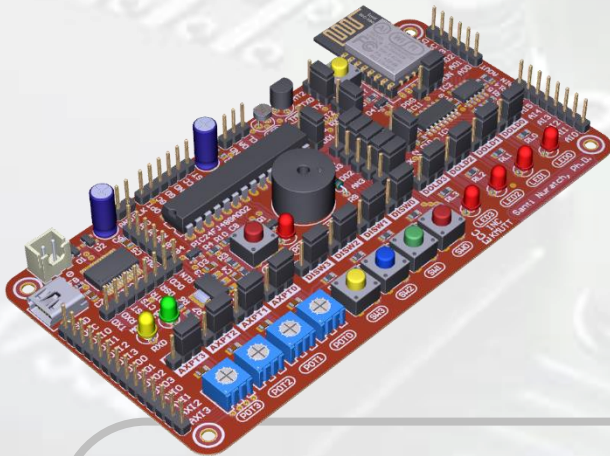


Day #2, Section #1

Node-RED Programming

System Setup & IoT Applications



Santi Nuratch., Ph.D.

Embedded Computing and Control Lab. @ INC-KMUTT

santi.inc.kmutt@gmail.com, santi.nur@kmutt.ac.th

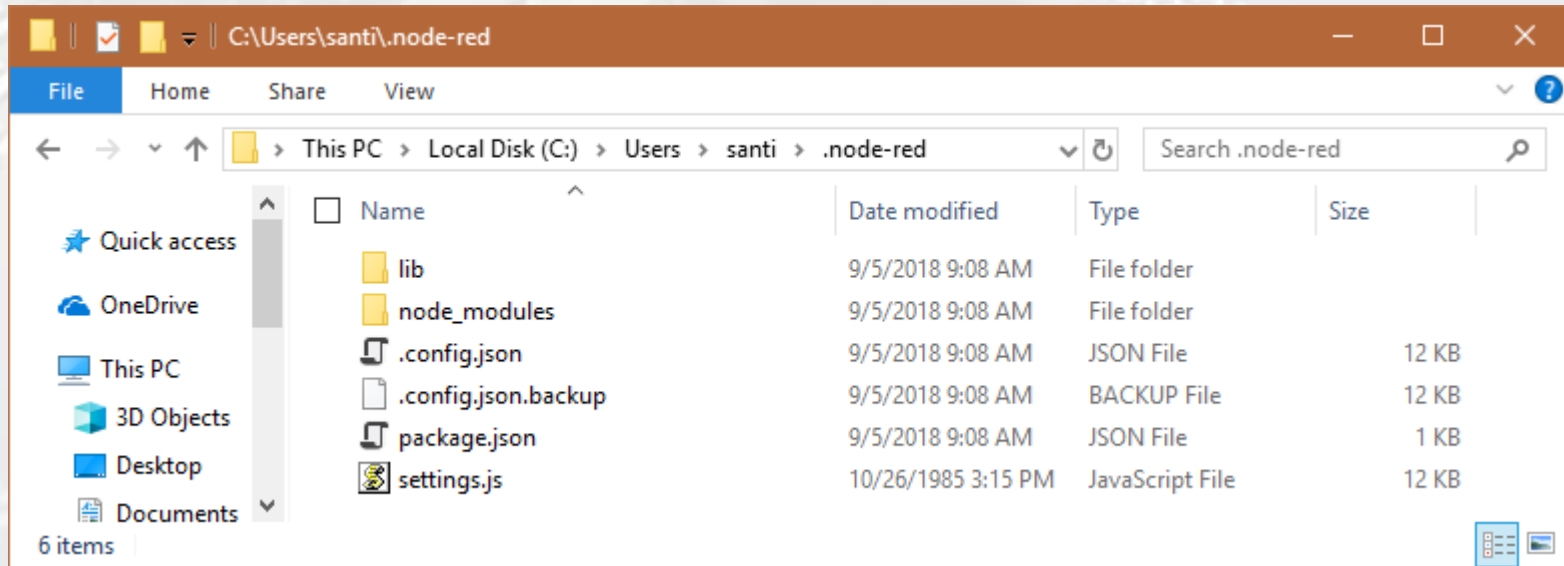
Department of Control System and Instrumentation Engineering,
King Mongkut's University of Technology Thonburi, **KMUTT**

Node-RED & ecc-iot-node-red Installation

Install ecc-iot-node-red Modules

01 : Be sure the **Node.js** and **Node-RED** were installed in your computer

02 : Go to the **.node-red** directory and check pre-installed files and folders



03 : Run the command-line, go to the **.node-red** directory and enter the command **npm i ecc-iot-node-red**

If no internet connection, copy the provided **node_modules** and replace in the **.node-red**

```
santi@DESKTOP-HP3ERRU MINGW64 ~/.node-red
$ npm i ecc-iot-node-red

> serialport@6.2.2 install C:\Users\santi\.node-red\node_modules\serialport
> prebuild-install || node-gyp rebuild

npm notice created a lockfile as package-lock.json. You should commit this file.
+ ecc-iot-node-red@0.0.3
added 253 packages from 122 contributors in 18.764s
```

