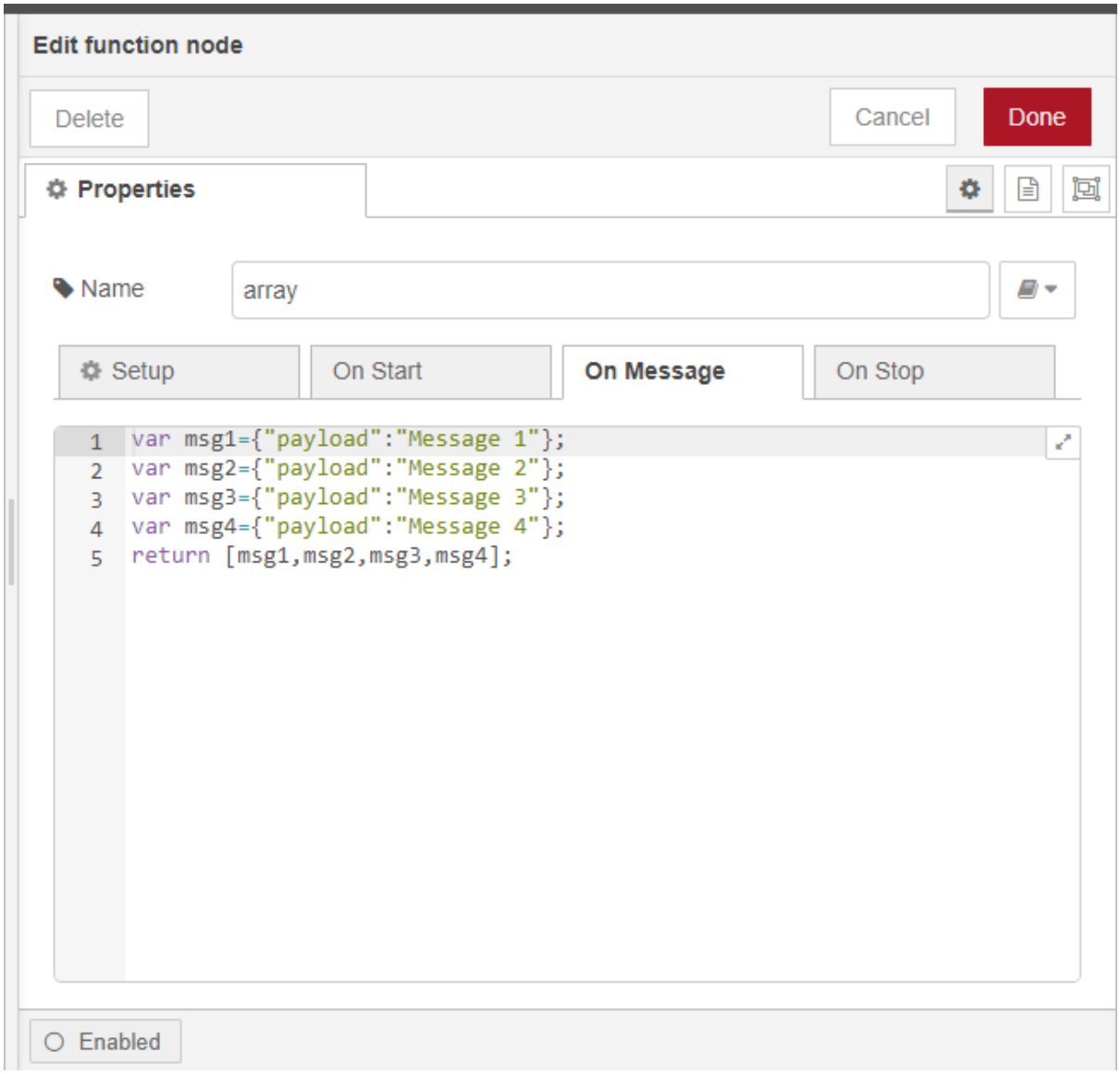
4. function node with array Flow 1



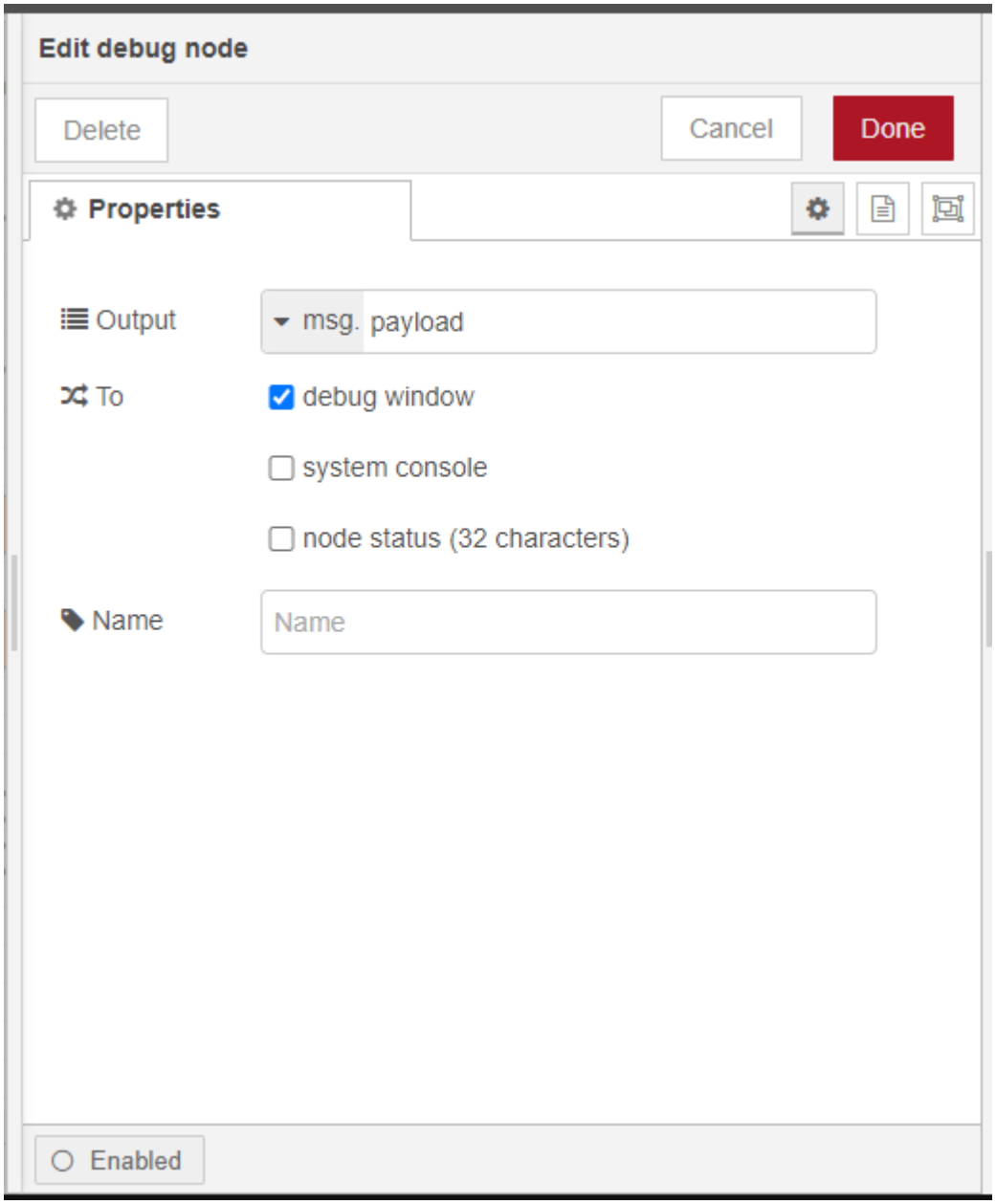
Inject Node:-



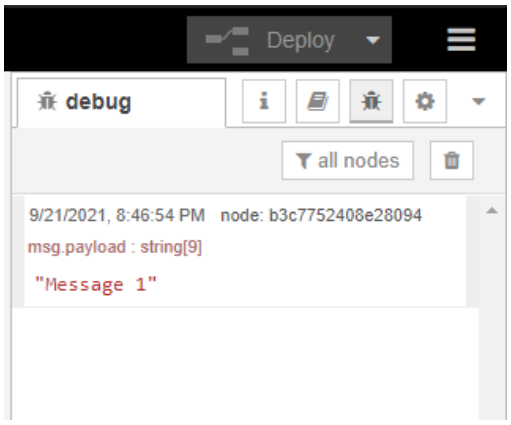
Function Node:-



Debug node:-

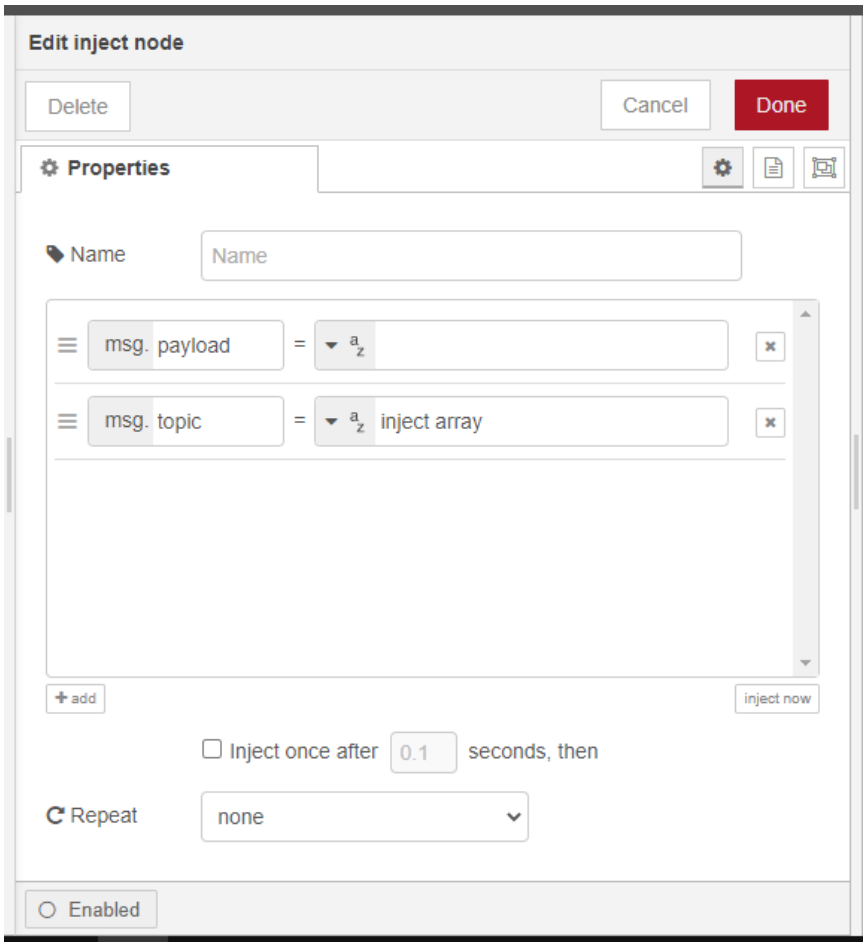


Debug window

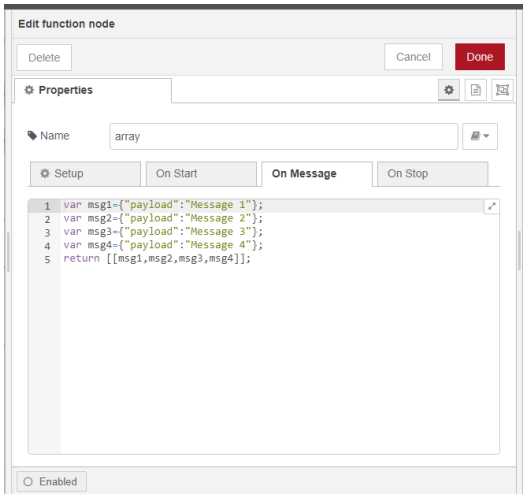


5. function node with array Flow 2

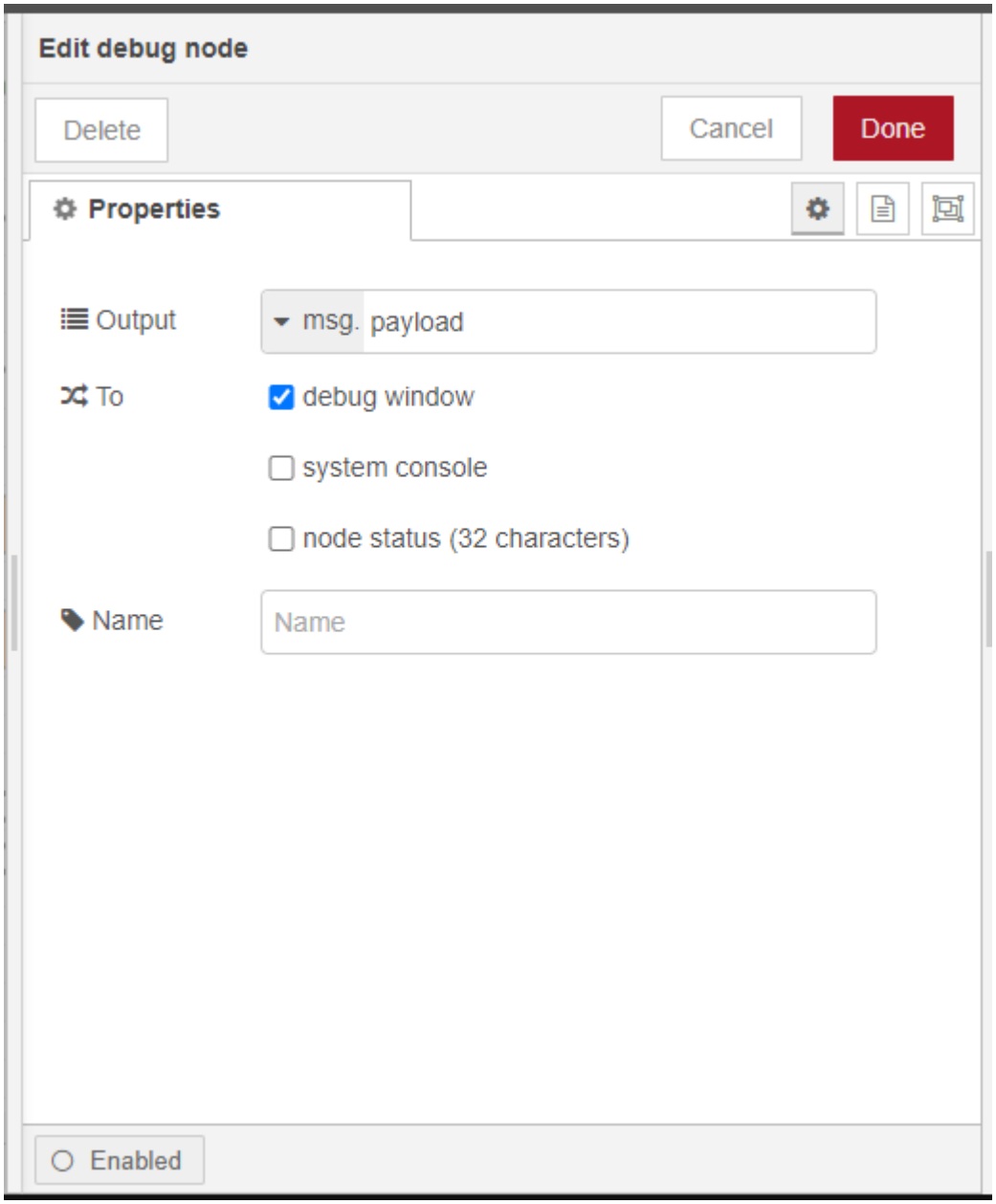
Inject Node:-



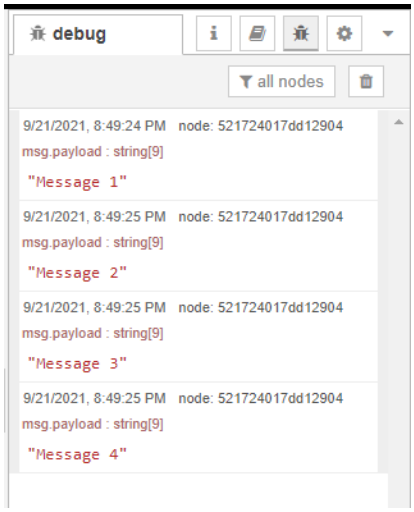
Function Node:-



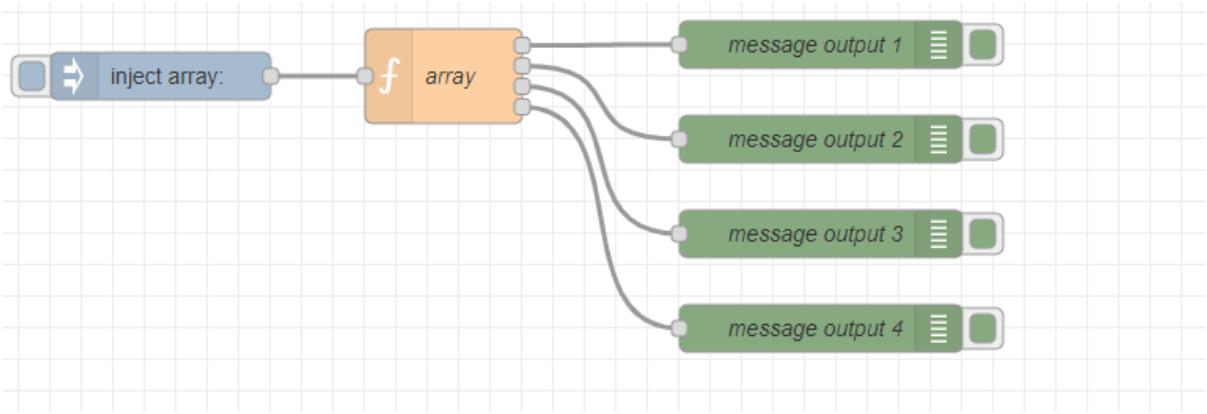
Debug node:-



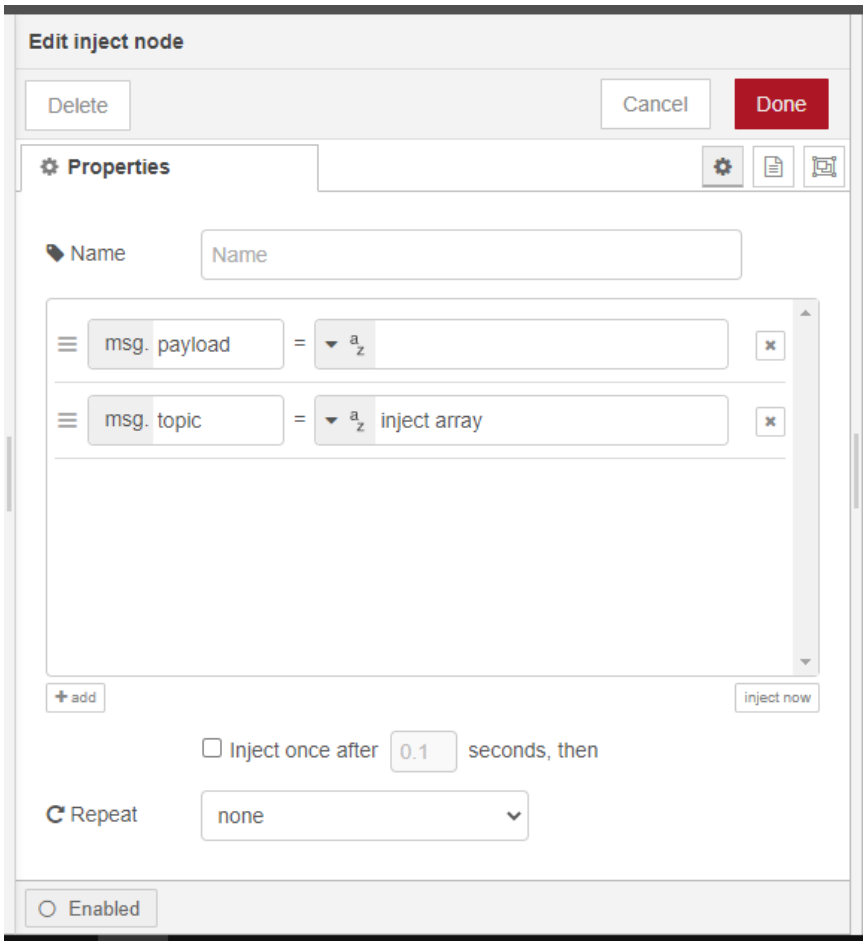
Debug window



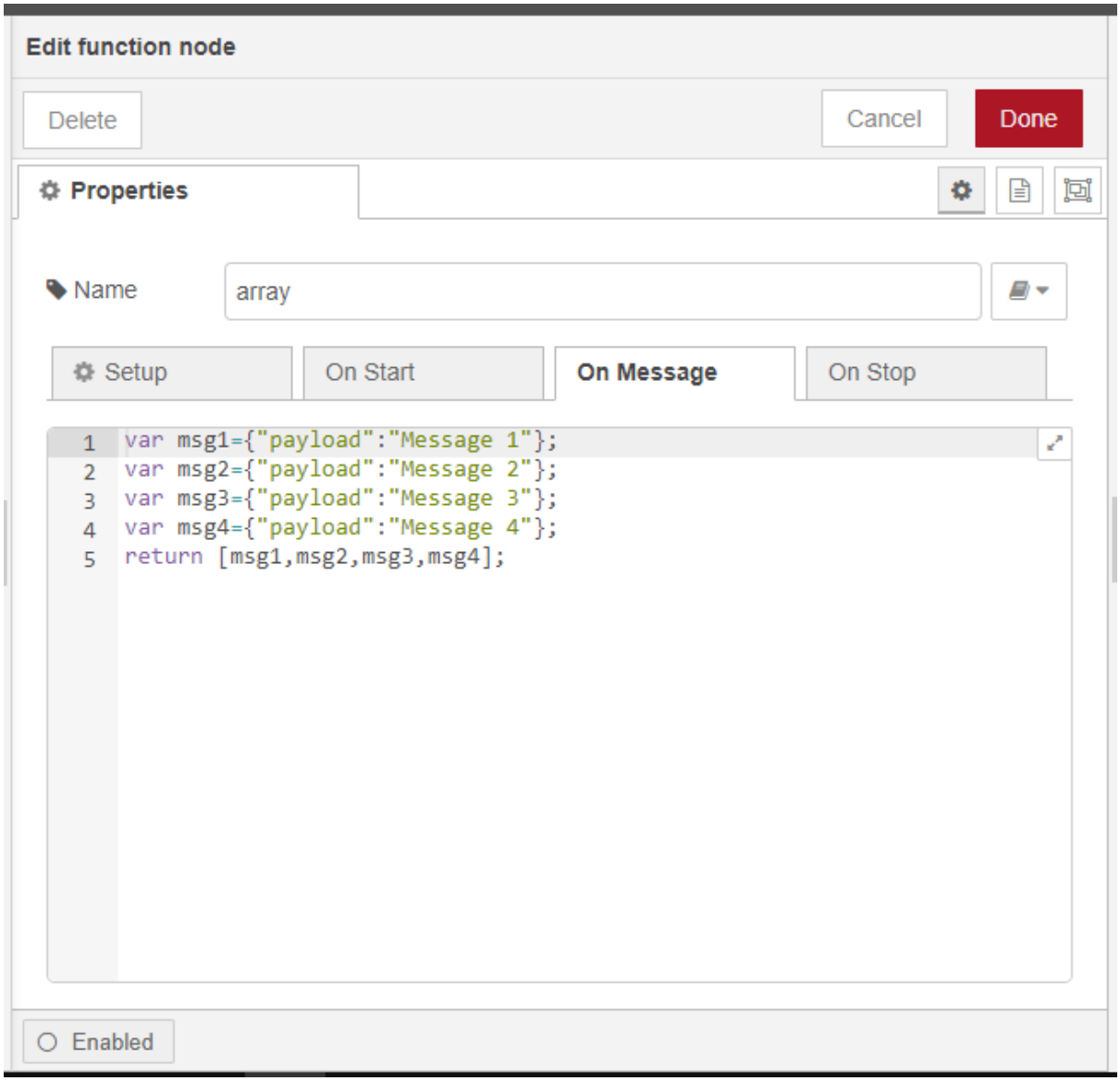
6. function node with array Flow 3



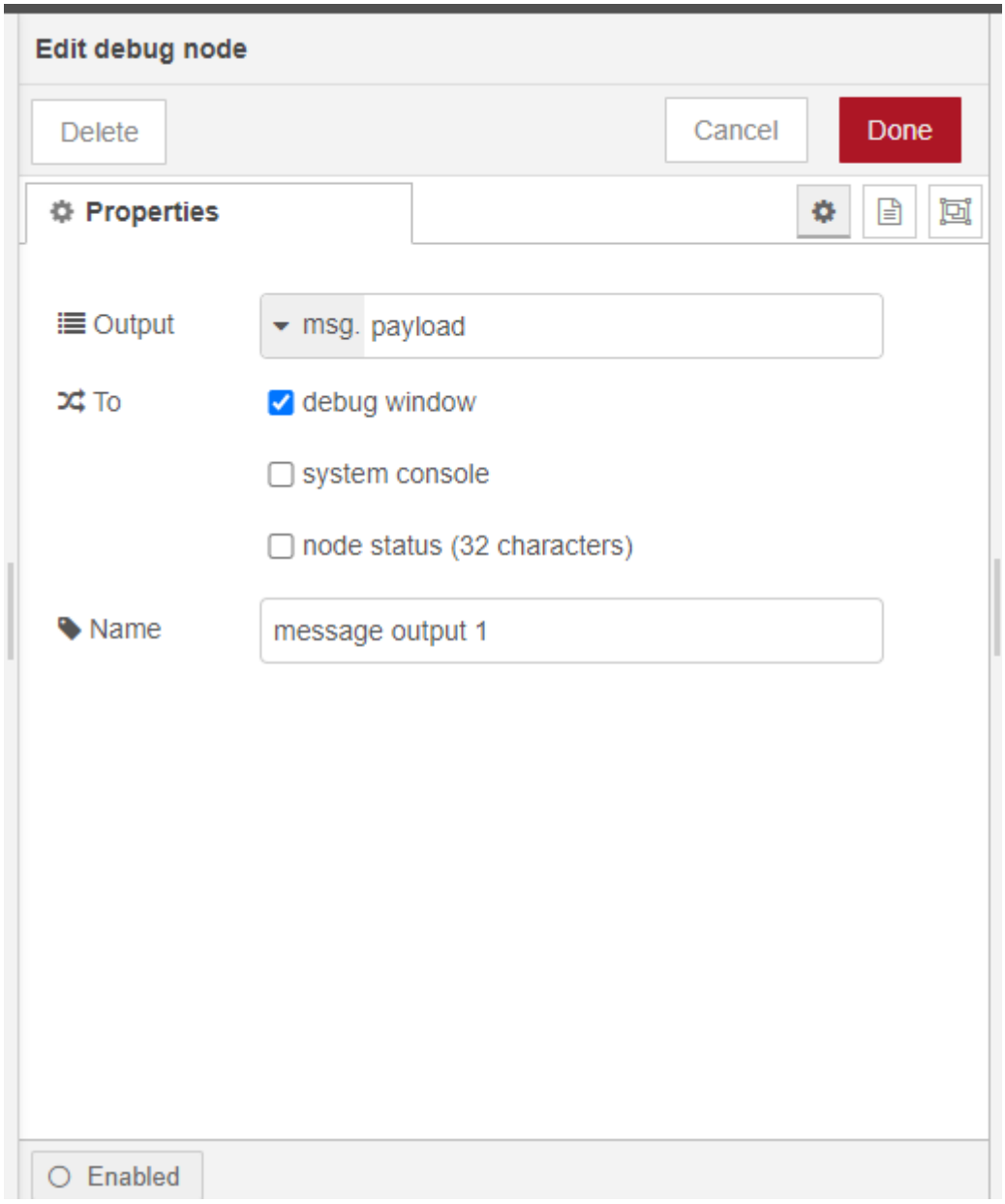
Inject Node:-

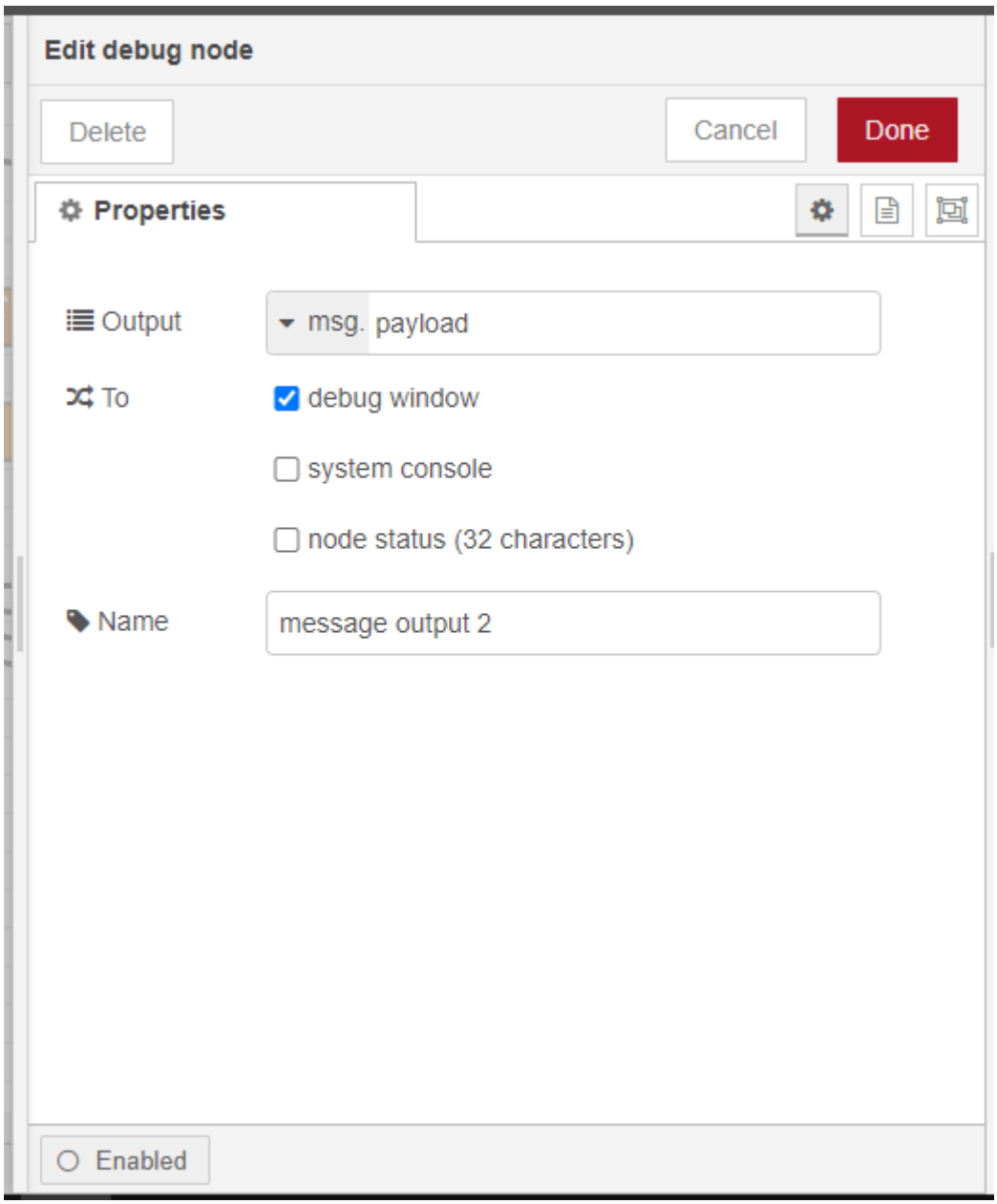


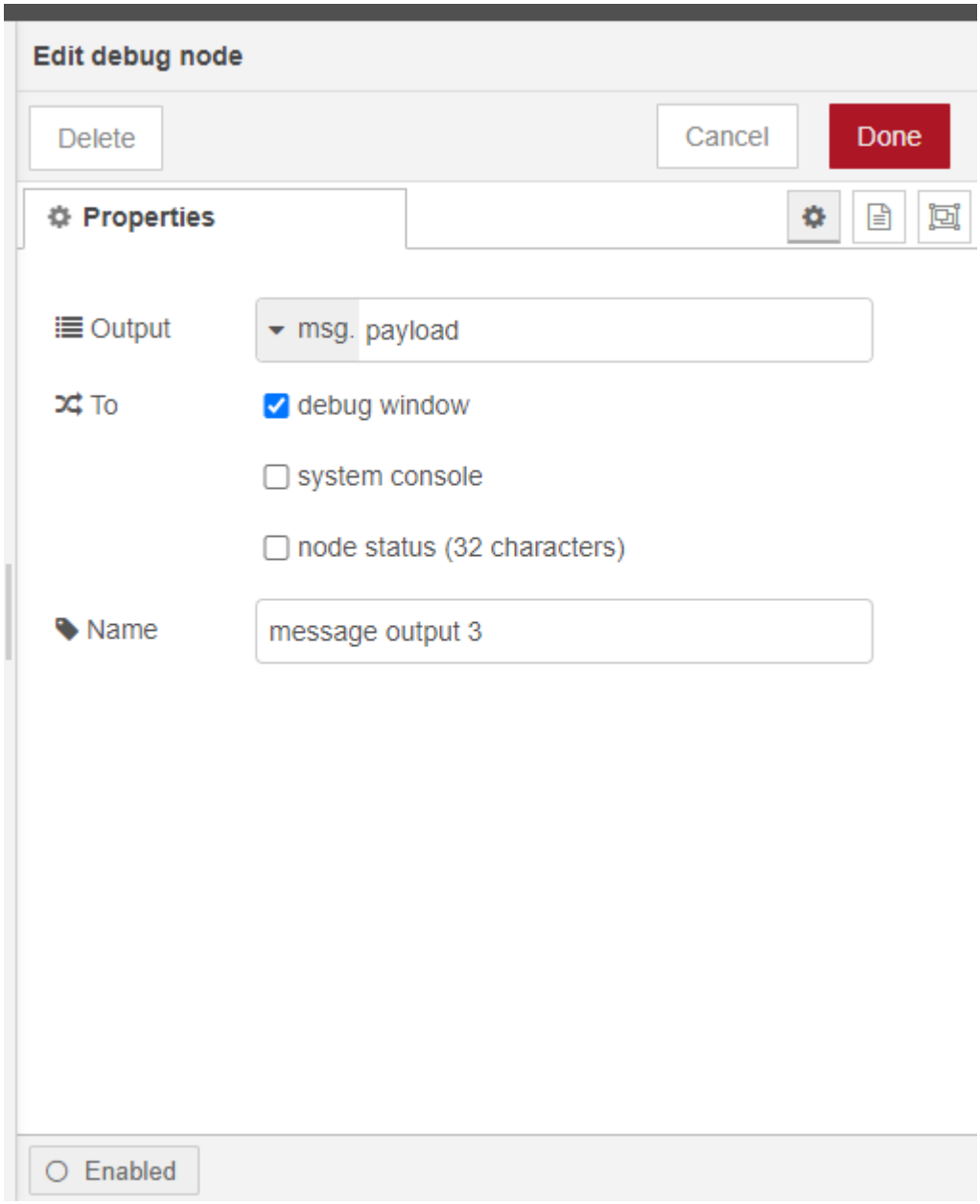
Function Node:-

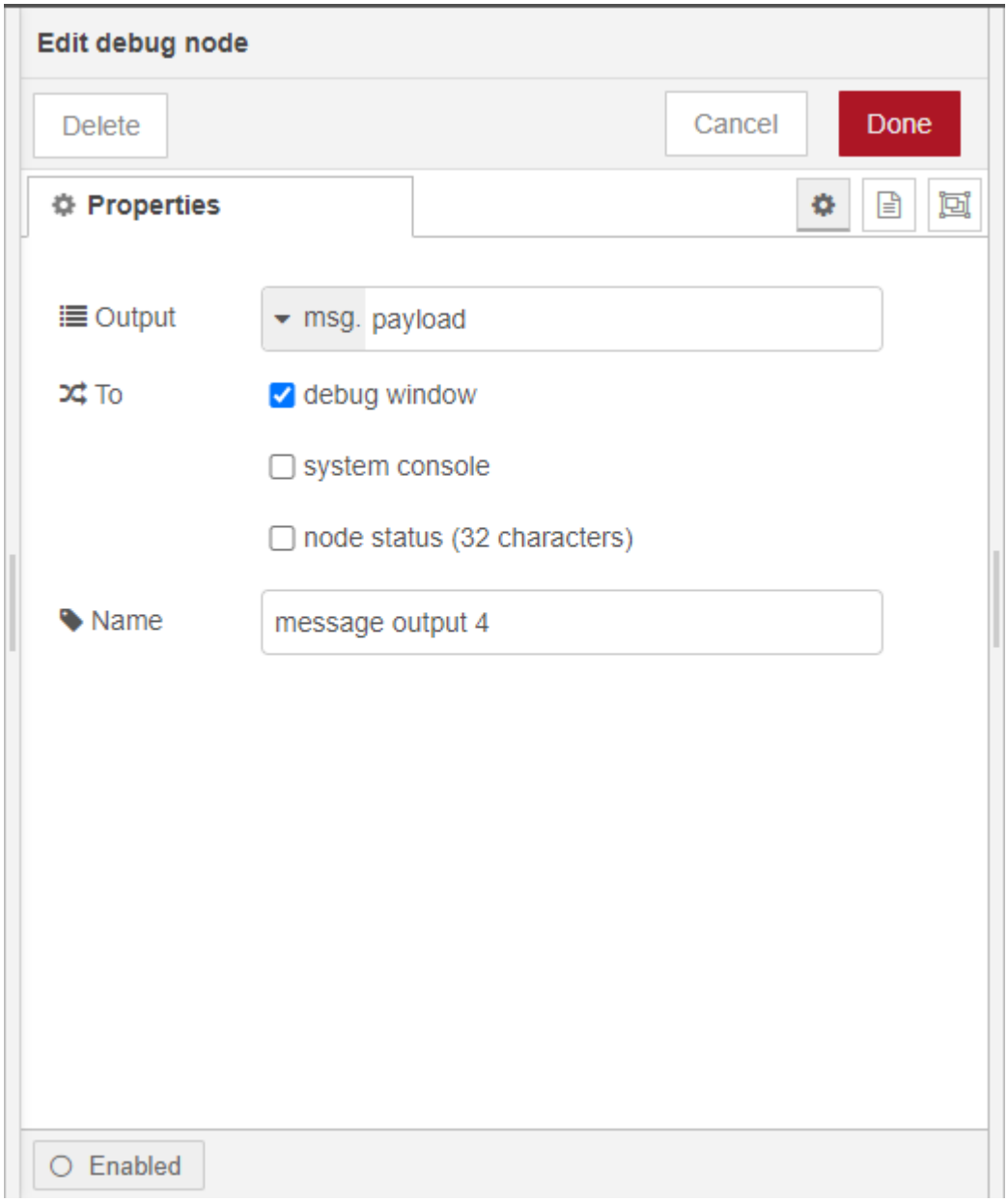


Debug node:-

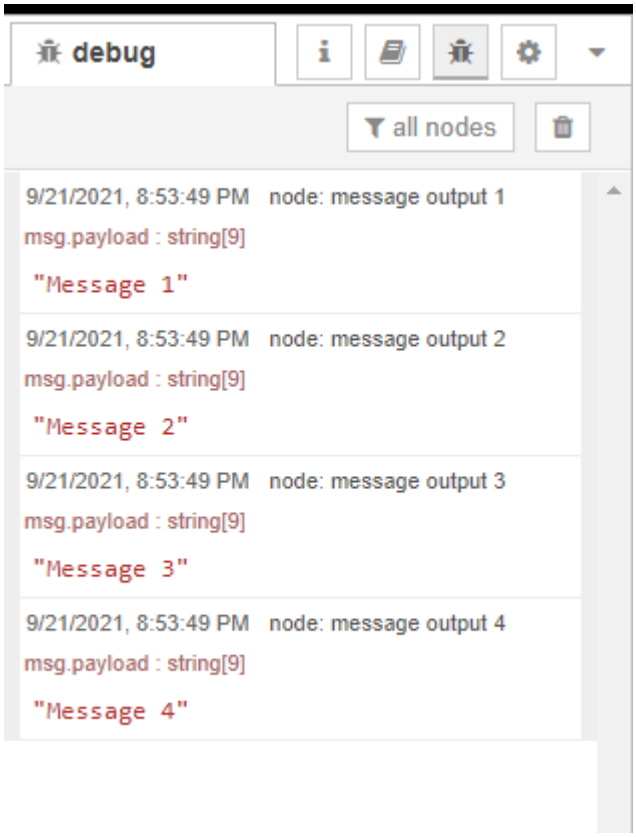

Edit debug node
Delete
Cancel
Done
Properties
Output
msg.
payload
To
debug window
system console
node status (32 characters)
Name
message output 1
Enabled


Edit debug node
Delete
Cancel
Done
Properties
Output
msg.
payload
To
debug window
system console
node status (32 characters)
Name
message output 2
Enabled


Edit debug node
Delete
Cancel
Done
Properties
Output
msg.
payload
To
debug window
system console
node status (32 characters)
Name
message output 3
Enabled

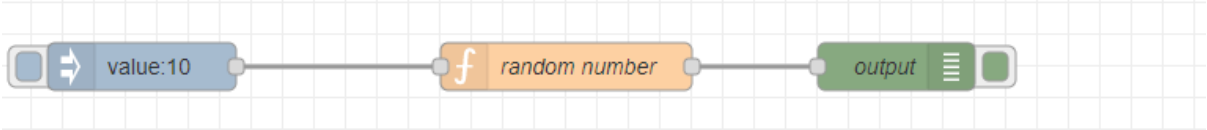

Edit debug node
Delete
Cancel
Done
Properties
Output
msg.
payload
To
debug window
system console
node status (32 characters)
Name
message output 4
Enabled

Debug window

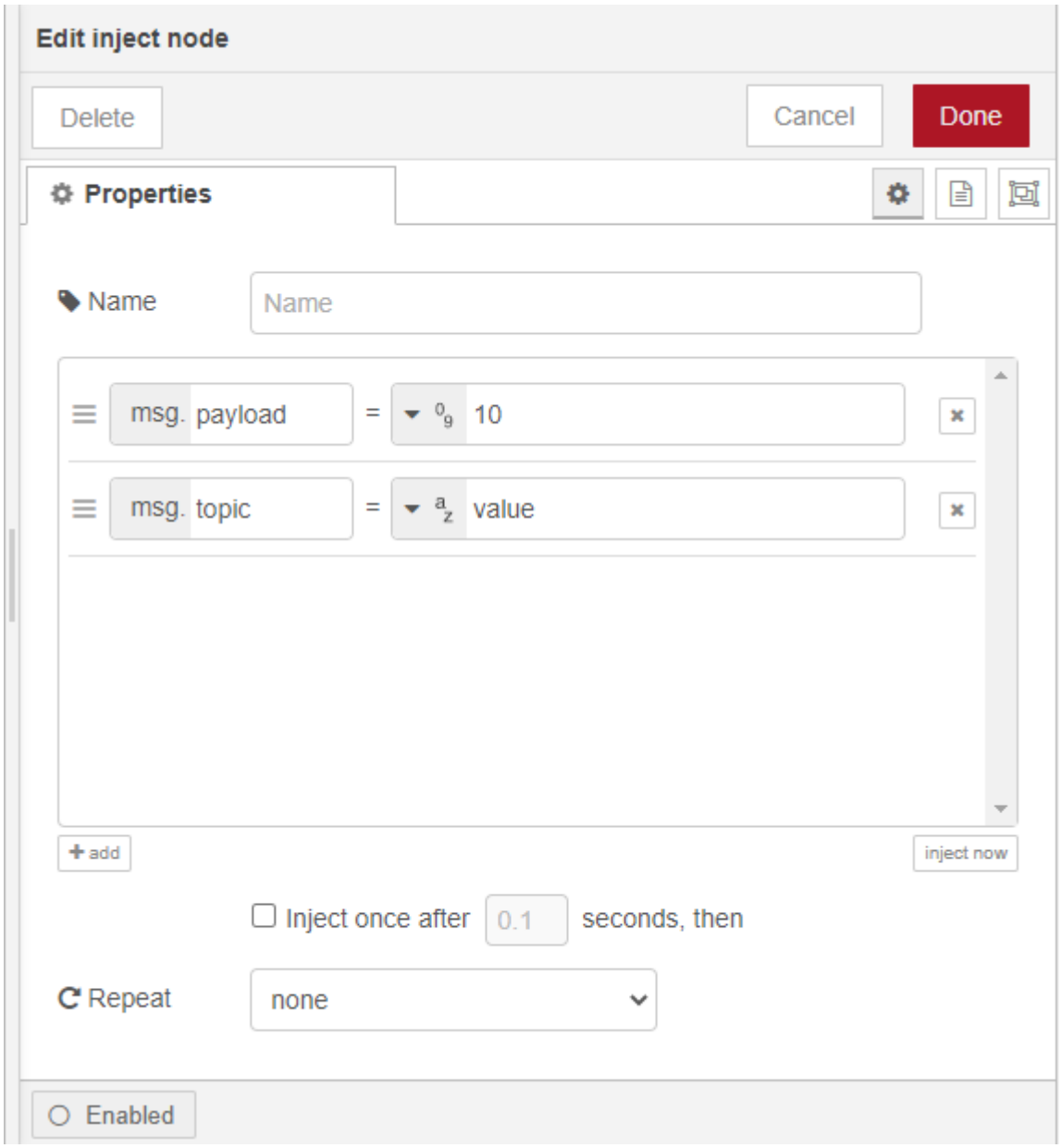


Practical 6 : Random number generator with selection of color. Introduce the HTTP node.

