

แสดงการประยุกต์ **NETPIE Freeboard** ในการควบคุมอุปกรณ์ โดยในเบื้องต้นนี้ เราจะควบคุมไฟ LED บนบอร์ด **NodeMCU** ซึ่งใช้หลักการทางานบนพื้นฐานของการ **Subscribe** ข้อความจาก **Topic** หรือหัวข้อที่ระบุ และการกำหนดตรรกะของการควบคุมทั้งในส่วนของ **Datasource** และส่วนของ **Widget** ที่ใช้ควบคุม โดยมีขั้นตอนดังต่อไปนี้

1 ในหน้า **NETPIE Freeboard** คลิกเพิ่ม **Datasource** ที่สร้างขึ้นก่อนหน้านี้เพื่อแก้ไข ตั้งชื่อ **Datasource** ใส่ค่า **KEY** และ **SECRET** และในช่อง **SUBSCRIBED TOPICS** ให้ใส่ **/pieled/state** หรือ **Topic** ที่ท่านระบุไว้สำหรับการ **publish**

รูปแสดงหน้าต่างตั้งค่า **Datasource** ใน **NETPIE Freeboard** สำหรับควบคุม LED

**DATASOURCE**

Connect to NETPIE as a microgear to communicate real-time with other microgears in the same App ID. The microgear of this datasource is referenced by microgear[DATASOURCENAME]

TYPE: **NETPIE Microgear**

NAME: **DataLab10**

APP ID: **TestLab10**  
NETPIE App ID obtained from <https://netpie.io/app>

KEY: **mEv9YqQdn...**  
Key

SECRET: **pvmzq1uFgtB...**  
Secret

SUBSCRIBED TOPICS: **/#**  
Topics of the messages that this datasource will consume, the default is /# which means all messages in this app ID.

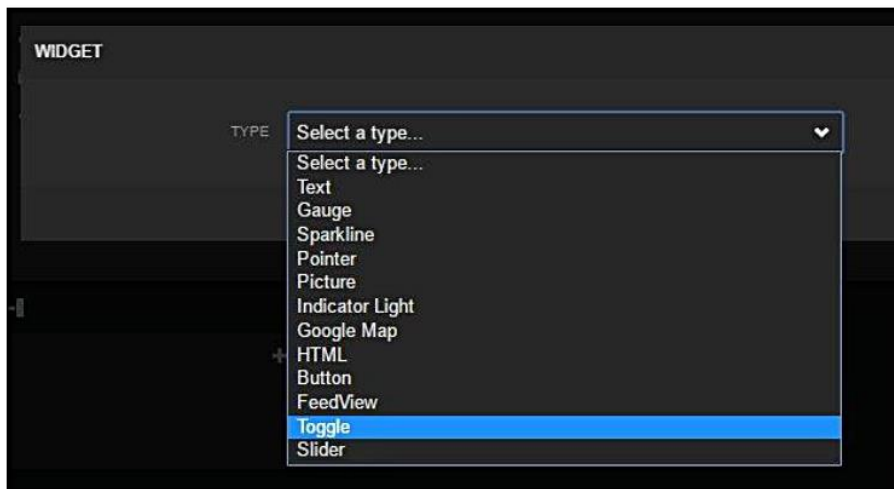
ONCREATED ACTION:   
JS code to run after a datasource is created

ONCONNECTED ACTION:   
JS code to run after a microgear datasource is connected to NETPIE

**SAVE** **CANCEL**

2.สร้าง Widget ขึ้นมาใหม่โดยกด + (ADD PANE) และเลือกชนิดใน Drop Down Box เป็นแบบ Toggle

