



Part 6: Conclusion

by NETPIE.io

Authentication: AppID, Key and Secret

NETPIE authenticate device ด้วย key และ secret

Key : SxYWm47l46eypBl

Secret Jlk9oWVKXfCmctRad7SsszLKv

แต่ละคู่ key/secret สามารถนำไปใช้กับ device ได้มากกว่า 1 ตัว เปรียบเสมือนกับการที่เรามี user/password ชุดเดียว แต่สามารถนำไปใช้ log in ได้พร้อมกันบนคอมพิวเตอร์หลายเครื่อง

Authentication: AppID, Key and Secret

การใช้ key ชุดเดียวกับทุก device มีข้อดีคือ สะดวก ทุก device สามารถ flash โดยใช้ firmware ตัวเดียวกันได้เลย

แต่มีข้อเสียคือ ถ้า key รั่วไหล การยกเลิก key (revoke) จะส่งผลกระทบต่อทุก device

ดังนั้น ในการนำไปใช้งานจริง ถ้าเป็นไปได้ แนะนำให้ใช้ 1 key กับ 1 device

Authentication: AppID, Key and Secret

Key ทุกอันบน APPID เดียวกันมีสิทธิ์เท่ากัน

- ตั้ง alias ตัวเองเป็นอะไรก็ได้
- chat ไปหาใครก็ได้ ภายใต APPID เดียวกัน
- publish ไปที่ไหนก็ได้ ภายใต APPID นี้
- subscribe รับข้อมูลจาก topic ได้ก็ได้

การ customize สิทธิ์ของ key เป็นฟีเจอร์อนาคต ที่กำลังจะเปิดให้ใช้

Authentication: AppID, Key and Secret

Key ของ NETPIE มี 2 ชนิด

- **Device Key** เหมาะสำหรับใช้บน physical device ที่ซึ่งเราอยากให้ identity ถูกฝังติดไปเพื่อที่จะมองว่าเป็น device เดิมไปตลอด ไม่ว่าจะ online หรือ offline ก็รอบ
- **Session Key** เหมาะสำหรับใช้กับ software หรือ browser ที่ซึ่ง identity มีอยู่เฉพาะตอน online เท่านั้น ไม่มีการฝังจำ หากปิดเปิดใหม่ก็จะมองเป็นตัวใหม่

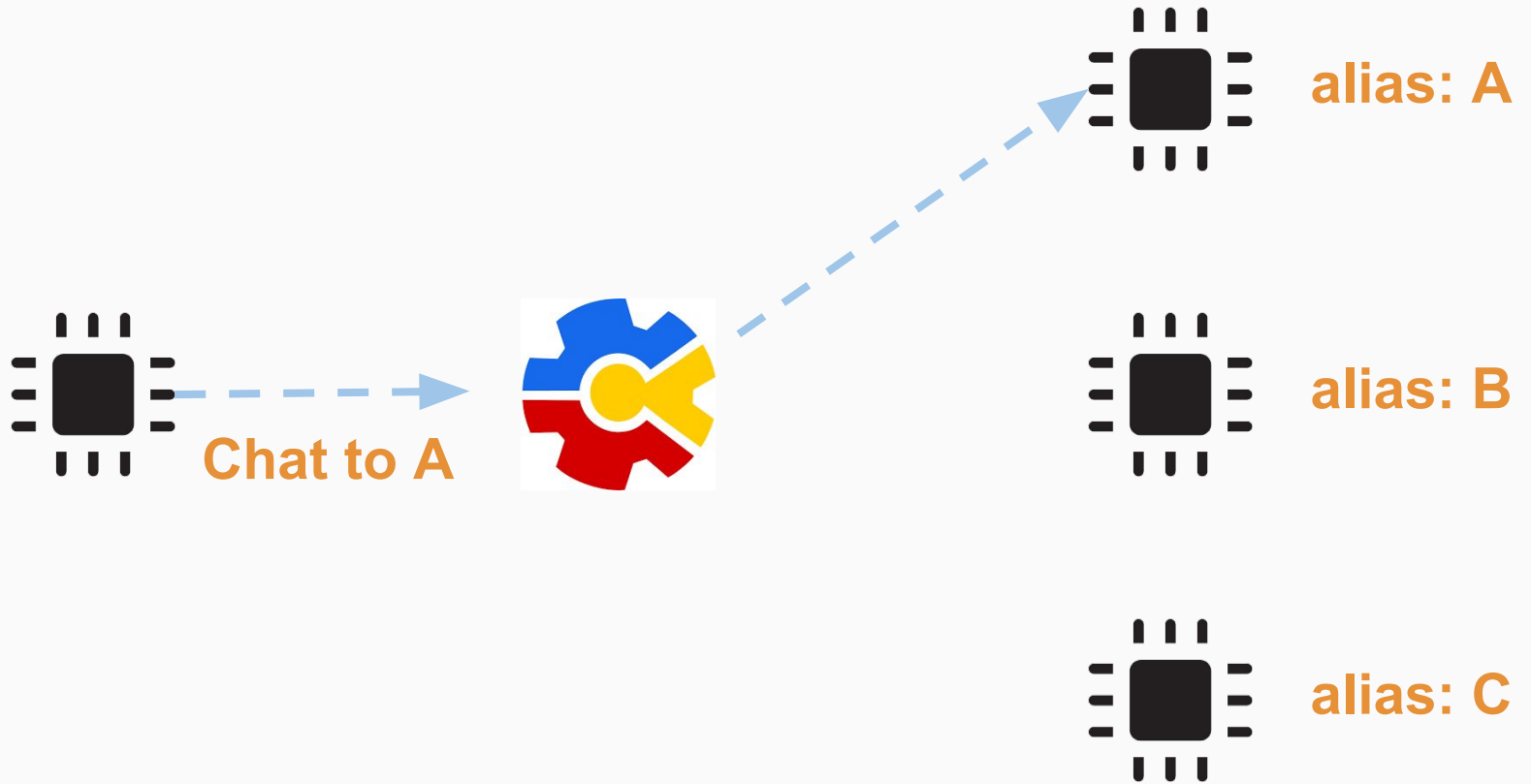
Authentication: AppID, Key and Secret

ความแตกต่าง

Session Key เมื่อเปิด client หรือ browser รายชื่อ thing ที่ online บน netpie application management จะหายไป ผู้ใช้จะได้ quota คืนมา

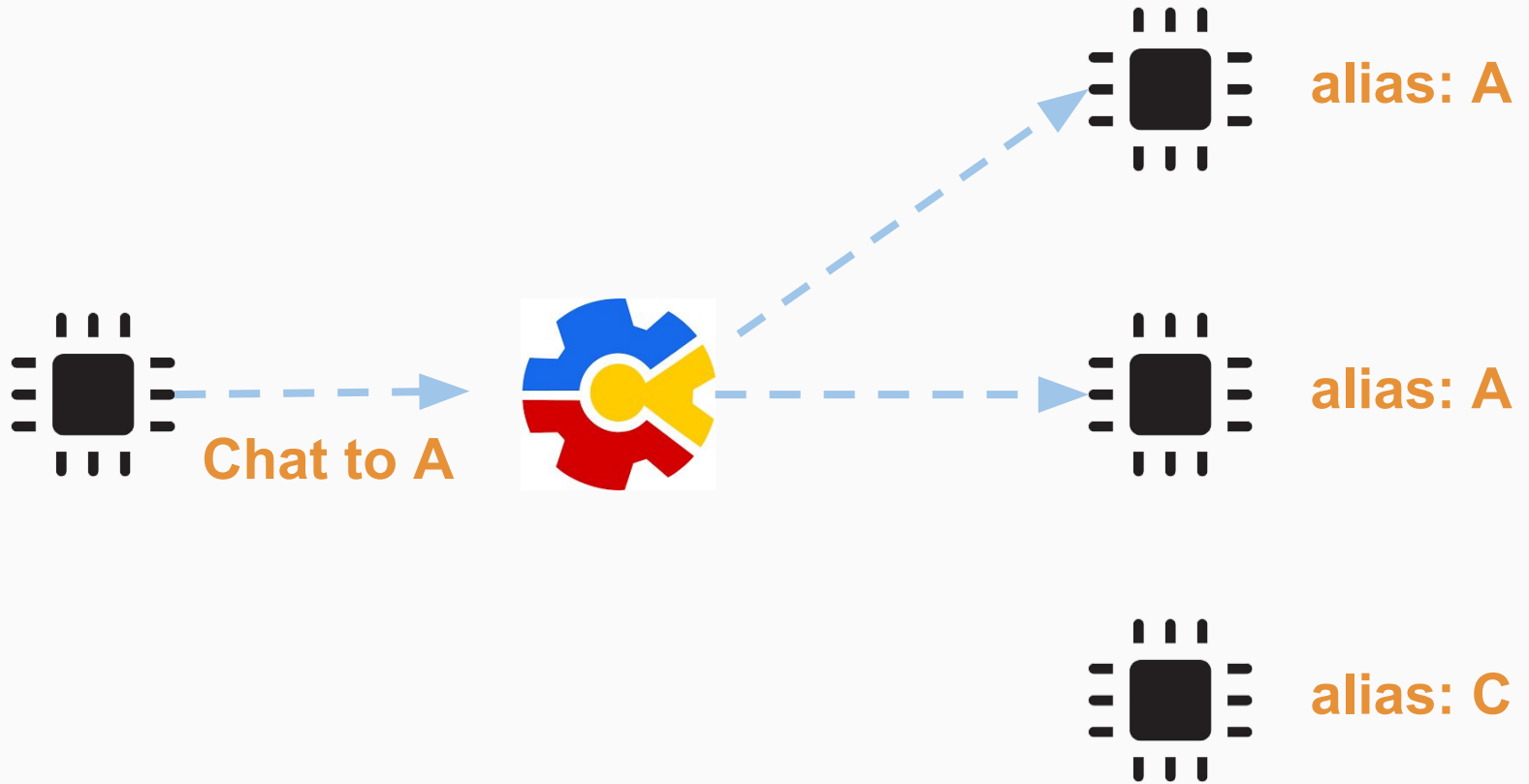
Device Key เมื่อ online ครั้งแรก id จะฝังไปบน device ทำให้ quota ลดลง จะไม่คืนมาแม้ว่าจะ offline device ลงไปก็ตาม ถ้าจะเลิกใช้ต้องลบออกเอง