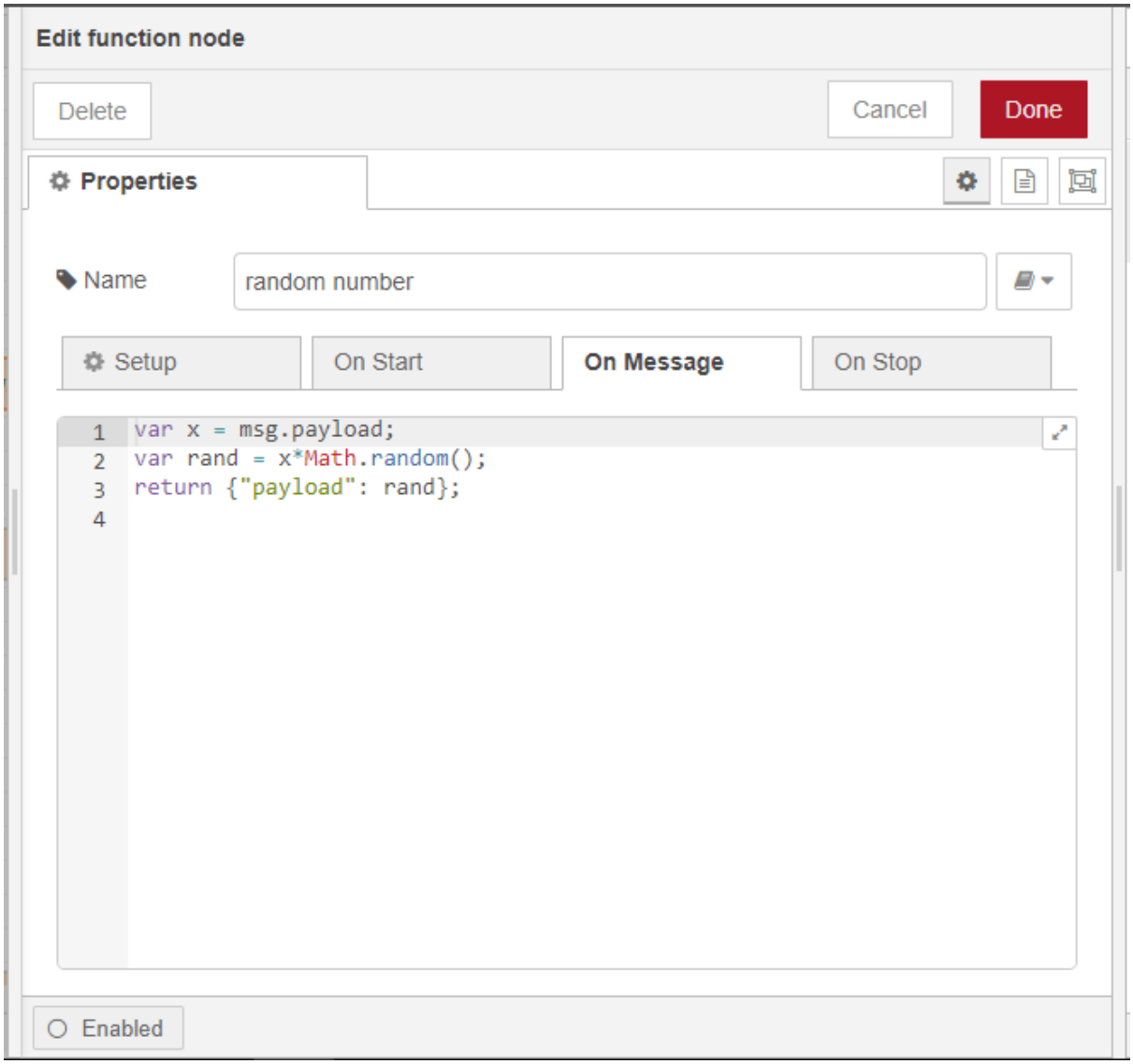
Flow 1:-



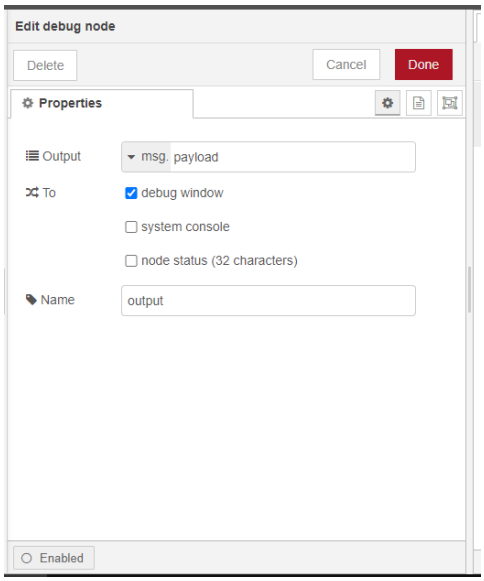
Inject node:-



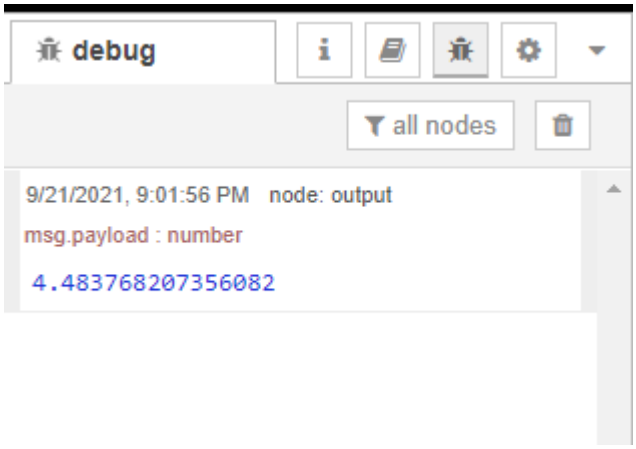
Function node:-



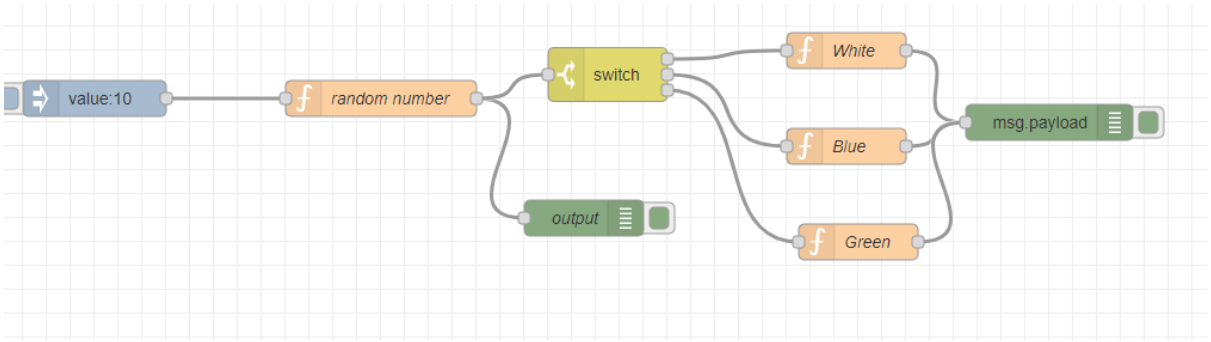
Debug node:-



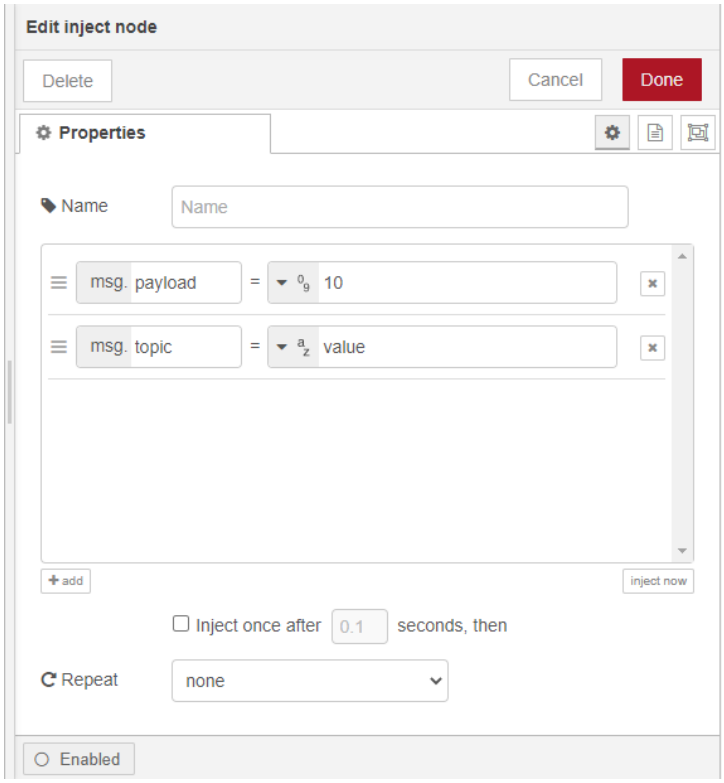
Debug window:-



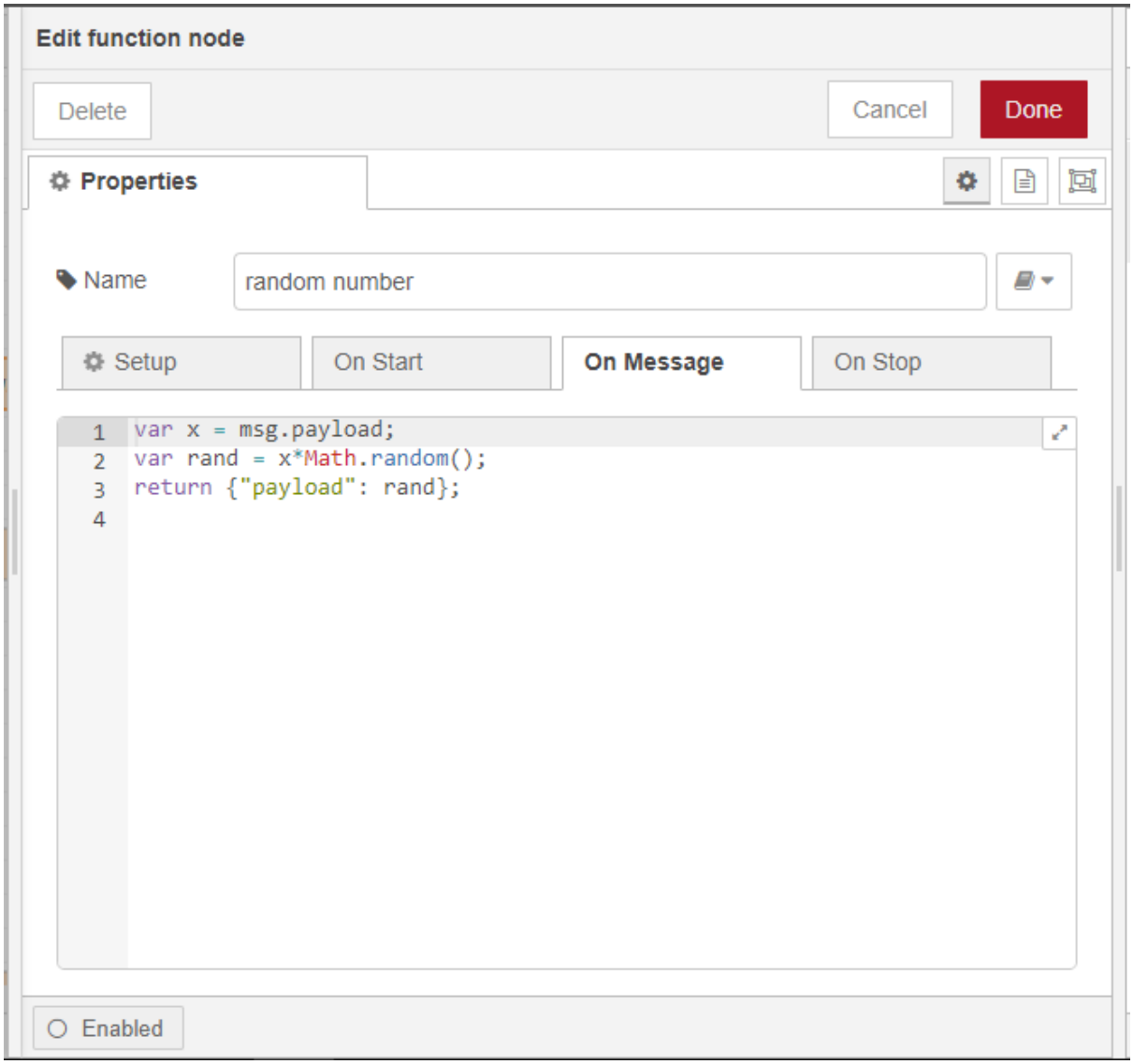
Flow 2:-



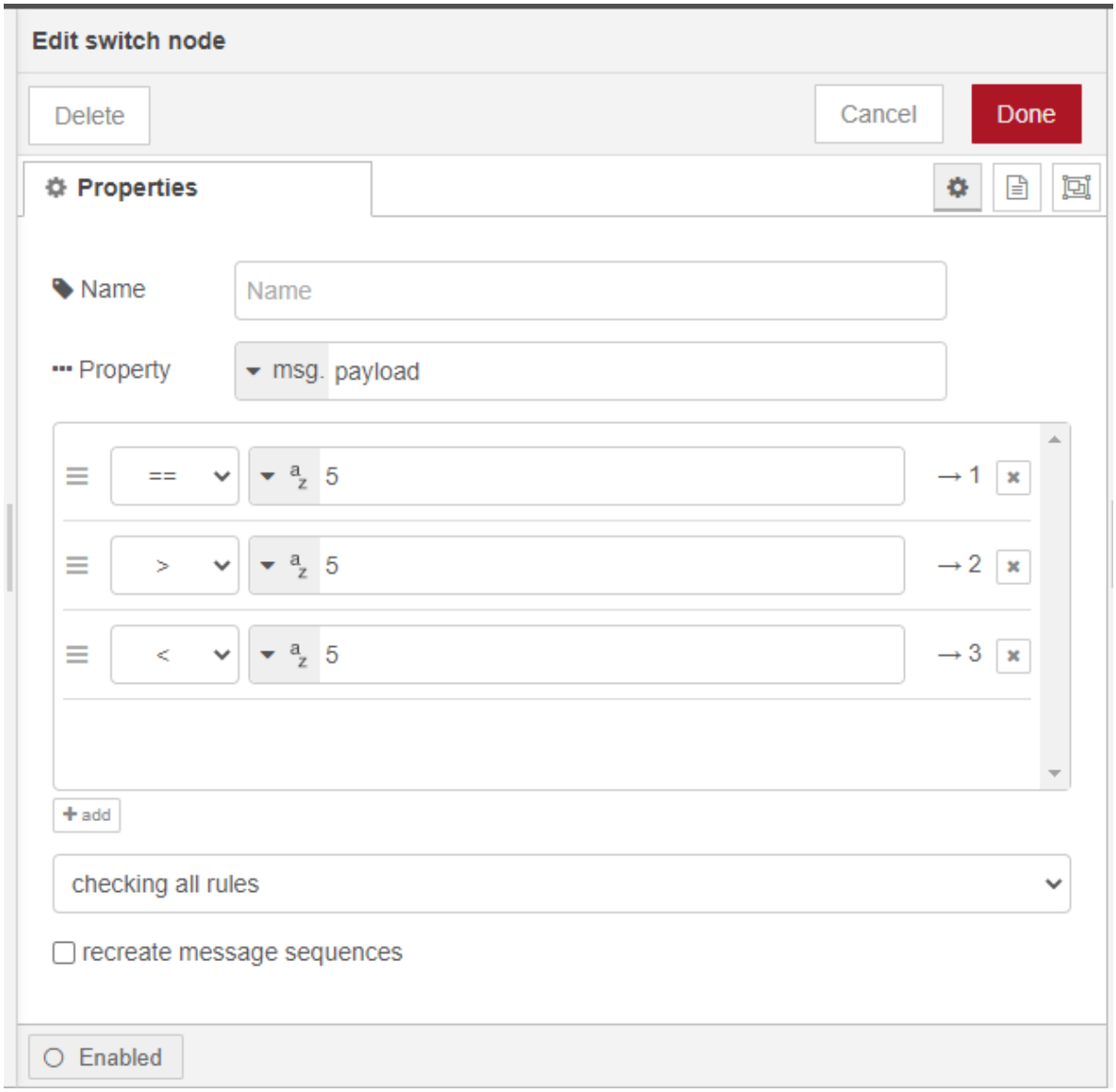
Inject node:-



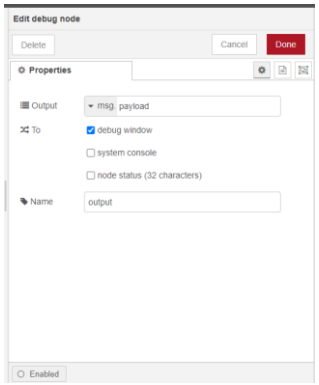
Function node:-



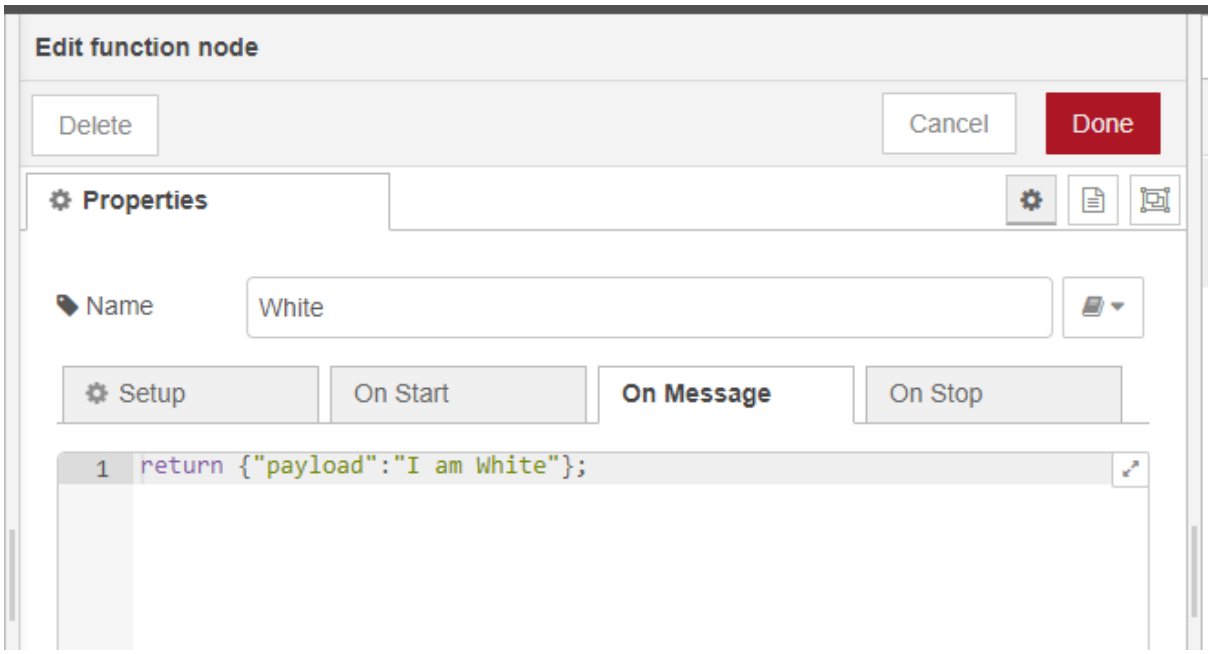
Switch node:-

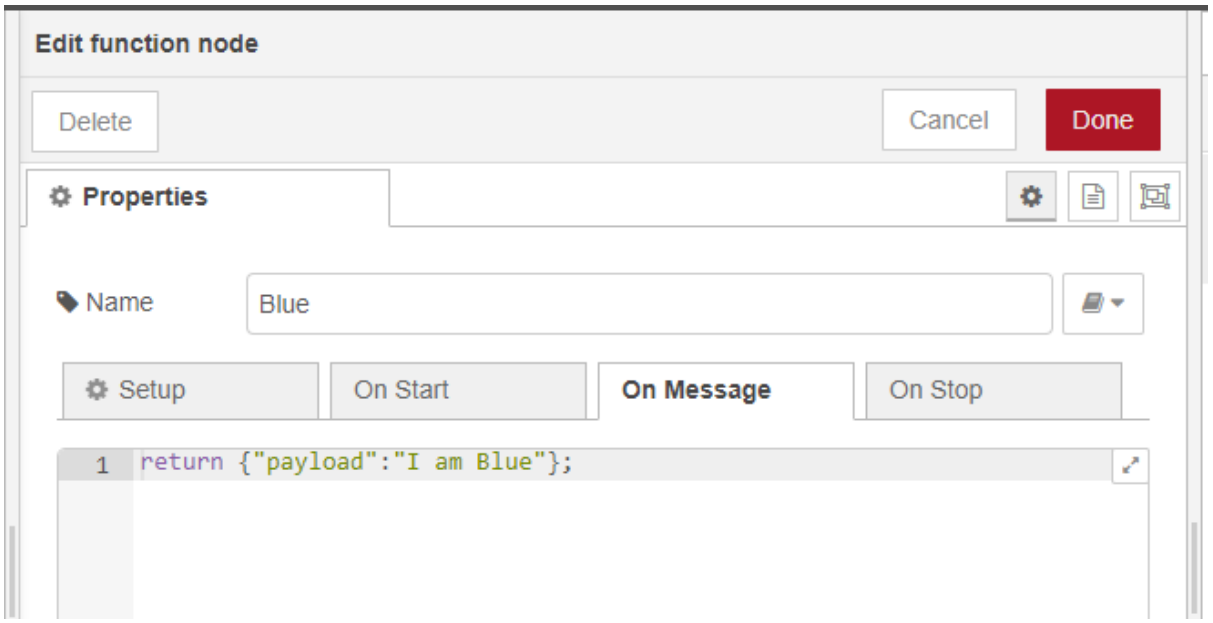


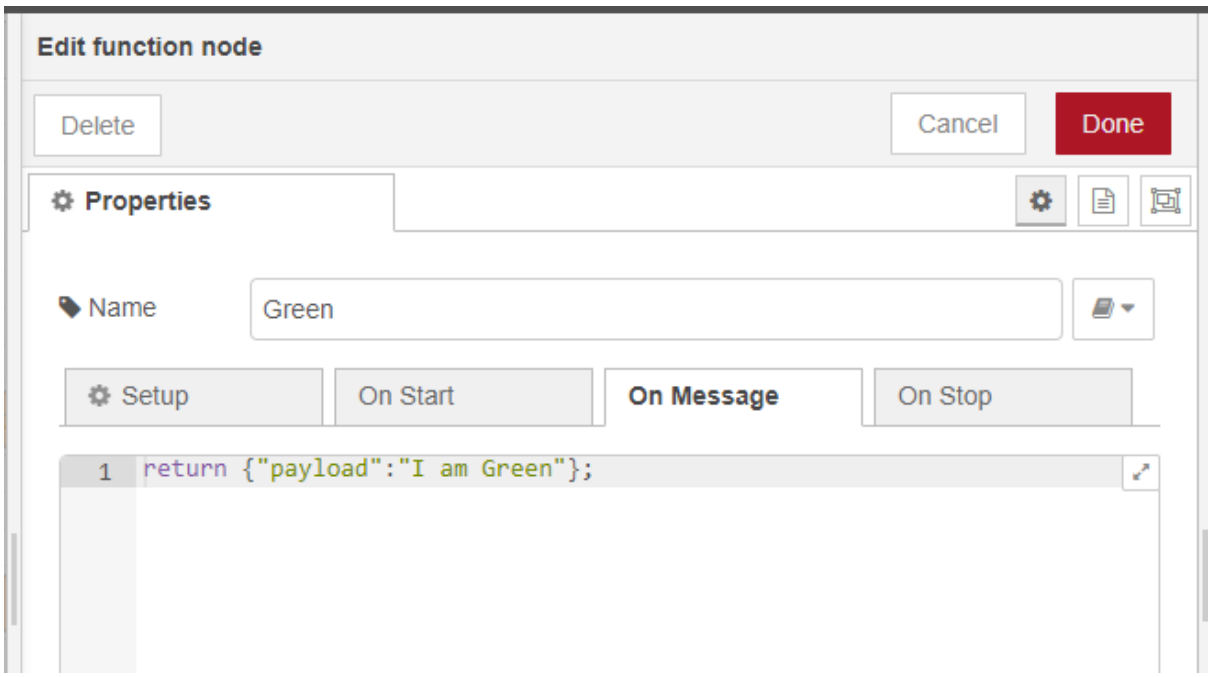
Debug node:-



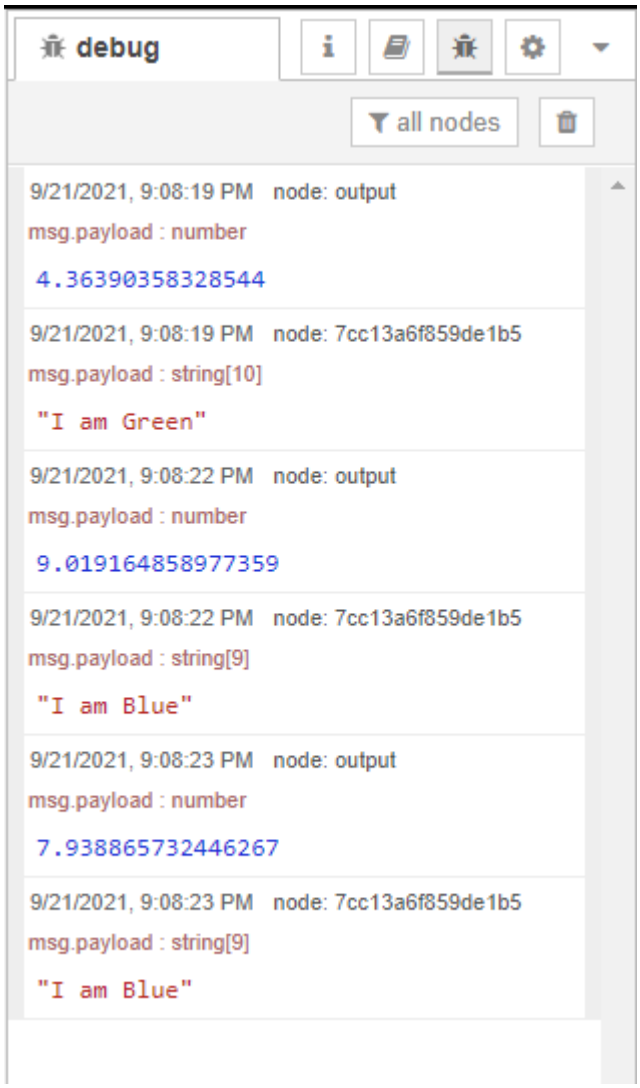
