WPM SDK manual

Version 04182018.

Cr. Chatpeth

# Introduction

## Download source code

Clone SDK related files from <https://github.com/chatpeth/WPM2018.git>

The SDK files are contain following folder in Fig. a

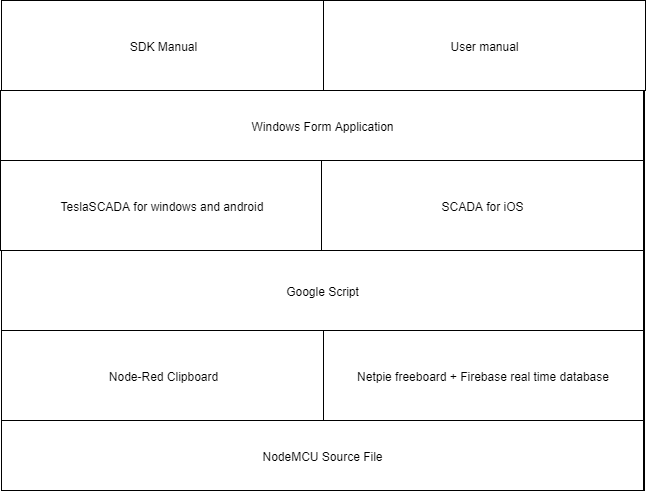


Fig.a

# MQTT protocols

## Get start with mqtt protocols

Go to <http://mqtt.org/> website to learn more about mqtt protocols.

## Mqtt library for ESP8266 with Arduino IDE

1. Download and install Arduino IDE 🡪 <https://www.arduino.cc/en/main/software>

2. Go to File🡪Preferences

3.Insert <http://arduino.esp8266.com/stable/package_esp8266com_index.json> into Addition Board Manager URLs.

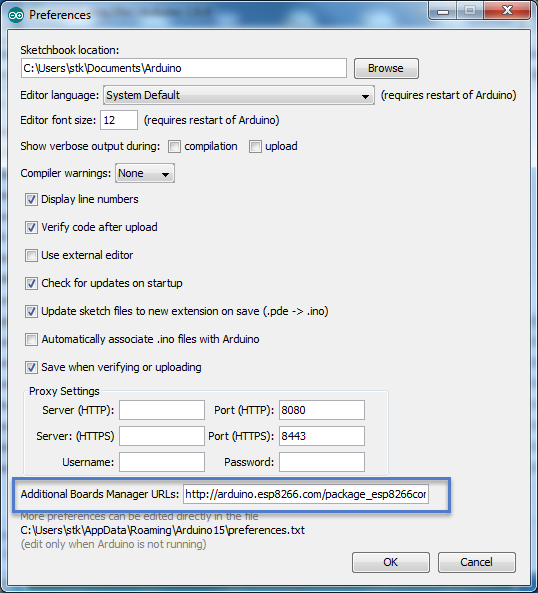


Fig.1 Install ESP8266 board

4. Click OK

5. Go to Tools🡪Board: “xxx”🡪Boards Manager

6. Select type to Contributed 🡪 ESP8266 then Install

7. When Installing complete, restart the IDE then go to Tools🡪Board🡪NodeMCU 1.0 (ESP-12E Modules)

8. Go to Sketch🡪Include library 🡪 Manage library to install PubSubClient library (mqtt library for ESP8266)

9. Open example project /sdk\_manual/samples/Basic\_mqtt

10. Build the application and upload to your NodeMCU.

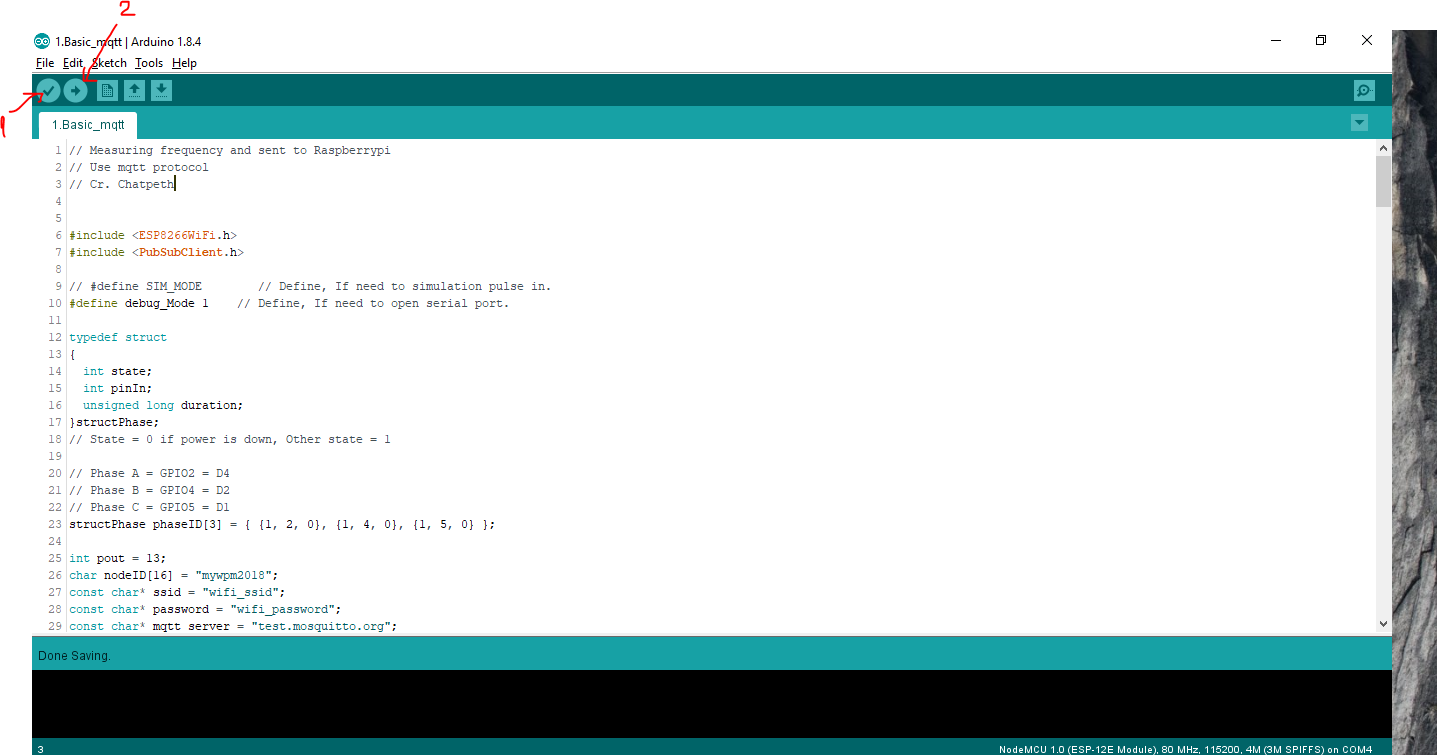


Fig.2 Build your first Application

11. Click on glass icon on the upper right corner to Open debug COM port

# Google Spread Sheet

## Install library

1. Copy all folder /Google\_script/lib to /Documents/Arduino/library

2. Open /sdk\_manual/samples/GoogleDocs

3. Build Application to check library are install correctly.

## Connect NodeMCU to Google spread sheet

1. Create Google web API by create new google spread sheet

2.Click Tools🡪Script editor, copy script from /Google\_script/script/ google\_spread\_sheet\_api.gs to the script editor