Messaging: Chat



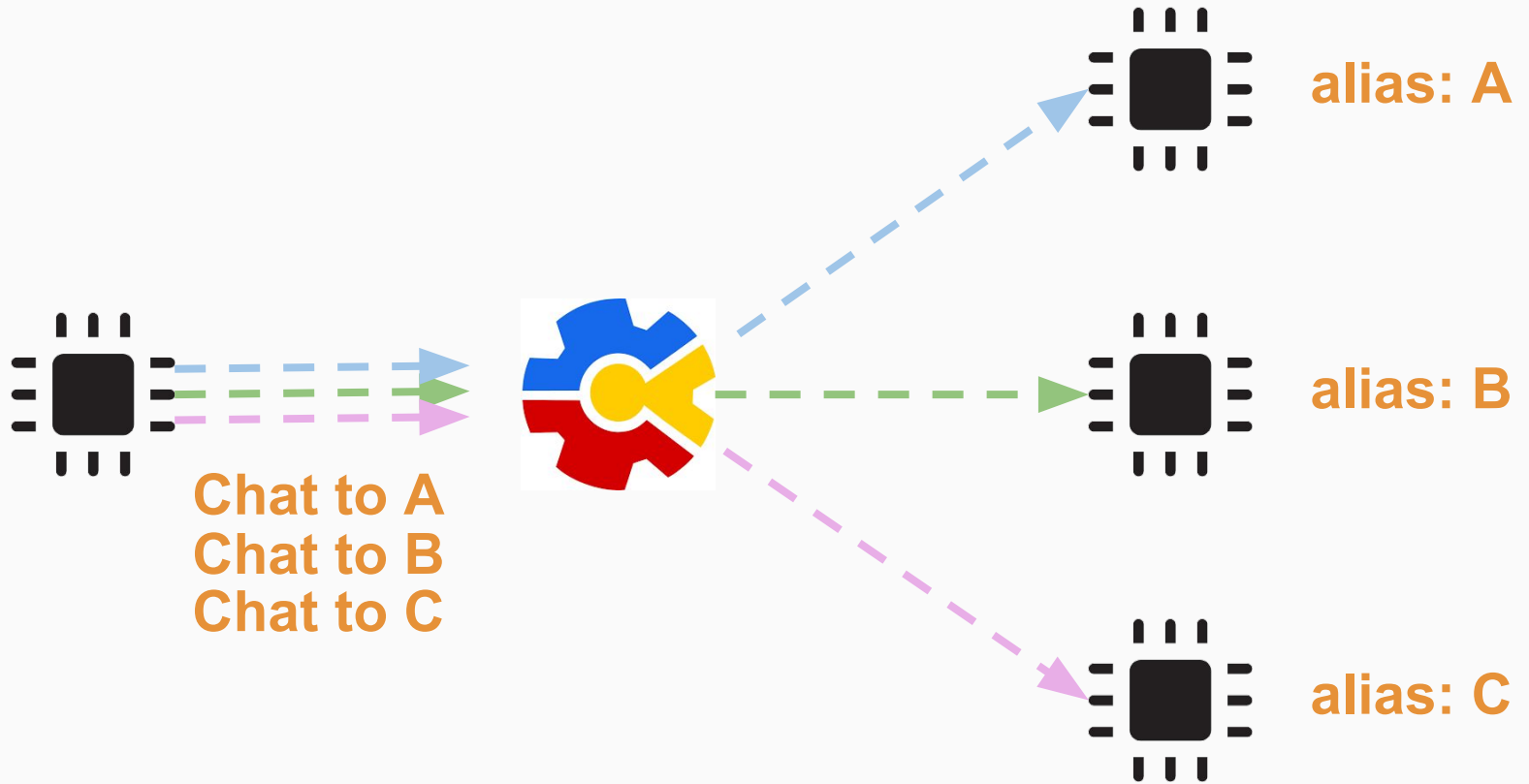
Chat เป็นการสื่อสารแบบเจาะจงผู้รับ

Messaging: Chat



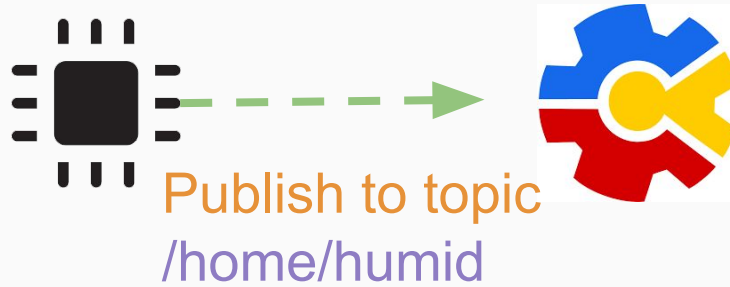
ถ้ามี device ตั้ง alias ซ้ำกัน จะได้รับ message ทั้งคู่

Messaging: Chat



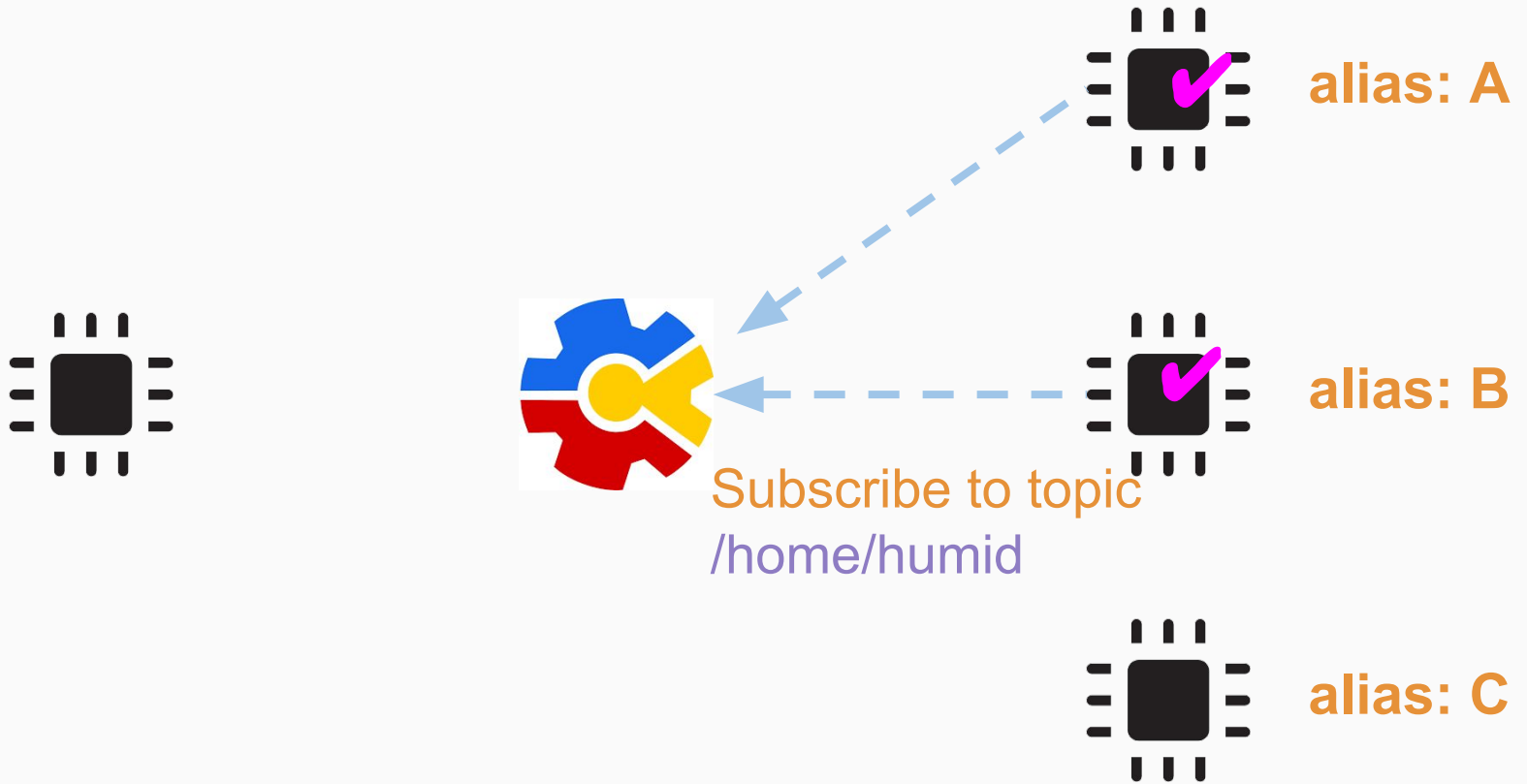
ถ้าจะส่งข้อมูลเดิมไปหลายๆ device ต้อง chat ซ้ำหลายครั้ง!

