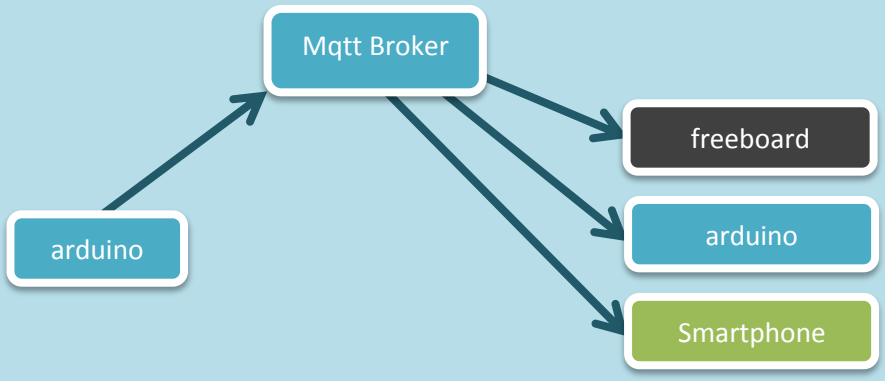
Arduino Manual para IoT

(openFramework)

Arduino+Mosquitto+NodeREd +Freeboard = Visualizacion Data en tiempo real



Flatbox

Guía de instalación de un framework abierto para IoT framework/dashboard con NodeRed, Mosquitto y Freeboard.io dashboard

Basado en el tutorial de primalcortex (febrero 25, 2015)

Conocimientos básicos Requeridos:
Arduino IDE
- Programación básica
- Programación con sensores.
- Programación con librería #include <ArduinoJson.h> // https://github.com/bblanchon/ArduinoJson/releases/tag/v5.0.7

Haber realizado los ejemplos por defecto para Comunicaciones por medio de clientes TCP (wifi/GSM/ETH)
- Haber realizado los ejemplos de las librerías de #include <PubSubClient.h> // https://github.com/knolleary/pubsubclient/releases/tag/v2.3

Conocimiento básico de Linux
Conocimiento básico de Git
Conocimientos básicos de nodejs
Conocimiento básico de Apache2
- Symlimks

Debido a su bajo costo se utilizara un chip esp8266 para el desarrollo del tutorial (cualquiera de las versiones de hardware disponibles esta bien)

Diagrama 1.0 (Modulo de ESP8266, NodeMCU v3) ($6 USD)

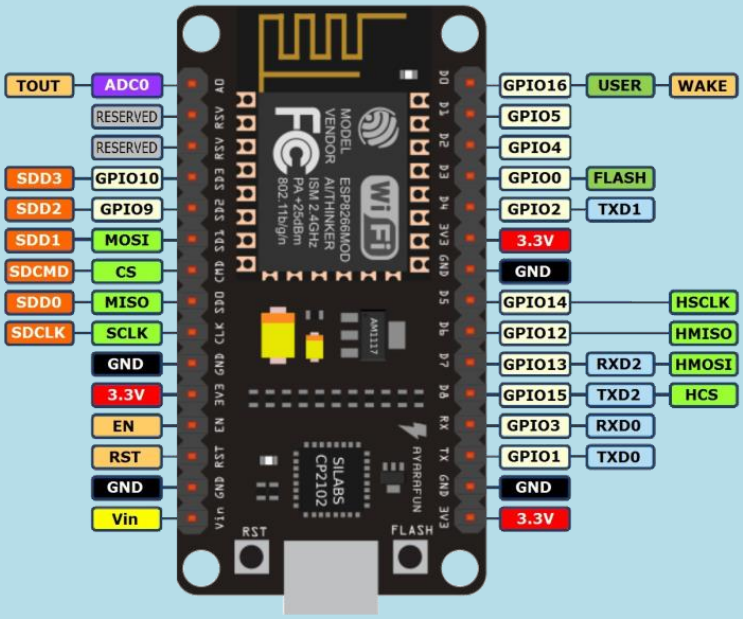
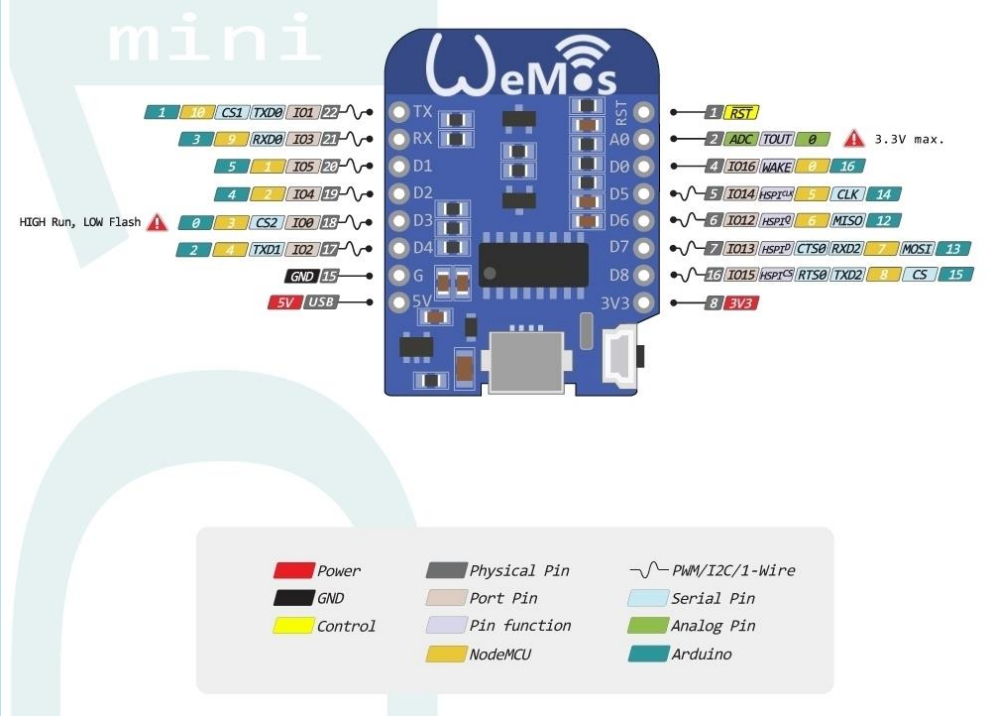