Function node:-



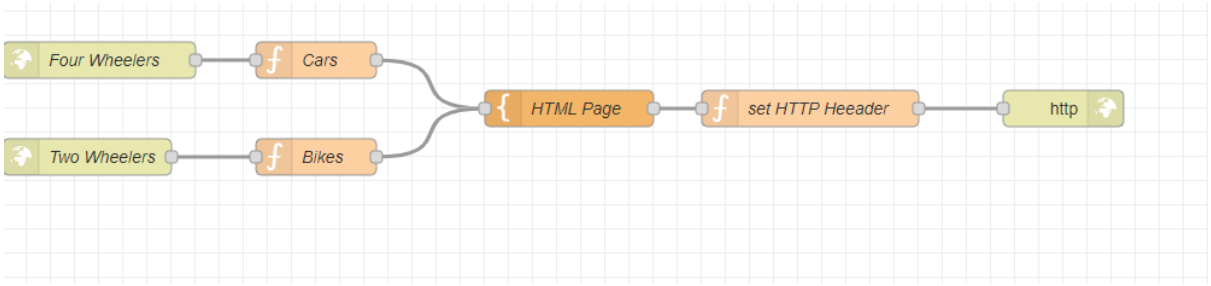




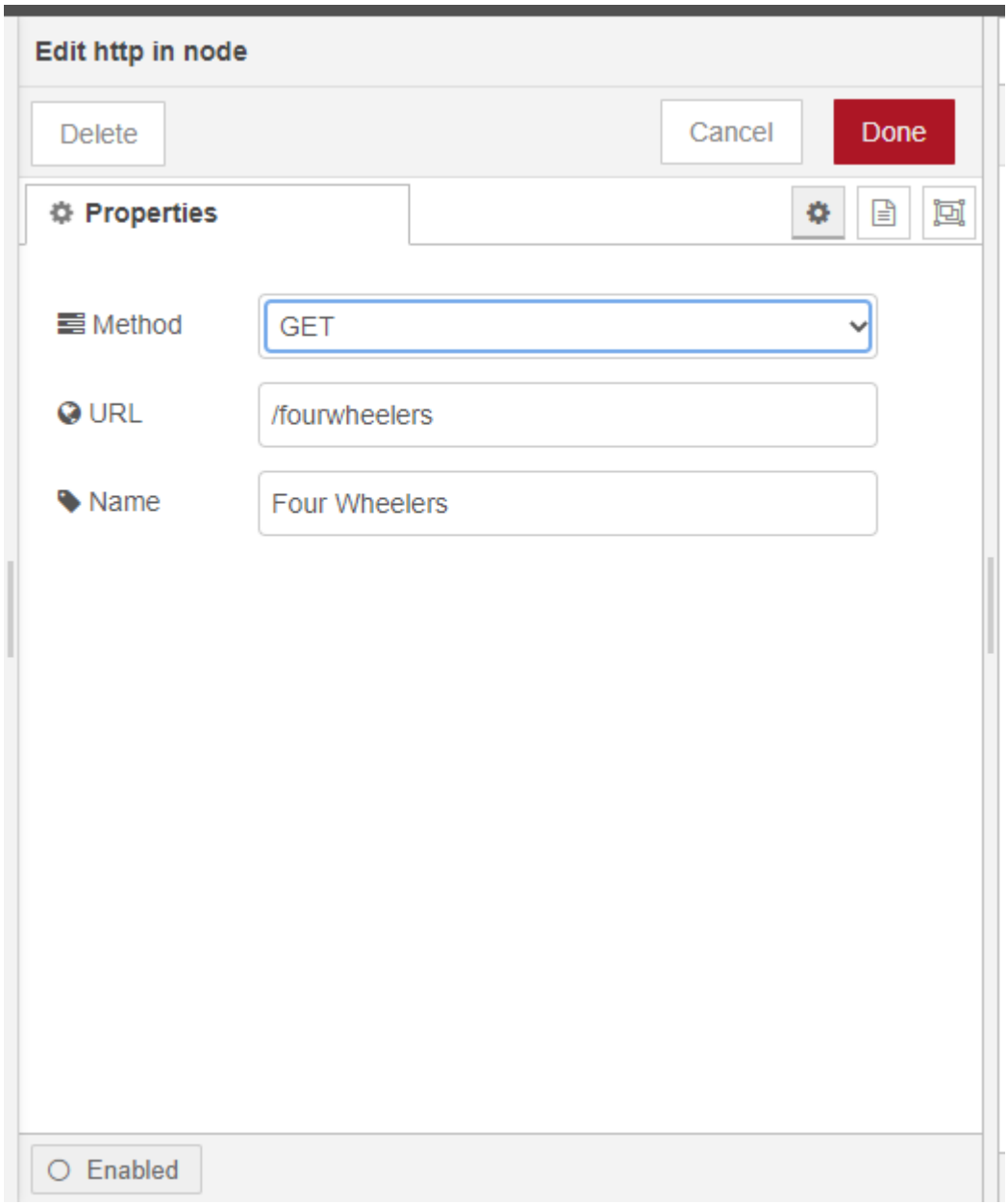
Debug window:-

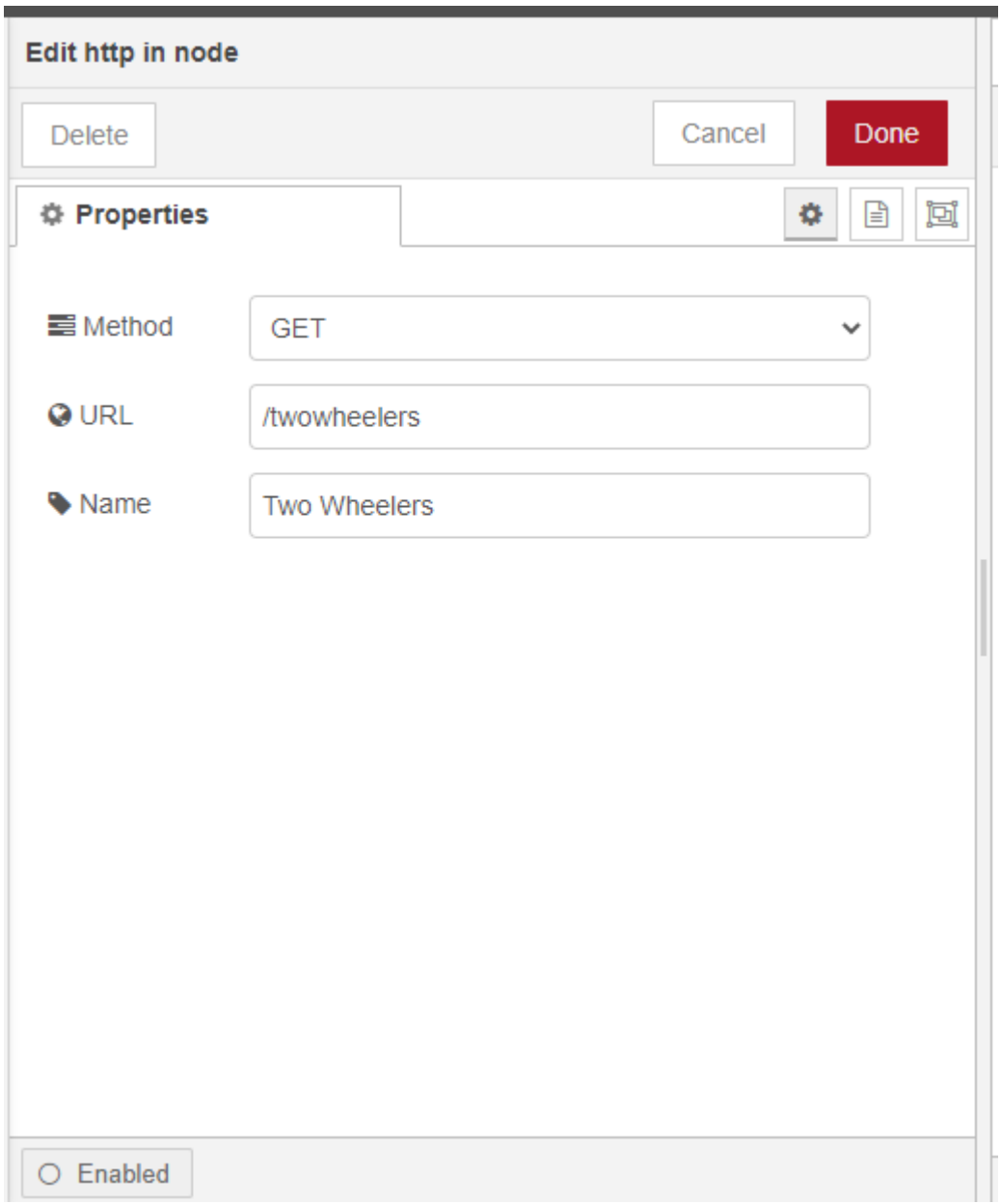


two wheelers and four wheelers flow

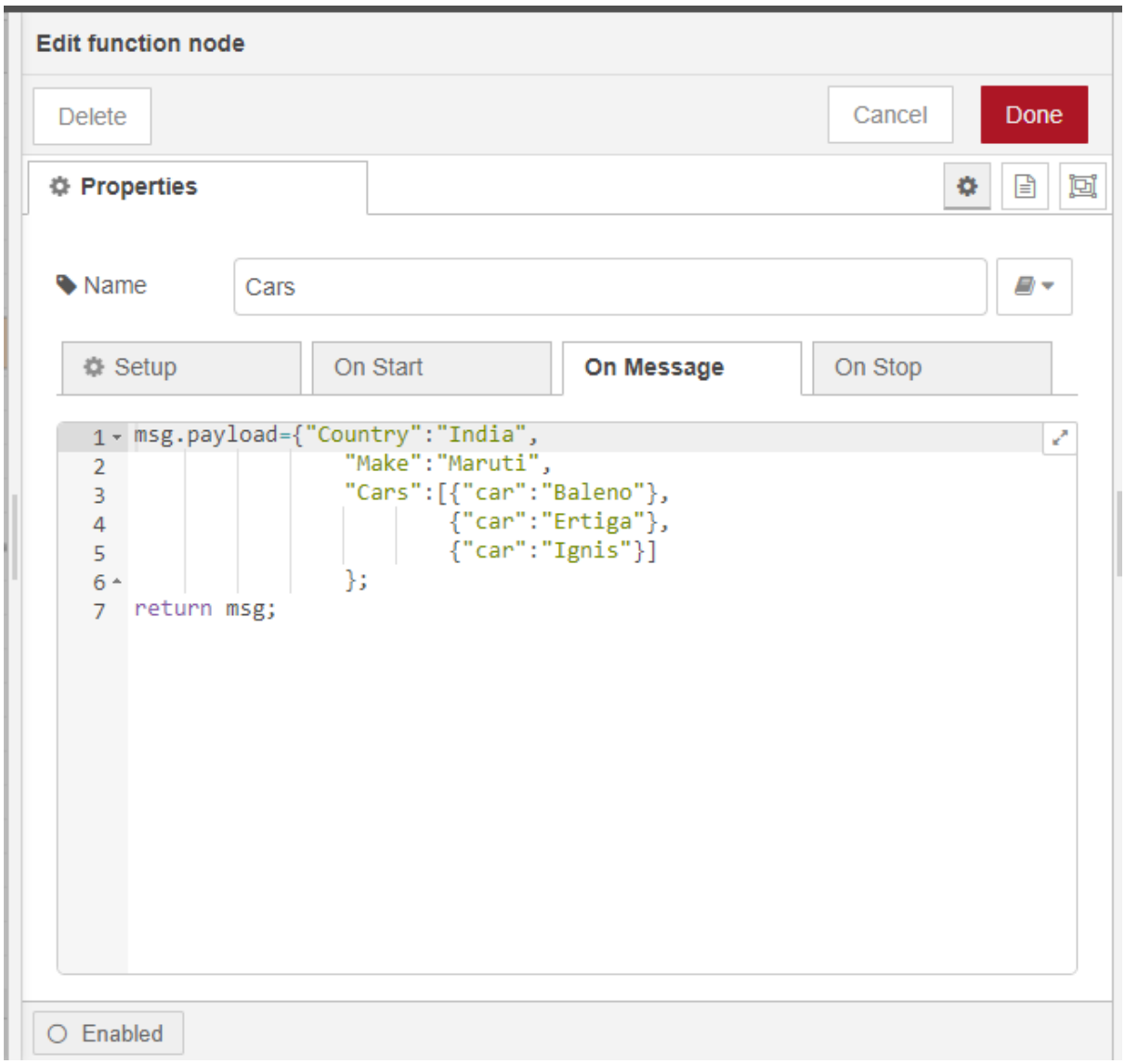


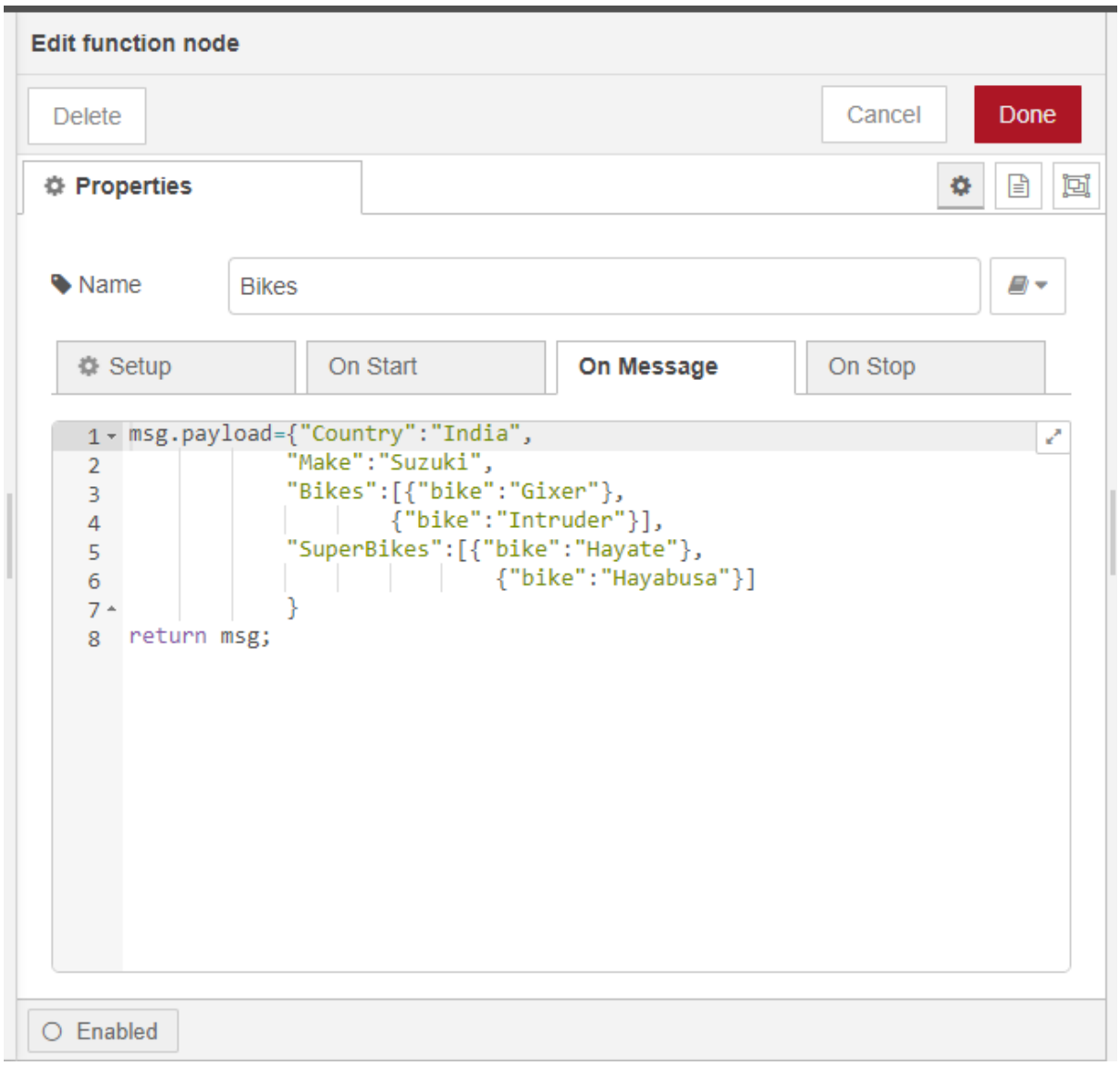
http in node:-

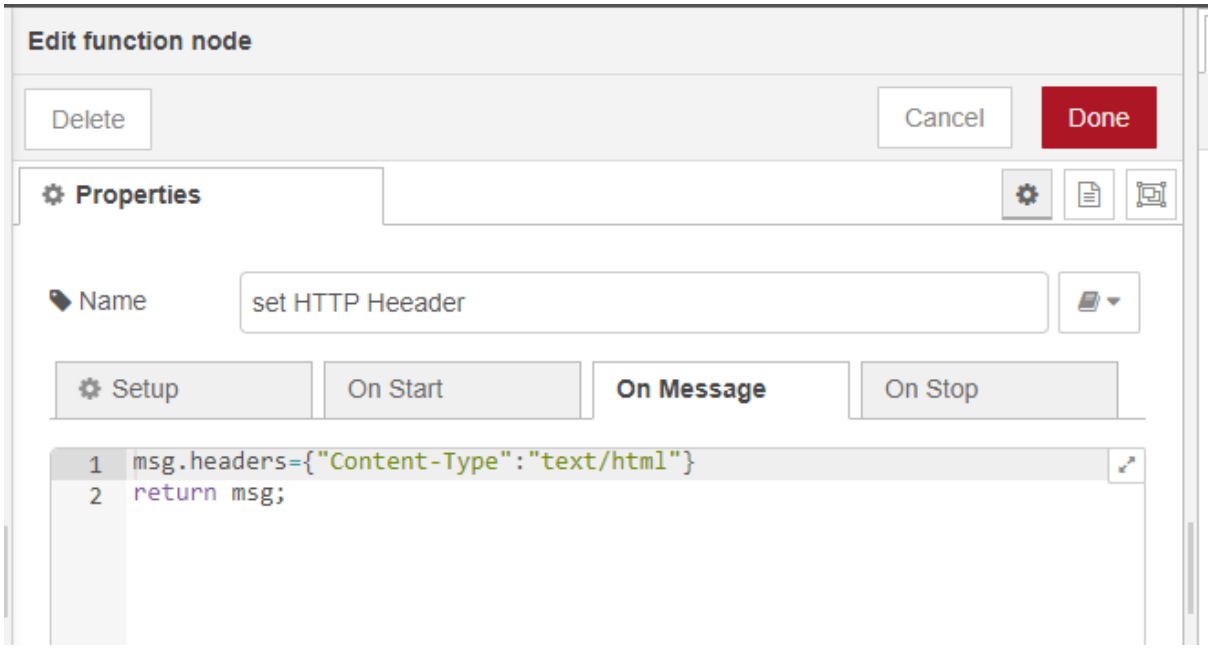




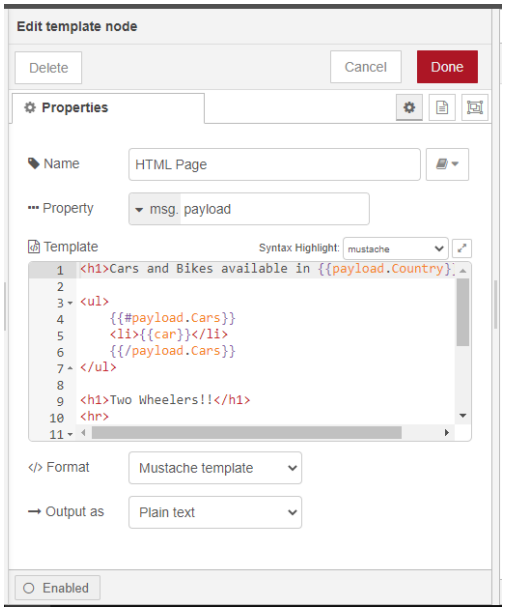
Function node:-


Edit function node
Delete
Cancel
Done
Properties
Name
Cars
Setup
On Start
On Message
On Stop
msg.payload={"Country":"India",
"Make":"Maruti",
"Cars":[{"car":"Baleno"},
{"car":"Ertiga"},
{"car":"Ignis"}]
};
return msg;
Enabled