Messaging: Publish



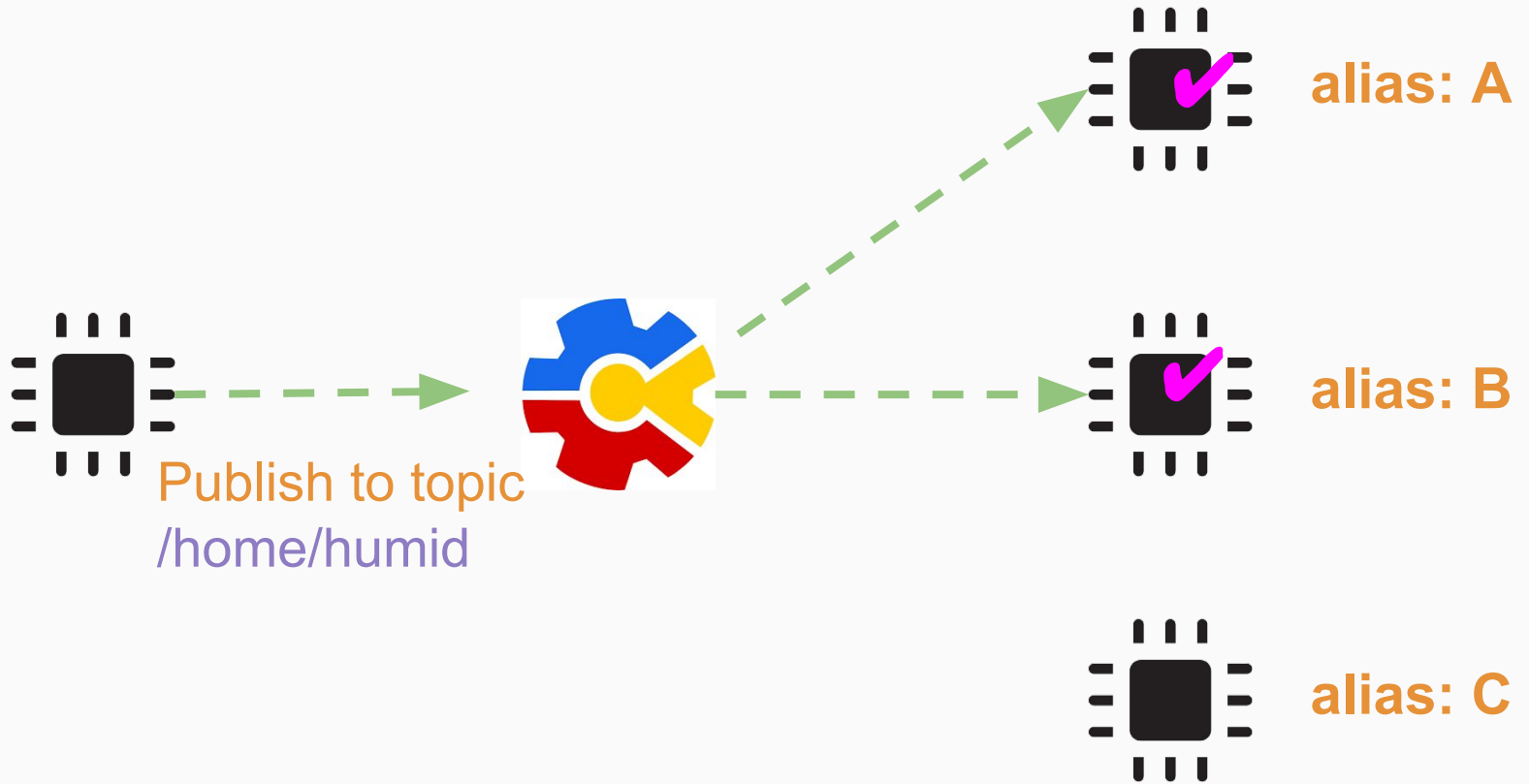
เราสามารถใช่วิธี publish ไปที่ topic แทนได้

Messaging: Publish



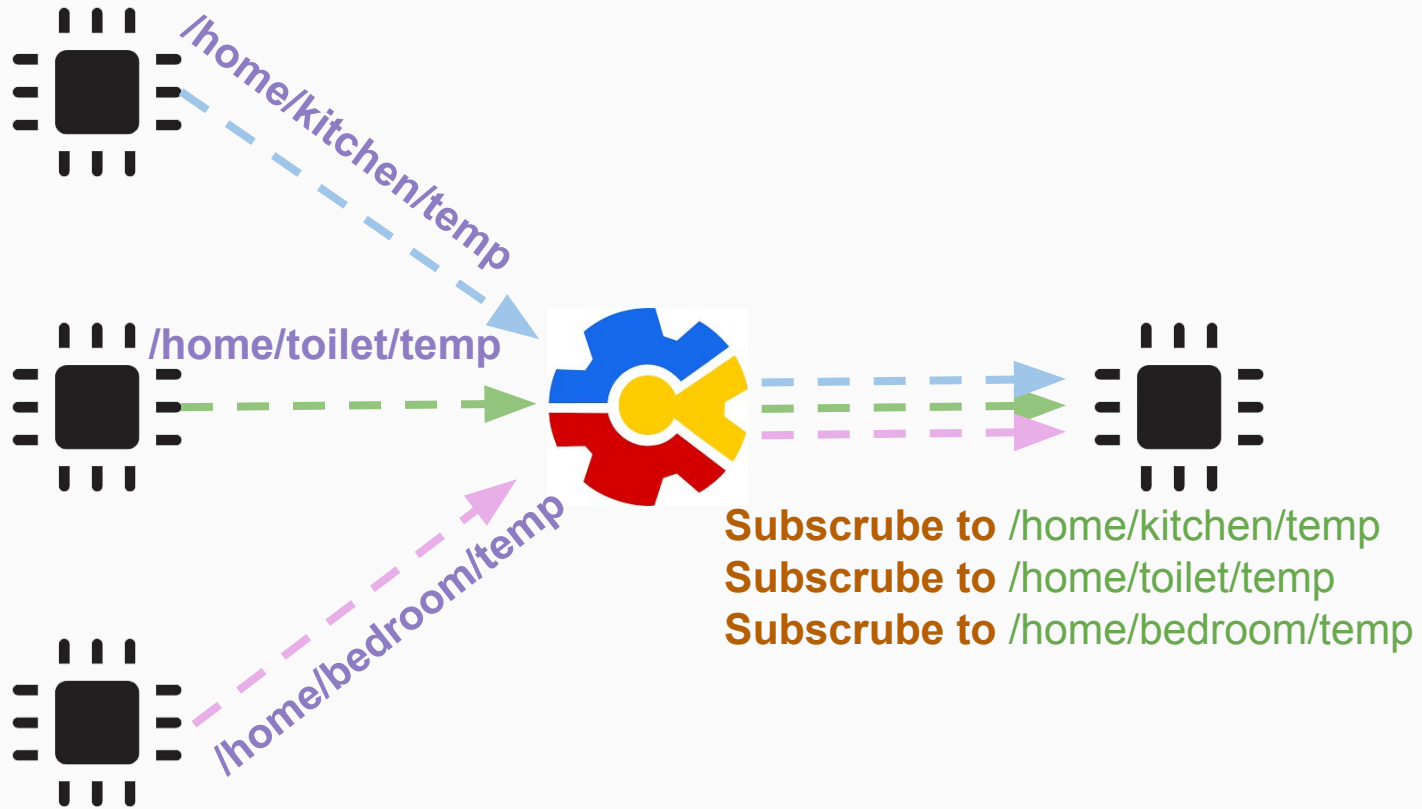
แต่คนที่จะได้รับจะต้อง subscribe topic นั้นๆ ก่อน