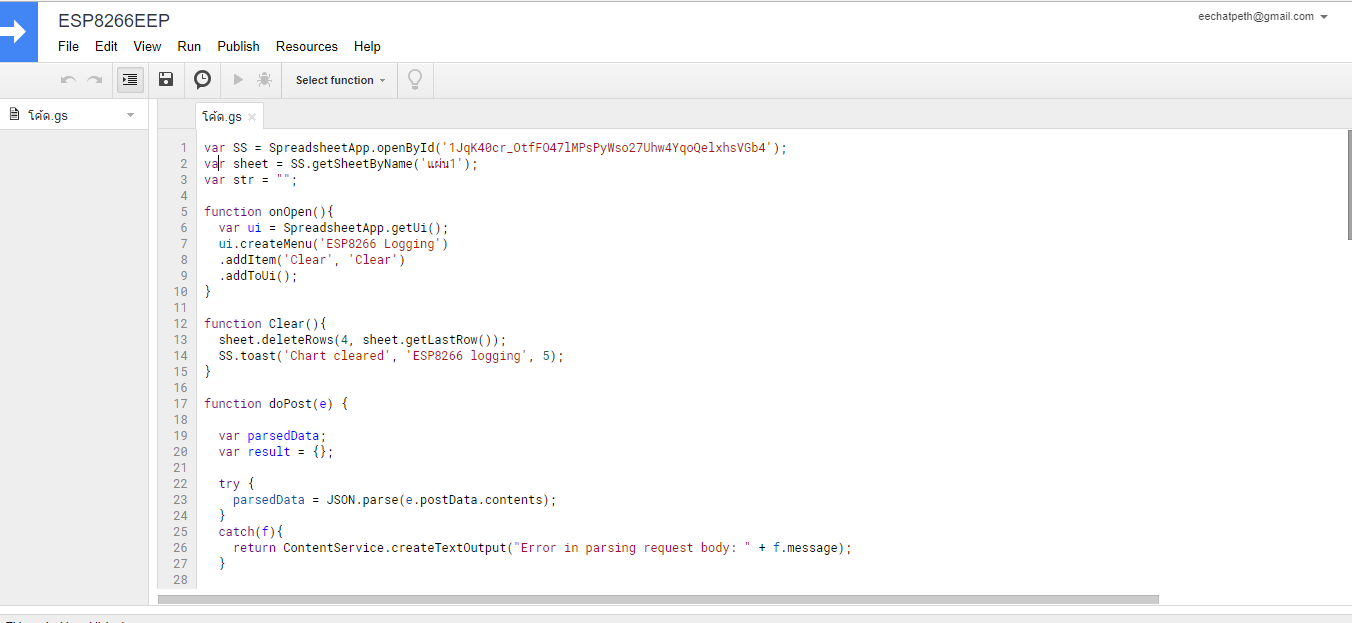


Fig.3 Create Google Web App

3. Copy your spread sheet id and put into var SS = SpreadsheetApp.openById(‘YOUR\_SHEET\_ID’)



Fig.4 Sheet ID

4. Edit var sheet = SS.getSheetByName(‘YOUR\_SHEET\_NAME’)

5. Click Publish🡪Deploy as web app (Require Authentication)

6. Allow permission and copy your web API



Fig. 5 GScripID

The highlight text is your GScriptID generated by Google.

7. Back to GoogleDocs.ino, Edit these parameter, GScriptID must receive from Step 9

//wifi

const char\* ssid = "";

const char\* password = "";

//Google script

const char\* host = "script.google.com";

const char \*GScriptId = "YOUR\_GSCRIPT\_ID";

8. Build the application and upload to NodeMCU. Open the spread sheet and observe the result.

# Firebase

## Create real time database

1. Go to <https://console.firebase.google.com> , sign in with Google account

2. Click on go to console. Click Add project

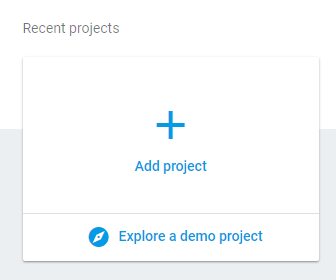


Fig.6 Create real time database

3. Give name to your project and then click on create project

4. Go to your project, select Database on left side

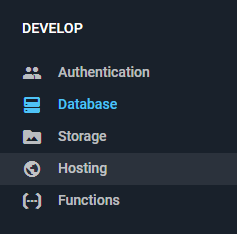


Fig. 7 Setup firebase

5. Click on Rules, provide with these basic rules (Non-Authentications)

{

"rules": {

".read": "auth == null",

".write": "auth == null"

}

}

## Import data structure

1. Import data structure by click Import JSON and direct to WPM2018/Node-Red/database/firebase.json

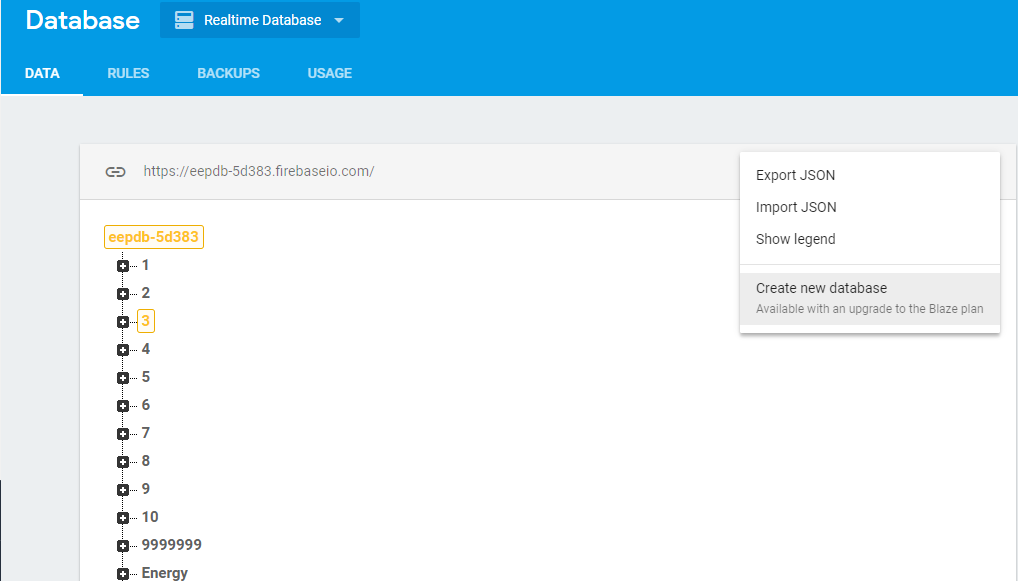


Fig.8 Import data structure

2. Learn more about firebase, go to <https://firebase.google.com/docs/?authuser=0>

# Cloud mqtt

CloudMQTT are managed Mosquitto servers in the cloud. Mosquitto implements the MQ Telemetry Transport protocol, MQTT, which provides lightweight methods of carrying out messaging using a publish/subscribe message queueing model.