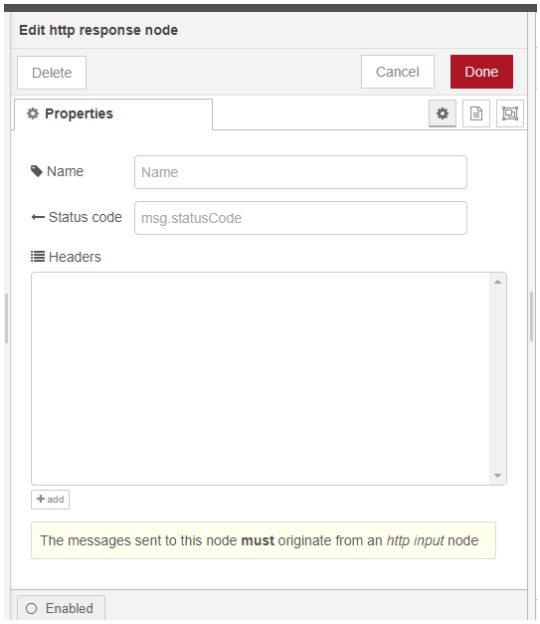

Edit function node
Delete
Cancel
Done
Properties
Name
Bikes
Setup
On Start
On Message
On Stop
msg.payload={"Country":"India",
"Make":"Suzuki",
"Bikes":[{"bike":"Gixer"},
{"bike":"Intruder"}],
"SuperBikes":[{"bike":"Hayate"},
{"bike":"Hayabusa"}]
}
return msg;
Enabled


Edit function node
Delete
Cancel
Done
Properties
Name
set HTTP Heeader
Setup
On Start
On Message
On Stop
msg.headers={"Content-Type":"text/html"}
return msg;

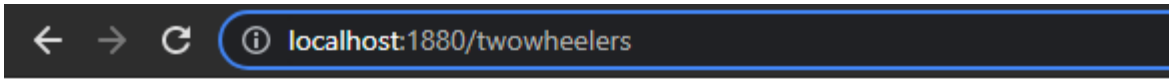
Template node:-



http node:-



Output :-



Cars and Bikes available in India

Two Wheelers!!

- Gixer
- Intruder

Cars and Bikes available in India

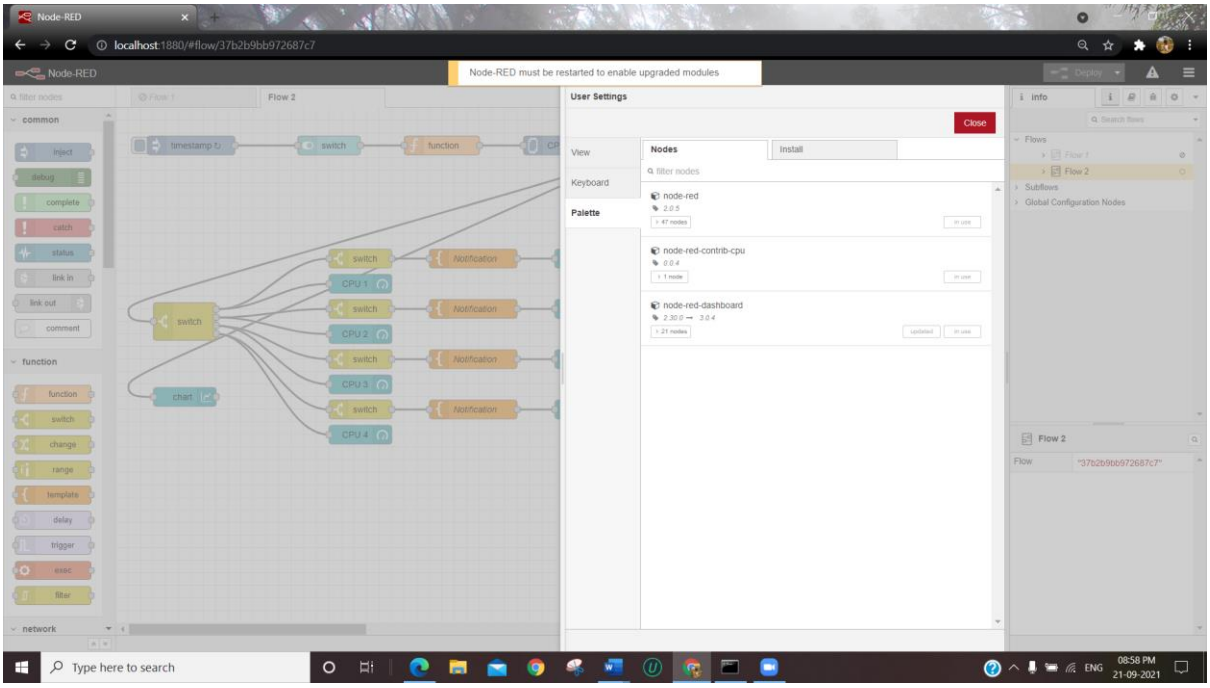
- Baleno
- Ertiga
- Ignis

Two Wheelers!!

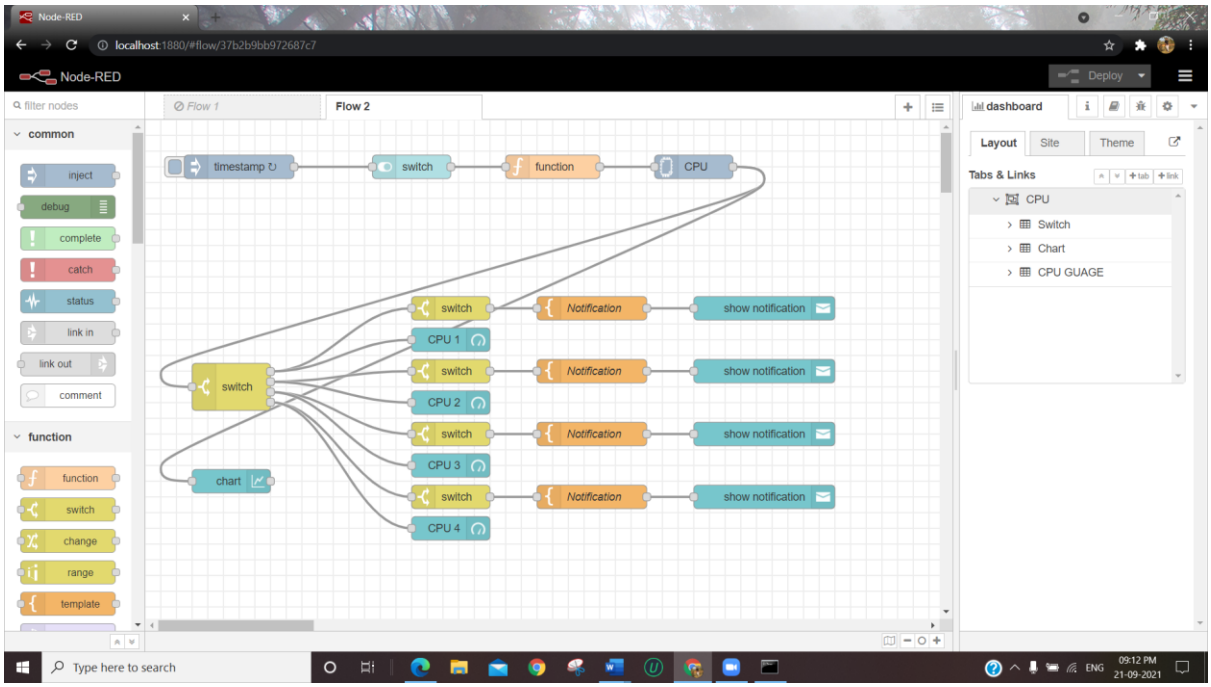
No bikes here!!

Practical 7:Practical Name: CPU utilization flow

1.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 screen shots)

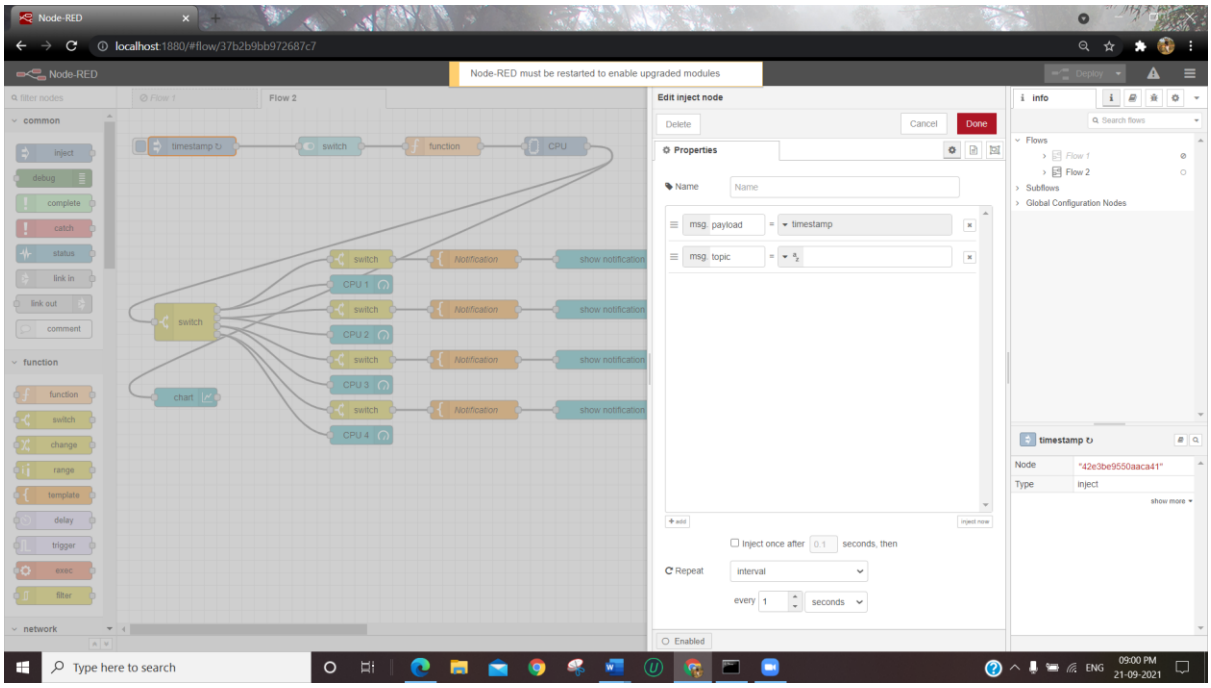


CPU Utilization flow

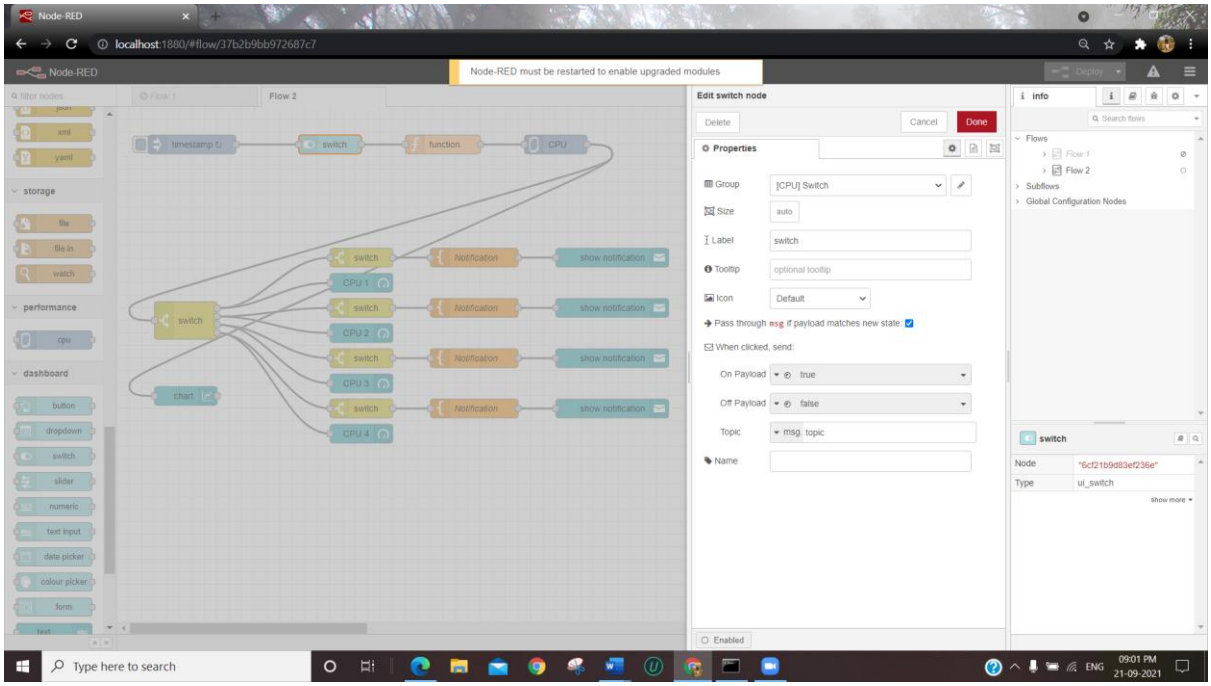


3) Explain all the nodes used in detail.