Messaging: Publish



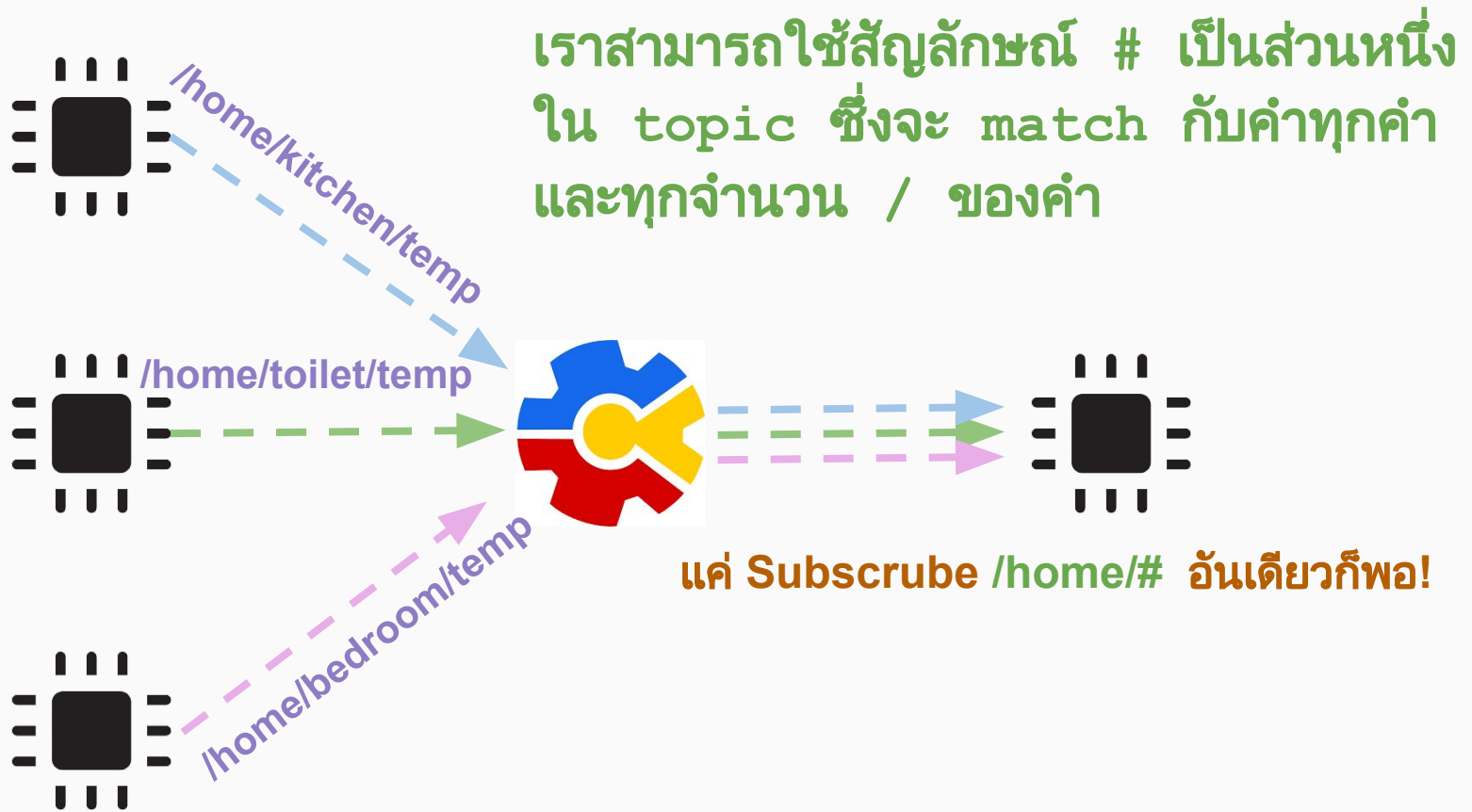
จากที่ต้องส่งหลายครั้ง เหลือส่งเพียงครั้งเดียว

Messaging: Topic Wildcard



ถ้าจะรับข้อมูลหลายจาก topic ต้อง subscribe หลายครั้ง!

Messaging: Topic Wildcard



ถ้าเรา subscribe topic **/#** จะเทียบเท่ากับการ Subscribe ALL

Messaging: Topic Wildcard

wildcard # ต้องใช้ที่ท้ายประโยคเท่านั้น
ตัวอย่าง การ subscribe topic นี้

`/home/ground/#`

จะได้รับ message ที่ publish เข้า topic ต่อไปนี้

✓ `/home/ground/kitchen/humid`

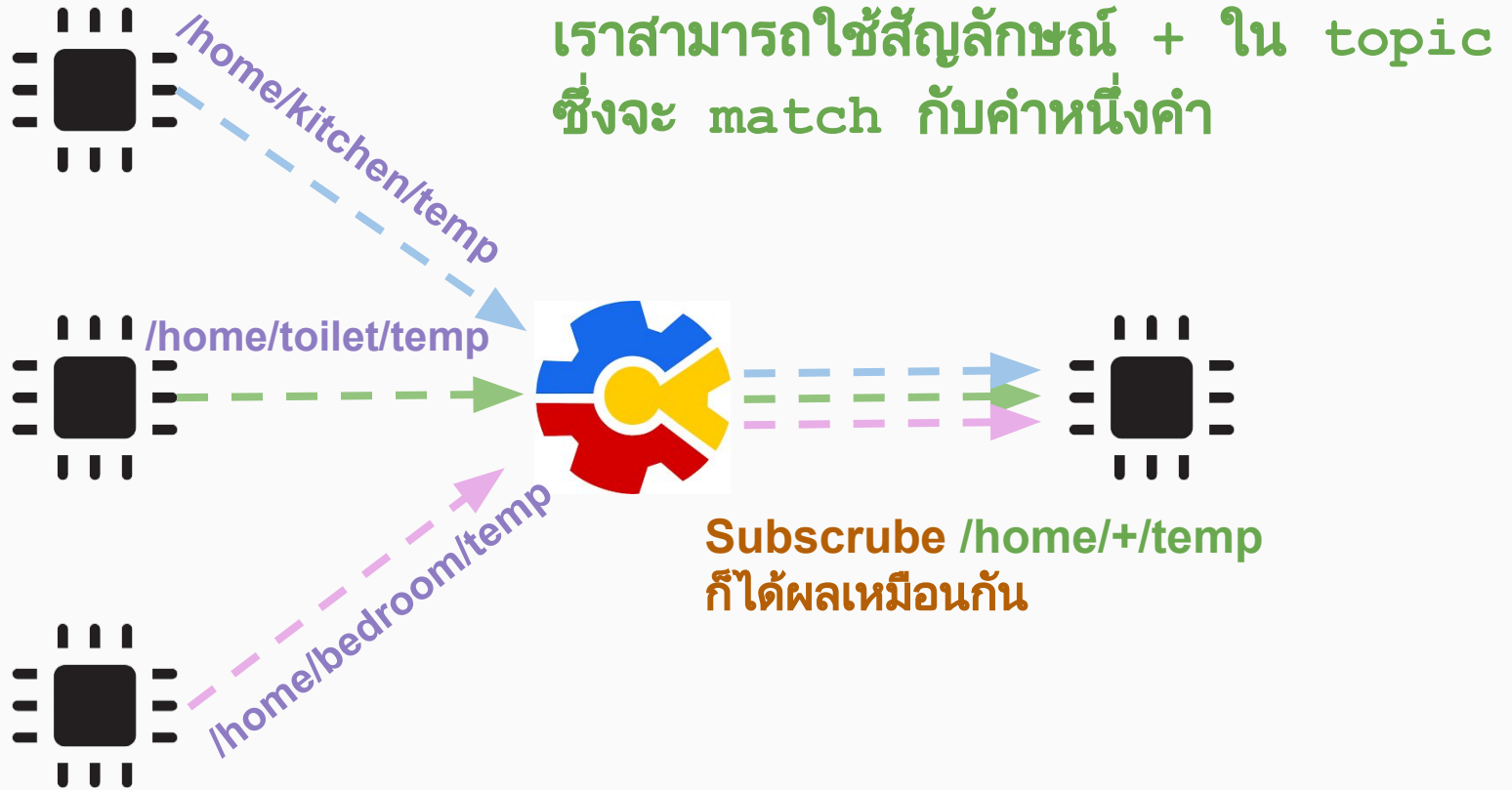
✓ `/home/ground/toilet/temp`

✓ `/home/ground/kitchen`

✓ `/home/ground`

✗ `/home/upstairs/bedroom/temp`

Messaging: Topic Wildcard



Messaging: Topic Wildcard

ตัวอย่าง การ subscribe topic นี้

`/home/ground/+/temp`

จะได้รับ message ที่ publish เข้า topic ต่อไปนี้

✓ `/home/ground/kitchen/temp`

✓ `/home/ground/toilet/temp`

✗ `/home/ground/bedroom`

✗ `/home/ground/kitchen/temp/oven`

✗ `/home/upstairs/bedroom/temp`

Messaging: Topic Wildcard

QUIZ : ถ้าเรา publish message ไปที่ topic นี้

/home/ground/kitchen/temp

Subscriber คนไหนจะได้รับ message บ้าง?