Diagrama 1.1 (Modulo de ESP8266, wemos D1 mini) ($4 USD)



Para este tutorial se recomienda una máquina virtual (local o en la nube) con los siguientes prerrequisitos:

Ubuntu14.04-64 Minimal for VSI:
Hardware:

1x2.66 GHz 2GB RAM 25GBHDD 100Mps 250MB Bandwidth
SSH enabled
IPV 4 IP address

Software:

Git:

```
$ sudo apt-get update
$ sudo apt-get install git
```

Configurado con tu cuenta de Git.

```
$ git config --global user.name "Your Name"
$ git config --global user.email "youremail@domain.com"
```

Node.js:

```
curl -sL https://deb.nodesource.com/setup | sudo bash -
```

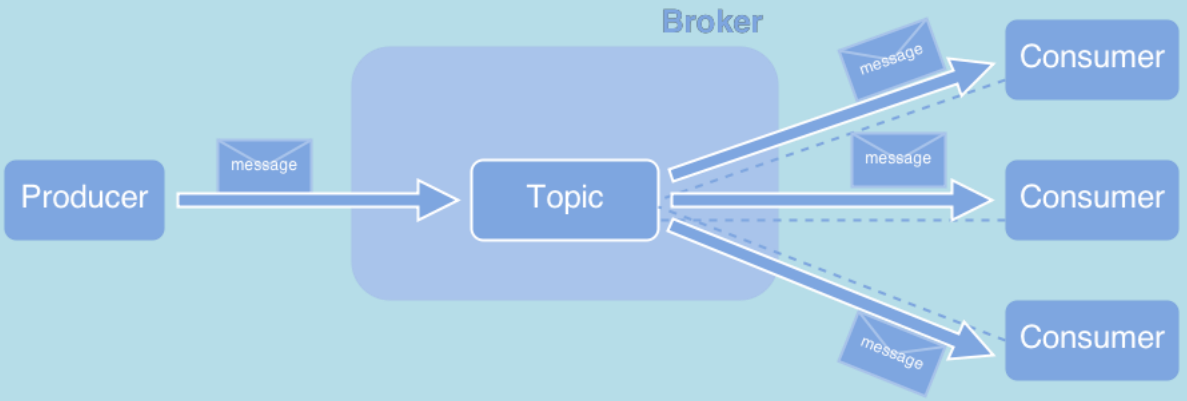
```
sudo apt-get install nodejs
```

```
sudo apt-get install build-essential
```

En la muchos de los casos también deberás instalar npm:

```
sudo apt-get install npm
```