Install **ecc-iot-node-red** Modules

04 : In the command line, enter the command **node-red** to run the Node-RED

```
C:\Users\santi>node-red
18 Sep 14:33:55 - [info]

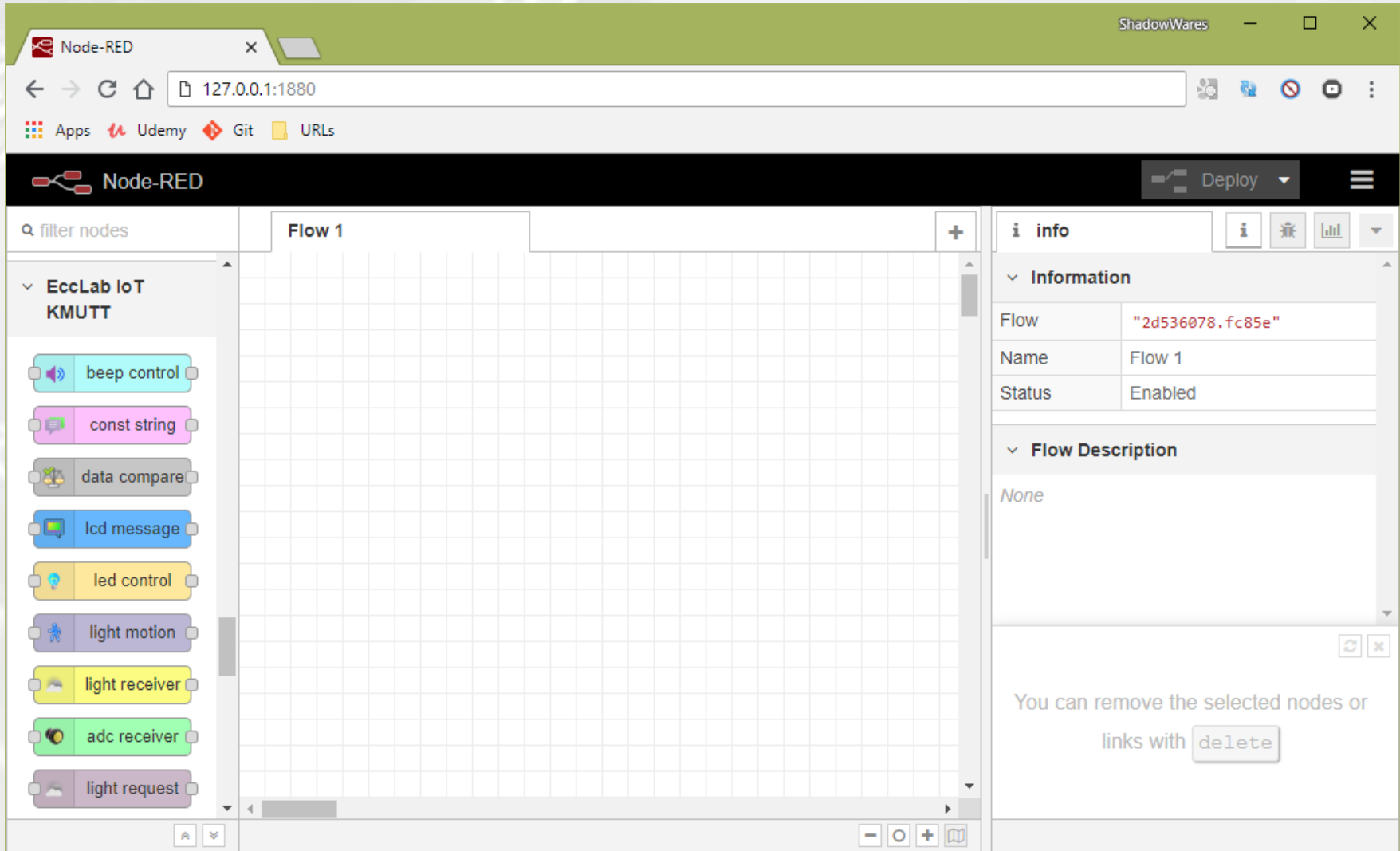
Welcome to Node-RED
=====

18 Sep 14:33:55 - [info] Node-RED version: v0.19.2
18 Sep 14:33:55 - [info] Node.js version: v10.10.0
18 Sep 14:33:55 - [info] Windows_NT 10.0.17134 x64 LE
18 Sep 14:33:56 - [info] Loading palette nodes
18 Sep 14:33:56 - [info] Dashboard version 2.9.8 started at /ui
18 Sep 14:33:56 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
18 Sep 14:33:56 - [warn] -----
18 Sep 14:33:56 - [warn] [node-red/tail] Not currently supported on Windows.
18 Sep 14:33:56 - [warn] -----
18 Sep 14:33:56 - [info] Settings file : \Users\santi\.node-red\settings.js
18 Sep 14:33:56 - [info] Context store : 'default' [module=memory]
18 Sep 14:33:56 - [info] User directory : \Users\santi\.node-red
18 Sep 14:33:56 - [warn] Projects disabled : editorTheme.projects.enabled=false
18 Sep 14:33:56 - [info] Flows file : \Users\santi\.node-red\flows_DESKTOP-HP3ERRU.json
18 Sep 14:33:56 - [info] Creating new flow file
18 Sep 14:33:57 - [warn] -----

18 Sep 14:33:57 - [info] Starting flows
18 Sep 14:33:57 - [info] Started flows
18 Sep 14:33:57 - [info] Server now running at http://127.0.0.1:1880/
```

Install ecc-iot-node-red Modules

05 : Open a Browser (e.g. Google Chrome) and give it the url, **127.0.0.1:1880**



The screenshot shows the Node-RED web interface in a browser window. The address bar displays `127.0.0.1:1880`. The interface includes a left sidebar with a search bar and a list of modules under the heading "EccLab IoT KMUTT". The central workspace shows "Flow 1" on a grid. The right sidebar contains an "info" tab with "Information" and "Flow Description" sections.