? /#

? /home/#

? /+/ground/#

? /home/ground/+

? /+/+/+/+/+

? /home/#/+

Messaging: Topic Wildcard

QUIZ : ถ้าเรา publish message ไปที่ topic นี้

`/home/ground/kitchen/temp`

Subscriber คนไหนจะได้รับ message บ้าง?

✓ `/#`

? `/home/#`

? `/+/ground/#`

? `/home/ground/+`

? `/+/+/+/+`

? `/home/#/+`

Messaging: Topic Wildcard

QUIZ : ถ้าเรา publish message ไปที่ topic นี้

`/home/ground/kitchen/temp`

Subscriber คนไหนจะได้รับ message บ้าง?

✓ `/#`

✓ `/home/#`

? `/+/ground/#`

? `/home/ground/+`

? `/+/+/+/+`

? `/home/#/+`

Messaging: Topic Wildcard

QUIZ : ถ้าเรา publish message ไปที่ topic นี้

`/home/ground/kitchen/temp`

Subscriber คนไหนจะได้รับ message บ้าง?

✓ `/#`

✓ `/home/#`

✓ `/+/ground/#`

? `/home/ground/+`

? `/+/+/+/+`

? `/home/#/+`

Messaging: Topic Wildcard

QUIZ : ถ้าเรา publish message ไปที่ topic นี้

`/home/ground/kitchen/temp`

Subscriber คนไหนจะได้รับ message บ้าง?

✓ `/#`

✓ `/home/#`

✓ `/+/ground/#`

✗ `/home/ground/+`

? `/+/+/+/+`

? `/home/#/+`

Messaging: Topic Wildcard

QUIZ : ถ้าเรา publish message ไปที่ topic นี้

`/home/ground/kitchen/temp`

Subscriber คนไหนจะได้รับ message บ้าง?

✓ `/#`

✓ `/home/#`

✓ `/+/ground/#`

✗ `/home/ground/+`

✓ `/+/+/+/+`

? `/home/#/+`

Messaging: Retained Publish

การ publish แบบคงค่า (retained)

เวลาที่เร publish ข้อมูลเข้าไปใน NETPIE ทาง device ฝั่งรับจะได้ข้อมูลก็ต่อเมื่อมัน online และ subscribe topic นั้นไว้รออยู่ก่อนแล้ว สำหรับ device ที่ online มาทีหลัง ไม่ทันได้รับ message ก็จะพลาดไป

เราสามารถใช้งานการ publish แบบ retained ในการบังคับให้ข้อมูลล่าสุดที่เรา publish ไปนั้น ถูกจัดเก็บบน NETPIE เพื่อที่ device ที่มา subscribe topic นั้นทีหลัง จะยังได้เห็นค่าสุดท้ายอยู่

Messaging: Retained Publish

การ publish แบบคงค่า (retained)

วิธีการสั่งให้การ publish ทำการคงค่า (retained) ล่าสุดไว้ ทำได้โดย
การส่ง parameter boolean เป็น true ต่อท้าย

```
microgear.publish("/home/bedroom/temp", "25.4", true);
```

Messaging: Retained Publish

Device ที่ subscribe topic /home/bedroom/temp หรือ topic แบบ wildcard ที่ match กับ /home/bedroom/temp จะได้รับ retained message เสมือนกับมี device อื่น publish message นี้เดี๋ยวนั้น

วิธีการเคลียร์ค่าที่ retain ทำได้โดย publish string เปล่า เข้าไปที่ topic เดิมแบบเซตเงื่อนไข retained = true เช่น

```
microgear.publish("/home/bedroom/temp", "", true);
```

REST API

REST API

นอกจาก microgear library แล้ว NETPIE ยังเตรียมช่องทางการเชื่อมต่ออีกหนึ่งช่องทาง คือผ่านทาง REST API วิธีนี้จะเหมาะสำหรับใช้กับ hardware platform ที่ยังไม่สามารถใช้ microgear library ได้ หรือการใช้ integrate กับ platform อื่น

<https://github.com/netpieio/microgear-restapi>

REST API

รูปแบบการ publish ผ่าน REST API

```
curl -X PUT
"https://api.netpie.io/topic/myappid/home/bedroom/light?retain"
-u jVjzJXaJwdJKHhF:StOAKIZhXB5CaqnIHeb7s1DfiW7mQj -d "ON"
```

Diagram illustrating the components of the REST API command:

- APPID**: Points to `myappid` in the URL.
- Topic**: Points to `/home/bedroom/light` in the URL.
- Key: Secret**: Points to the username `jVjzJXaJwdJKHhF` in the authentication string.
- Message content**: Points to the data `"ON"` in the `-d` flag.

จากตัวอย่าง เป็นการ publish เข้า topic `/home/bedroom/light` ภายใต้ AppID `myappid` โดย authenticate ด้วย `key:secret jVjzJXaJwdJKHhF:StOAKIZhXB5CaqnIHeb7s1DfiW7mQj` และมีข้อความที่ส่งไปว่า `ON` ส่วนของ `?retain` เป็นตัวเลือกที่ใส่เฉพาะตอนที่อยากจะทำให้การคงค่าล่าสุดไว้ใน topic นี้

REST API

รูปแบบการ subscribe ผ่าน REST API

```
curl -X GET "https://api.netpie.io/topic/myappid/home/bedroom/light"  
-u jjVjzJXaJwdJKHhF:StOAKIZhXB5CaqnIHeb7s1DfiW7mQj
```



Key: Secret

การ subscribe ด้วย REST API จะใช้ได้เฉพาะกับ topic ที่มีการ publish แบบคงค่า retained เท่านั้น การเรียกคำสั่ง subscribe ผ่าน REST API จะเป็นการอ่านค่าสุดท้ายของข้อมูลล่าสุดที่ถูก publish เข้า topic นั้น

REST API

รูปแบบการ chat ผ่าน REST API

```
curl -X PUT  
"https://api.netpie.io/microgear/myappid/myalias"  
-u jVjzJXaJwdJKHhF:StOAKIZhXB5CaqnIHeb7s1DfiW7mQj -d "ON"
```

Diagram labels:

- APPID** points to **myappid**
- Alias** points to **myalias**
- Key:Secret** points to **jVjzJXaJwdJKHhF:StOAKIZhXB5CaqnIHeb7s1DfiW7mQj**
- Message content** points to **"ON"**

คำสั่งข้างต้นเป็นการ chat ไปหา alias ชื่อ myalias ภายใต้ APPID myappid โดย message content คือคำว่า ON

REST API



ตัวอย่าง

หลอดไฟ IOT ต่อ NETPIE

APPID ชื่อ **deerdev**

มี alias ว่า **pieslampher2**

เราสร้าง key/secret ไว้ดังนี้

Key: **LWLq4ijV5Kq66Jp**

Secret: **4GL5gU270mteQPtS9RDqafuPI**

คำสั่งเปิดไฟ:

```
curl -X PUT
```

```
"https://api.netpie.io/microgear/deerdev/pieslampher2"
```

```
-u LWLq4ijV5Kq66Jp:4GL5gU270mteQPtS9RDqafuPI -d "1"
```

REST API



คำสั่งปิดไฟ:

```
curl -X PUT
```

```
"https://api.netpie.io/microgear/deerdev/pieslampher2"
```

```
-u LWLq4ijV5Kq66Jp:4GL5gU270mteQPtS9RDqafuPI -d "0"
```


Freeboard Tricks

- **Private-Host Freeboard**
 - Download freeboard package from github.com/netpieio
 - Host it on your web browser
 - Add user management as you wish
- **Multiple applications on the same freeboard**
 - Add multiple datasources (from multiple applications)

Network Requirements

- WIFI with WEP, WPA, WPA2 Personal security
- Doesn't work with WPA2 Enterprise
- Doesn't work with Web-based Authentication

- Allow special ports 1883, 8080, 8081, 8083, 8084, 8883
- Best practice: 1 Access point per 30 devices (NodeMCUs)