publish subscribe example

## Create cloud mqtt account

1. Open <https://www.cloudmqtt.com/> to sign up your account

## Construct new broker

1. Sign in to your account. Click on Create New Instance 

2. After building your instance, you will see the account detail

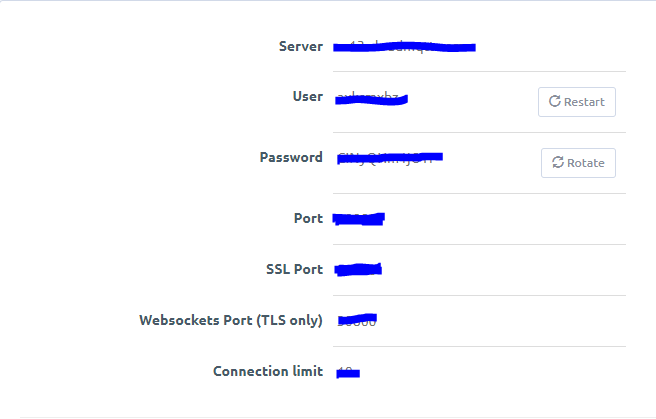


Fig.9 Cloud mqtt account

3. Back to sdk folder, Open /sdk\_manual/samples/3.Cloud\_mqtt

4. Build Application and upload to NodeMCU.

5. Back to your cloud mqtt instance, Click on Websocket UI

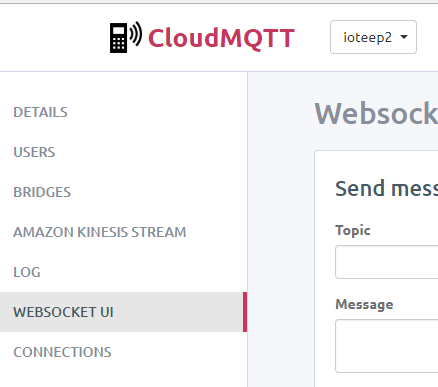


Fig.10 Websocket UI of cloudmqtt

You can see data which sent from nodemcu to the broker look like Fig.11

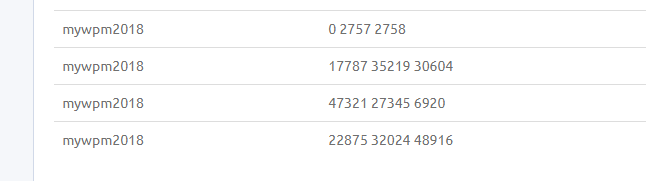


Fig.11 mqtt protocols data publishing

6. Since we subscribe for topic nodeID on line 203 in Cloud\_mqtt.ino “client.subscribe(nodeID);”

On websocket UI, Insert value of nodeID (In default code is mywpm2018), In sert message and click Send.

7. Back to debug COM port, NodeMCU was receive data from broker, because they were subscribe nodeID topic before.



Please go to <https://www.cloudmqtt.com/docs.html> , If you want to learn more about Cloud Mqtt.

# Netpie

NETPIE platform is a cloud-based platform-as-a-service that facilitates interconnecting IoT devices (“things”) together in a most seamless and transparent manner possible by pushing the complexity of connecting IoT devices from the hands of application developers or device manufacturers to the cloud.

## Create Netpie account

1. Go to [https://netpie.io/#](https://netpie.io/) then sign up new account

2. Click on Resources🡪Application🡪Create new application

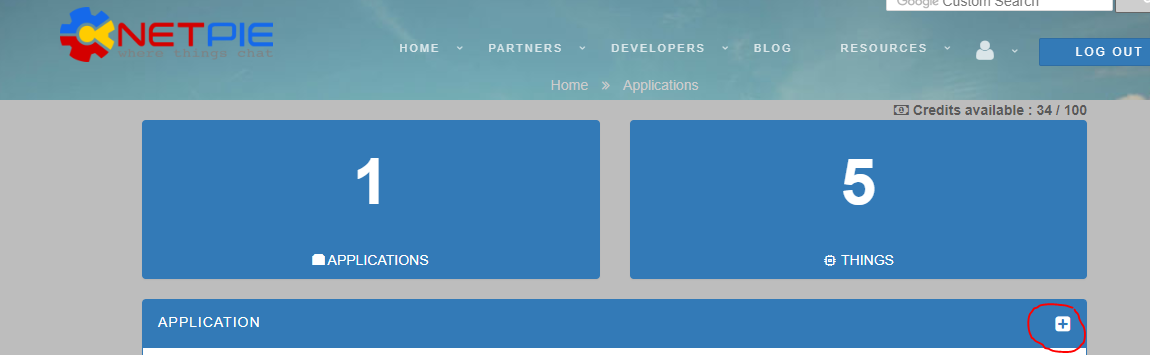


Fig.12 Create Netpie application

3. In your application, create new key then copy AppID, Key and Secret, we will use on node-red section.

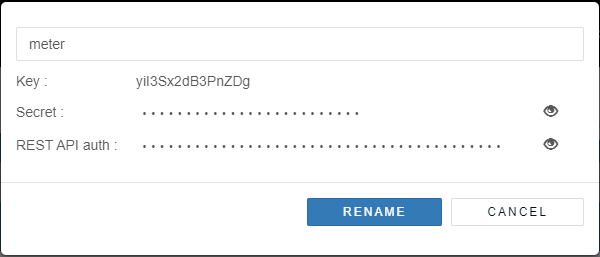


Fig.13 Netpie Authentication

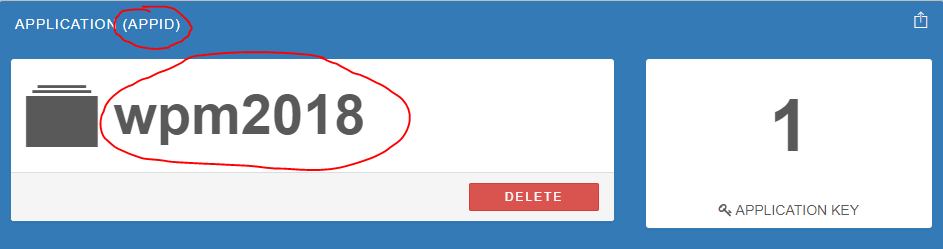


Fig.14 AppID

## Netpie feed

Netpie feed is long term data storage. You can collect data one-year age in here. Not only free Netpie provide api in many langiage programing including C#, Java, Python, Node.js, html5. However in this section we will focus on Node-Red API