Information

Flow	"2d536078.fc85e"
Name	Flow 1
Status	Enabled

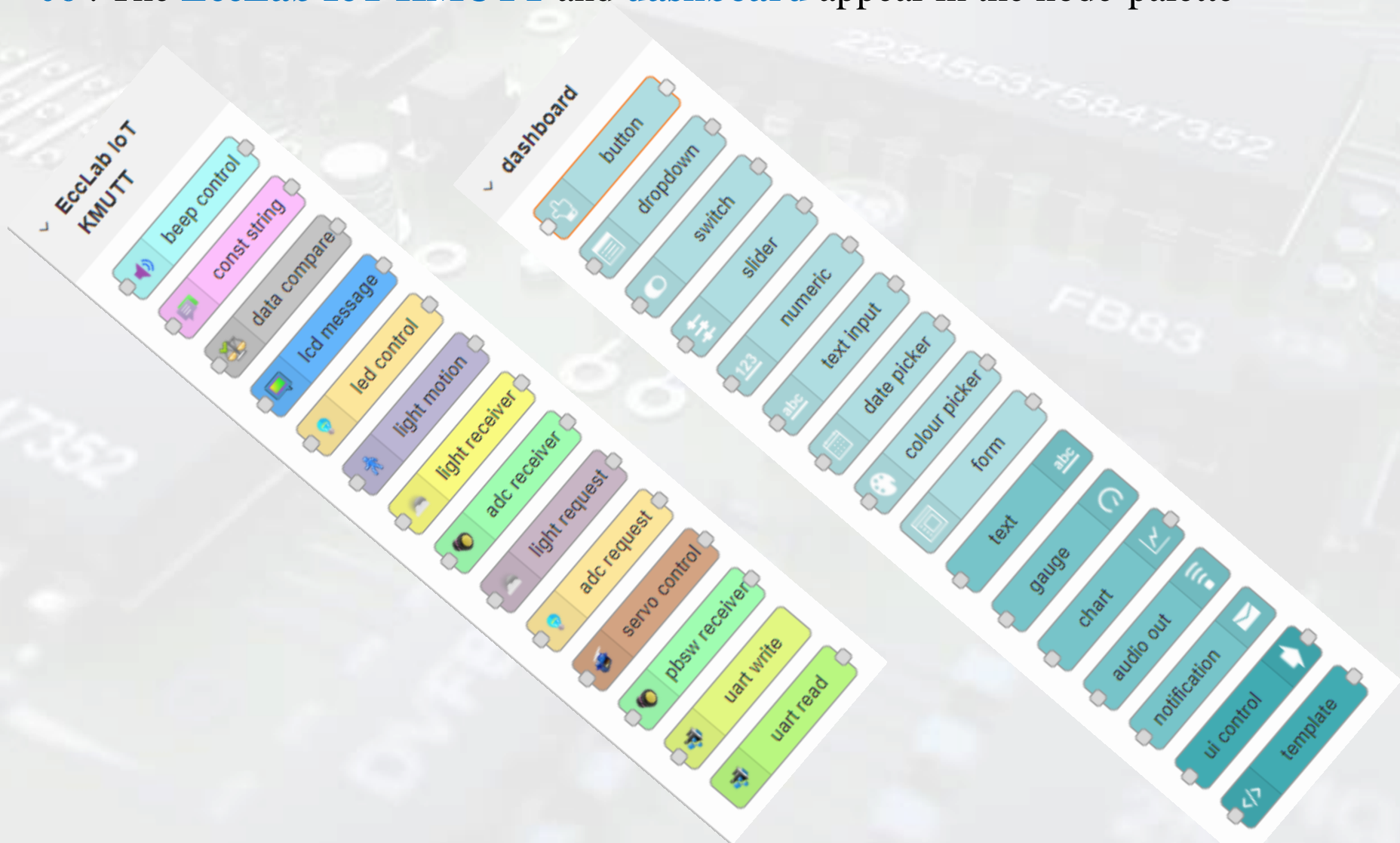
Flow Description

None

You can remove the selected nodes or links with delete

Install ecc-iot-node-red Modules

06 : The **EccLab IoT KMUTT** and **dashboard** appear in the node-palette

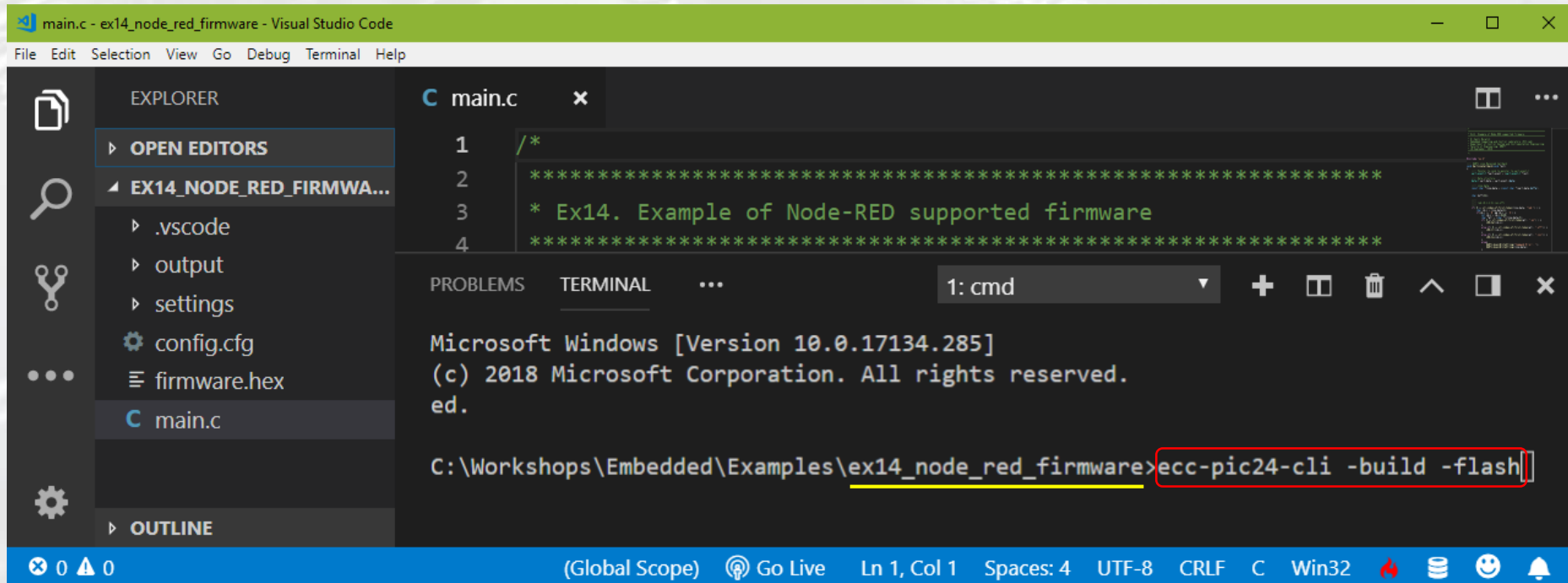


Flash the `node_red_firmware`

Embedded Node-RED Firmware

Flash the `node_red_firmware`

01 : Build and Flash the `ex14_node_red_firmware` using the command `ecc-pic24-cli -build -flash`