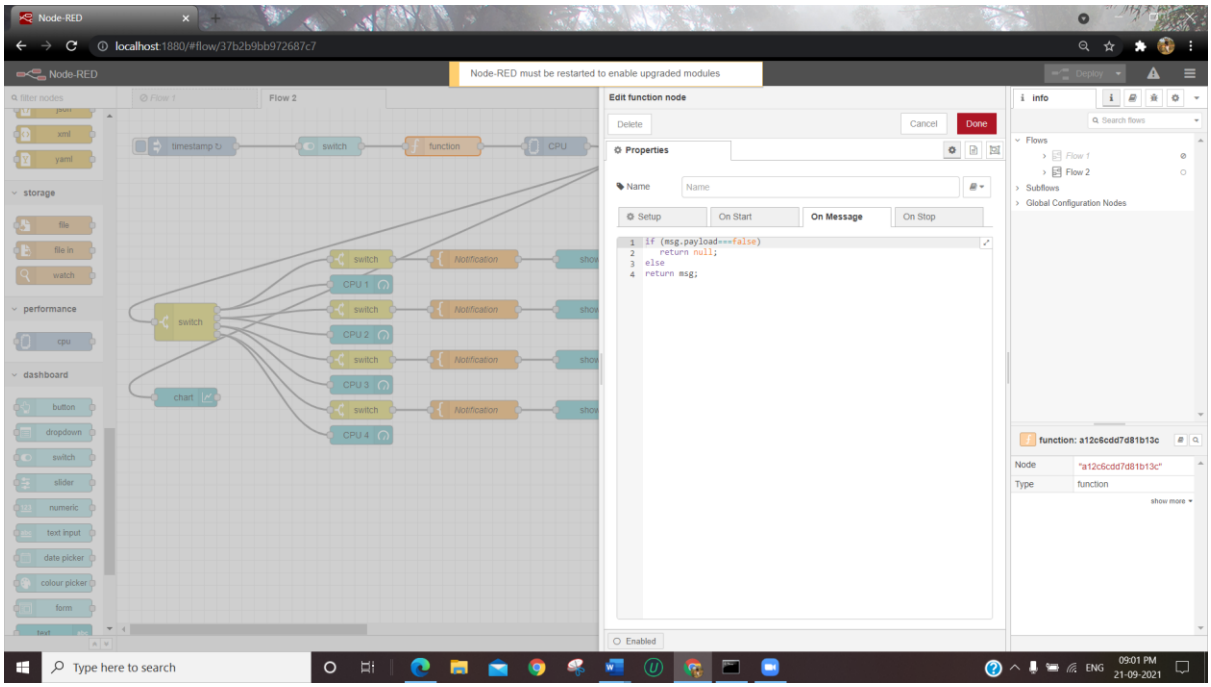
Inject Node: -



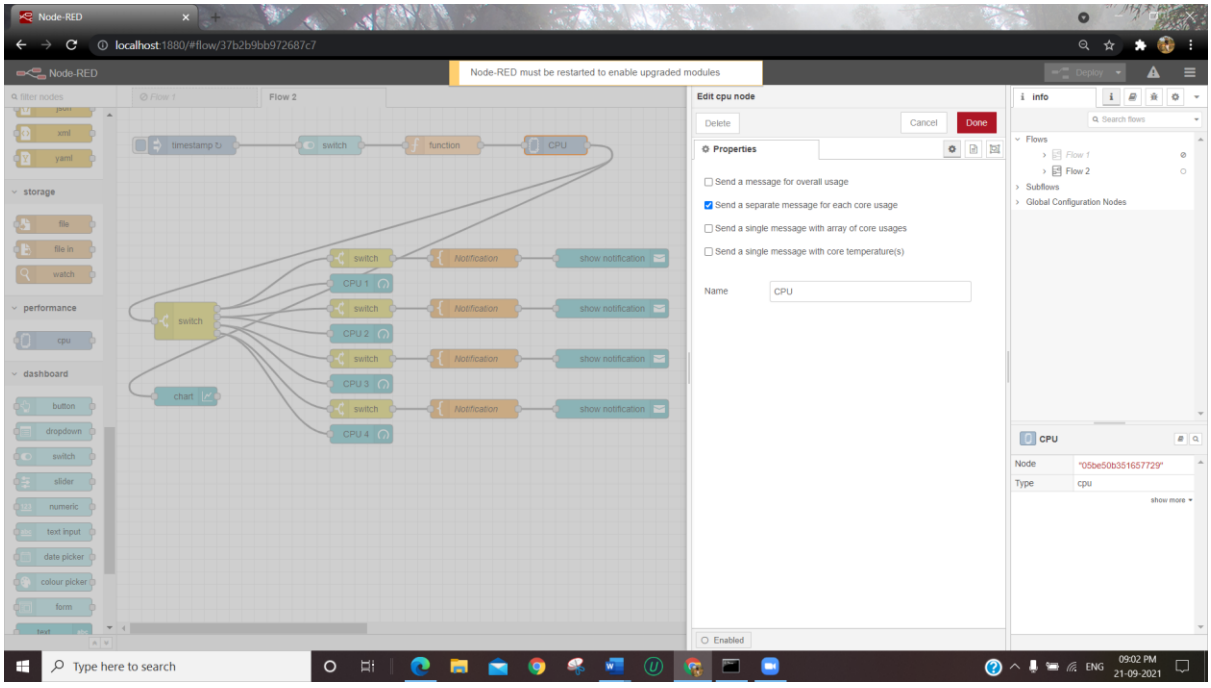
Switch Node: -



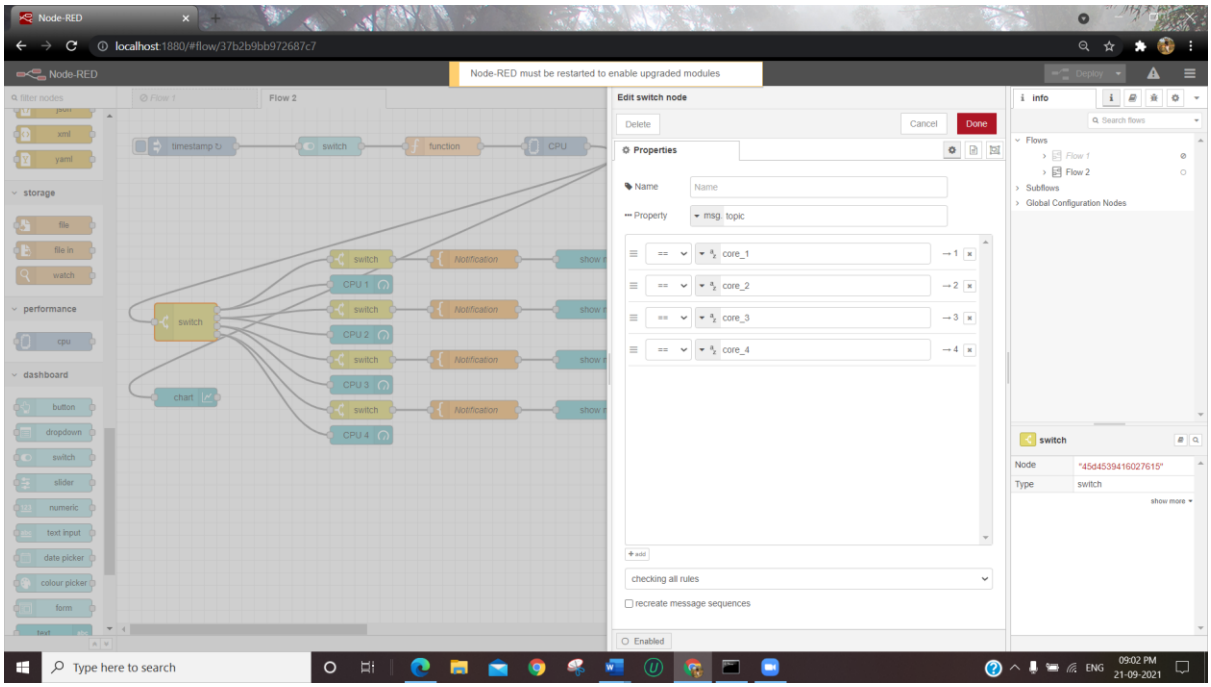
Function Node: -



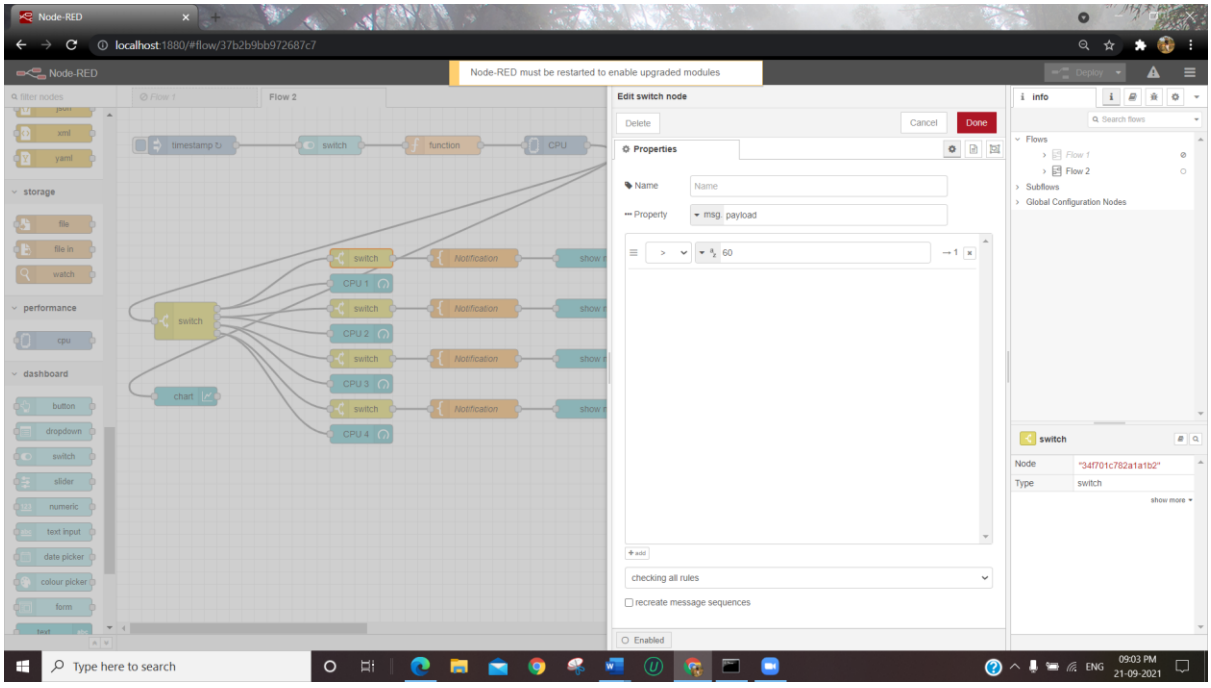
CPU: -



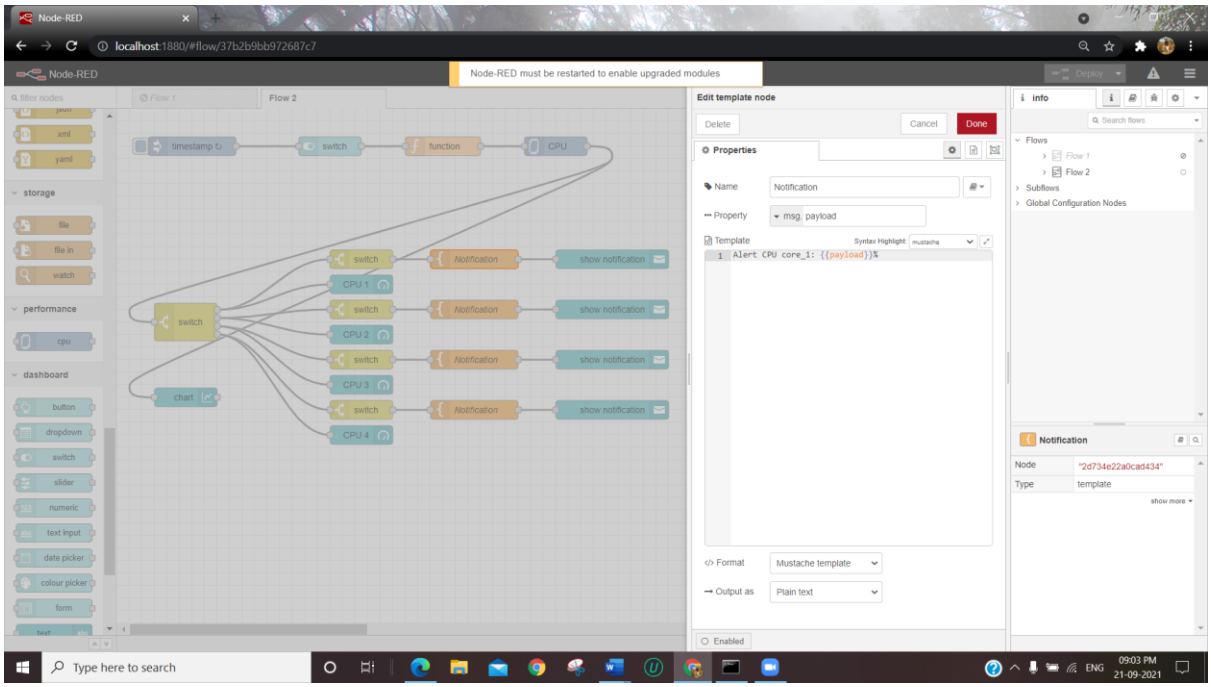
Switch: -



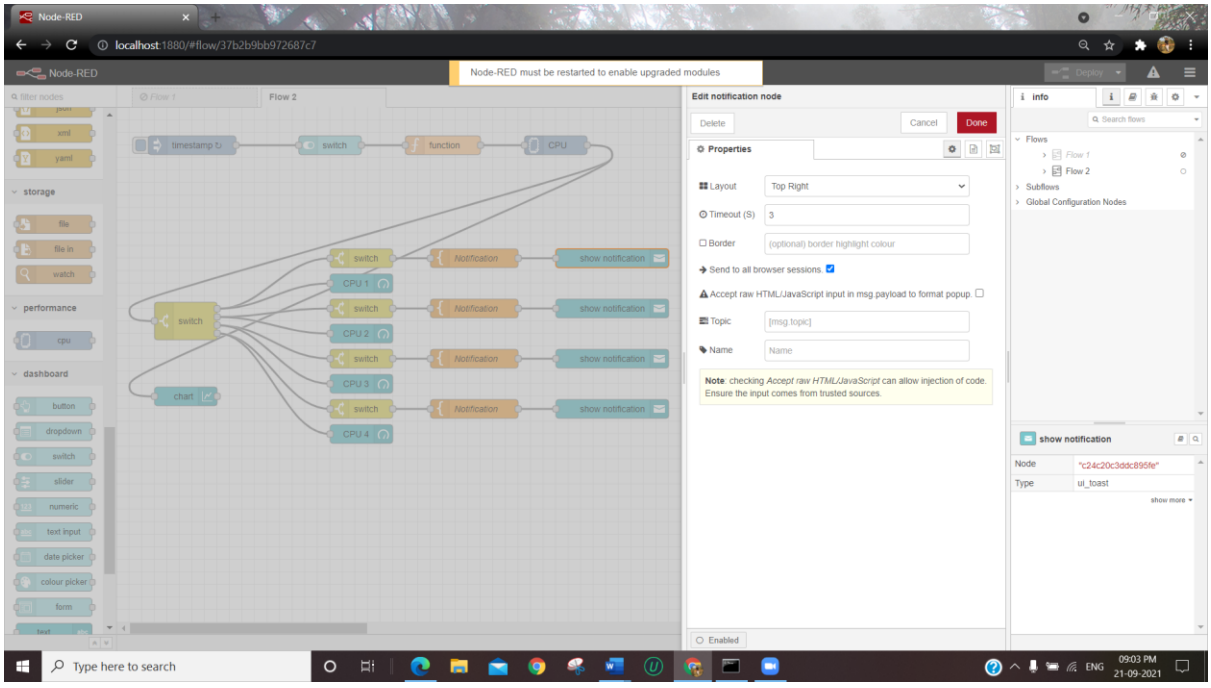
Switch: -



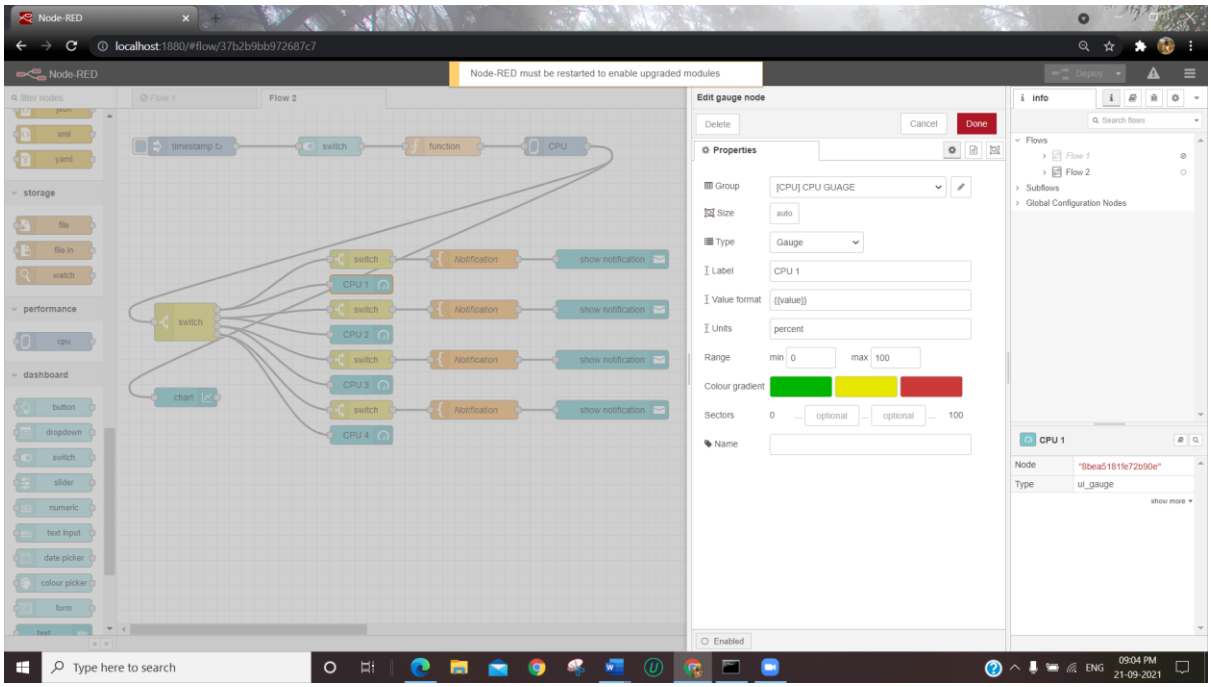
Notification: -



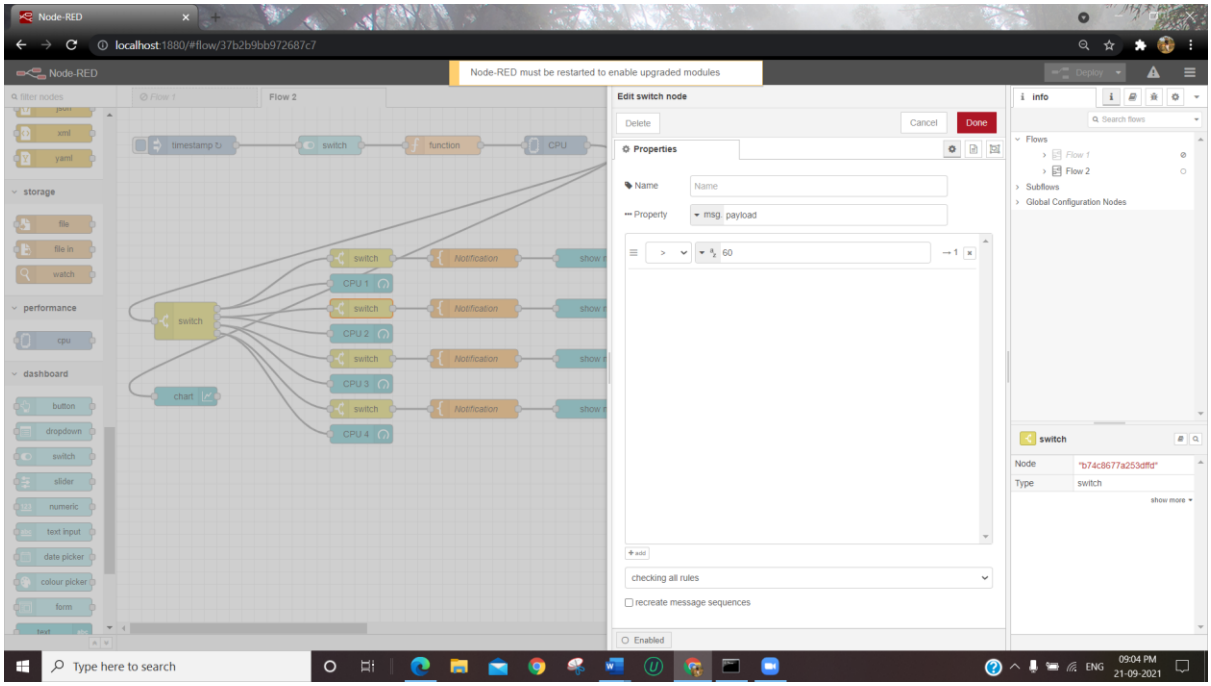
Show Notification: -



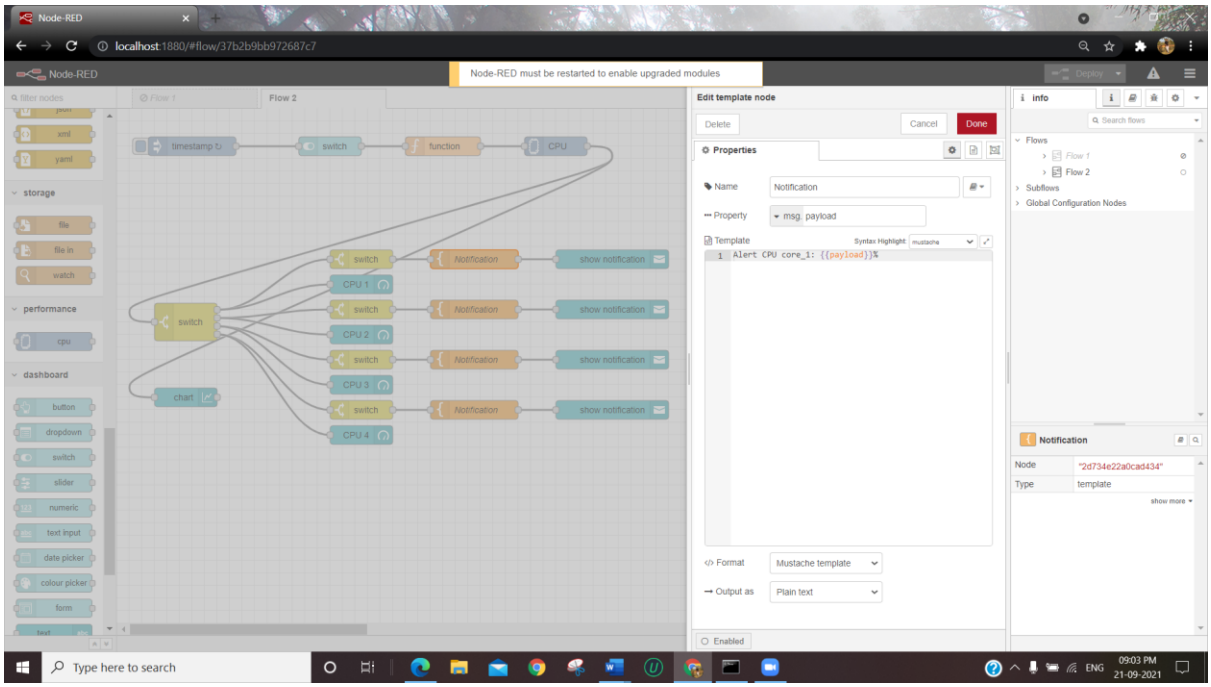
CPU1: -


Node-RED
localhost:1880/#flow/37b2b9bb972687c7
Node-RED must be restarted to enable upgraded modules
Edit gauge node
Delete
Cancel
Done
Properties
Group
[CPU] CPU GUAGE
Size
auto
Type
Gauge
Label
CPU 1
Value format
{{value}}
Units
percent
Range
min 0
max 100
Colour gradient
Sectors
Name
Enabled
info
Flows
Flow 1
Flow 2
Subflows
Global Configuration Nodes
CPU 1
Node
"8bea5181fe72b90e"
Type
ui_gauge
show more
timestamp
switch
function
CPU
Notification
show notification
chart
Type here to search
ENG
09:04 PM
21-09-2021

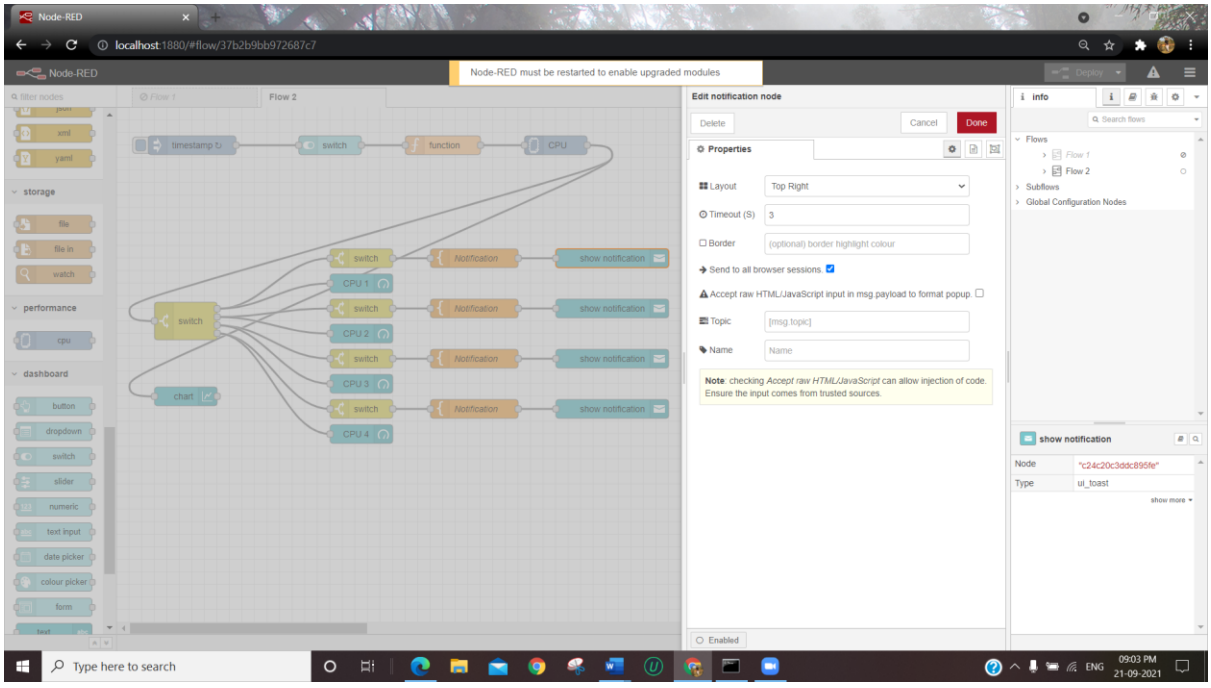
Switch: -


Node-RED
localhost:1880/#flow/37b2b9bb972687c7
Node-RED must be restarted to enable upgraded modules
Edit switch node
Delete
Cancel
Done
Properties
Name
Property
msg. payload
60
checking all rules
recreate message sequences
Enabled
info
Flows
Flow 1
Flow 2
Subflows
Global Configuration Nodes
switch
Node
"b74c8677a253dffd"
Type
switch
show more
timestamp
switch
function
CPU
Notification
chart
Type here to search
ENG
09:04 PM
21-09-2021

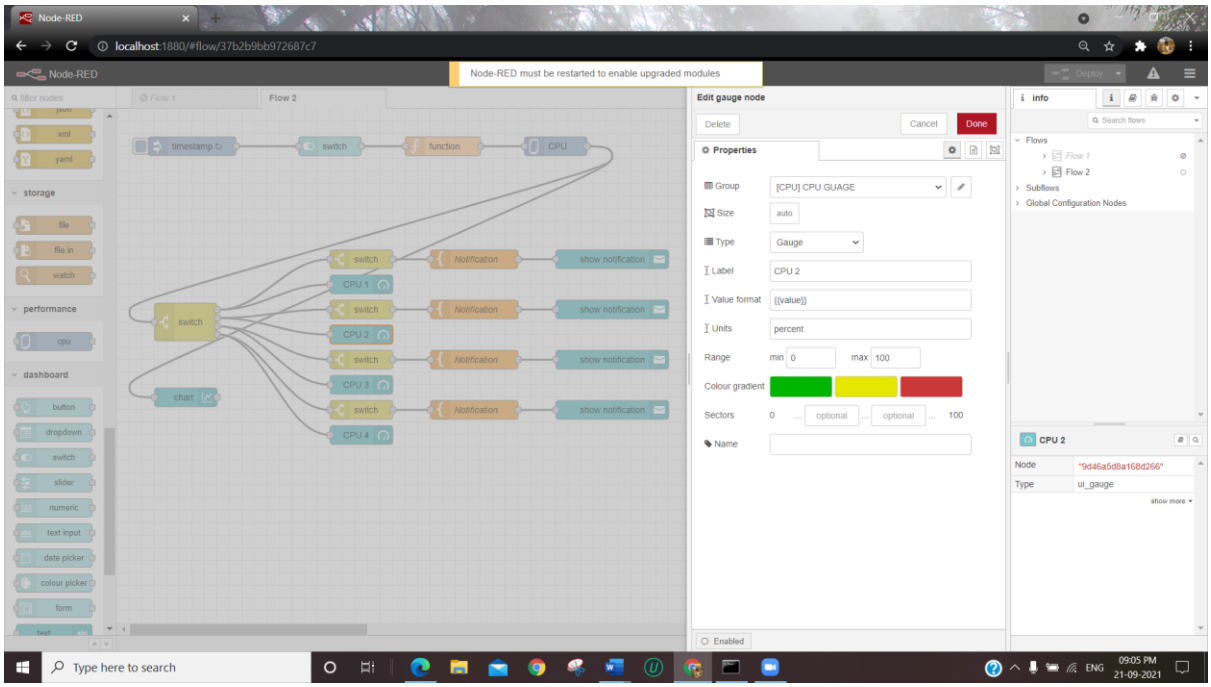
Notification: -



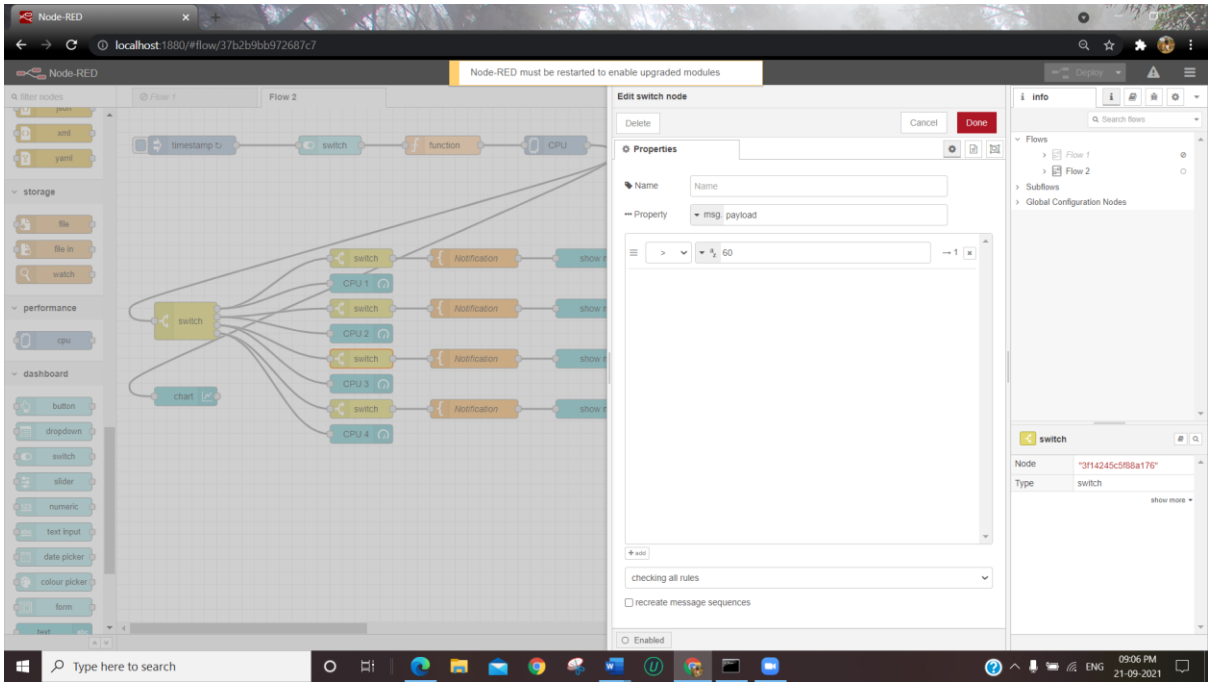
Show Notification: -



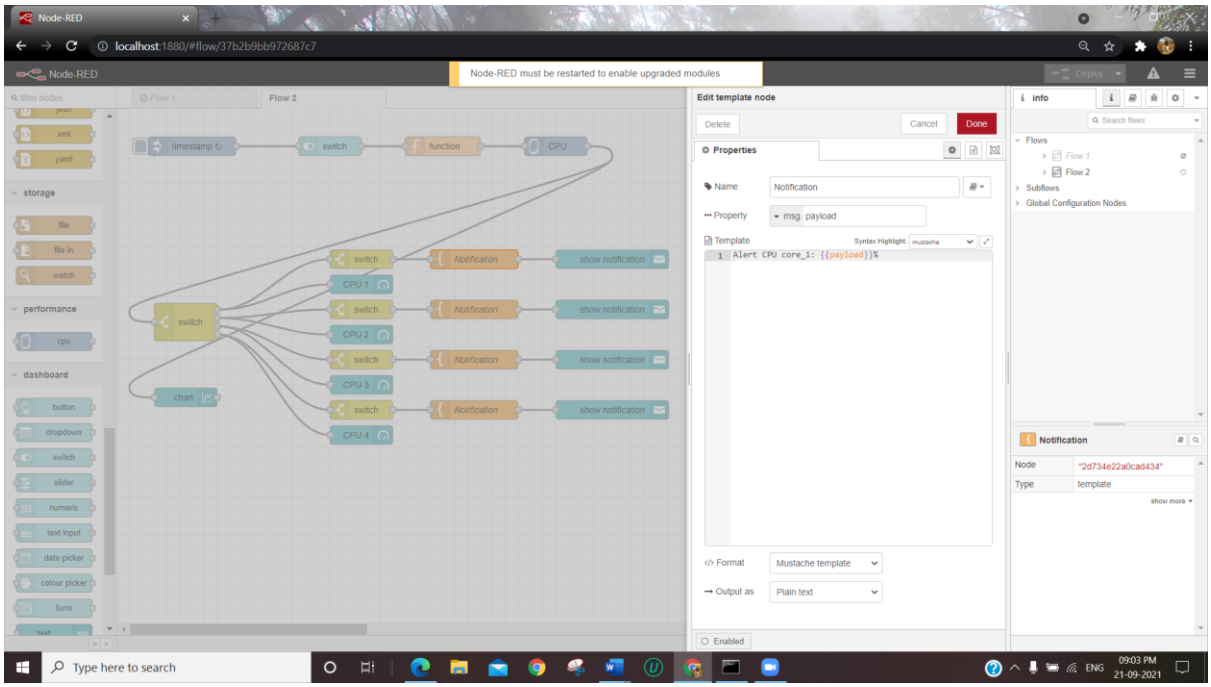
CPU2: -


Node-RED
localhost:1880/#flow/37b2b9bb972687c7
Node-RED must be restarted to enable upgraded modules
Edit gauge node
Delete
Cancel
Done
Properties
Group
[CPU] CPU GUAGE
Size
auto
Type
Gauge
Label
CPU 2
Value format
{{value}}
Units
percent
Range
min 0
max 100
Colour gradient
Sectors
0
optional
optional
100
Name
Enabled
info
Flows
Flow 1
Flow 2
Subflows
Global Configuration Nodes
CPU 2
Node
"9d46a5d8a168d266"
Type
ui_gauge
show more
Type here to search
ENG
09:05 PM
21-09-2021

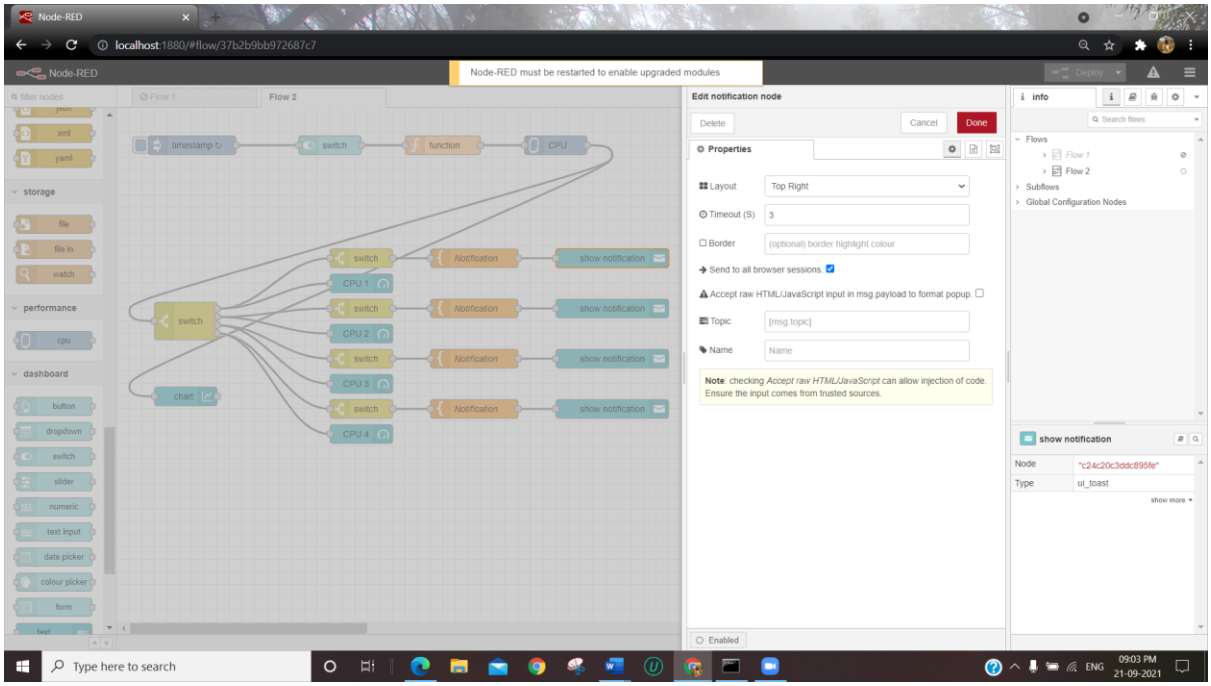
Switch: -


Node-RED
localhost:1880/#flow/37b2b9bb972687c7
Node-RED must be restarted to enable upgraded modules
Edit switch node
Delete
Cancel
Done
Properties
Name
Property
msg. payload
60
checking all rules
recreate message sequences
Enabled
info
Flows
Flow 1
Flow 2
Subflows
Global Configuration Nodes
switch
Node
"3f14245c5f88a176"
Type
switch
show more
Type here to search
ENG
09:06 PM
21-09-2021