1. click on resource🡪 Feed them assign variable you want to store

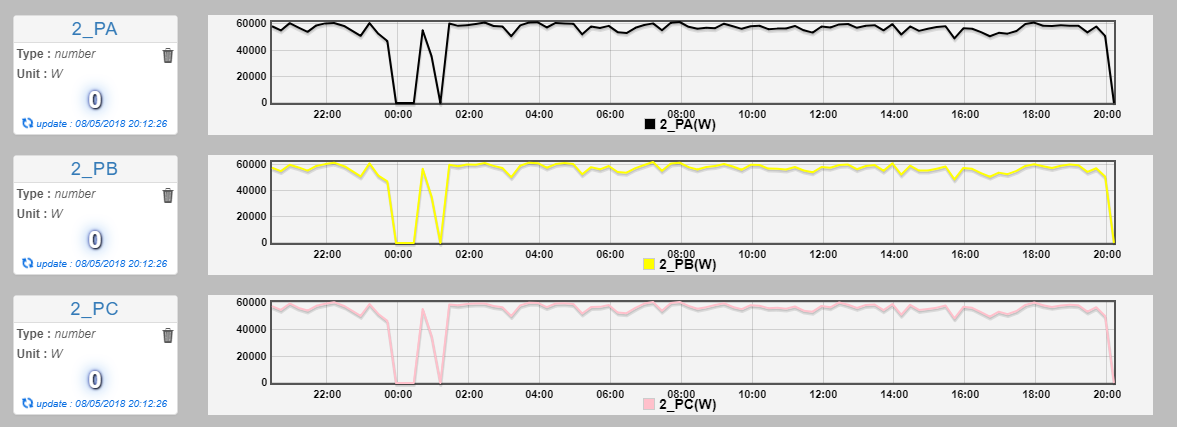


Fig.15 Netpie feed

2. Copy API key from permission tab for next section (Node-Red), set allowed application ID to allow all

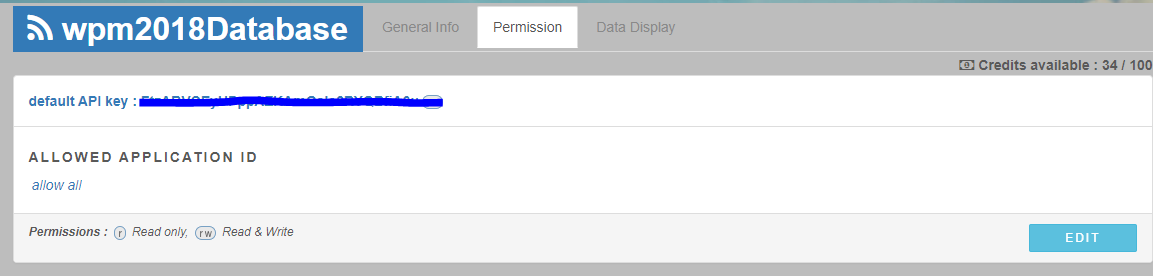


Fig.16 Feed permission

## Import Netpie freeboard

1. Click on Netpie🡪 Resource🡪Freeboard🡪 Settings🡪 Import, direct to /WPM2018/Node-Red/database/ netpie\_freeboard.json

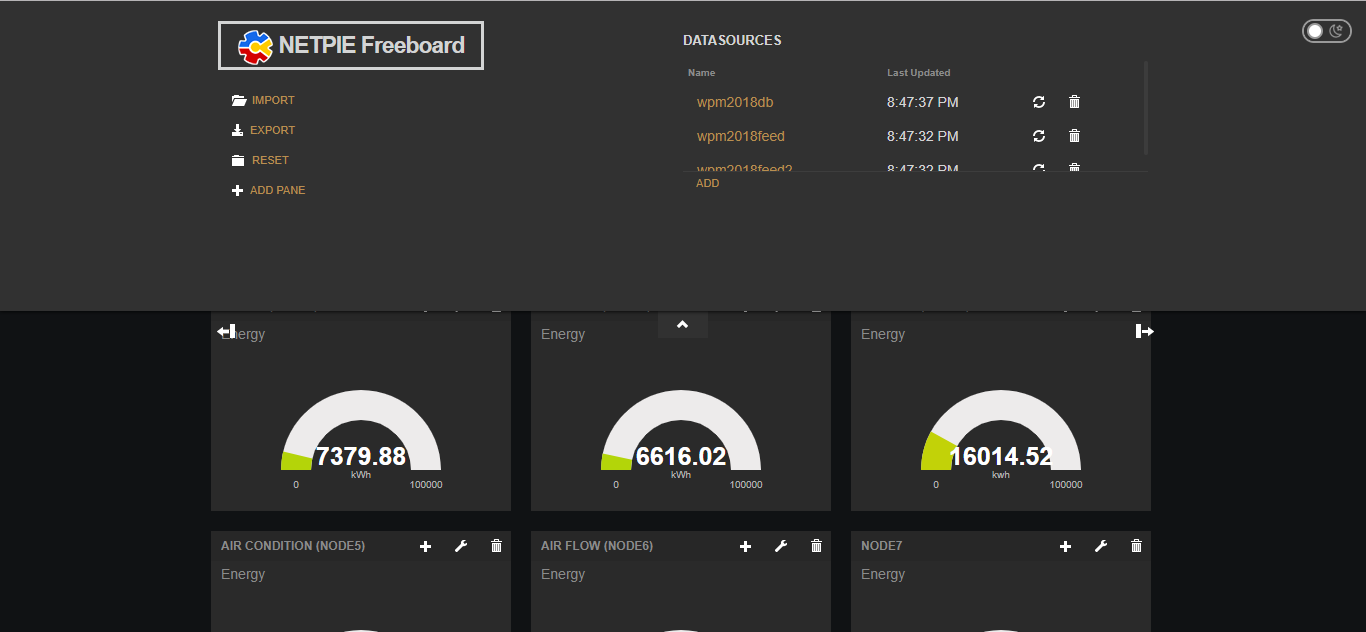


Fig.16.1 Netpie Freeboard

2. Setting data source on the upper right corner, setup parameter following your app.

# Raspberry Pi

## Get started with Raspberry pi.

1. Preparing Raspberry pi board, SD card and card reader.
2. Download Raspbian OS from
3. Install SD card writing program e.g. [win32DiskImager](https://sourceforge.net/projects/win32diskimager/) on Windows.
4. Extract Raspbian file, you will get \*.img file.
5. Run win32DiskImager, put SD card on your computer select device, \*.img file then start writing.