The screenshot shows the Visual Studio Code editor with the file `main.c` open. The Explorer sidebar on the left shows the project structure, including `EX14_NODE_RED_FIRMWA...`, `.vscode`, `output`, `settings`, `config.cfg`, `firmware.hex`, and `main.c`. The main editor area shows the `main.c` file with the following content:

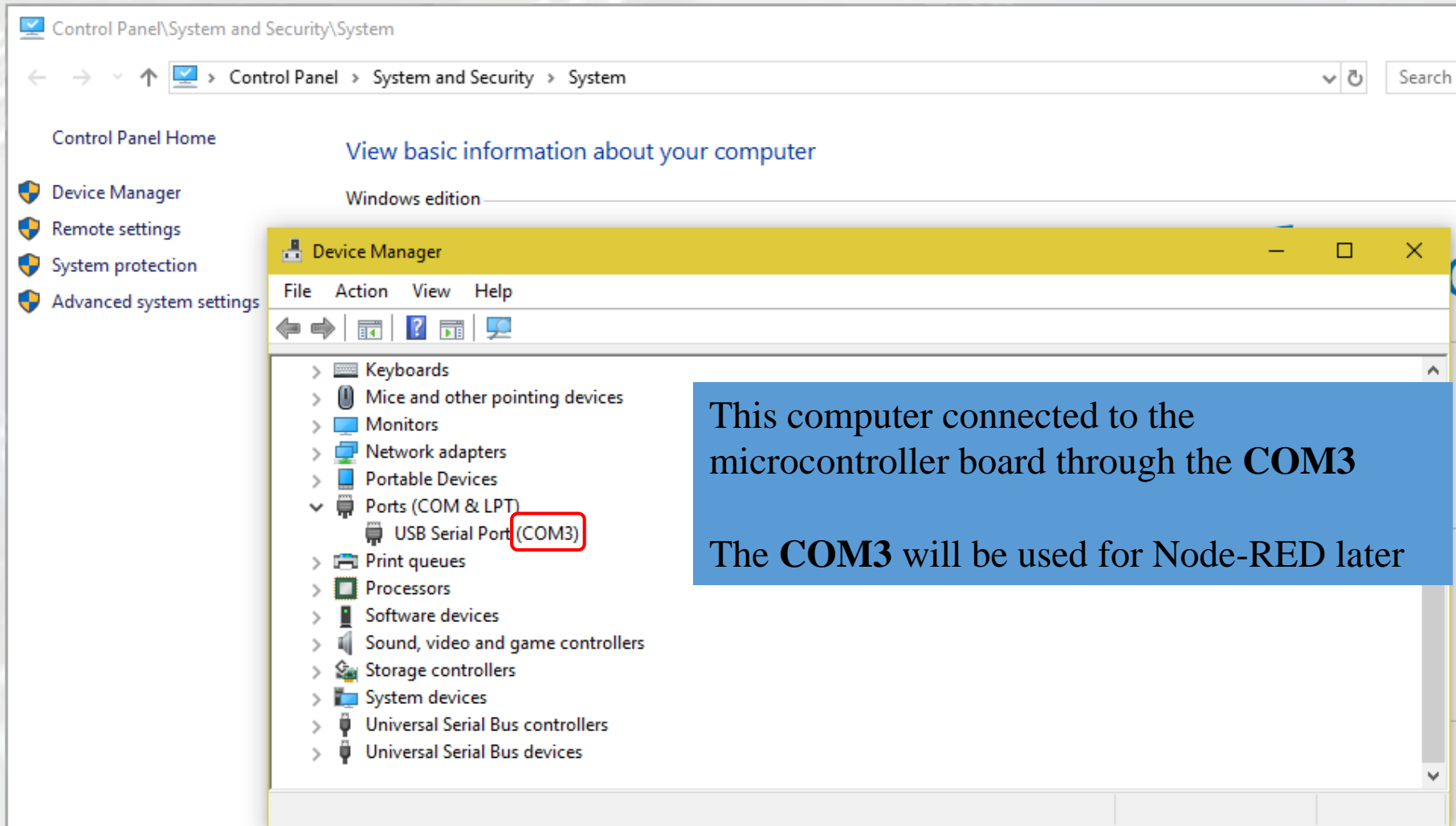
```
1 /*
2 *****
3 * Ex14. Example of Node-RED supported firmware
4 *****
```