Free Account Limits

100 credits

10 Applications

1 credit : 1 device

1 credit : 1 Freeboard datasource

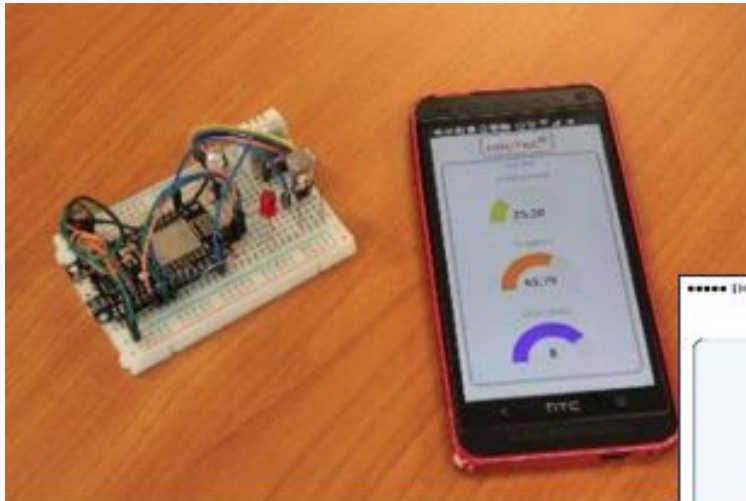
4 credits : 1 data field

Write feed 4 times/minute

Applications

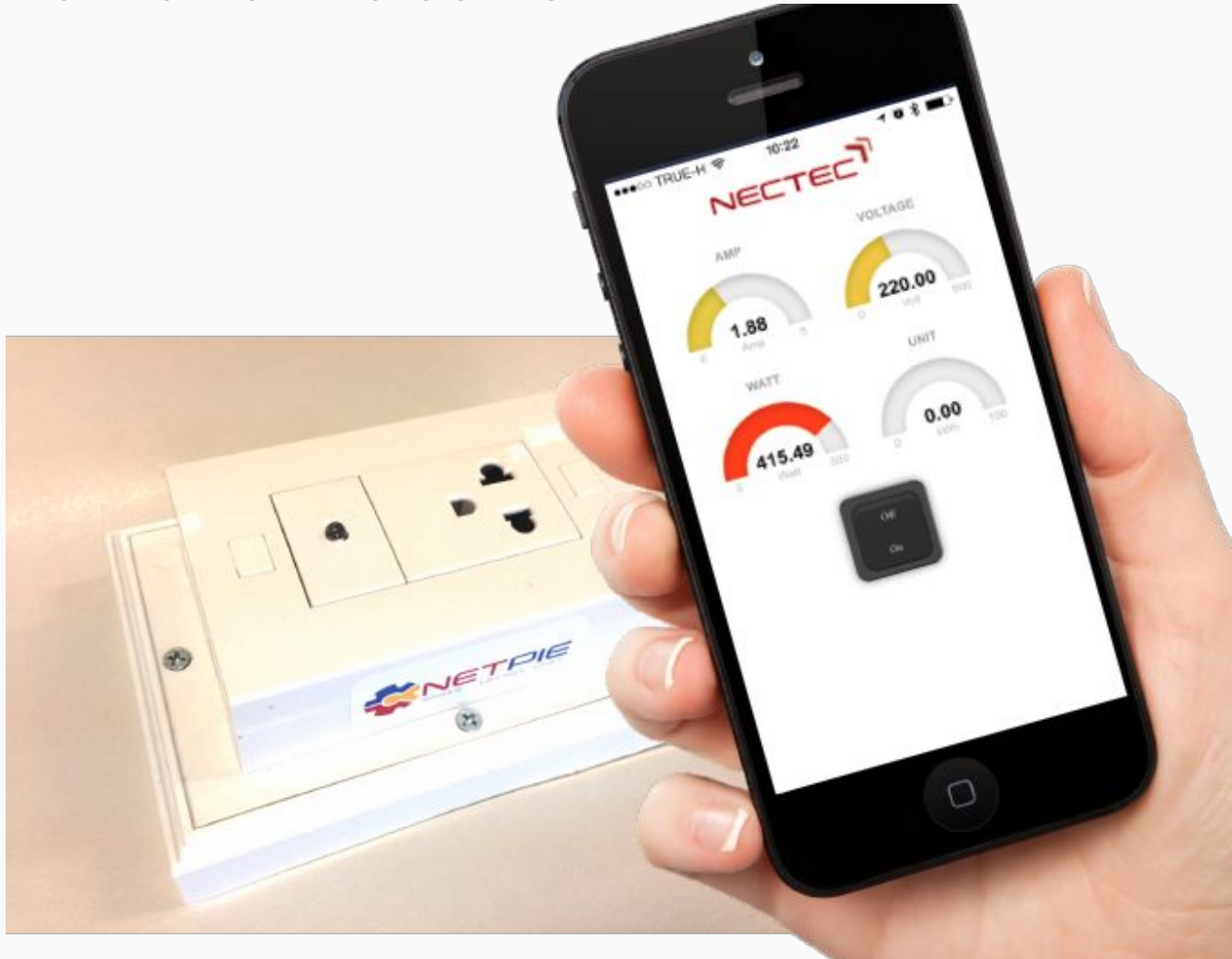
Some Use Cases

Sensor วัดความชื้นดิน



Some Use Cases

ปลั๊กไฟวัดพลังงาน



Some Use Cases

Sensor วัดการใช้พลังงานไฟฟ้าแบบคล่อง



Some Use Cases

T-WiLS Sensor วัดอุณหภูมิแบบพกพา



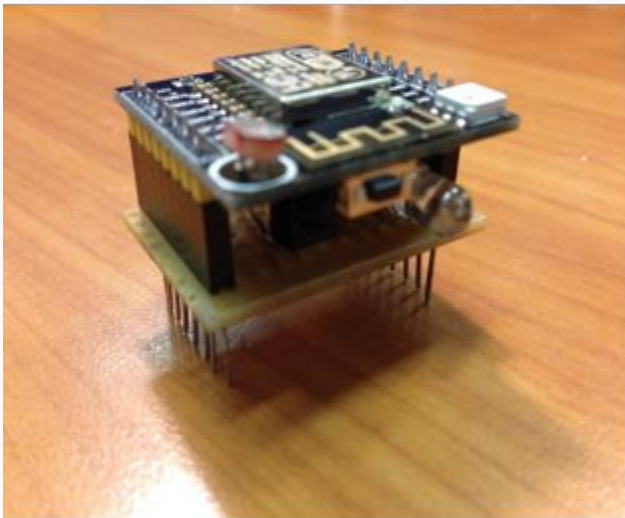
Some Use Cases

เครื่องให้อาหารปลา IoT



Some Use Cases

รีโมตคอนโทรล IOT

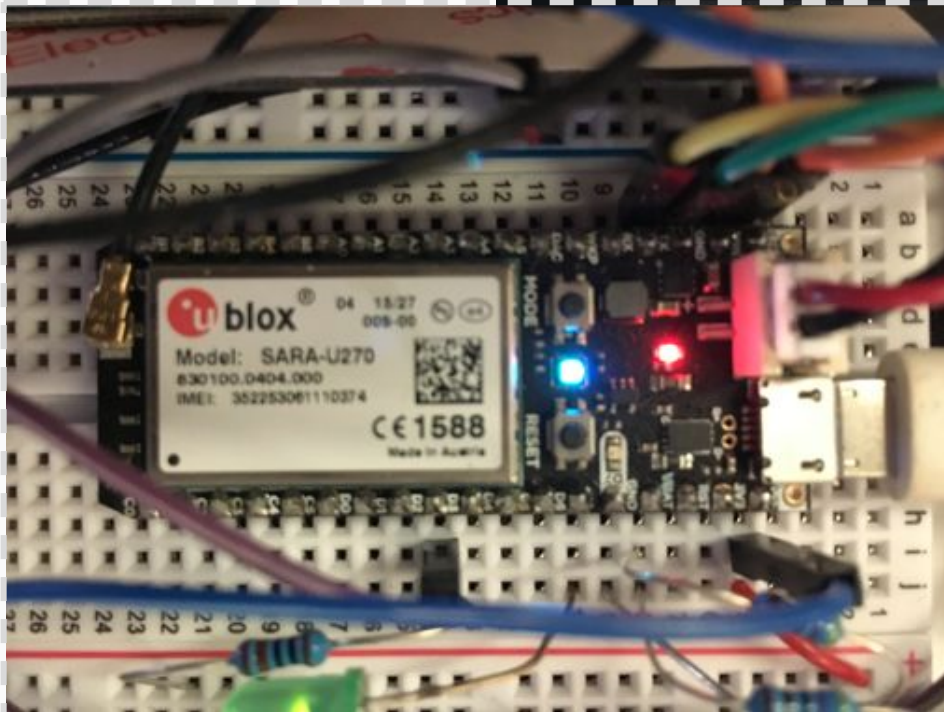
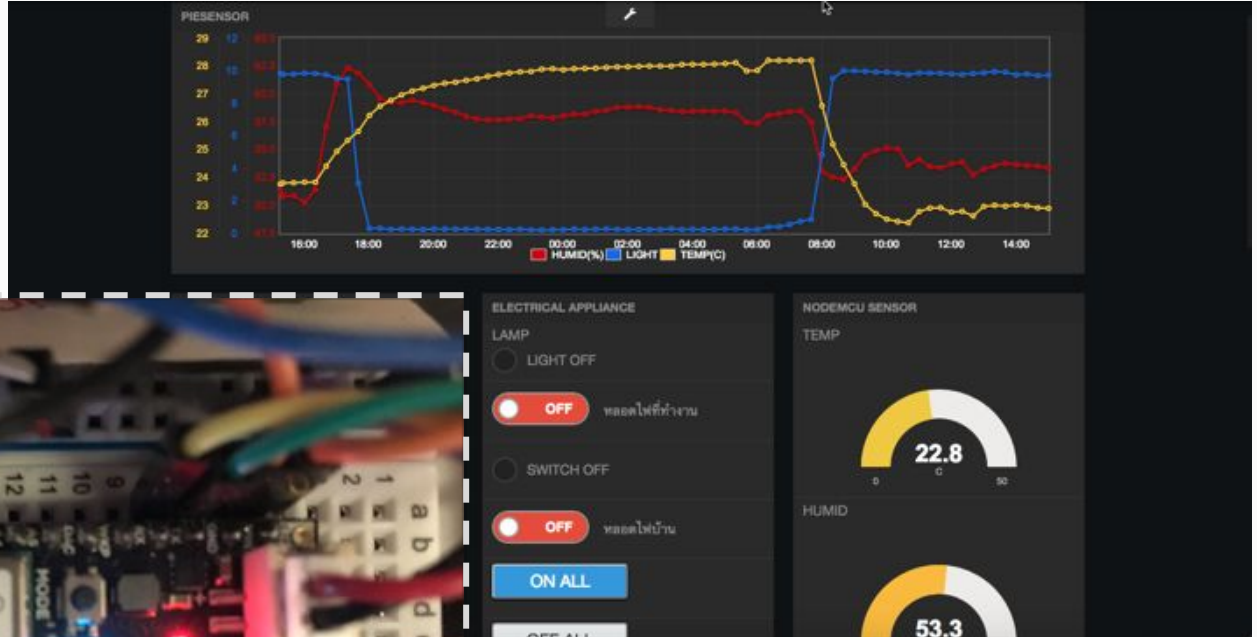