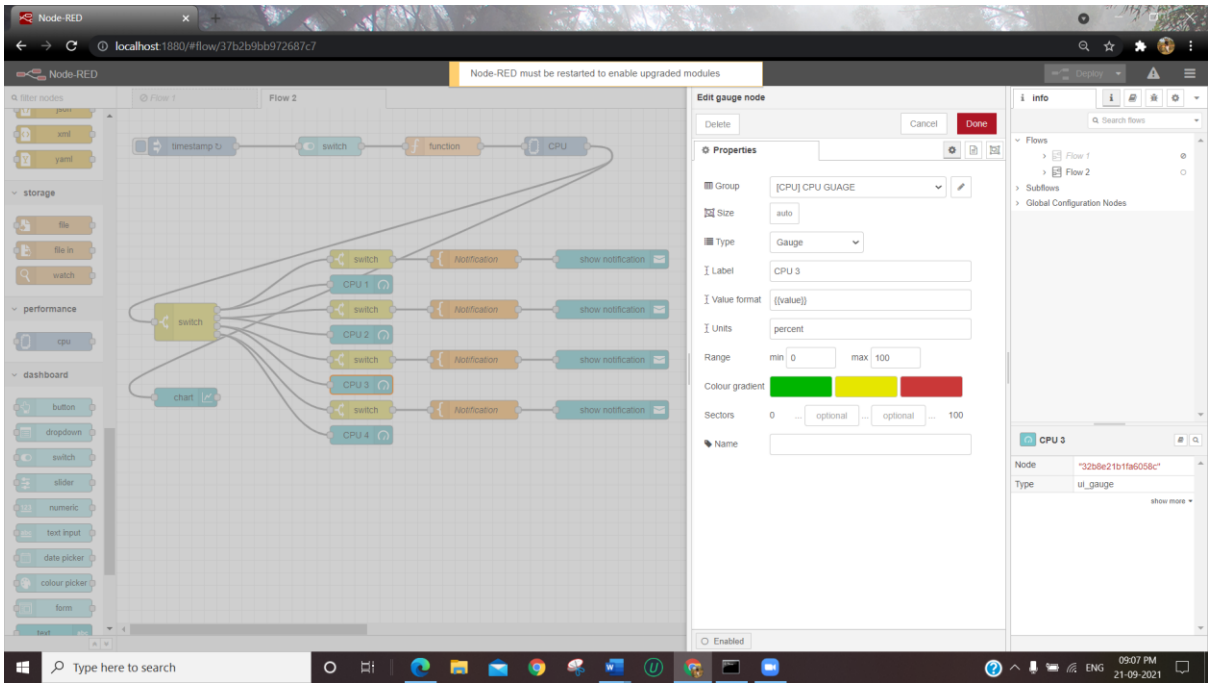
Notification: -



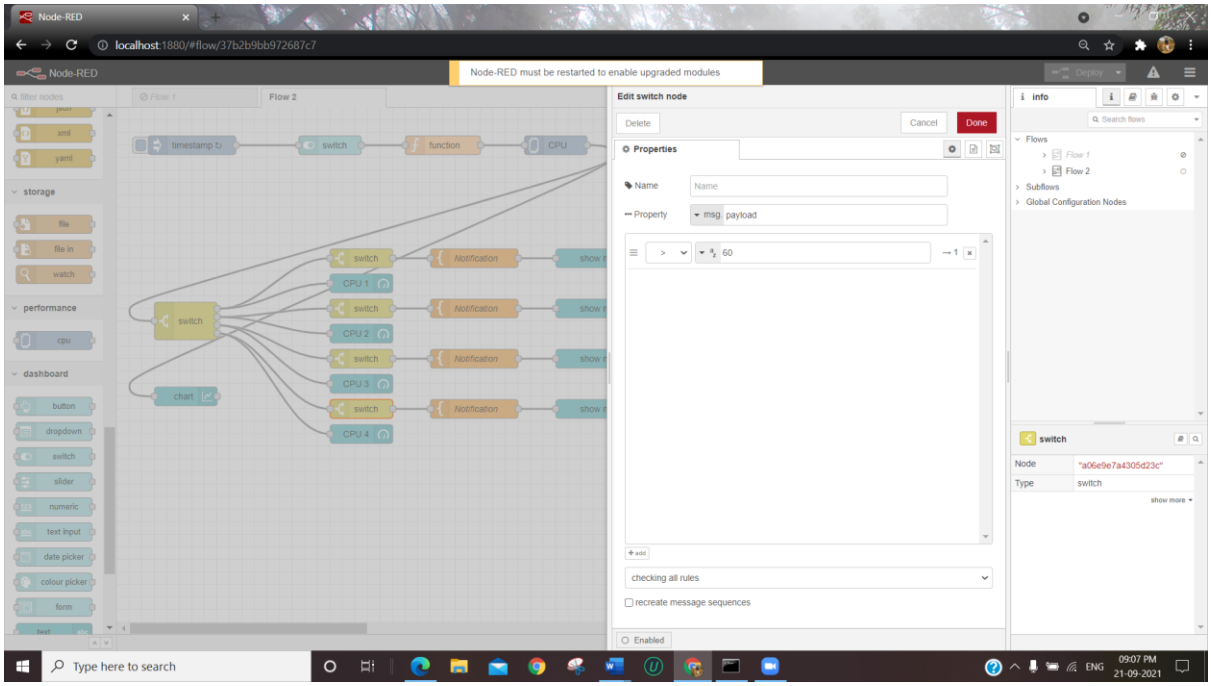
Show Notification: -



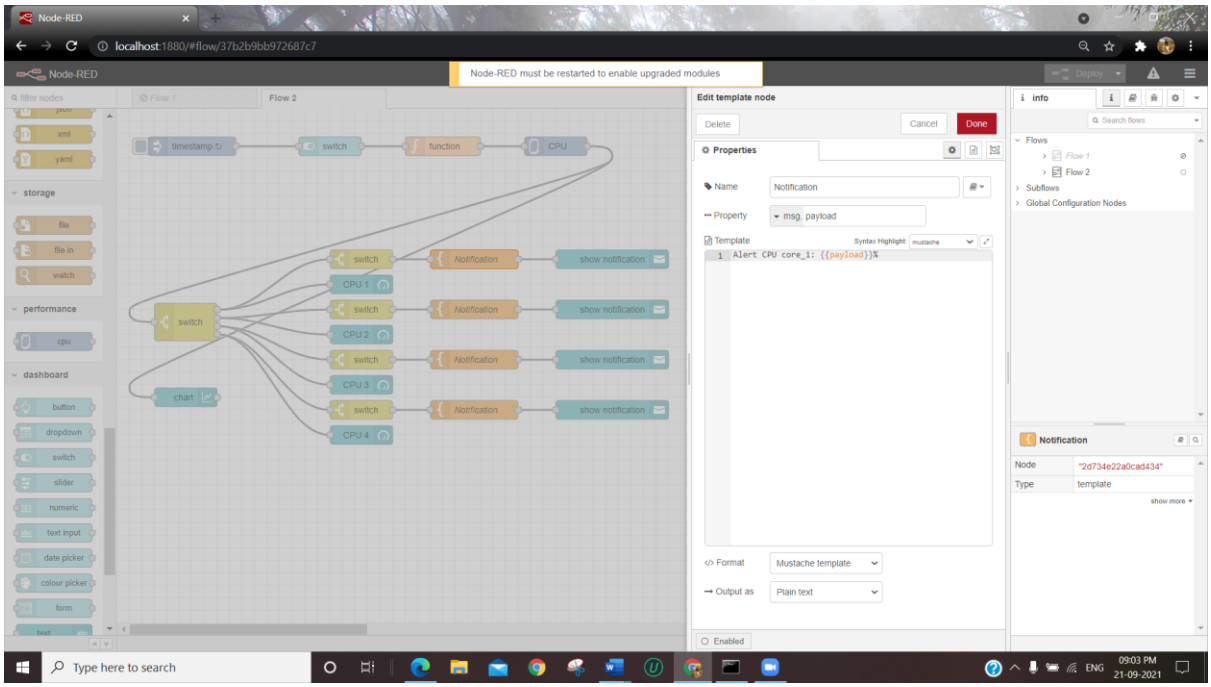
CPU3: -


Node-RED
localhost:1880/#flow/37b2b9bb972687c7
Node-RED must be restarted to enable upgraded modules
Edit gauge node
Delete
Cancel
Done
Properties
Group
[CPU] CPU GUAGE
Size
auto
Type
Gauge
Label
CPU 3
Value format
{{value}}
Units
percent
Range
min 0
max 100
Colour gradient
Sectors
Name
Enabled
info
Flows
Flow 1
Flow 2
Subflows
Global Configuration Nodes
CPU 3
Node
"32b8e21b1fa6058c"
Type
ui_gauge
show more
Type here to search
ENG
09:07 PM
21-09-2021

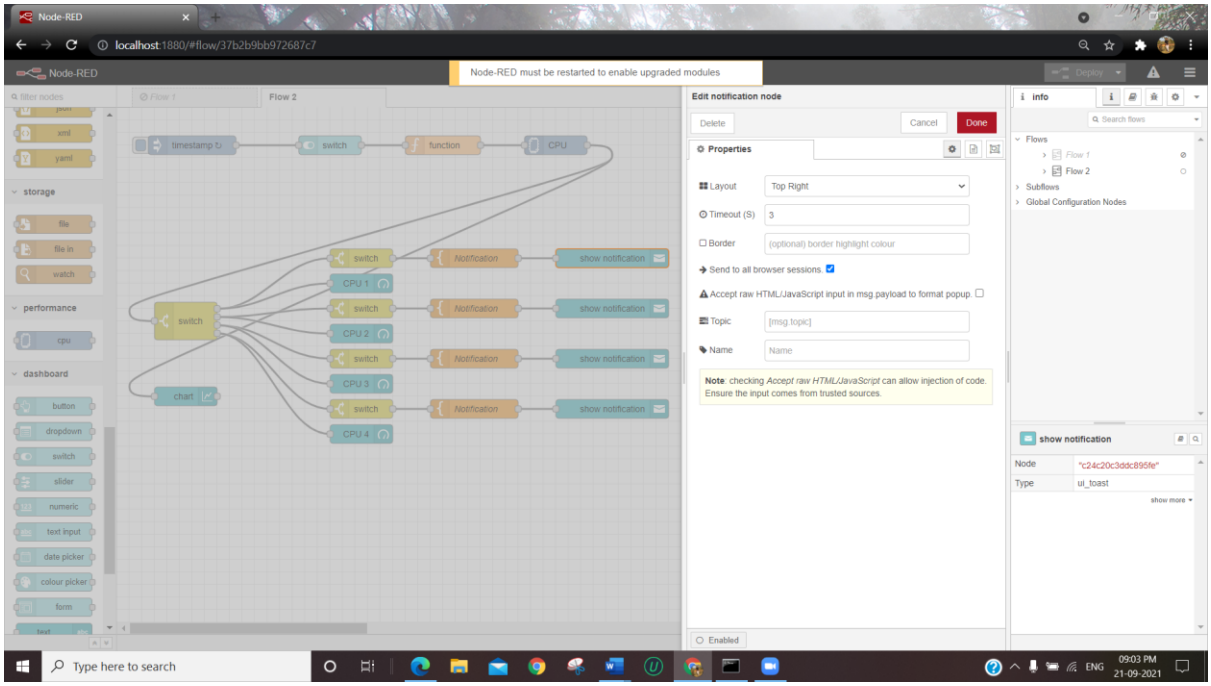
Switch: -


Node-RED
localhost:1880/#flow/37b2b9bb972687c7
Node-RED must be restarted to enable upgraded modules
Edit switch node
Delete
Cancel
Done
Properties
Name
Property
msg payload
60
checking all rules
recreate message sequences
Enabled
info
Flows
Flow 1
Flow 2
Subflows
Global Configuration Nodes
switch
Node
"a06e9e7a4305d23c"
Type
switch
show more
Type here to search
ENG
09:07 PM
21-09-2021

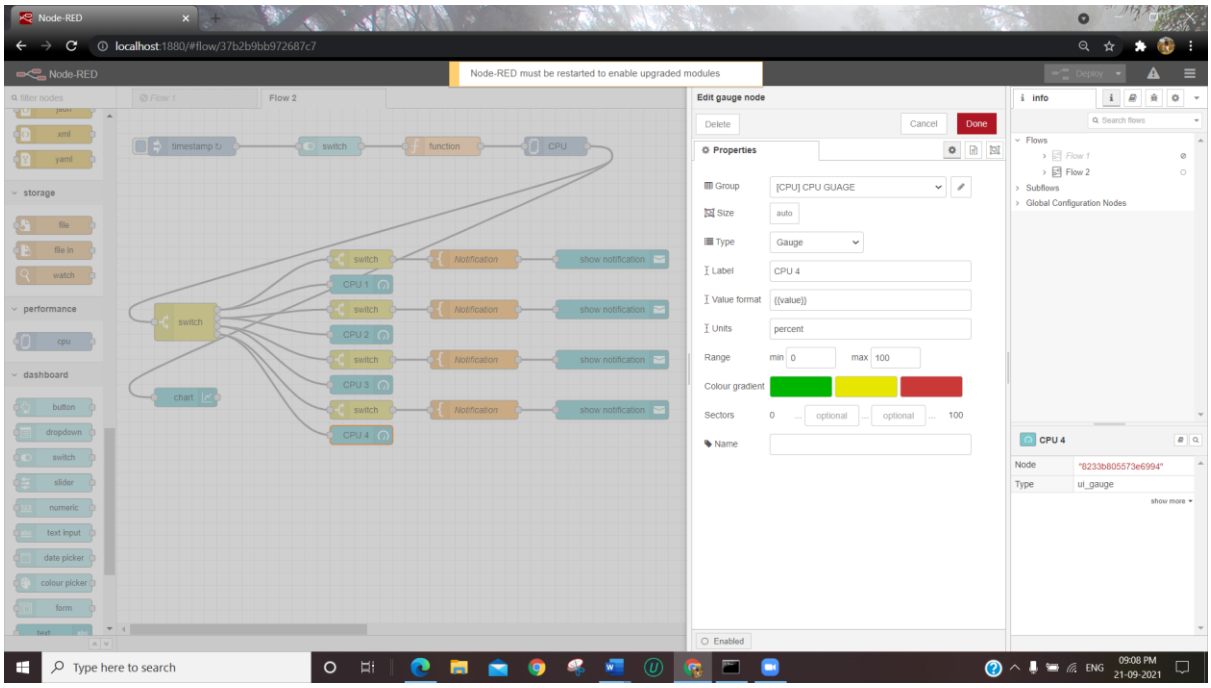
Notification: -



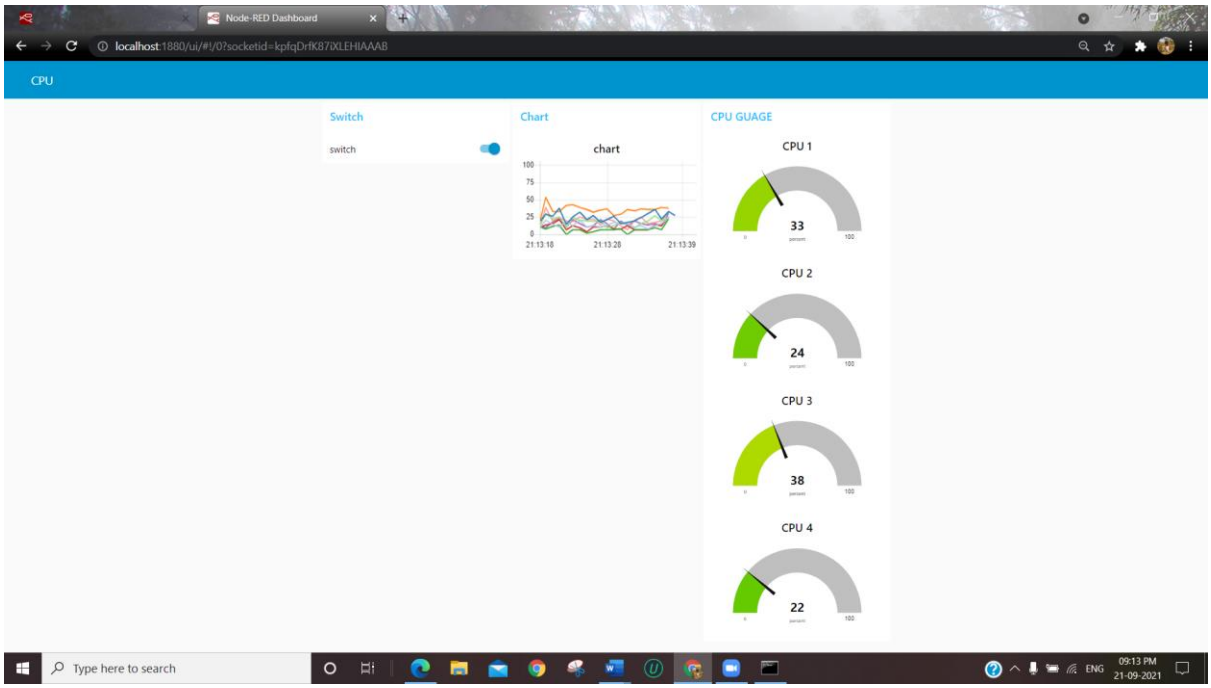
Show Notification: -



CPU4: -



final dashboard



Practical 8

Practical Name: WiFi Setup

Code:

Energia:

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

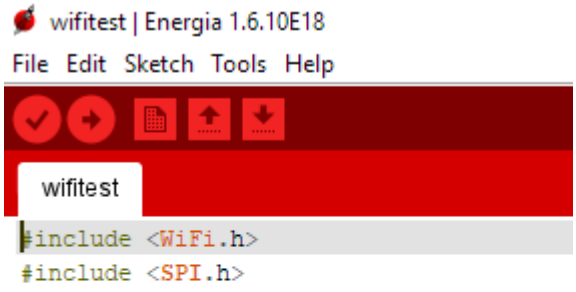
CCS:

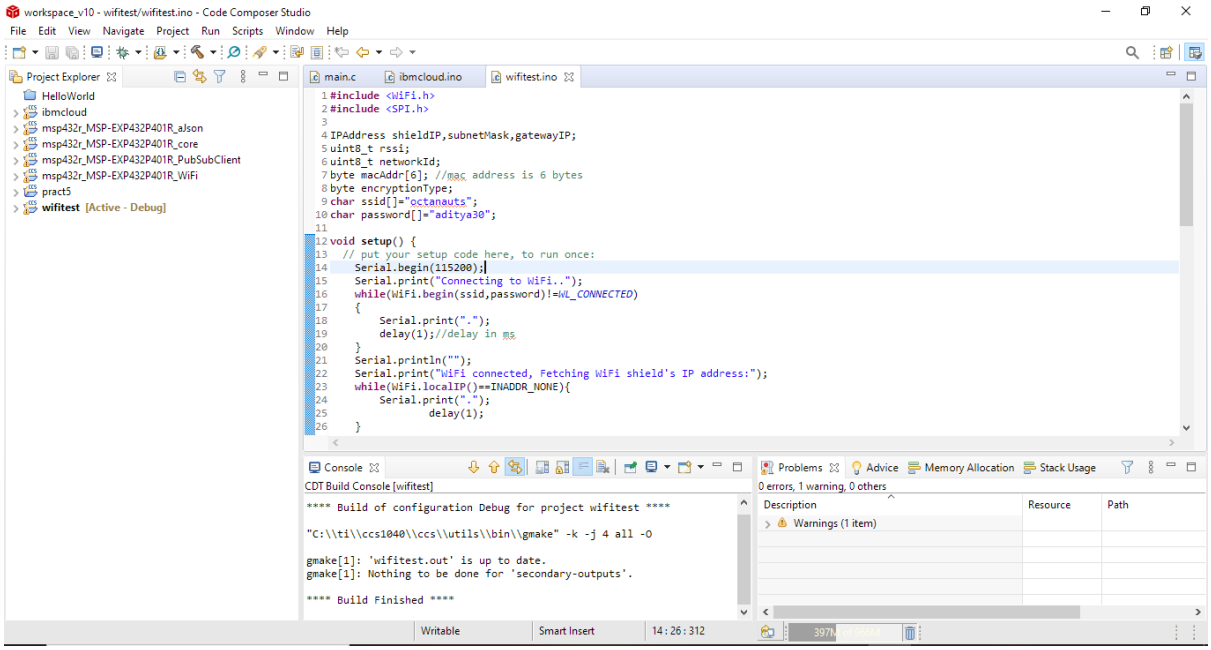
```
#include <WiFi.h>
#include <SPI.h>

IPAddress shieldIP,subnetMask,gatewayIP;
uint8_t rssi;
uint8_t networkId;
byte macAddr[6]; //mac address is 6 bytes
byte encryptionType;
char ssid[]="octanauts";
char password[]="aditya30";

void setup() {
  // put your setup code here, to run once:
    Serial.begin(115200);
    Serial.print("Connecting to WiFi..");
    while(WiFi.begin(ssid,password)!=WL_CONNECTED)
    {
       Serial.print(".");
       delay(1);//delay in ms
    }
    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.print(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.print(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 loop is used here as it takes mac addr in rev order. to put in correct order
```

```
  for(int i=5;i>-1;i--){
    Serial.print(macAddr[i],HEX);
    if(i>0)
      Serial.print(":");
    else
      Serial.println();
  }
}

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

}
```

Output and Screenshots:





Explain the API required in the WiFi setup on the board

Ans)

<Wifi.h>

With the Arduino WiFi Shield, this library allows an Arduino board to connect to the internet. It can serve as either a server accepting incoming connections or a client making outgoing ones.

<SPI.h>

Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers.With an SPI connection there is always one master device (usually a microcontroller) which controls the peripheral devices

Practical 9

Practical Name: Analog To Digital Converter

Code:

```
#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[]="hardik";
//char password[]="";

WiFiClient wclient;

//byte server[] = {198,41,30,241};



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
//while(WiFi.begin(ssid,password)!=WL_CONNECTED)
{
 Serial.print(".");
 delay(1);
 }
 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 Points in range: ");
  Serial.println(networkId);
  for(int i=1;i<=networkId;i++)
  {
```

```
  Serial.print("Number 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=5;i>-1;i++ )
 {
  Serial.print(macAddr[i],HEX);
  if(i>0)
  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_MODUL
E_FUNCTION);

ADC14_initModule(ADC_CLOCKSOURCE_MCLK,ADC_PREDIVIDER_1,ADC_PREDIVIDER_1,ADC_NORO
UTE);
 ADC14_configureSingleSampleMode(ADC_MEM6,true);
```

```
ADC14_configureConversionMemory(ADC_MEM6,ADC_VREFPOS_AVCC_VREFNEG_VSS,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();
}
```

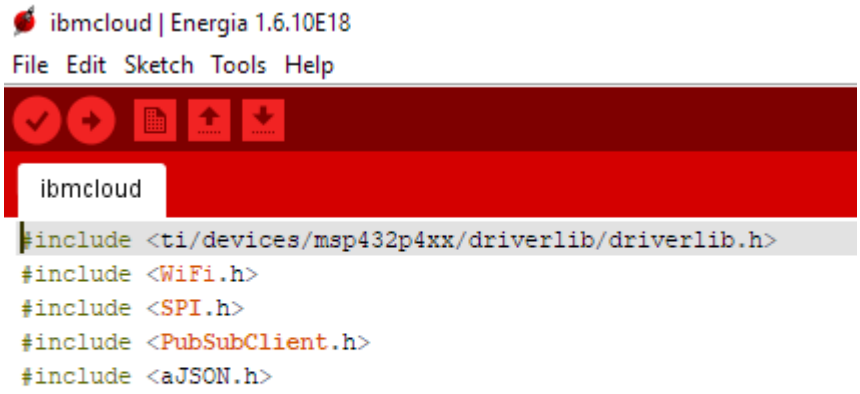
```
void loop() {
 // put your main code here, to run repeatedly:
 if(!client.connected())
 {
  Serial.println("Disconnected:Reconnection..");
  if(!client.connect("MSP432"))
  {
   Serial.println("Connection Failed");
  }
  else
  {
     Serial.println("Connection Success");
     if(client.subscribe("TurnOnOffLED1"))
     {
      Serial.println("Subscription Successful");
     }
  }
 }
```

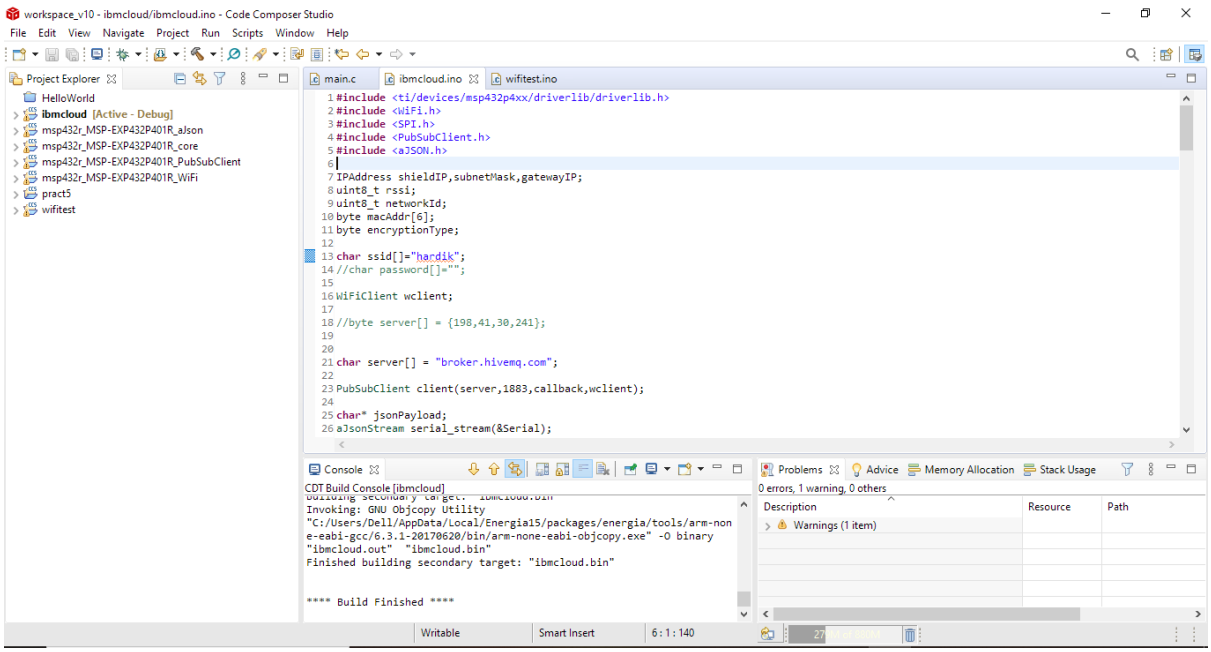
```
int result,regressedData1;
float regressedData;

while(!ADC14_isBusy());
result=ADC14_getResult(ADC_MEM6);

ADC14_toggleConversionTrigger();
regressedData = ((result*0.866848575)-2.12705)/1000;
Serial.println(regressedData);
regressedData1 = ((result*0.866848575)-2.12705);
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]='.';
```

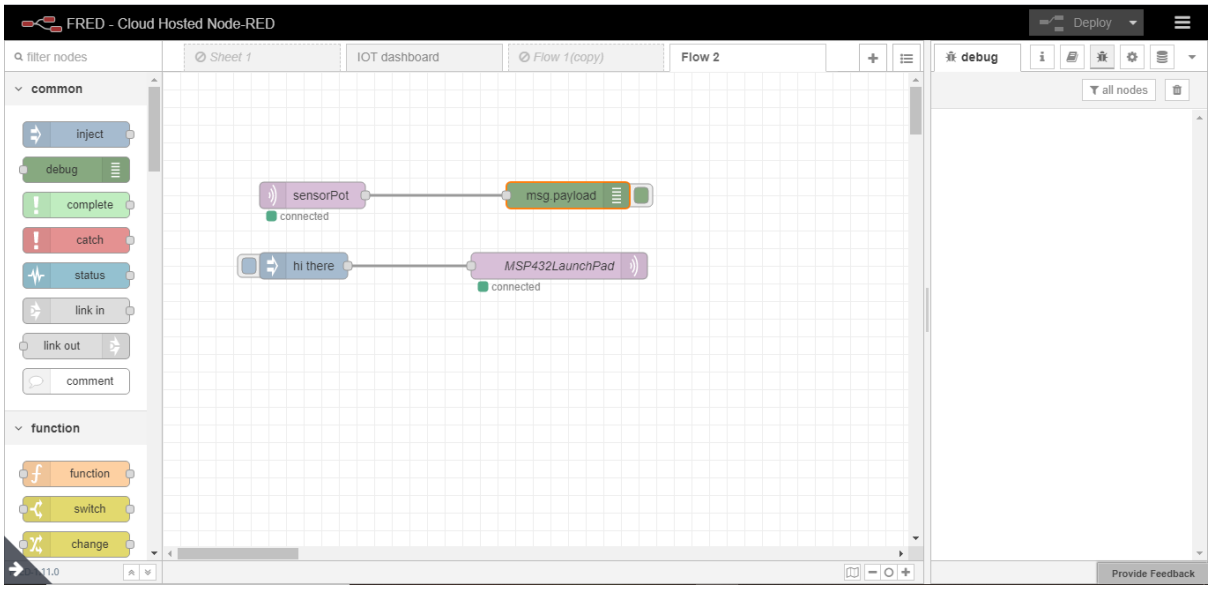
```
      for(int i=2;i<=str_len;i++)
      {
        pot_reading[i]= char_array[i-1];
       }
      }

  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';

  aJson.deleteItem(d);
  aJson.deleteItem(root);
  client.publish("sensorPot",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++)
```

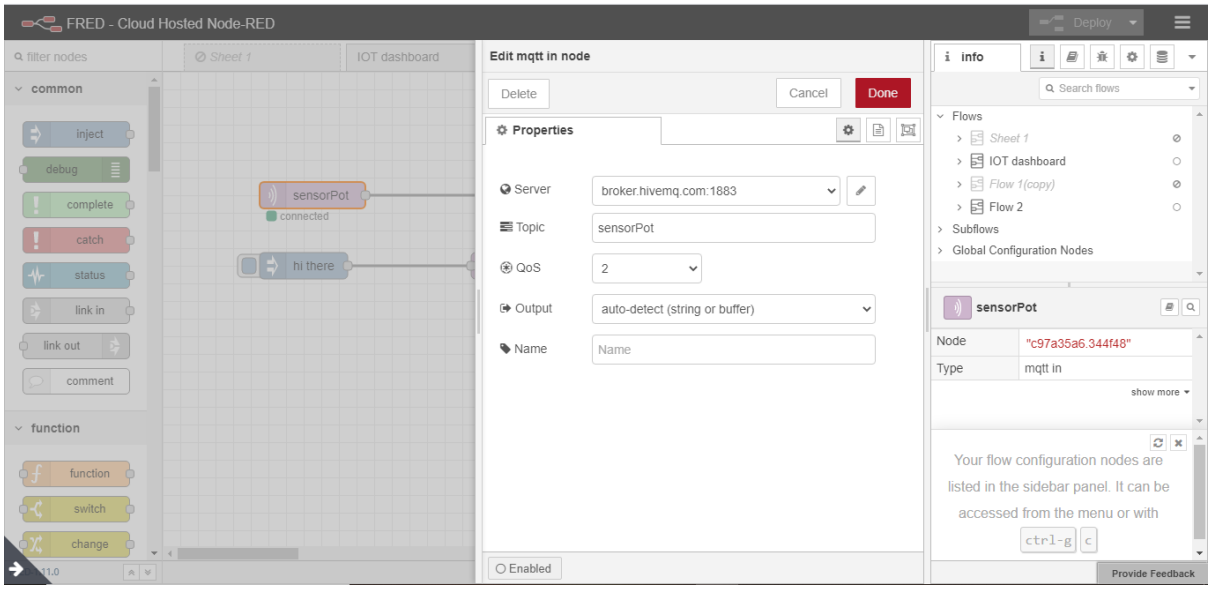
```
{
  Serial.print((char)(payload[i]));
  arrivedMessage+=(char)payload[i];
  }
 Serial.println();
}
```