The TERMINAL panel at the bottom shows the command prompt with the command `ecc-pic24-cli -build -flash` entered. The command is highlighted with a red box.

```
Programming flash memory: 88.89 %
Programming flash memory: 94.44 %
Programming flash memory: 100.00 %
Firmware has been updated [OK, 5.60 seconds]
The microcontroller is now running...
```


Flash the `node_red_firmware`

02 : Check the communication port, COM port, using the **Device Manager**



The screenshot shows the Windows Control Panel window open to the 'System' section. The 'Device Manager' link is selected in the left-hand navigation pane. The Device Manager window is open, displaying a list of hardware categories. Under the 'Ports (COM & LPT)' category, the 'USB Serial Port (COM3)' is listed and highlighted with a red rectangular box. To the right of the Device Manager window, there is a blue text box with white text.

This computer connected to the microcontroller board through the **COM3**

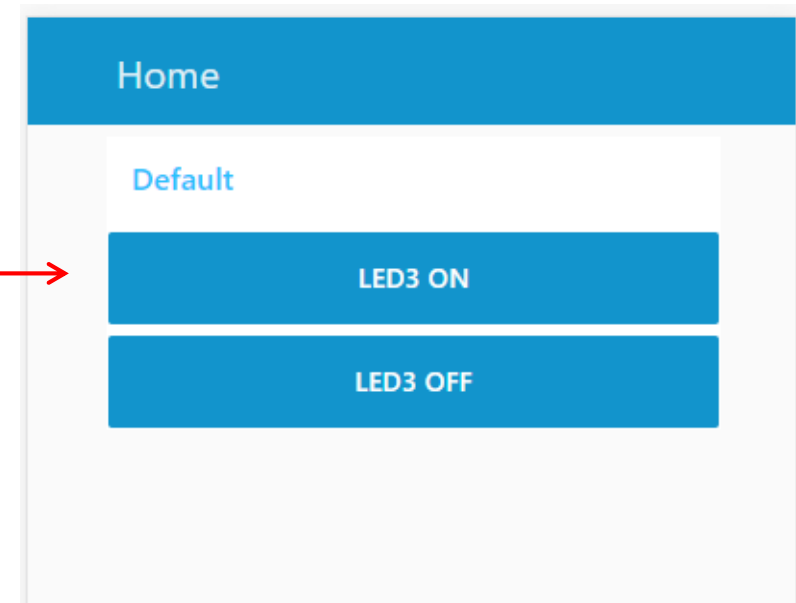
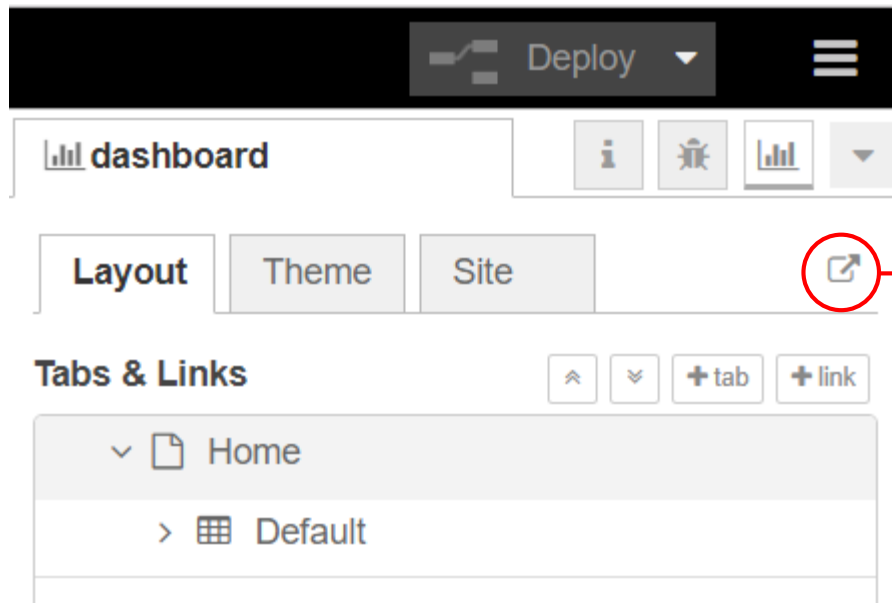
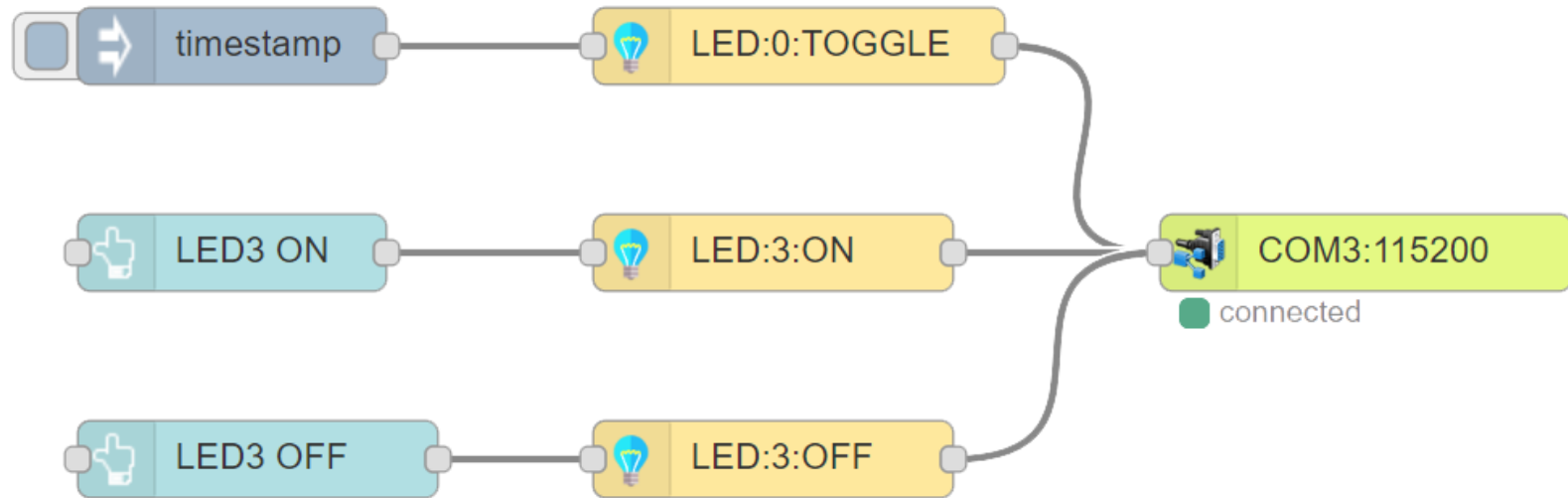
The **COM3** will be used for Node-RED later