Que es un open framwork para IoT, es un conjunto de programas con licencias de código abierto los cuales nos ayudaran a capturar, estructurar y presentar los datos que transmitamos al internet desde nuestros módulos de Arduino. Existen diversos protocolos de IoT como AMQP, MQTT, STOMP, para este caso en particular estaremos utilizando MQTT para conocer a profundidad sobre el mismo pueden leer más en : http://mqtt.org/ este es un simple servicio de publicar suscripción:



El software que estaremos usando dentro de Linux será:

IBM NodeRed: (http://nodered.org/) Node-RED nos provee de una interfaz basada en un navegador web que nos permite crear flujos de eventos e interconectarlos todos ellos a través de un ligero entorno de trabajo y desarrollo. Está construido en node.js, lo que le permite funcionar justo al borde de la red o en la nube, dotándole de una notable flexibidad. El ecosistema del gestor de paquetes de node (npm) puede ser usado para extender de forma sencilla la paleta de nodos disponibles, permitiendo conexiones entre nuevos dispositivos y servicios. Node-RED ha sido desarrollado como un proyecto open-source en GitHub y está bajo una licencia Apache 2

.**MQTT Broker Mosquitto:** (http://mosquitto.org/) Mosquitto es un corridor de Mqtt para linux el cual es liviano y facil de instalar

Freeboard.io Dashboard: (http://freeboard.io/) es un motor dinámico de tableros opensource project GitHub page (https://github.com/Freeboard/freeboard). El cual nos permite diseñar y administrar los datos capturados por el corredor de MQTT por medio de WEbSockets

Paso 1: instalando NodeRed:
Primero verificamos que se encuentre instlado node.js y npm con el commandos **node -v** and **npm -v** .

Luego creamos nuestra estructura de archivos llamado IOTServer en el cual mantendremos todas nuestras configuraciones de software.

```
cd ~
mkdir IoTServer
cd IoTServer
```

Luego corremos desde el bash:

```
sudo npm install -g --unsafe-perm node-red
```

Ahora podemos correr el servidor de nodered con el comando: **node red.js -v. utilizamos el** -v al final para saber si existe algún error o si algún modulo se encuentra mal instalado:

```
$ node-red

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

25 Feb 22:51:09 - [info] Node-RED version: v0.13.1
25 Feb 22:51:09 - [info] Node.js  version: v4.2.6
25 Feb 22:51:09 - [info] Loading palette nodes
25 Feb 22:51:10 - [warn] ---------------------------------------
25 Feb 22:51:10 - [warn] Failed to register 1 node types
25 Feb 22:51:10 - [warn] Run with -v for details
25 Feb 22:51:10 - [warn] ---------------------------------------
25 Feb 22:51:10 - [info] User Directory : /home/nol/.node-red
25 Feb 22:51:10 - [info] Server now running at http://127.0.0.1:1880/
25 Feb 22:51:10 - [info] Creating new flows file : flows_noltop.json
25 Feb 22:51:10 - [info] Starting flows
25 Feb 22:51:10 - [info] Started flows
```

Podremos verificar el funcionamiento del nuestro sitio de nodered en la dirección : http://localhost:1880

Podemos encontrar más información sobre como configurar NodeRed en http://nodered.org/docs/configuration.html ahora podemos empezar a usar nodered con: **node red.js -v -s settings.js**

En node red podemos diseñar el flujo de información para la data del ESP8266 recibida por el MQTT e incluso elaborar un servicio simple REST basado en HTTP protocolo, procesar la data y almacenarla.

Para hacer que el servicio corra desde el boot:

```
sudo npm install -g pm2
```

```
pm2 start /usr/bin/node-red -- -v
```

```
pm2 info node-red
pm2 logs node-red
```

```
pm2 save
pm2 startup
```

Reboot!!!