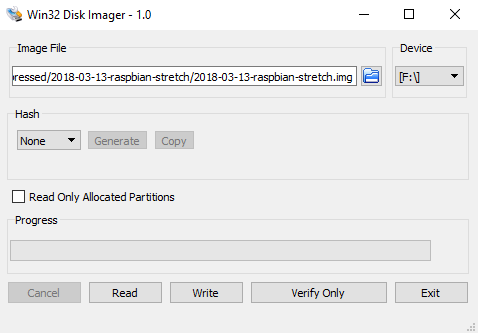


Fig.17

1. After writing success, remove the SD card, put in to raspberry pi.

You will get Raspberry Pi with Raspbian OS. If you have screen, mouse and keyboard you can use like general linux. In the case of without screen please follow step below.

1. Create new file without extension on PC, give file name as ssh (Be sure you don’t add any extention like \*.txt or any)
2. Copy the ssh file to the SD card.

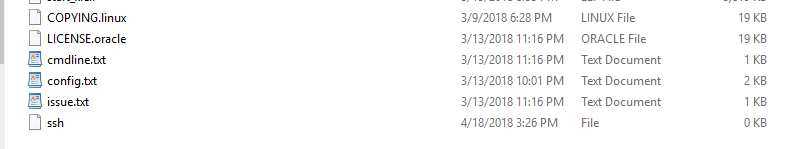


Fig.18

1. Remove the SD card from PC and put into Raspberry Pi.
2. Plug power and Ethernet to them. (Make it in the same LAN with your PC)
3. Run ssh remote program e.g. [putty](https://www.putty.org/) to remote to Raspbian.

Now, you can use the raspberry pi with linux base command. If you prefer to use remote desktop, please follow the step below which will guide you to VNC server installation for remote desktop to the Raspbian.

1.Enable VNC server

On your Raspberry Pi, run the following commands to make sure you have the latest version of VNC Connect.

sudo apt-get update

sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer

You can enable VNC Server at the command line using raspi-config.

sudo raspi-config

Now, enable VNC Server by doing the following:

* Navigate to Interfacing Options.
* Scroll down and select VNC > Yes.

Connecting to your Raspberry Pi with VNC Viewer

ESTABLISHING A DIRECT CONNECTION

Direct connections are quick and simple providing you're joined to the same private local network as your Raspberry Pi. For example, this might be a wired or wireless network at home, at school, or in the office).

On your Raspberry Pi (using a terminal window or via SSH) use these instructions or run ifconfig to discover your private IP address.

On the device you'll use to take control, download VNC Viewer. For best results, use the compatible app from RealVNC.

Enter your Raspberry Pi's private IP address into VNC Viewer.

The default username and password are pi, raspberry.

## Install npm.

All new Raspbian OS install node-red, node.js by default. You just install NPM by yourself, for install additional node to node-red api.

Run following command:

sudo apt-get install nodejs npm

## Running on Node-Red.

sudo node-red-start //Start node-red

sudo node-red-stop //stop node-red

Access node-red API

<Raspberry pi ip address>:1880

Editor & Admin API security

The Editor and Admin API supports two types of authentication:

username/password credential based authentication

since Node-RED 0.17: authentication against any OAuth/OpenID provider such as Twitter or GitHub

Username/password based authentication

To enable user authentication on the Editor and Admin API, add the following to your settings.js file: (cd $HOME/pi/.node-red)

adminAuth: {

type: "credentials",

users: [{

username: "admin",

password: "$2a$08$zZWtXTja0fB1pzD4sHCMyOCMYz2Z6dNbM6tl8sJogENOMcxWV9DN.",

permissions: "\*"

}]

}

Username: admin, Password: password

Generating the password hash

To generate a suitable password hash, you can use the node-red-admin command-line tool:

node-red-admin hash-pw

Setup node-red to auto start on boot:

sudo systemctl enable nodered.service

## Install additional palette

1. node-red-dashboard
2. node-red-contrib-netpie
3. node-red-contrib-firebase
4. node-red-contrib-aggregator

## Basic mqtt protocol with Node-Red

Let’s get started build basic mqtt flow on Node-Red.

1. Import clipboard bellow to workspace.

[{"id":"d13f07bf.3fbba8","type":"mqtt out","z":"c8bf9571.63b078","name":"","topic":"testwpm","qos":"","retain":"","broker":"4f5b7853.78fbc8","x":567,"y":140,"wires":[]},{"id":"25af8d63.b5ca92","type":"mqtt in","z":"c8bf9571.63b078","name":"","topic":"testwpm","qos":"2","broker":"4f5b7853.78fbc8","x":276,"y":56,"wires":[["53717a76.094784"]]},{"id":"53717a76.094784","type":"debug","z":"c8bf9571.63b078","name":"","active":true,"console":"false","complete":"false","x":572,"y":63,"wires":[]},{"id":"e48ca6f8.e47158","type":"inject","z":"c8bf9571.63b078","name":"","topic":"testwpm","payload":"Hello world!","payloadType":"str","repeat":"","crontab":"","once":false,"x":296,"y":138,"wires":[["d13f07bf.3fbba8"]]},{"id":"4f5b7853.78fbc8","type":"mqtt-broker","z":"","broker":"m13.cloudmqtt.com","port":"10800","clientid":"","usetls":false,"compatmode":true,"keepalive":"60","cleansession":true,"willTopic":"","willQos":"0","willPayload":"","birthTopic":"","birthQos":"0","birthPayload":""}]

2. Set broker url, port, username, password follow your mqtt account

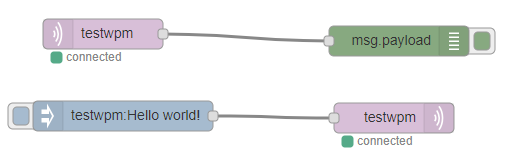


Fig.19 Basic mqtt with Node-Red

## Connect Node-MCU and Node-Red together with mqtt protocols

1. Open sample project /sdk\_manual/samples/3.Cloud\_mqtt set url, port, username and password of cloud mqtt to the same with mqtt flow on Node-Red. Build and upload to NodeMCU.