The IoT Applications

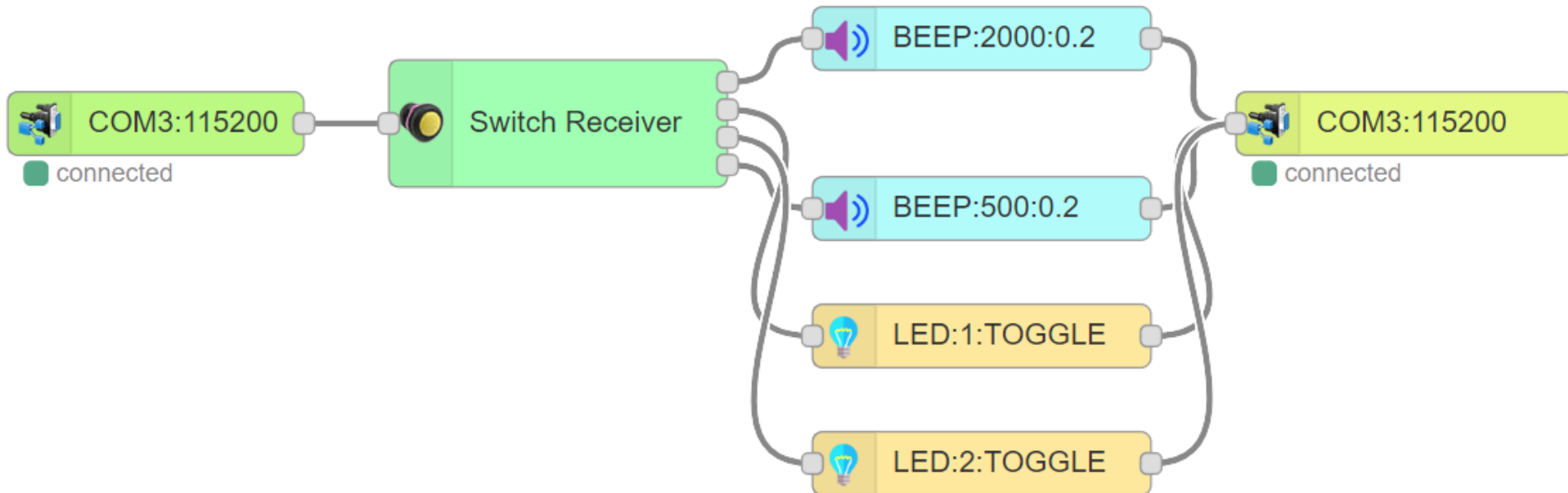
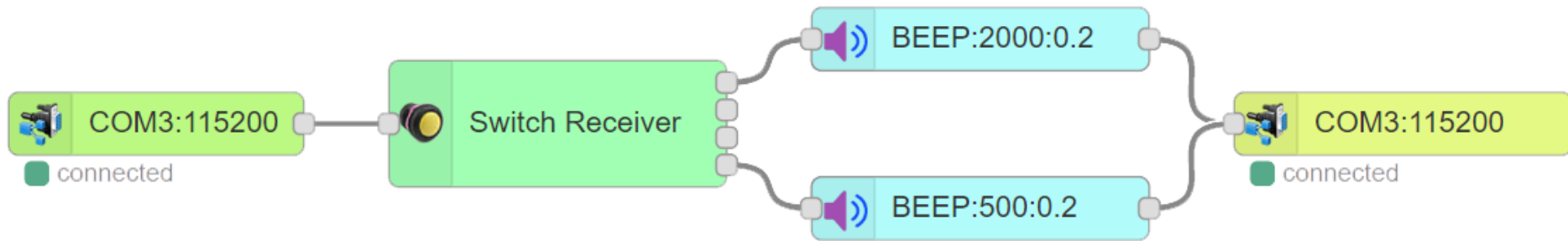
Node-RED & ecc-iot-node-red

