

IoTtweet.com ก็เป็นเว็บที่เราสมัครไปสร้างหน้า dashboard โดยทางเราสามารถทำการส่งข้อมูลผ่านทาง arduino หรือ raspberry pi เข้า concept IoT ที่เพิ่งไปอบรมมาก่อนหน้านี้

concept

1. หลังจากเราพัฒนา arduino หรือ raspberry pi งานบางอย่างเราต้องการเก็บข้อมูลจาก sensor มาเก็บไว้เพื่อทำการวิเคราะห์
2. บางครั้งเราทำหน้า dashboard เองจะเสียเวลามาก อีกอย่างการเก็บข้อมูลจำนวนมากก็ไม่ง่ายที่จะดูแล ถ้าข้อมูลไม่ได้เป็นความลับผมก็เห็นด้วยที่จะผลัดภาระการแสดงผลและการจัดการฐานข้อมูลออกไปบ้าง
3. หลักการของ IoTtweet คือเปิดช่องทางให้เราส่งข้อมูลจาก arduino หรือ raspberry ผ่าน internet เข้ามาเก็บที่ IoTtweet แล้วเราก็สามารถ login เข้าไป customize หน้า screen ได้ ระบบจะมี graph รูปแบบต่างๆเตรียมไว้ให้เราเลือกมาใช้
4. จากนั้นเราก็สามารถ monitor ข้อมูลของเราจากที่ไหนก็ได้ ขอแค่มี internet

นอกจากนี้ยังทำให้ผมเห็นว่า คนรุ่นใหม่เริ่มมีความคิดที่จะทำตัวเป็นผู้ให้บริการและมี community ของนักพัฒนามากขึ้น ไม่ใช่ต้องเข้าเวปต่างประเทศอย่างเดียว

การใช้งานคำสั่งต่างๆใน APIs ของ IoTtweet

WiFi connecting command

การเชื่อมต่อ WiFi network

```
.begin(ssid,password);
```

ตัวแปรในคำสั่ง	ประเภทของข้อมูล	ความหมาย
SSID	const char	ชื่อของเครือข่าย WiFi ที่คุณต้องการเชื่อมต่อกับ IoT device.
Password	const char	รหัสผ่านของเครือข่าย WiFi ที่คุณต้องการเชื่อมต่อ.

ผลลัพธ์	ประเภทของข้อมูลที่ส่งกลับมา	ค่าที่ส่งกลับมา
สามารถเชื่อมต่อ WiFi ได้สำเร็จ	bool	true
ไม่สามารถทำการเชื่อมต่อ WiFi ได้	bool	false

การส่งข้อมูลจาก IoT มายัง Dashboard

การส่งข้อมูลจาก IoT device ไปยัง Dashboard IoTtweet

ผลลัพธ์	ประเภทของข้อมูลที่ส่งกลับมา	ค่าที่ส่งกลับมา
สร้างฐานข้อมูลได้สำเร็จ (จะแสดงเฉพาะครั้งแรกที่ส่งข้อมูลมายัง IoTtweet)	JSON	{"iottweet":[{"create":"success"}]}
บันทึกข้อมูลลงฐานข้อมูลได้สำเร็จ	JSON	{"iottweet":[{"record":"success"}]}

จากไฟล์ที่เราดาวน์โหลดมาจาก Github:IoTtweet, ในนั้นจะมีไฟล์ Example อยู่ด้วยครับ และในแต่ละไฟล์ ก็จะมีคำสั่งการใช้งานดังนี้.

```
.WriteDashboard(userid,key,data0,data1,data2,data3,private_tweet,public_tweet);
```

ตัวแปรในคำสั่ง	ประเภทของข้อมูล	ความหมาย
userid	const char	เลขประจำตัวผู้ใช้งานของสมาชิก IoTtweet.com , เลขนี้จะมีความยาว 6 หลัก เช่น 000001 เป็นต้น.
key	const char	รหัสประจำอุปกรณ์ของเรา ที่ได้จากการลงทะเบียนในระบบ "My IoT garage".
data0-data3	float	ข้อมูลตัวเลขแบบมีทศนิยม ที่เราต้องการส่งขึ้นไปเก็บในฐานข้อมูล และแสดงผลบนกราฟในหน้า dashboard.
private_tweet	const char	ข้อความที่ต้องการส่งไปแสดงผลบนหน้า dashboard เป็นแบบส่วนตัว.
public_tweet	const char	ข้อความที่ต้องการส่งไปแสดงผลบนหน้า dashboard เป็นแบบสาธารณะ(User ผู้อื่นจะสามารถเห็นได้จากหน้า Home ของ IoTtweet.com).