Demo: Remote Sensor

LinkIt Smart 7688Duo using Python



Demo: 3G Realtime Sensor

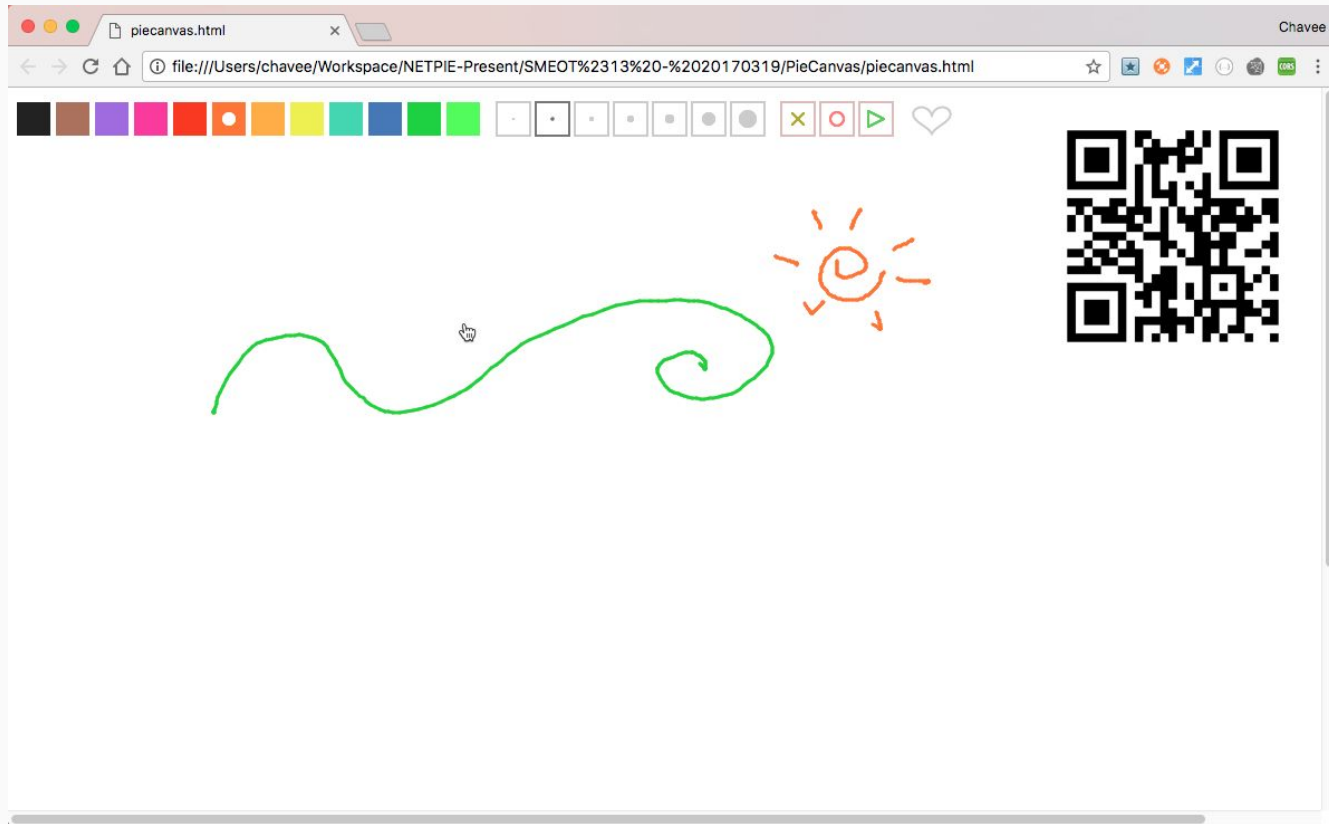


Particle Electron 3G

SET Index Bulb



Demo: Web Canvas



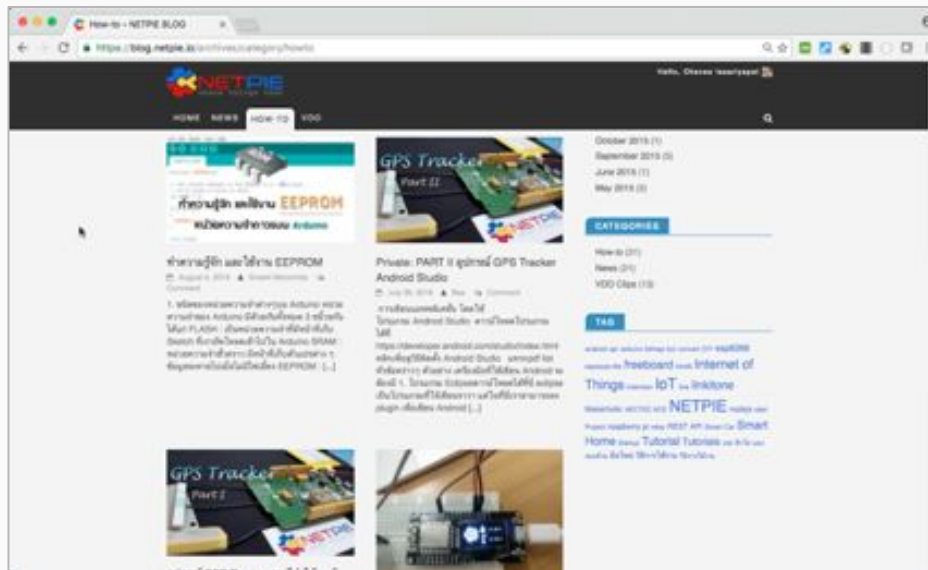
<http://chavee.com/piecanvas.html>

Demo: PieBingo



Further Study

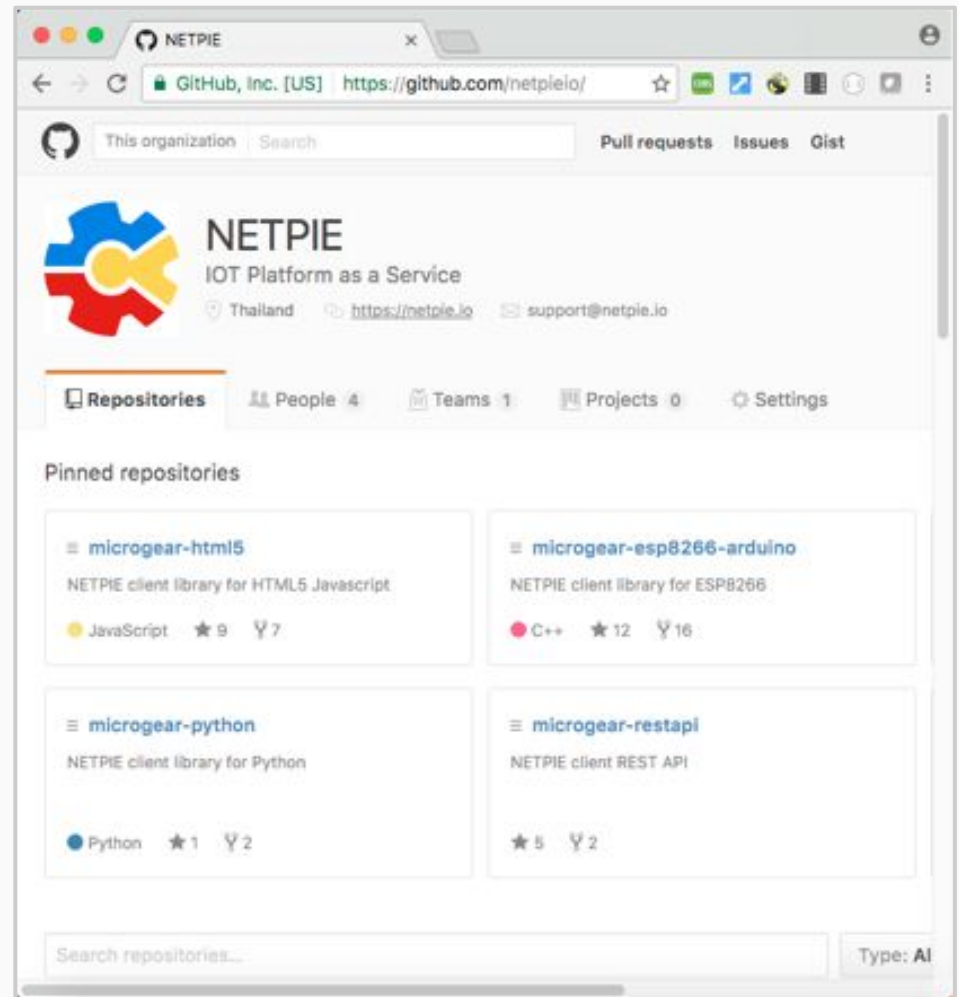
<https://blog.netpie.io>



Further Study


- NodeJS
- Python
- HTML5 Javascript
- Arduino
- ESP8266 Arduino
- Android Studio
- C#
- JAVA
- RESTful API
- More coming soon..

<https://github.com/netpieio>



Further Study

<https://netpie.io/tutorials>




Google Custom Search

HOME PARTNERS DEVELOPERS BLOG

LOG IN SIGN UP FREE


Home » Tutorials

E-BOOK




Official Guide

HARDWARE E-LEARNING




Arduino
LAB & VDO




NodeMCU / ESP8266
LAB & VDO


NETPIE FEATURE TUTORIAL




Feed




Freeboard



REST API



Secure Connection



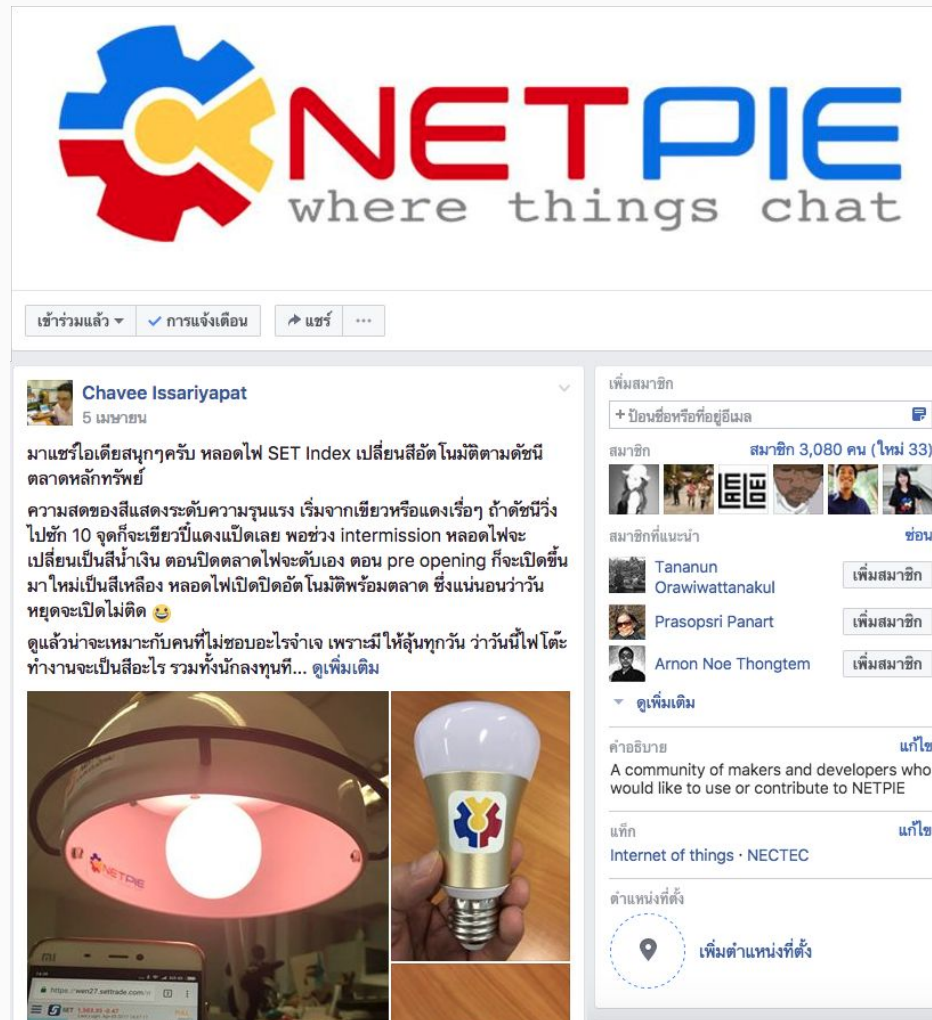
Get Started

DIY Projects

- Smart Plug
- PIE Sensor
- Plant Sensor
- CPU and Memory Load Monitor
- Alan chat bot (by Comdet Phudphut)
- Wifi Smart Plug (by Soft Power Group)
- Control LED with HTML5 (by TESR)
- Read ADC data with HTML5 (by TESR)
- Bingo Game
- Smart Car Robot
- Smart Home with NodeMCU (by Narin Bannasan)
- NETPIE REST API on Freeboard (by Anuchit Chalothorn)