The IoT Applications

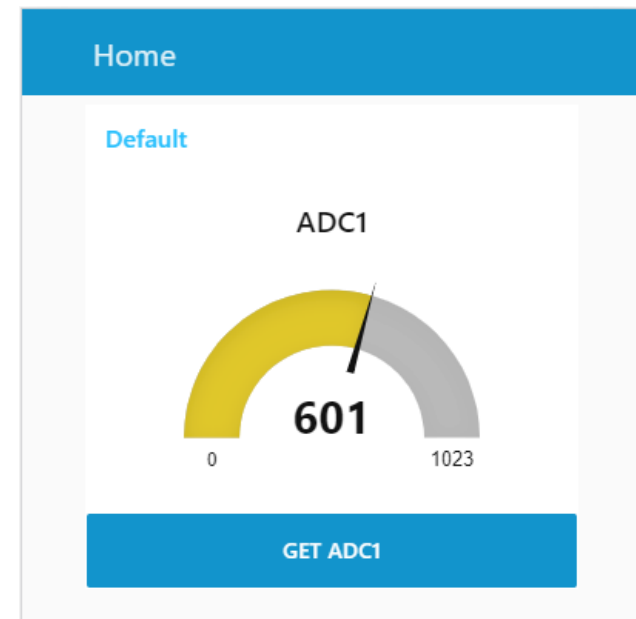
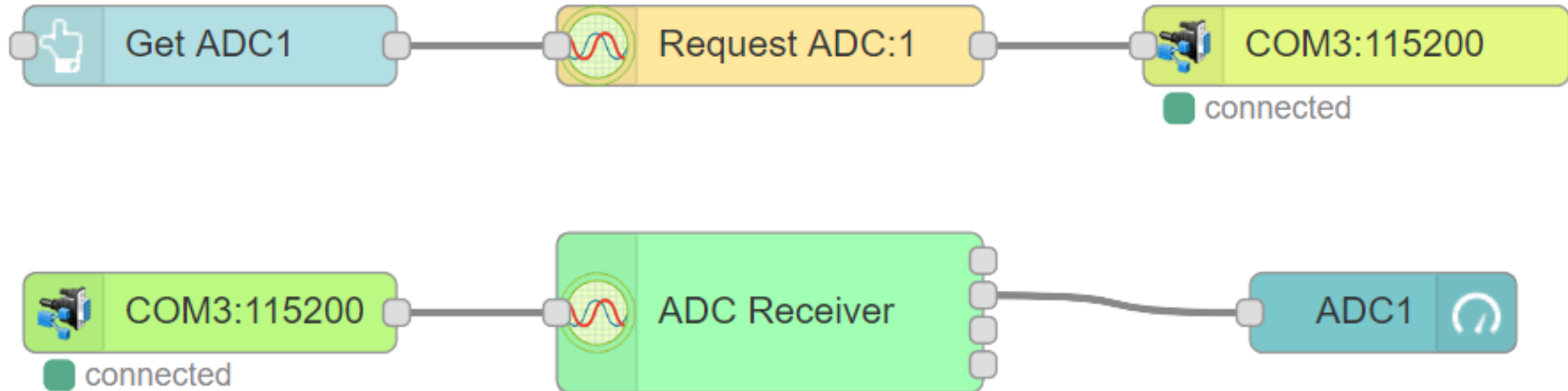
LEDs Control Application