[คำแนะนำ]

สำหรับการส่งข้อมูลยังฐานข้อมูลบน IoTtweet, ในการส่งแต่ละครั้ง ควรจะเว้นระยะเวลาในการส่งอย่างน้อย 15 วินาที. เพื่อป้องกันการบันทึกข้อมูลลงฐานข้อมูลผิดพลาด.

วันนี้จะมาสอนวิธีการส่งข้อมูลจาก IoT device ไปยังเว็บ loTtweet.com นะครับ ซึ่งใน loTtweet.com นี้ จะมี
จอแสดงผล (Dashboard) ให้ดูเป็นแบบ Line chart, Bar chart และ Gauge วัตถุประสงค์ที่สามารถปรับแต่งหน้าตาได้ตามต้องการ
(อันนี้จะไว้มาเขียนบทความการใช้งานให้อีกทีนะครับ)

[สิ่งที่ต้องจัดเตรียมก่อนเริ่ม]

1.User account loTtweet

- แน่นนอนครับว่าต้องมี User account ของเว็บ loTtweet ก่อน, ว่าแล้วก็เข้าไป [Sign up](https://loTtweet.com) ได้เลยครับ. หลังจาก Sign up แล้ว อย่า
ลืมตามไป Verify activation e-mail ด้วยนะครับ

(ถ้าหาใน Inbox ไม่เจอ, ลองดูตามใน Junk mail ก็ได้ครับ บางทีก็เข้าไปอยู่ในนั้น)

- หลังจาก activate แล้ว, ให้เข้าไปที่หน้าเว็บ loTtweet.com ครับ Login ให้เรียบร้อย แล้วไปที่ Dashboard คลิกที่ปุ่ม “My
IoT garage” ด้านบนครับ.

- กรอกข้อมูล device ของเราให้เรียบร้อย ชื่อ IoT, ชื่อ Model ที่ใช้ แล้วก็กด Add เพื่อทำการเพิ่มเข้าไปในโรงเก็บ IoT ของเรครับ

* สังเกตว่า จะมี key ที่ทางเว็บ สร้างขึ้นมาให้ เดี่ยวเราจะเอาไปใช้ใน code ของ Arduino IDE นะครับ

ปุ่ม My IoT garage.

My IoT garage

Chart Preferences

Export data to JSON

Export data to CSV

Get Live feed

เพิ่มอุปกรณ์ IoT เข้าไป

Add your IoT device

TEST

IoT model.

NODE MCU

Automatic generated keys for device [What is this ?](#)

h5fg1c0qhdu@

Add

อุปกรณ์ของเราจะเข้ามาในลิสต์แล้ว

My registered IoT device

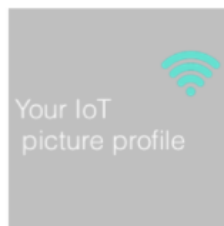
You can register your IoT maximum to 10 devices

No.	Thingname	Keys ?	IoT model.	Register date	Remove device
1				2016-09-08 22:53:14	Delete
2				2016-09-18 12:06:51	Delete
3				2016-09-26 20:32:26	Delete
4	TEST	h5fg1c0qhdu9	NODE MCU	2016-09-30 15:58:59	Delete
5					Delete
6					Delete

กลับมาที่หน้า Dashboard ก็จะได้เห็นอุปกรณ์ พร้อมใช้งาน



My IoT name : TEST



My IoT device list TEST

Graph refresh rate 5 sec.

Number of data plotting

☒ Latest 30 data samples

☐ All data (underconstruction)

2.Node MCU board

- ในตัวอย่างนี้ จะใช้งาน **Node MCU v0.9** นะครับ มาพร้อมกับ **ESP8266 WiFi module ESP-12**



3.เครื่องคอมพิวเตอร์ ลง Arduino IDE

4.ไฟล์ IoTtweet.h Library

- IoTtweet.h Library สามารถดาวน์โหลดได้จาก [github:IoTtweet library](https://github.com/IoTtweet/library) ครับ (เข้าไปดาวน์โหลดแล้ว ให้กำลังใจด้วยการกด star ไปด้วยนะครับ)

[ขั้นตอนการติดตั้ง Library]

ในการใช้งานนี้ จำเป็นจะต้องมีการลง Library ใน Arduino IDE ไว้ 2 ตัวนะครับ คือ **IoTtweet.h** และ **ESP8266WiFi.h** ถ้าเครื่องใครเคยลง ESP8266WiFi.h แล้ว ก็ข้ามขั้นตอนนี้ได้เลย

** การติดตั้ง ESP8266WiFi.h **

1. เปิด Arduino IDE software, แล้วไปที่ Preferences
2. ในช่อง **Board manage urls**, ให้ใส่ link นี้ลงไป

http://arduino.esp8266.com/stable/package_esp8266com_index.json

เสร็จแล้วกด OK.