2. Change topic of mqtt node to (mywpm2018). Deploy  application.

3. Click on debug tab on Node-Red. You will see data from NodeMCU.

## Connect NodeMCU and Node-Red to firebase

1. Import following flow, setup mqtt node to the same with 3.Cloud\_mqtt.ino.

[{"id":"25af8d63.b5ca92","type":"mqtt in","z":"c8bf9571.63b078","name":"","topic":"mywpm2018","qos":"2","broker":"4f5b7853.78fbc8","x":286,"y":56,"wires":[["53717a76.094784","e232fd6e.57605"]]},{"id":"53717a76.094784","type":"debug","z":"c8bf9571.63b078","name":"","active":true,"console":"false","complete":"false","x":555,"y":57,"wires":[]},{"id":"e232fd6e.57605","type":"firebase modify","z":"c8bf9571.63b078","name":"firebase","firebaseconfig":"","childpath":"testvalue","method":"set","value":"msg.payload","priority":"msg.priority","x":548,"y":100,"wires":[[]]},{"id":"4f5b7853.78fbc8","type":"mqtt-broker","z":"","broker":"m13.cloudmqtt.com","port":"10800","clientid":"","usetls":false,"compatmode":true,"keepalive":"60","cleansession":true,"willTopic":"","willQos":"0","willPayload":"","birthTopic":"","birthQos":"0","birthPayload":""}]

2. Set firebase name then deploy application.

3. Run NodeMCU again. You can check result in firebase 

Now we can write data to firebase, next let check read function

[{"id":"53717a76.094784","type":"debug","z":"c8bf9571.63b078","name":"","active":true,"console":"false","complete":"false","x":667,"y":370,"wires":[]},{"id":"1fe6d6c.cb21629","type":"firebase.once","z":"c8bf9571.63b078","name":"read\_from\_firebase","firebaseconfig":"","childpath":"testvalue","repeatifnull":false,"eventType":"value","queries":[],"x":458,"y":372,"wires":[["53717a76.094784"]]},{"id":"65dc3144.b45cc","type":"inject","z":"c8bf9571.63b078","name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"x":250,"y":373,"wires":[["1fe6d6c.cb21629"]]}]

## Connect Node-Red to Netpie

### Write data to feed

1. Login to Netpie, Select Resources🡪Feeds

2. Create new feed

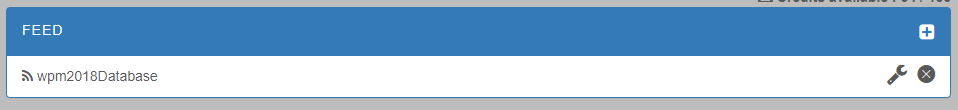


Fig.20 Netpie Feed

3. Add new variables follow fig.21

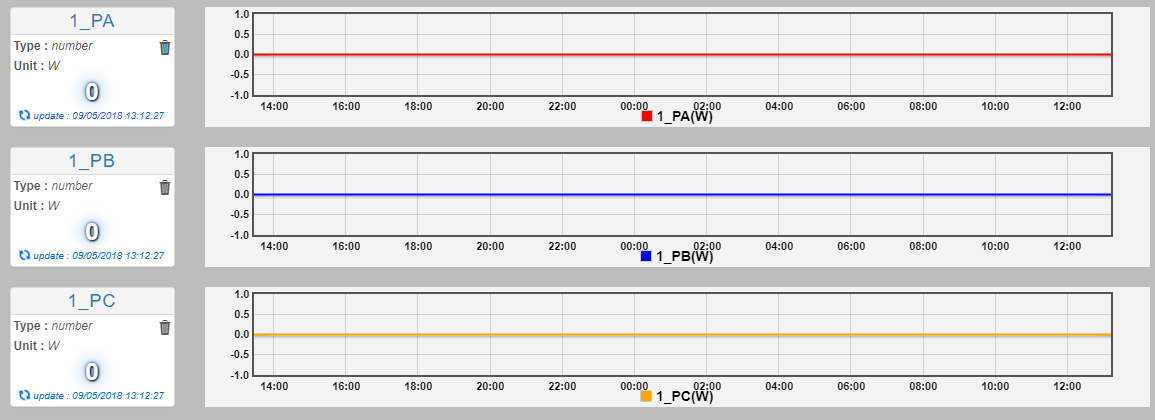


Fig.21 Add new variable to Netpie feed

4. Graph resolution on Data Display tab

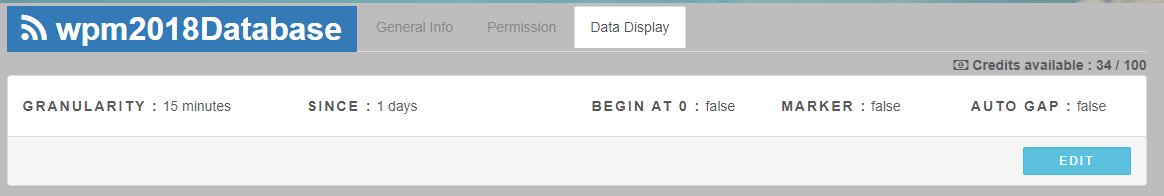


Fig.22 Change data display

5. As we mention on Netpie topic before (Learn Netpie section first), change permission to allow all, copy Feed ID and default api key to Node-Red flow bellow.

[{"id":"e30eb1c6.a23fb","type":"function","z":"c8bf9571.63b078","name":"feed","func":"msg.payload = \"1\_PA: \" + msg.payload + \",1\_PB: \" + msg.payload + \",1\_PC: \" + msg.payload;\nreturn msg;","outputs":1,"noerr":0,"x":342,"y":390,"wires":[["f4fddb20.998e58","2f94c89e.8720e8"]]},{"id":"f4fddb20.998e58","type":"writefeed","z":"c8bf9571.63b078","name":"writefeed","feedid":"","apikey":"","data":"payload","dataType":"msg","x":530.9478759765625,"y":387.8853759765625,"wires":[]},{"id":"46b87bbd.92e6c4","type":"inject","z":"c8bf9571.63b078","name":"write\_to\_db","topic":"testvalue","payload":"1000","payloadType":"num","repeat":"","crontab":"","once":false,"x":162,"y":389,"wires":[["e30eb1c6.a23fb"]]},{"id":"2f94c89e.8720e8","type":"debug","z":"c8bf9571.63b078","name":"","active":true,"console":"false","complete":"false","x":536,"y":338,"wires":[]}]