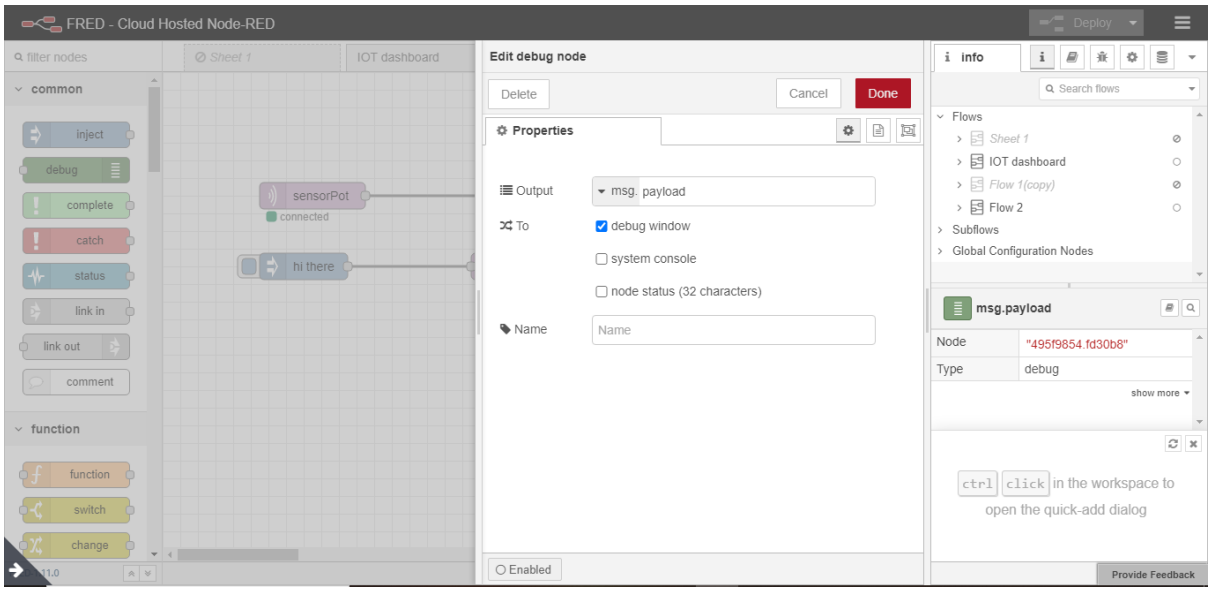
Output and Screenshots:

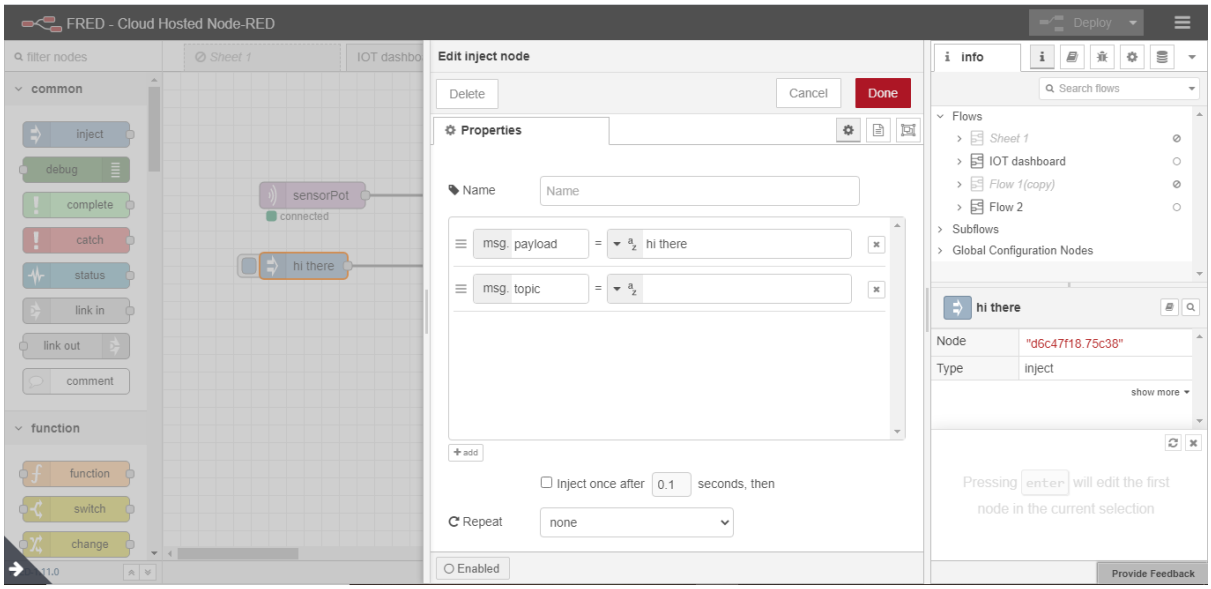


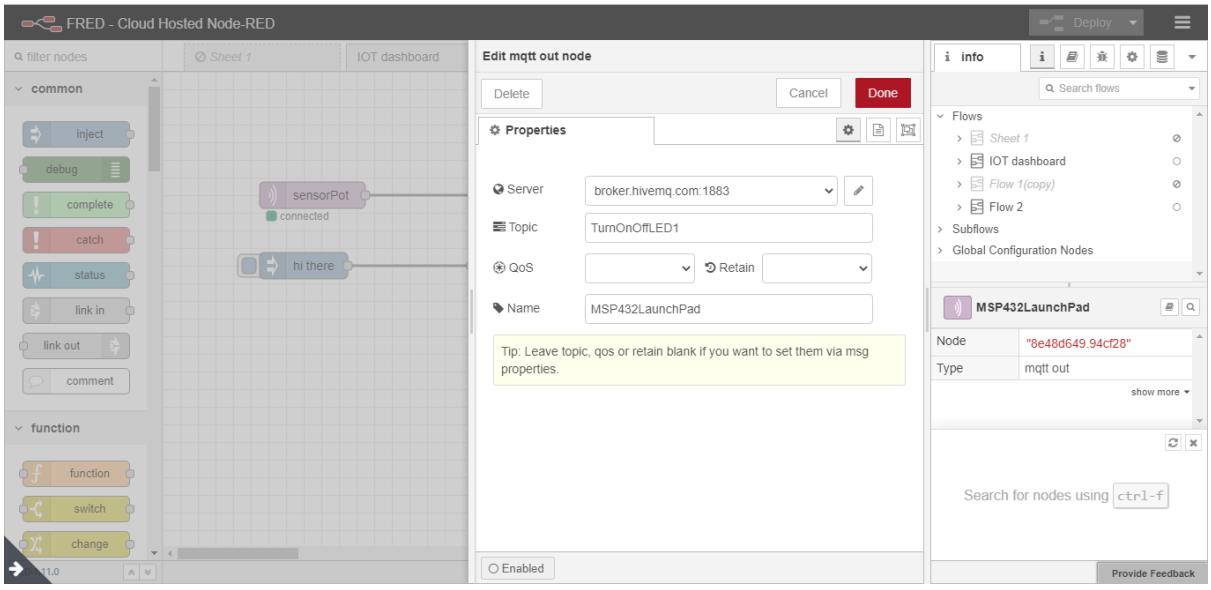



FRED - Cloud Hosted Node-RED
Deploy
filter nodes
Sheet 1
IOT dashboard
Flow 1(copy)
Flow 2
debug
all nodes
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
sensorPot
connected
msg.payload
hi there
MSP432LaunchPad
connected
Provide Feedback


FRED - Cloud Hosted Node-RED
Deploy
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
info
Search flows
Flows
Sheet 1
IOT dashboard
Flow 1(copy)
Flow 2
Subflows
Global Configuration Nodes
sensorPot
Node
"c97a35a6.344f48"
Type
mqtt in
show more
Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the menu or with
ctrl-g
c
Provide Feedback


FRED - Cloud Hosted Node-RED
Deploy
Edit debug node
Delete
Cancel
Done
Properties
Output
msg. payload
To
debug window
system console
node status (32 characters)
Name
Name
Enabled
info
Search flows
Flows
Sheet 1
IOT dashboard
Flow 1(copy)
Flow 2
Subflows
Global Configuration Nodes
msg.payload
Node
"495f9854.fd30b8"
Type
debug
show more
ctrl
click
in the workspace to open the quick-add dialog
Provide Feedback





1) Explain the ADC properties

Ans)

Resolution

The resolution of the converter indicates the number of discrete values it can produce. It is usually expressed in bits.

Accuracy

Accuracy depends on the error in the conversion. If the ADC is not broken, this error has two components: quantization error and (assuming the ADC is intended to be linear) non-linearity.

Sampling rate

The analogue signal is continuous in time and it is necessary to convert this to a flow of digital values. It is therefore required to define the rate at which new digital values are

sampled from the analogue signal. The rate of new values is called the sampling rate or sampling frequency of the converter.

Aliasing

All ADCs work by sampling their input at discrete intervals of time. Their output is therefore an incomplete picture of the behaviour of the input.

2) Explain the API used for configuring the ADC

Ans)

<Wifi.h>

With the Arduino WiFi Shield, this library allows an Arduino board to connect to the internet. It can serve as either a server accepting incoming connections or a client making outgoing ones.

<SPI.h>

Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers.With an SPI connection there is always one master device (usually a microcontroller) which controls the peripheral devices

<PubSubClient.h>

A client library for MQTT messaging. MQTT is a lightweight messaging protocol ideal for small devices. This library allows you to send and receive MQTT messages.

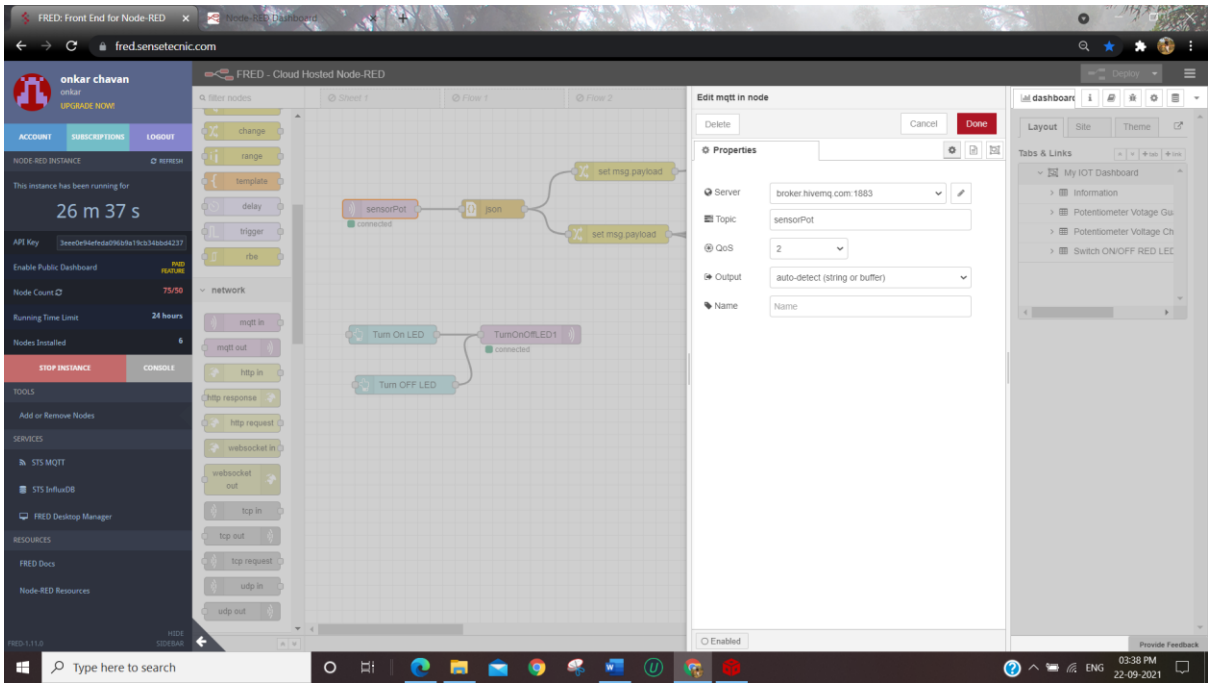
Practical 10

Connecting MSP432 to the cloud

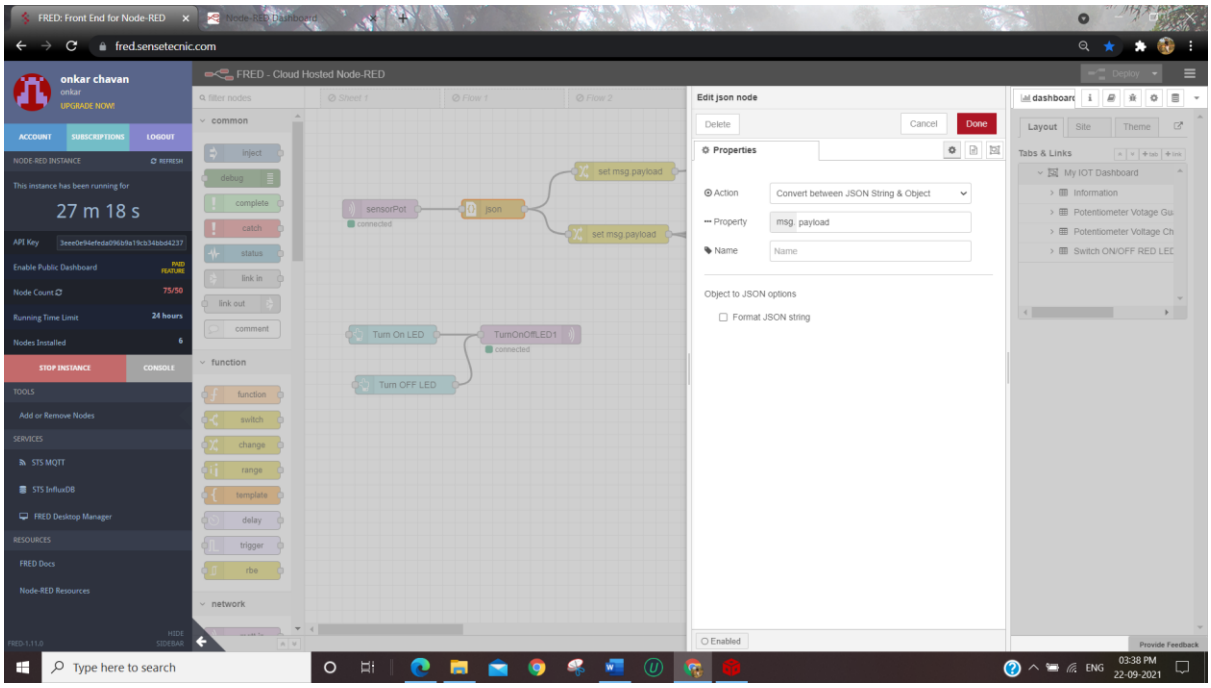
Points to be covered in the following order:

1.Explain the cloud dashboard creation with screenshots of the configuration of various nodes.

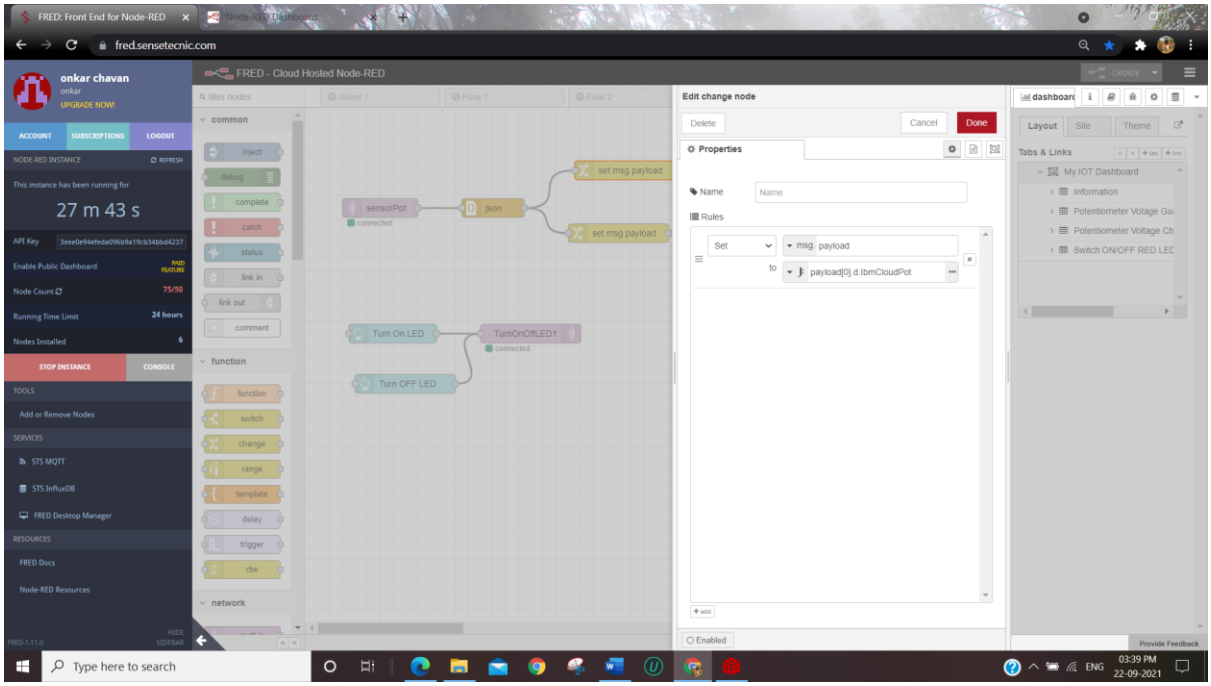
Mqtt in Node: -



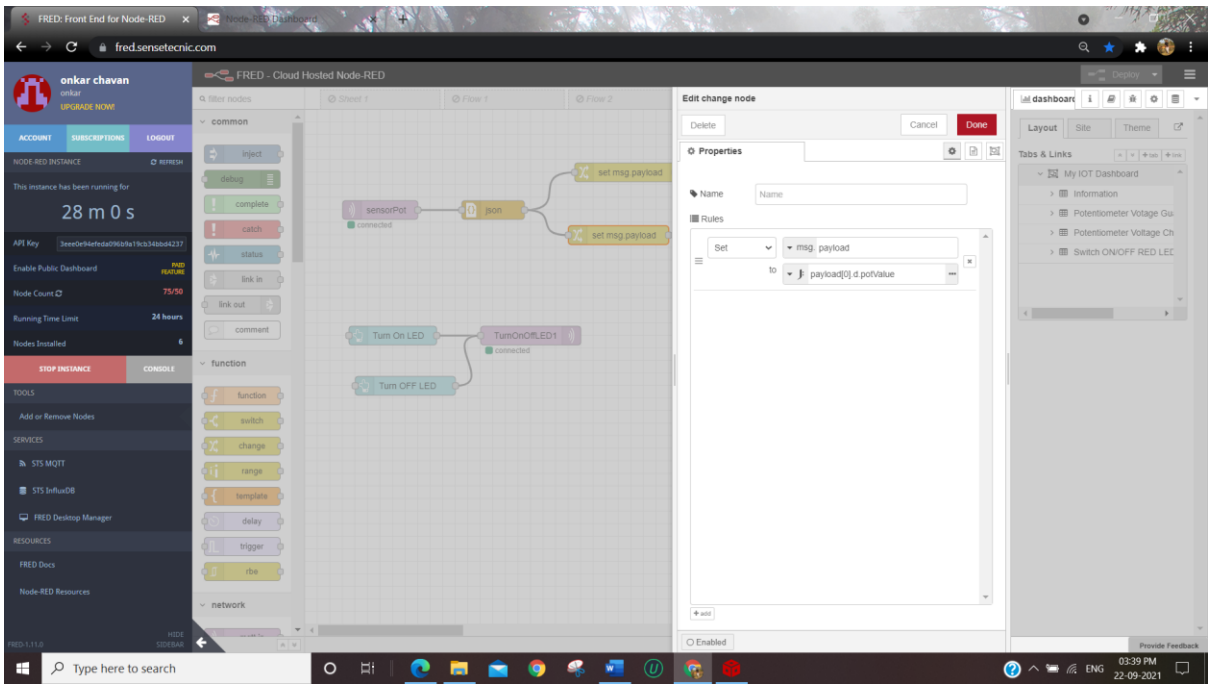
Json Node: -



Change Node: -



Change Node: -



Text Node: -

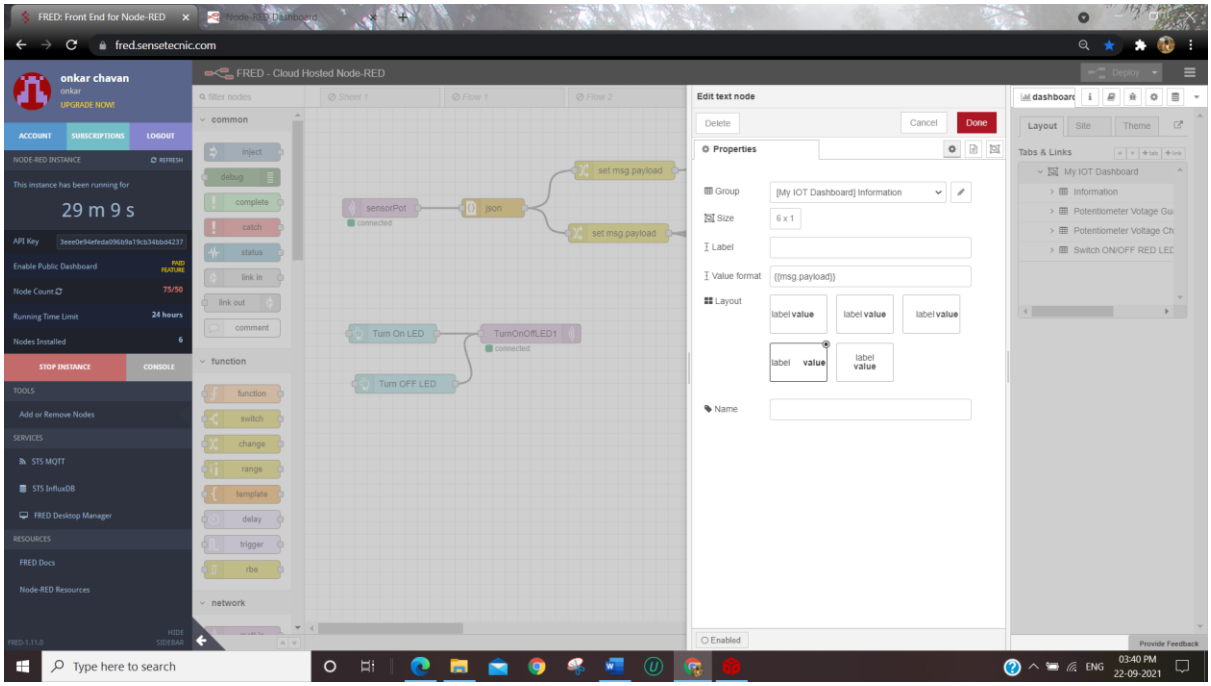
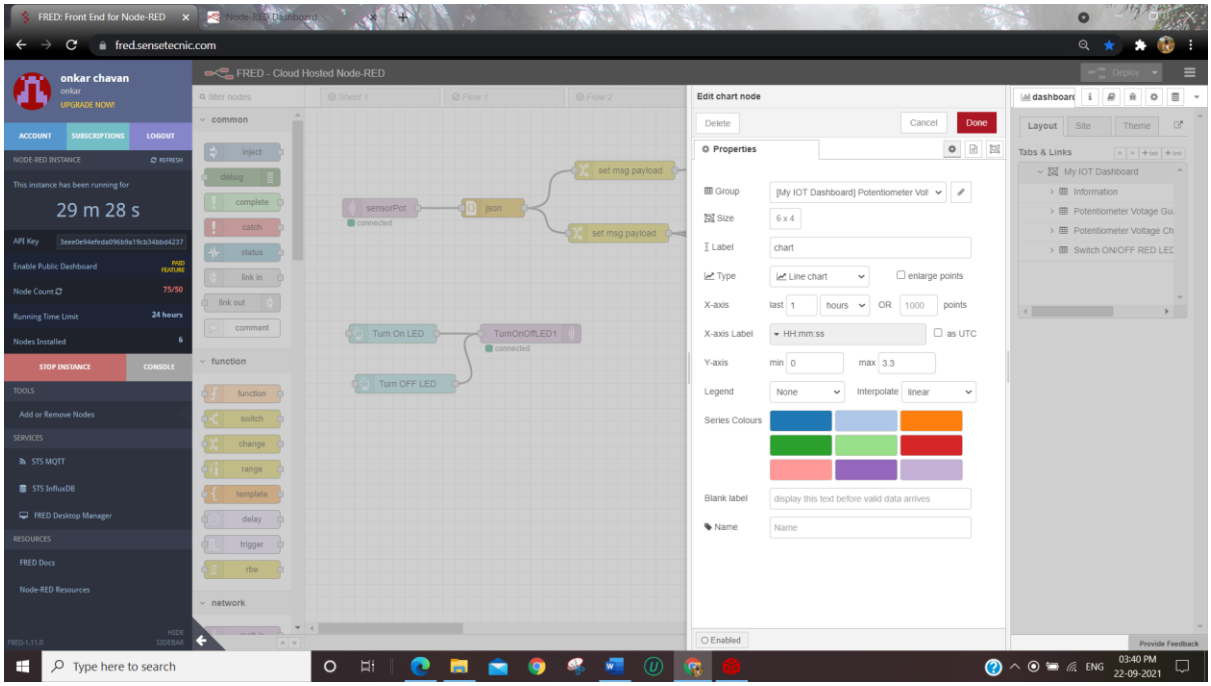
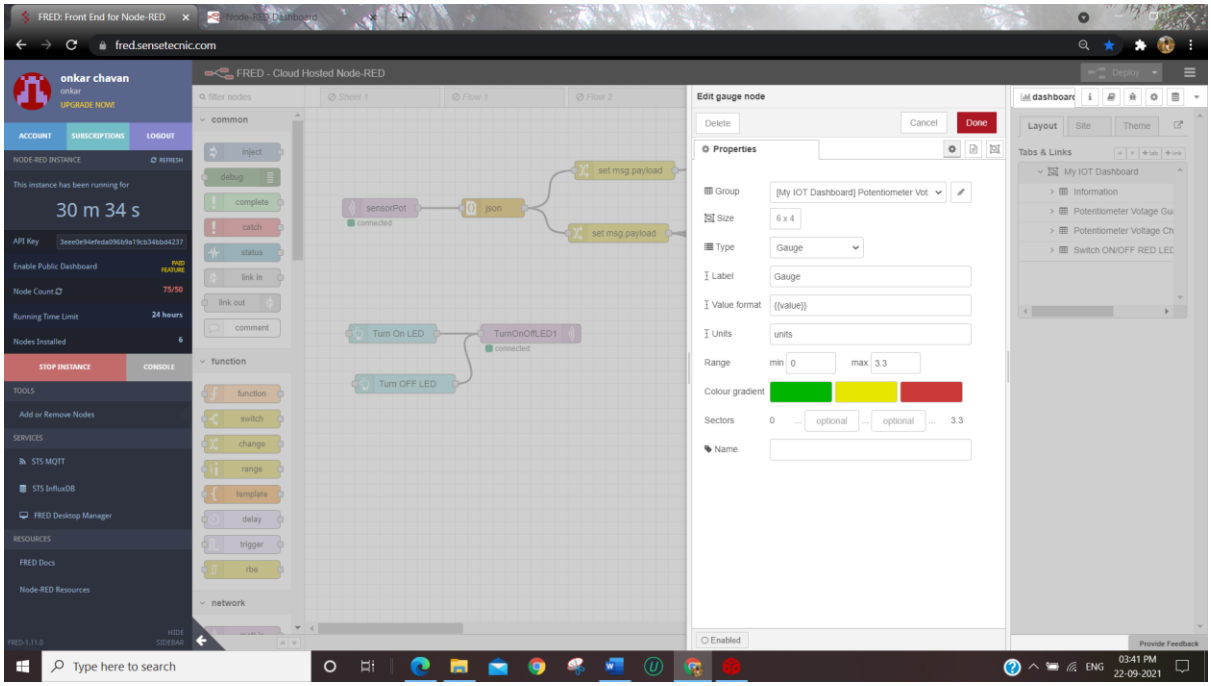