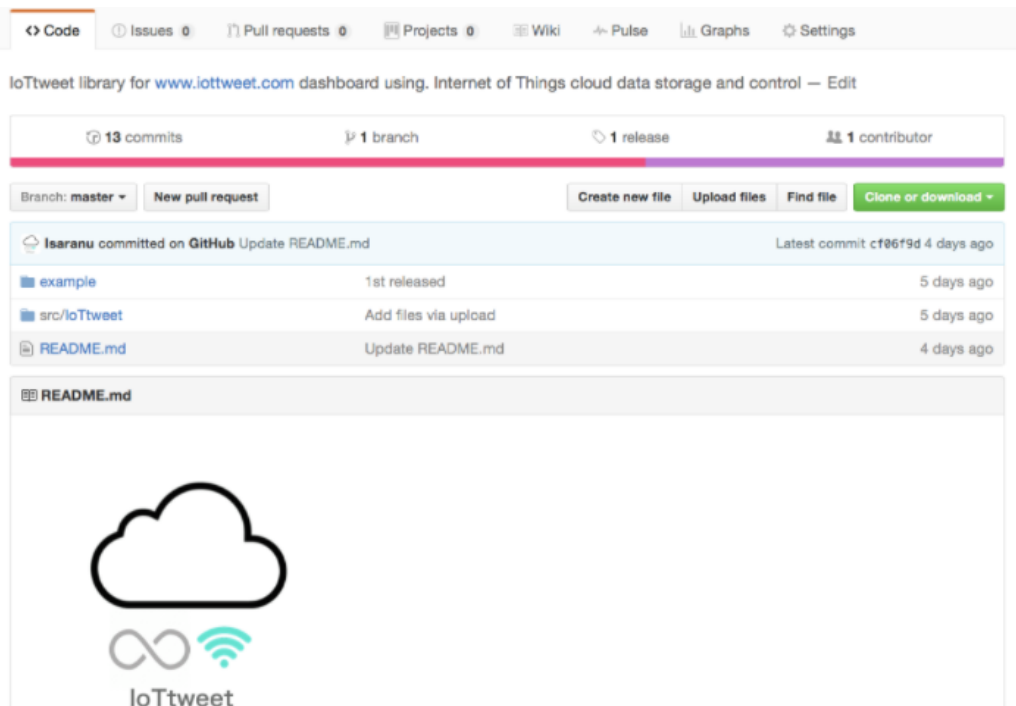
3. ปิดโปรแกรม Arduino IDE แล้วเปิดขึ้นมาใหม่
4. ไปที่ **Tool > Boards > Boards Manager**, แล้วเลื่อนลงมา หา package ที่ชื่อ **"esp8266 by ESP8266 community"**. จากนั้นกด Install ครับ
5. Install เสร็จแล้ว ก็ปิด แล้วก็เปิด Arduino IDE ใหม่อีกรอบ เป็นอันเสร็จสิ้นกระบวนการ ตอนนี้เราก็จะเห็นว่าใน Example จะมีตัวอย่าง code ของ ESP8266 เข้ามาแล้ว รวมไปถึงในรายการอุปกรณ์ ก็มีบอร์ดต่างๆเพิ่มเข้ามาด้วย

** การติดตั้ง ArduinoJson.h **

1. เปิด Arduino IDE software, แล้วไปที่ **Sketch > Include Libraries > Manage Libraries**
2. ค้นหา **"ArduinoJson.h"** libraries, แล้วกด **Install** ได้เลยครับ