Para almacenar la data en una base de datos podemos usar MongoDB instalando los módulos

Que hacen falta de la siguiente manera:

Primero debemos instalar mongodb

1) Importar la llave publica usada el manejo de paquetes:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
EA312927
```

2) Crear una lista por MongoDB

```
echo "deb http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3
.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2
.list
```

3) Recagar la base de datos de paquetes:

```
sudo apt-get update
```

4) Instalar los paquetes de MongoDB

```
sudo apt-get install -y mongodb-org
```

5) Run MongoDB

```
sudo service mongod start
```

Luego localizamos el directorio de ~/ .node-red: por medio de cd ~/.node-red e instalamos el plug in en el directorio utlizando:

```
npm install node-red-node-mongodb
```

Paso2: Inatalar Mosquitto MQTT:

Instalar el repositorio y actualizar el listado

- sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
- sudo apt-get update

Ir al folder:

```
cd ~/IoTServer
```

E nstalar por medio de:

apt-get install mosquitto

Después de la instalación echamos andar el servicio por defecto con:

nohup mosquitto -c ./etc/mosquitto/mosquitto.conf &

y confirmamos que el servicio este corriendo con:

netstat -nap | grep 1883

Paso 3: instala el plugin de freeboard.io dashboard para node red

Luego localizamos el directorio de ~/ .node-red, por medio de

cd ~/.node-red

Instalamos el plugin en el directorio utilizando:

npm install node-red-contrib-freeboard

Reiniciamos y verificamos que se encuentre corriendo

http://iotarduinodaygt.flatbox.io:1880/freeboard/

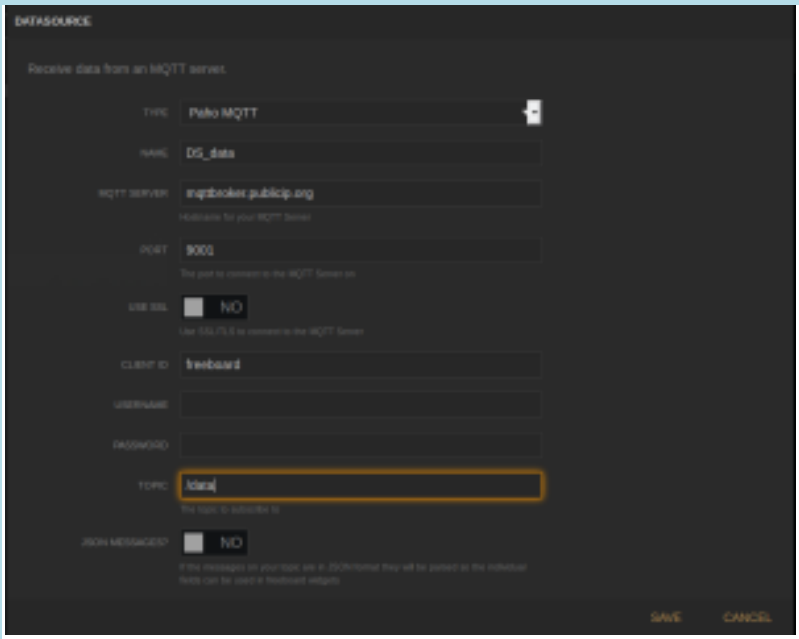
Paso 4. Cargar el código al arduino.

Bajar el Zip con el Codigo para arduino de: https://github.com/EdwinKestler/IoTarduinoDayGT/
Compilarlo y cargarlo al ESP desde el IDE de Arduino.

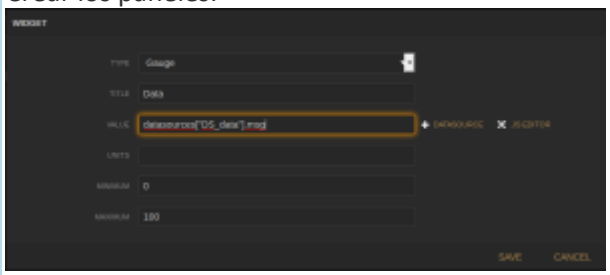
Paso 5. Configuramos el flujo de nuestro servicio en Node Red

Pado 6. Configuramos nuestra visualización de data en Freeboard:

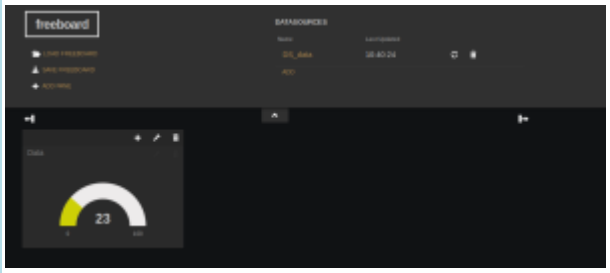
Seleccione la Fuente de datos (PahoMQTT)
Configurar el servidor y la fuente de dato:



Crear los paneles.



Y VEr La Data Fluir…..



Felicidades han creado su primer openframework de IoT

IOT Arduino Day Guatemala 2016!