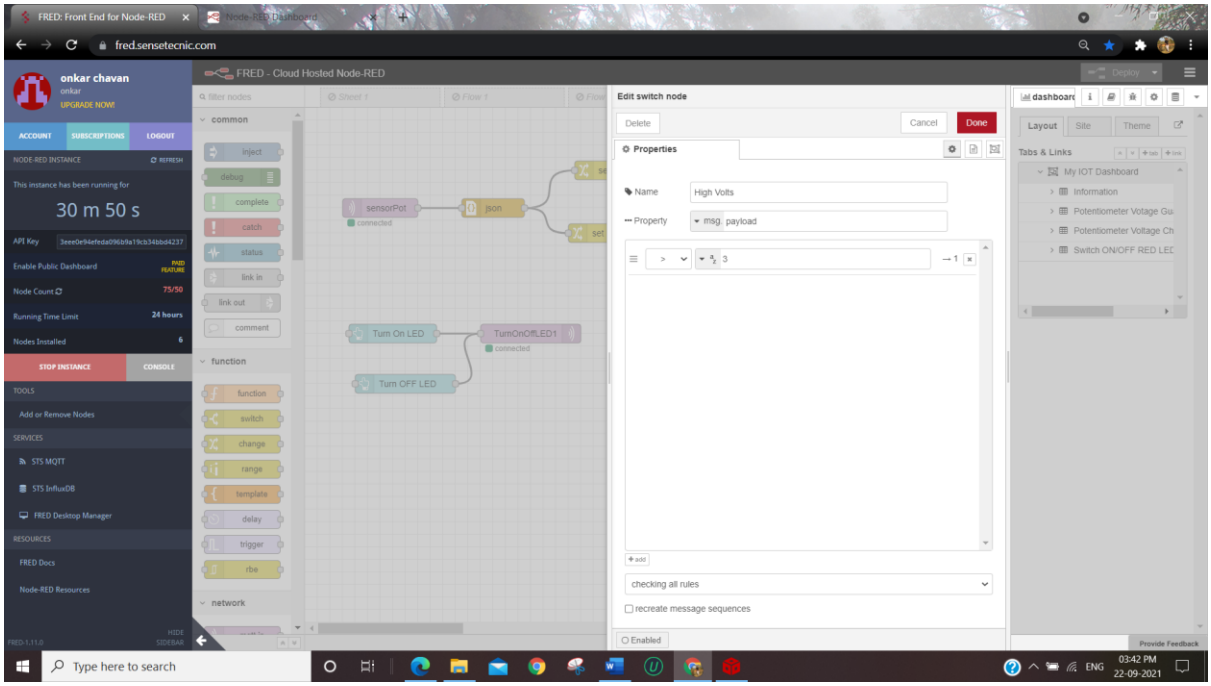
Chart Node: -



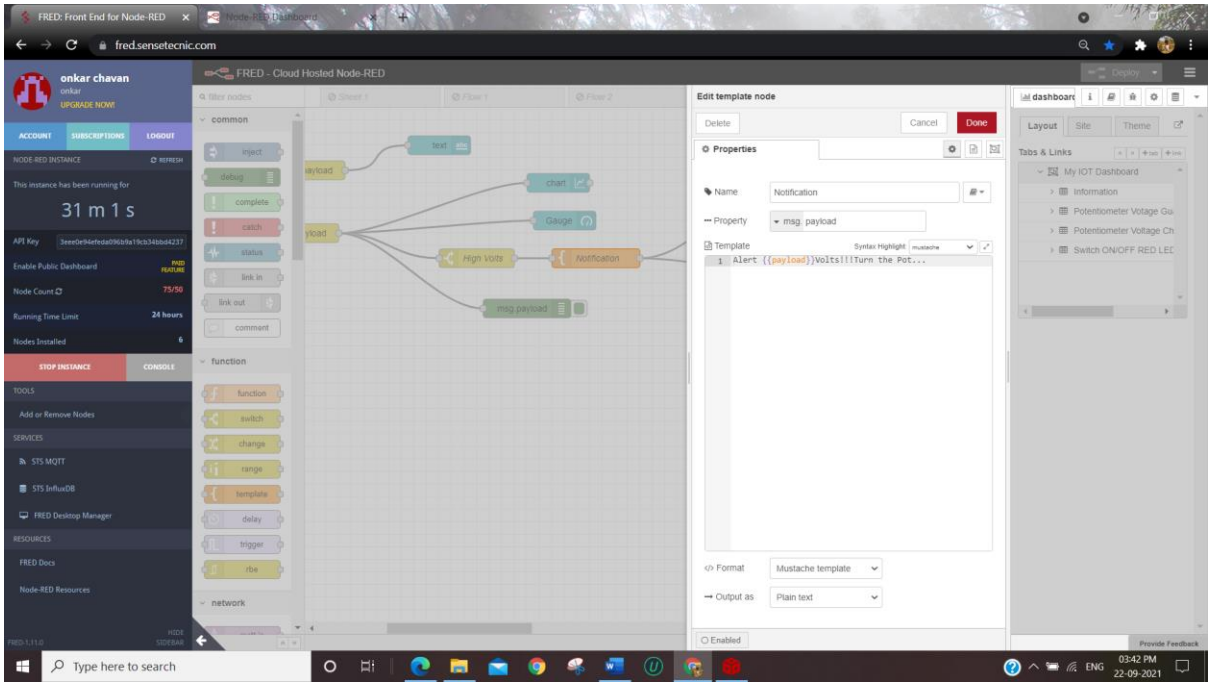
Gauge Node: -



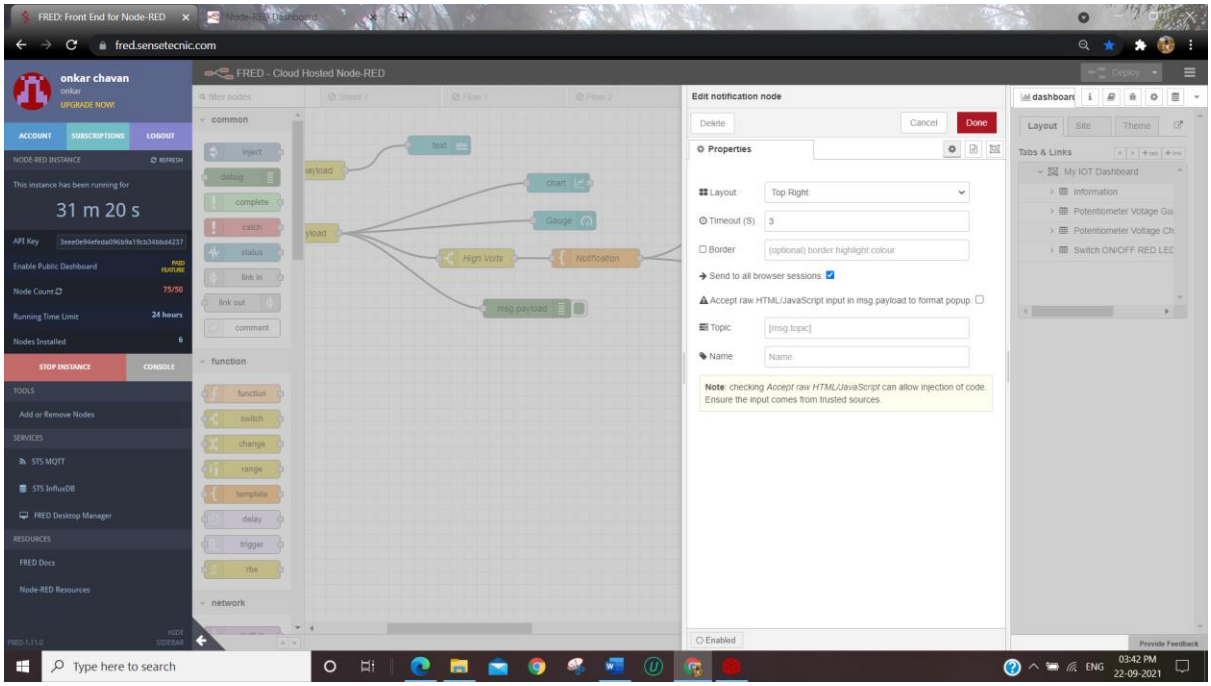
Switch Node: -



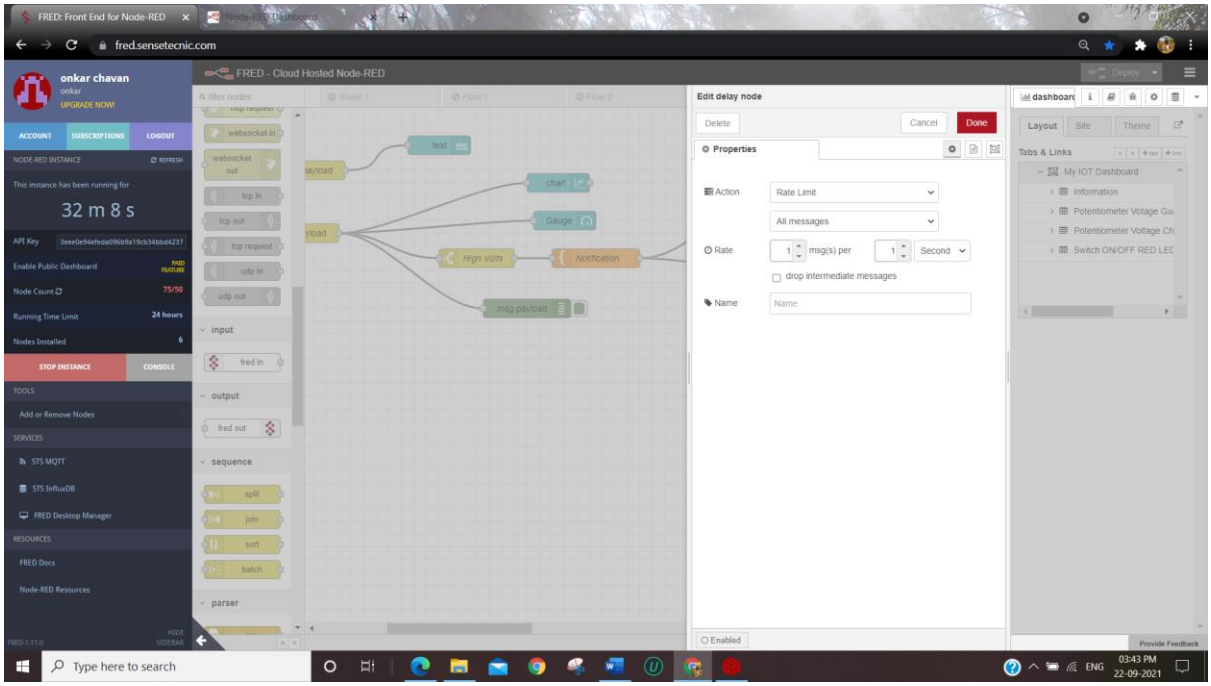
Notification Node: -



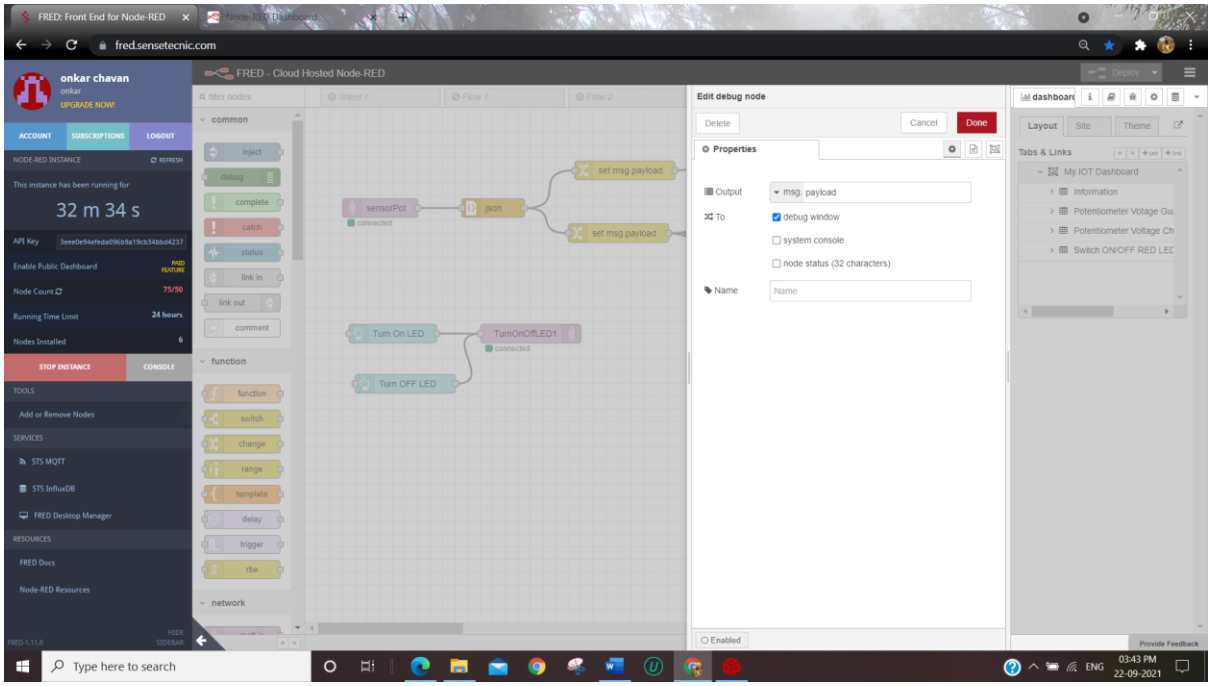
Show Notification Node: -



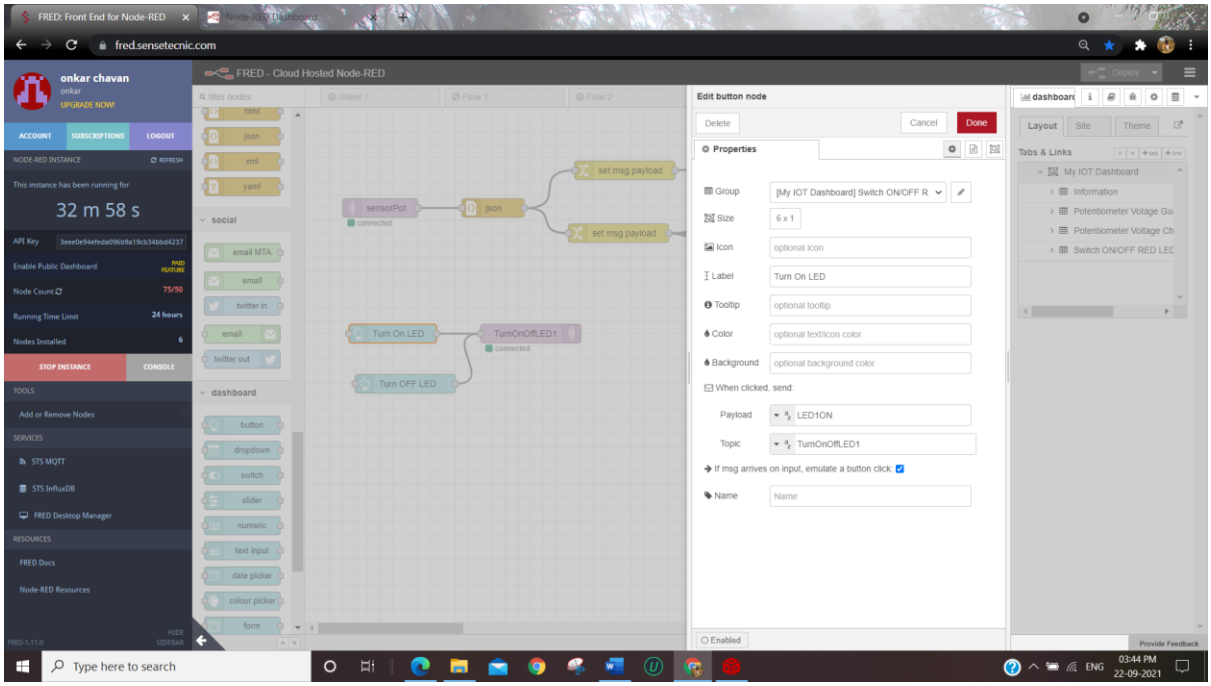
Delay Node: -



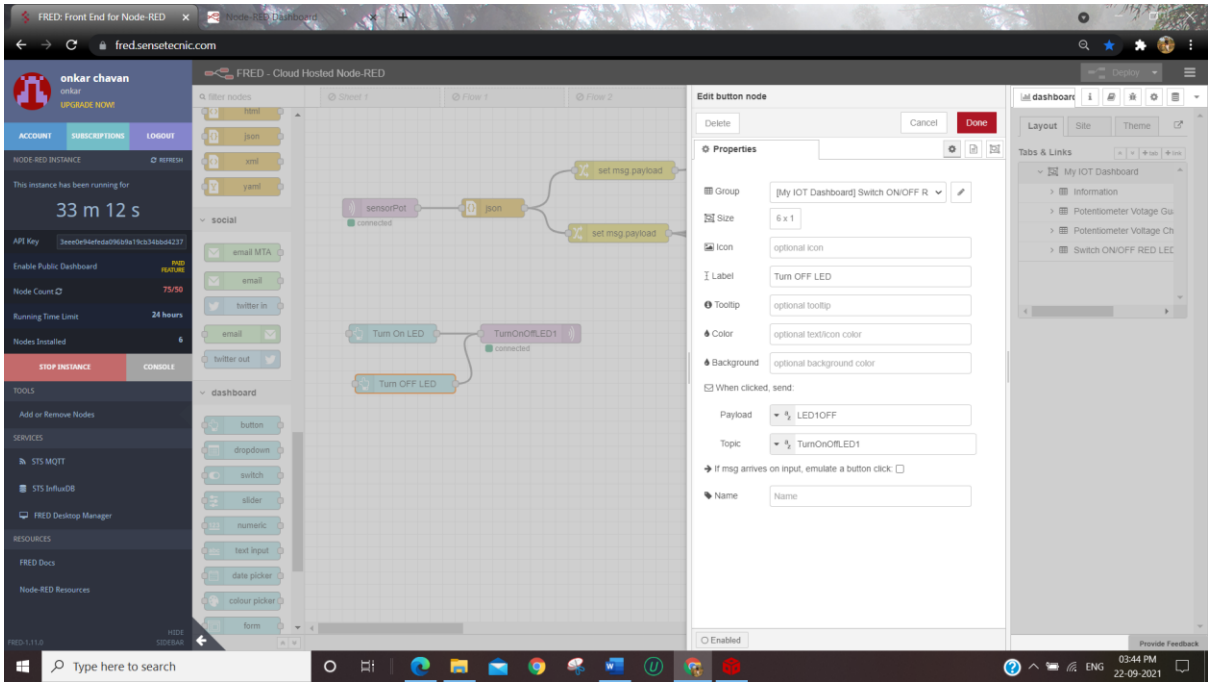
Debug Node: -



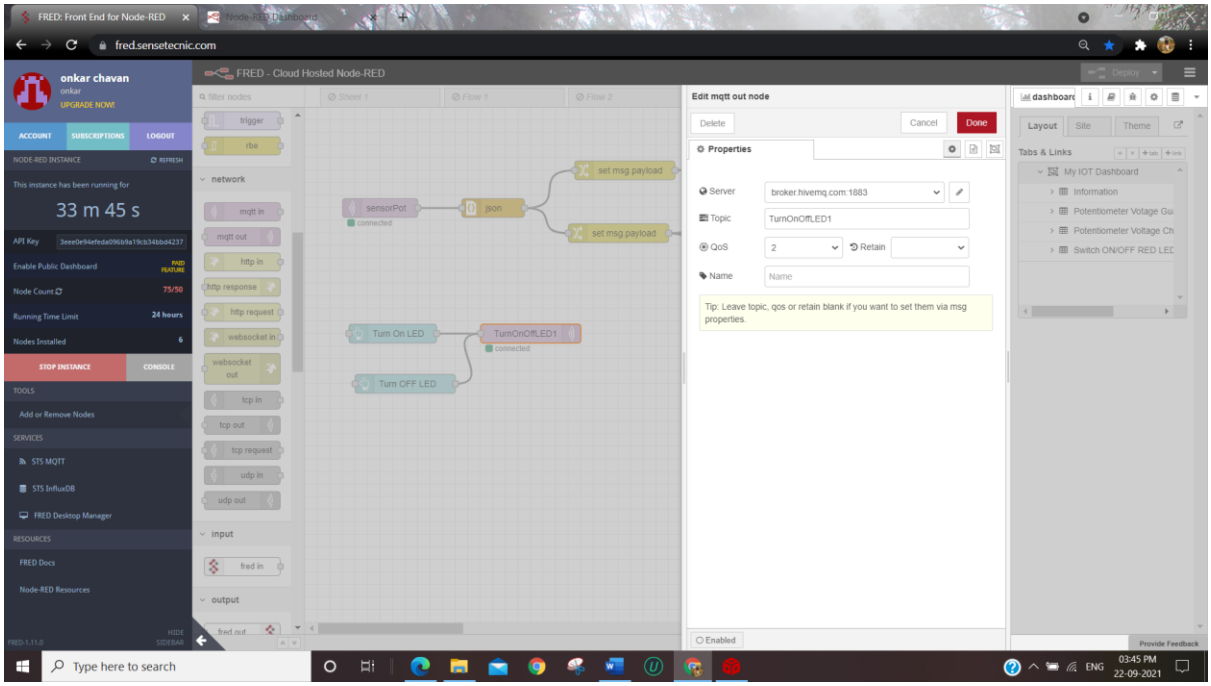
Button Node: -



Button Node: -



Mqtt out Node: -



2) Explain the setup of the various subsections of the firmware code on MSP432 .Start with WiFi configuration followed by ADC configuration ,followed by JSON data and MQTT protocol used for communication.

Code:-

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

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

char ssid[]="octanauts";
char password[]="";

WiFiClient wclient;

//byte server[] = {198,41,30,241};
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,password)!=WL_CONNECTED)
  {
    Serial.print(".");
    delay(1);
    }
    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 Points in range: ");
  Serial.println(networkId);
  for(int i=1;i<=networkId;i++)
  {
    Serial.print("Number 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:");
```

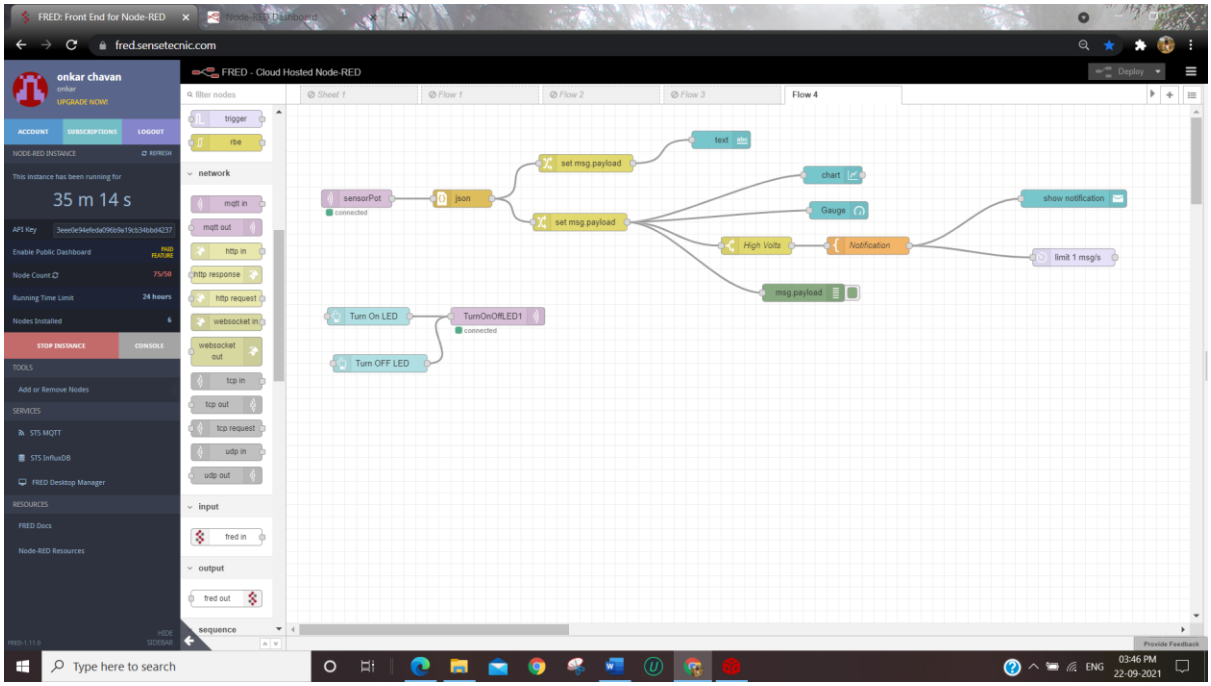
```
    Serial.println(gatewayIP);


    WiFi.macAddress(macAddr);
    Serial.print("Mac Address of shield: ");
    for(int i=5;i>-1;i++ )
    {
     Serial.print(macAddr[i],HEX);
     if(i>0)
     Serial.print(':');
     else
     Serial.println();
    }
    GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0);
    GPIO_setAsOutputPin(GPIO_PORT_P2,GPIO_PIN0|GPIO_PIN1|GPIO_PIN2);

GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P4,GPIO_PIN7,GPIO_TERTIARY_MODUL
E_FUNCTION);

ADC14_initModule(ADC_CLOCKSOURCE_MCLK,ADC_PREDIVIDER_1,ADC_PREDIVIDER_1,ADC_NORO
UTE);
    ADC14_configureSingleSampleMode(ADC_MEM6,true);

ADC14_configureConversionMemory(ADC_MEM6,ADC_VREFPOS_AVCC_VREFNEG_VSS,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();
 }
```

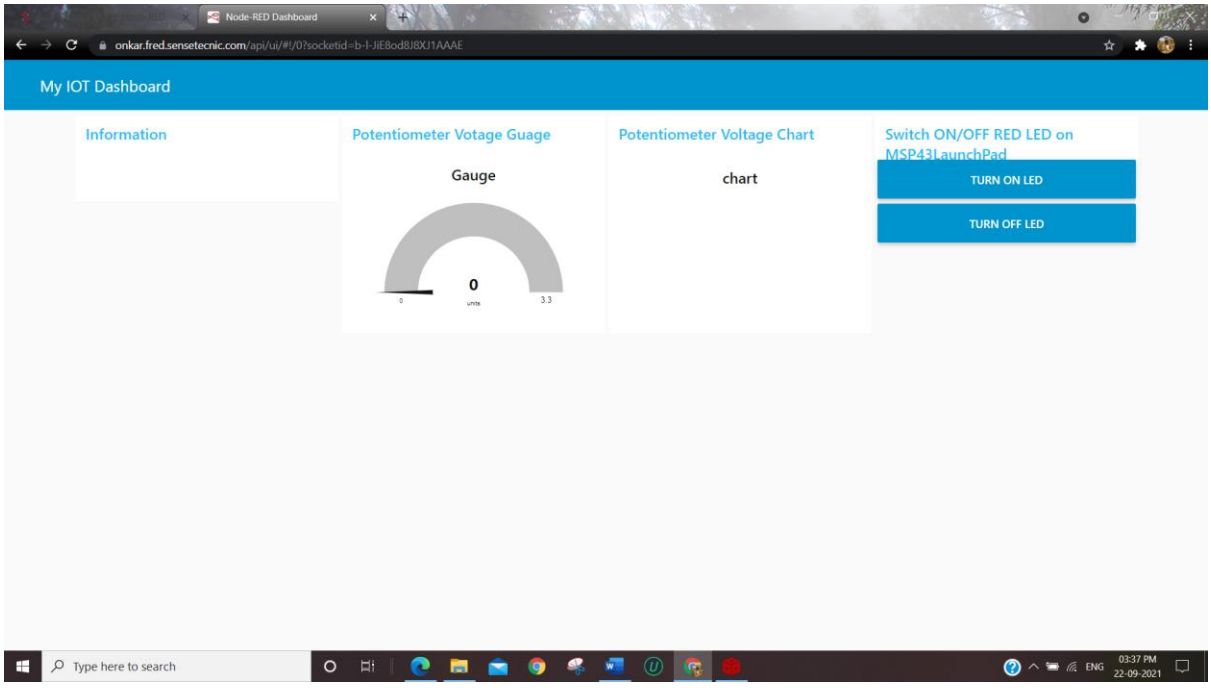
```
void loop() {
 // put your main code here, to run repeatedly:
 if(!client.connected())
 {
  Serial.println("Disconnected:Reconnection");
  if(!client.connect("Msp432"))
  {
   Serial.println("Connection Failed");
   }
   else
   {
    Serial.println("Connection Success");
    if(client.subscribe("TurnOnOffLED1"))
    {
     Serial.println("Subscription Successful");
     }
    }
  }
  int result,regressedData1;
  float regressedData;
  while(!ADC14_isBusy());
  result=ADC14_getResult(ADC_MEM6);
  ADC14_toggleConversionTrigger();
  regressedData = ((result*0.866848575)+4.999033)/1000;
  Serial.println(regressedData);
  regressedData1 = ((result*0.866848575)+4.999033);
  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';
aJson.deleteItem(d);
aJson.deleteItem(root);
```

```
 client.publish("sensorPot",jsonPayload);
 client.poll();
 delay(1000);



}


void callback(char* inTopic,byte* payload,unsigned int length)
{
 Serial.print("Messsage 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);
}
```

3) Put the screen shot of the final dashboard and explain each section of the dashboard


FRED: Front End for Node-RED
fred.sensetecnic.com
FRED - Cloud Hosted Node-RED
onkar chavan
35 m 14 s
sensorPot
json
set msg.payload
text
chart
Gauge
show notification
High Volts
Notification
limit 1 msg/s
msg.payload
Turn On LED
TurnOnOffLED1
Turn OFF LED
Type here to search
03:46 PM
22-09-2021


Node-RED Dashboard
onkar.fred.sensetecnic.com/api/ui/#!/0?socketid=b-I-JiE8odBJ8XJ1AAAE
My IOT Dashboard
Information
Potentiometer Votage Guage
Gauge
0
Potentiometer Voltage Chart
chart
Switch ON/OFF RED LED on MSP43LaunchPad
TURN ON LED
TURN OFF LED
Type here to search
03:37 PM
22-09-2021