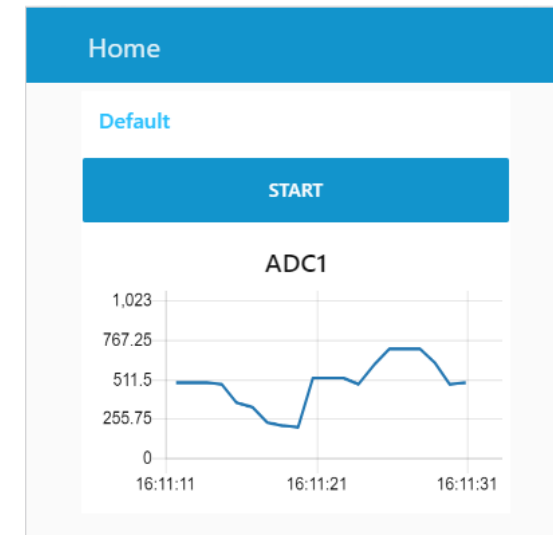
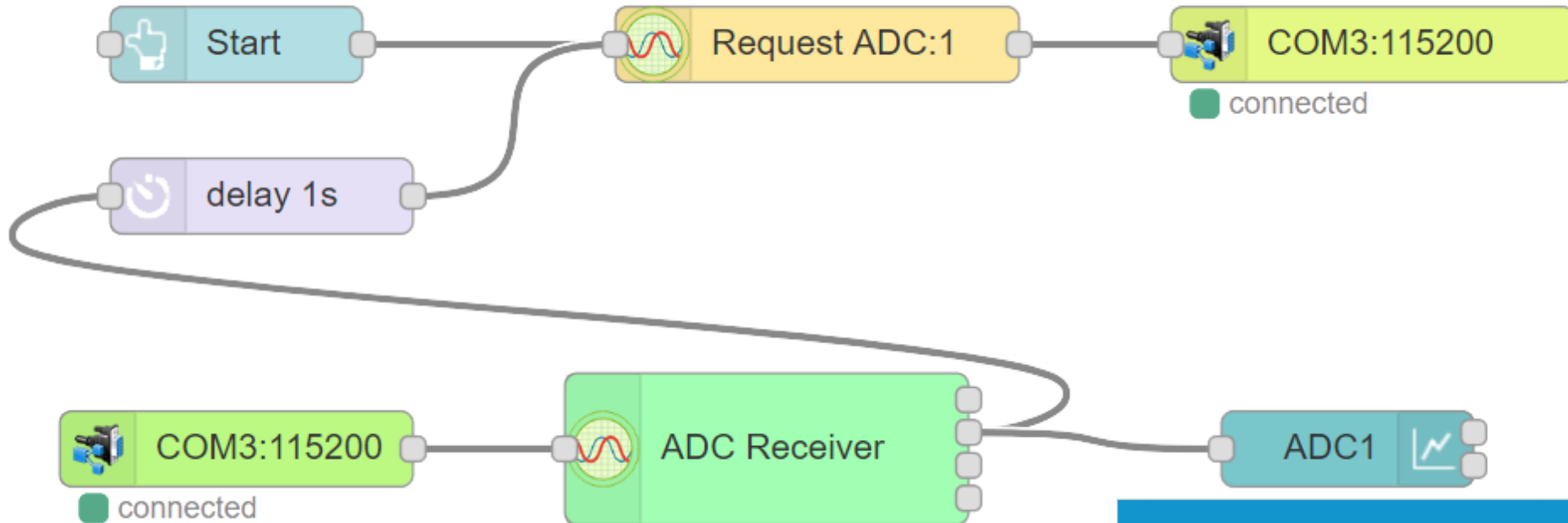
Switches Status Checking & Beep



ADC/Voltage Reading



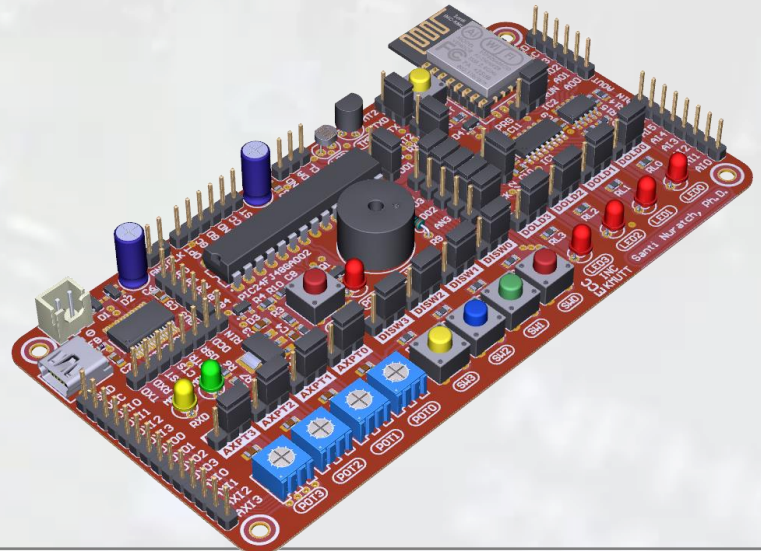
Light Intensity Visualization



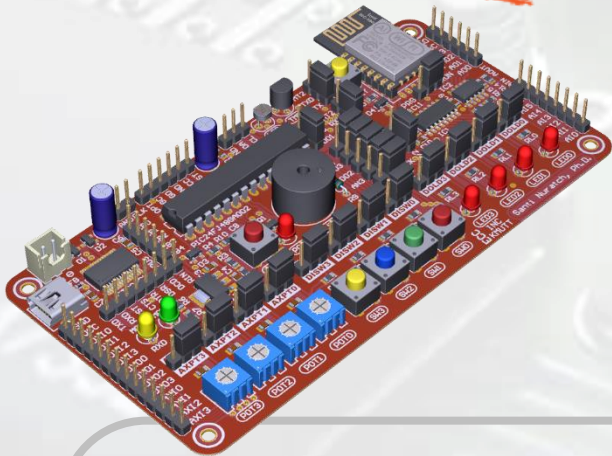
Let's Coding...

the MORE
YOU PRACTICE
THE BETTER
YOU GET

```
543 int main(void)
544 {
545     OS_Init();
546     AT_Init();
547     ESP_Init();
548     WiFi_Init();
549     Internet_Init();
550     OS_TimerCreate("AT_Service", 100, 1, AT_Service);
551     OS_WorkerCreate("WiFi_Init", Worker_ESPInitialise);
552     OS_Uart2SetLineReceivedCallback(ESP_LineReceived);
553     UART1_AsyncWriteString("\r\nMQTT Client...\r\n");
554     Beep(100);
555     OS_Start();
556 }
557
```



THANK YOU!



Santi Nuratch., Ph.D.

Embedded Computing and Control Lab. @ INC-KMUTT

santi.inc.kmutt@gmail.com, santi.nur@kmutt.ac.th

Department of Control System and Instrumentation Engineering,
King Mongkut's University of Technology Thonburi, **KMUTT**