### Import data from feed to Netpie freeboard.

1. Select resource🡪Freeboard🡪YOUR\_FREEBORD\_NAME

2. Click setting icon on top of page.

3. On Datasource section, click Add then select type to Netpie feed

4. Insert your Feed ID and API key.

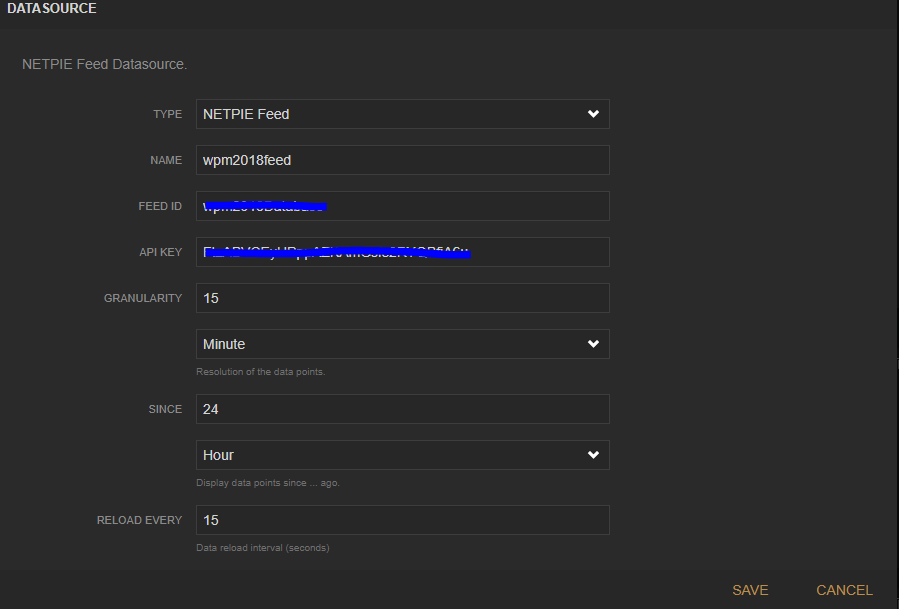
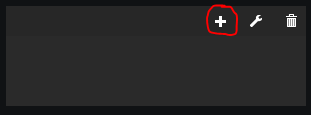


Fig.23 Netpie feed data source

5. Change Granularity (like sampling rate) to small value (15 second) for observe testing result. Set Since (Data span) value to 5 minutes.

Now let try creating simple widget.

6. Add Pane 🡪 Add new widget 

7. Select widget type to Feed view. Click on +Datasource and point to your feed data source

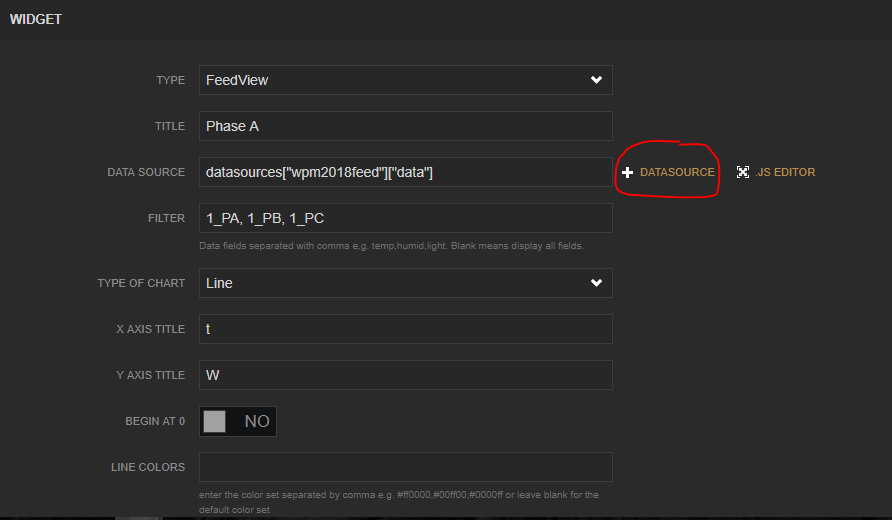


Fig.24 Feed View setting

8. Back to Node-Red, change inject node to inject every 15 second (Maximum rate for Netpie feed)

9. Observe result on your freeboard

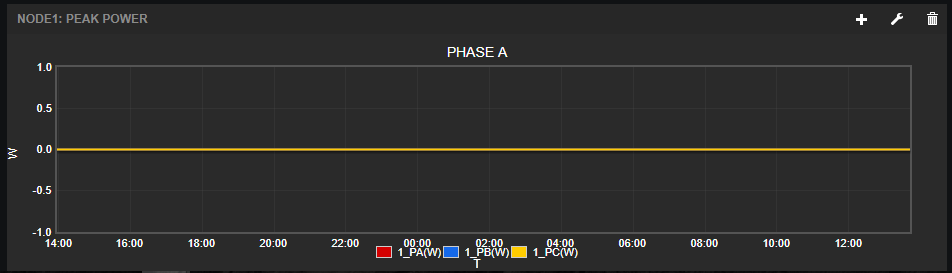


Fig. 25 Feed view

Let try to connect between mqtt node and Netpie feed by yourself. The topology looks like this:

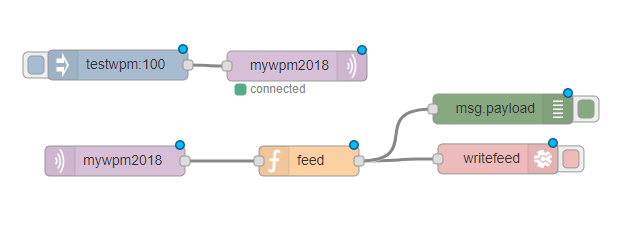


Fig. 25 Connect mqtt node to Netpie feed

### Microgear

Microgear is mqtt api from Netpie

1. Click Resource🡪Freeboard

2. On datasource pane, add🡪Netpie Microgear then copy App ID, Key and Secreat from we mention before (Resource🡪 Application🡪 Key)

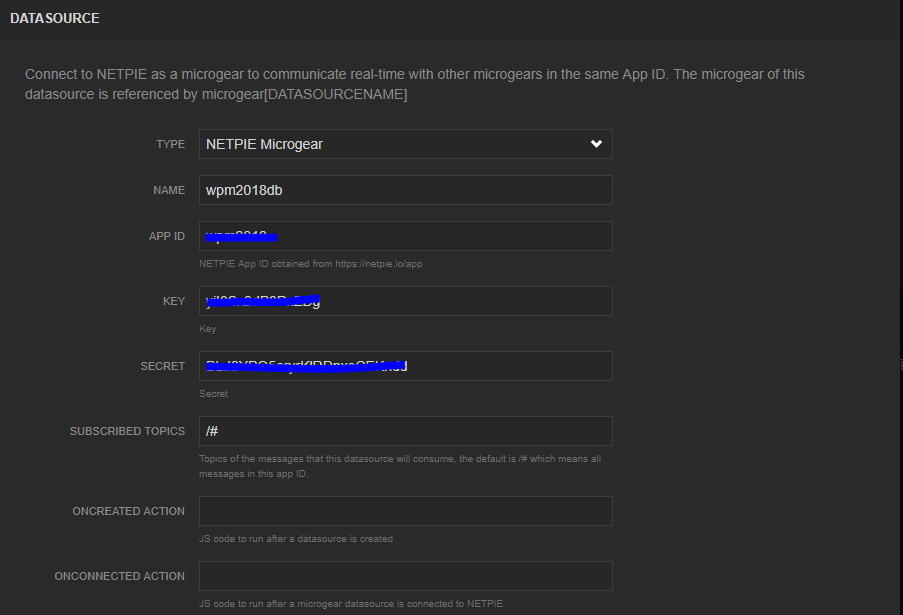
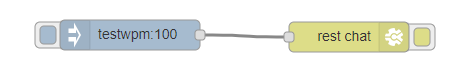