รูปแสดงหน้าจอเลือกชนิด **Widget** ให้เป็นแบบ **Toggle**



จากนั้นตั้งค่า Widget ดังนี้โดยหน้าจอการตั้งค่าแสดงดังรูป

- TOGGLE CAPTION : ตั้งชื่อปุ่ม Toggle (ในตัวอย่างตั้งเป็น PIE\_LED)
  - TOGGLE STATE : ใส่ข้อมูลตามชื่อของ Datasource และ Topic เช่น `datasources["DataLab10"]["/TestLab10/pieled/state"]==1`
- ON TEXT : ON
- OFF TEXT : OFF
- ONTOGGLEON ACTION : `microgear[DataLab10].chat('pieled','1')`
- ONTOGGLEOFF ACTION : `microgear[ DataLab10].chat('pieled','0')`

รูปแสดงหน้าจอการตั้งค่า **Widget** ชนิด **Toggle**

**WIDGET**

A simple toggle widget that can perform Javascript action.

TYPE: Toggle

TOGGLE CAPTION: LED

TOGGLE STATE: datasources["DataLab10"]["/TestLab10/pieled/state"]==1 + DATASOURCE JS EDITOR  
Add a condition to switch a toggle state here. Otherwise it just toggle by click.

ON TEXT: ON

OFF TEXT: OFF

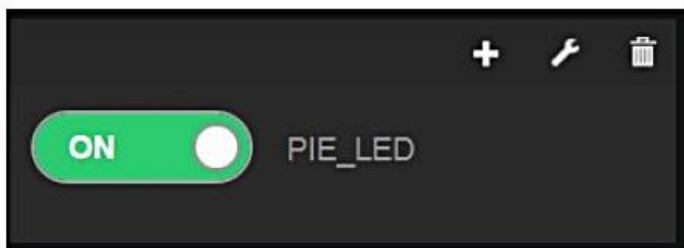
ONTOGGLEON ACTION: microgear["DataLab10"].chat("pieled","1")  
JS code to run when a toggle is switched to ON

ONTOGGLEOFF ACTION: microgear["DataLab10"].chat("pieled","0")  
JS code to run when a toggle is switched to OFF

ONCREATED ACTION:   
JS code to run after a toggle is created

SAVE CANCEL

แล้วกด **Save** จะได้ Widget ที่มีปุ่มควบคุมดังแสดงในภาพ เพื่อทดสอบเปิดปิด LED บน NodeMCU



คำอธิบายเพิ่มเติม

**TOGGLE STATE** เป็นสถานะ On/Off ซึ่งสามารถผูกกับตรรกะของ **Datasource** ในที่นี้เราตั้งค่าให้ Toggle เปลี่ยนสถานะตามค่าที่ส่งมาใน Topic ชื่อ `/pieled/state`

**ONTOGGLEON** และ **ONTOGGLEOFF** เป็นคำสั่งที่จะถูกเรียก เมื่อ Toggle เปลี่ยนสถานะไปเป็น ON และ OFF ตามลำดับ

ในหน้า **Datasources** ตรงช่อง **SUBSCRIBED TOPICS** นั้น นอกจากจะสามารถระบุค่าแบบเจาะจงเป็น topic `/pieled/state` แล้ว ยังสามารถระบุค่าเป็น `/pieled/+` ก็ได้ โดยใช้เครื่องหมาย (+) ซึ่งเป็น **Single-Level Wildcard** เพื่อรับค่าของ State

เราสามารถใช **Wildcard** เพื่อช่วยในการ **Subscribe Topic** ต่างๆ เช่น หากต้องการ **Subscribe Topic** ตามที่ระบุแบบเจาะจงดังนี้: `"/home/kitchen/temp"`, `"/home/bedroom/temp"`, และ `"/home/livingroom/temp"` ก็สามารถยุบเหลือ **1 Topic** คือ `"/home/+temp"` นอกจาก `+` แล้วยังใช้เครื่องหมาย `#` ได้ด้วย โดยที่เครื่องหมาย `+` จะแทนค่าอะไรก็ได้ระดับชั้นเดียว ส่วน `#` จะแทนค่าอะไรก็ได้ในระดับชั้นยาวต่อไปเท่าไรก็ได้ เช่นจะให้ **match 3 Topic** ข้างต้น ก็อาจจะเขียนระบุเป็น **Topic** คือ `"/home/#"`