** การติดตั้ง IoTtweet.h **

1. หลังจากดาวน์โหลดไฟล์ Library package IoTtweet จาก github มาแล้ว, ให้ Extract ไฟล์ลงในเครื่องคอมพิวเตอร์
2. ในโฟลเดอร์ src, ให้ copy โฟลเดอร์ชื่อ **IoTtweet** ไปวางไว้ที่ **Arduino > libraries** ในเครื่องคอมพิวเตอร์ของเรา (ในโฟลเดอร์ IoTtweet จะมีไฟล์ 2 ตัว ชื่อ **IoTtweet.h** และ **IoTtweet.cpp**)



เอาละ, ติดตั้ง Library เสร็จเรียบร้อยแล้ว ก็กลับมาที่ไฟล์ที่ดาวน์โหลดจาก github กันนะครับ ในนั้นจะมีตัวอย่างของโค้ดอยู่ 2 ไฟล์ คือ **WriteToDashboard.ino** และ **ControlMyIoT.ino** ครับ ในตัวอย่างนี้ จะใช้ไฟล์ **WriteToDashboard.ino** นะครับ ว่าแล้วก็เปิดไฟล์นี้ใน Arduino IDE ได้เลย

..
ControlMyIoT.ino
WriteToDashboard.ino
readme.md

สำหรับค่าต่างๆที่จะต้องกรอกลงไปในไฟล์นี้ เพื่อให้เป็นอุปกรณ์ของเรา ก็มีดังนี้ครับ

SSID – ชื่อ WiFi network ที่เราต้องการเชื่อมต่อ

Password – รหัสผ่านเข้า WiFi ที่เราจะเชื่อมต่อ

userID – เลข User ID จาก IoTtweet.com ซึ่งสามารถดูได้จาก มุมบนขวาของหน้าเว็บไซต์ครับ เป็นตัวเลข 6 หลัก ต้องกรอกให้ครบนะครับ ถึงแม้จะเป็นเลข 0

key – นี่ละครับ, key ที่เราได้จากการลงทะเบียน IoT device ของเราใน **"My IoT garage"** เอามากรอกลงตรงนี้เลยครับ

key นี้จะเป็นรหัสเฉพาะตัวสำหรับอุปกรณ์ IoT device แต่ละตัวที่เราลงทะเบียนไว้ในโรง IoT ของเราครับ

```

WriteToDashboard | Arduino 1.7.10

/*
 * This example demonstrate how to write data from your "Internet of Things"
 * coding from IoTtweet.com
 * Created : 2016.Sep.25
 * By Isaranu Janthong
 * IoTtweet Founder.
 * Visit us at : www.iottweet.com
 */

#include <ESP8266WiFi.h>
#include <IoTtweet.h>

const char *userid = "your_IoTtweet_userID"; //IoTtweet account u
const char *key = "your_registered_IoT_device_key"; //IoTtweet registere
const char *ssid = "your_wifi_ssid"; //Your-WiFi-router-S
const char *password = "your_wifi_password"; //Your-WiFi-password

float data0, data1, data2, data3; //Your sendi
const char *private_tweet = "Hello%20World"; //Your priva
const char *public_tweet = "I%20am%20Internet%20of%20Things"; //Your publi

IoTtweet myiot; //naming your devices

void setup() {

```

หลังจากเปลี่ยนค่าต่างๆด้านบนแล้ว ก็ทำการ Upload code ไปยังอุปกรณ์ เราก็เลือกอุปกรณ์ให้ตรงตามที่เราเชื่อมต่อไว้ละครับ.

[ดู data ของเรา ที่ Dashboard]

หลังจากที่เรา upload code ลง Node MCU แล้ว เราก็ปล่อยให้มันรันไปเรื่อยๆ ในตัวอย่าง sketch file จะทำการ **delay 15 วินาที** ทุกๆการส่งค่าครับ

ส่วนค่าที่ส่งไป **data0 ถึง data3** จะใช้การ random เลขขึ้นมา ตรงนี้เวลาเอาไปใช้จริง เราก็เปลี่ยนเป็นค่าที่เราต้องการได้ทันทีครับ

ส่วน tweet message ก็เช่นเดียวกัน เราต้องการให้ข้อความใดไปแสดงผล ก็แก้ไขได้เช่นกันครับ

(เวอร์ชัน Dashboard 2.0 ตอนนี้ ยังไม่รองรับ Blank space ระหว่างค่าละครับ ถ้าจะเว้นวรรค, ให้ใส่ %20 ลงไปครับ)

ใน Dashboard เราก็จะเห็นกราฟแสดงค่าล่าสุด 30 จำนวนที่ส่งไปครับ เหนือนี้ก็เป็นอันเรียบร้อยแล้ว