Fig. 26 Netpie Microgear

3. Back to node-red, test Microgear datasource with this flow (Do not forget insert authenticate in rest chat node) 

[{"id":"ab910fae.01705","type":"rest chat","z":"c8bf9571.63b078","name":"rest chat","appid":"","auth":"","alias":"mywpm","aliasType":"str","payload":"payload","payloadType":"msg","x":479,"y":454,"wires":[]},{"id":"467d5257.f0ba4c","type":"inject","z":"c8bf9571.63b078","name":"","topic":"testwpm","payload":"100","payloadType":"str","repeat":"","crontab":"","once":false,"x":260,"y":452,"wires":[["ab910fae.01705"]]}]

4. After click inject, add new widget (e.g. gauge) to your freeboard.

5. In widget setting click +datasource and point to your data (datasources["YOUR\_DATASOURCE"]["YOUR\_TOPIC"] e.g. datasources["wpm2018db"]["/mywpm"])

### Additional

NodeMCU able to connect to Netpie directly, click on <https://netpie.gitbooks.io/netpie-lab-4-arduino-on-netpie/content/> for learn more.

## Import Wireless Power Meter project

1. Go to /WPM2018/Node-Red/node-red\_clipboard/all-node.txt. Copy all clipboard (All text in all-node.txt) to Node-Red by select import menu.

2. Check mqtt account follow your account, Firebase name follow your firebase account and Netpie authentication follow your Netpie account.

3. Make sure you understand how the flow work. All topics details are in /sdk\_manual/all\_topic.

4. Edit m (slope) and C node value, from the equation P = m\*f + C then deploy application.

5. Open /WPM2018/NodeMCU\_Source\_File/wpm2018\_releasexxxxxxxx

6. Change the highlight value

//#define SIM\_MODE // Define, If need to simulation pulse in.

#define ON 1

#define OFF 0

#define NID "2"

#define SW "sw" NID

#define ALM "Alarm" NID

#define T\_MAX 1000000

#define T\_MIN 500

#define MAX\_CONNECT 100000

#define CLID "fff" NID

#define NUM\_PHASE 3

#define MAX\_ERR 3

#define TIME\_OUT 15000

#define SLOPE "m" NID

#define CONST "C" NID

#define ENCRIPT\_TOPIC "fff" NID

#define LOG\_SETTING "log\_enable" NID

#define LOG\_INTERVAL "log\_interval" NID

#define POLLING\_INTERVAL "polling\_interval" NID

#define MAX\_EQUATION\_REQ 10

#define DEFAULT\_SLOPE 60

#define DEFAULT\_CONST -2

#define SHEET\_NAME "\"แผ่น" NID "\", \"values\": "

#define URL\_BASE "{\"command\": \"appendRow\",\"sheet\_name\": " SHEET\_NAME

#define NUMBER\_OF\_SAMPLE 20

#define HIGH\_POWER

int sw\_status = ON;

int count\_connect = 0;

char nodeID[5] = NID;

const char\* ssid = "atop802.11x";

const char\* password = "atop3352";

//IPAddress ip(192, 168, 1, 140); //set static ip

//IPAddress gateway(192, 168, 1, 1); //set getteway

//IPAddress subnet(255, 255, 255, 0);//set subnet

const char\* mqttServer = "m12.cloudmqtt.com";

const int mqttPort = 19574;

const char\* mqttUser = "qonihivg";

const char\* mqttPassword = "UNIemBQQpGw8";

const char\* client\_id = CLID;

7. Build and upload project to NodeMCU.

8. Check result on Node-Red dashboard and debug COM port.

const char\* host = "script.google.com";

// Link to script of google spread sheet

const char \*GScriptId = "AKfycbwZagpM\_hvvOnNXGWQtSiiLalvMtdFgTtrdm-segr6QVqRqxuw";

//const char \*GScriptId\_backup = "AKfycbyJW6L0XCxnaaUaIblDZWprteCtFoaRhPemmnIvFw3vR31t75co";

const int httpsPort = 443;

const char\* fingerprint = "‎46 b2 c3 44 9c 59 09 8b 01 b6 f8 bd 4c fb 00 74 91 2f ef f6";

String url2 = String("/macros/s/") + GScriptId + "/exec?cal";

//String urlbackup = String("/macros/s/") + GScriptId\_backup + "/exec?cal";

// Read from Google Spreadsheet

//String url3 = String("/macros/s/") + GScriptId + "/exec?read";

String payload\_base = URL\_BASE;

String payload = "";

HTTPSRedirect\* clientg = nullptr;

# TeslaSCADA

## Installing TeslaSCADA

Go to TestlaSCADA web site, download and install TeslaScada2 IDE and TeslaScada2 runtime.

## Create your first application

Follow /sdk\_manual/TeslaSCADA\_IDE\_UserManual.pdf

## Examples

The example projects are in WPM2018/TeslaScada

## Import to Andriod

Follow /sdk\_manual/TeslaSCADA2\_Runtime\_Android.pdf

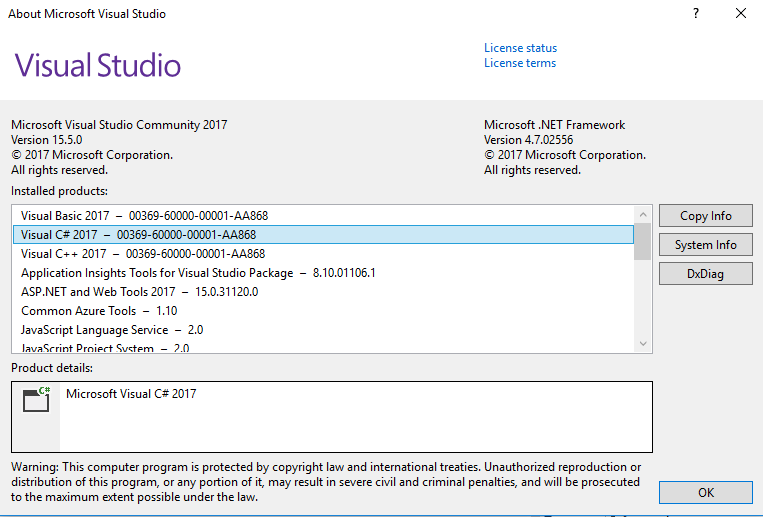
## Import for iOS

Follow/TeslaSCADA2\_Runtime\_iPhone.pdf

# Windows Form Application

## Installing Visual Studio IDE

Referrence version



## Build Open Source Application

Open /WPM2018/Windows\_Form\_Application/wpm…/\*.sln with visual studio 2017