- plant watering system (by Thanakorn Yodrat)
- สนุกกับ smart plug ตอนที่ 1 : มา hack สวิตช์ Sonoff! ให้ต่อ NETPIE กันเถอะ
- สนุกกับ smart plug ตอนที่ 2 : สวิตต์ HTML5
- สนุกกับ smart plug ตอนที่ 3 : REST API ใครๆก็คุยกับปลั๊กได้
- มาโปรแกรมหุ่นขี้หอดไฟ Slampher ให้เป็น NETPIE device กันเถอะ
- สอนการใช้งาน NETPIE บน ESP8266 Lite 2
- แนะนำการใช้งาน NETPIE Freeboard Part 1 (by Narin Bannasan)
- แนะนำการใช้งาน NETPIE Freeboard Part 2 (by Narin Bannasan)
- แนะนำการใช้งาน NETPIE Freeboard Part 3 (by Narin Bannasan)
- เล่นกับ NETPIE ง่ายๆ ด้วย Smart Light Indicator! (by Narin Bannasan)
- มาทำ Smart Car ด้วย NETPIE กันเถอะ! (by Narin Bannasan)
- อุปกรณ์ GPS Tracker คุณก็ทำได้นะ รู้ยัง...!! (by Chutikam Siriwimonwan)
- มาเล่น RGB NeoPixel Ring + NETPIE กันเถอะ
- มาทำพัดลมให้เป็นพัดลม IOT กันเถอะ
- See more projects

Further Study

<https://www.facebook.com/groups/netpie>



NETPIE
where things chat

เข้าร่วมแล้ว ▾ ✓ การแจ้งเตือน ➔ แชร์ ...

Chavee Issariyapat
5 เมษายน

มาแชร์ไอเดียสนุกๆครับ หลอดไฟ SET Index เปลี่ยนสีอัตโนมัติตามดัชนีตลาดหลักทรัพย์

ความสดของสีแสดงระดับความรุนแรง เริ่มจากเขียวหรือแดงเรื่อยๆ ถ้าดัชนีวิ่งไปซีก 10 จุดก็จะเขียวบีบแดงแปดเลย พอช่วง intermission หลอดไฟจะเปลี่ยนเป็นสีน้ำเงิน ตอนปิดตลาดไฟจะดับเอง ตอน pre opening ก็จะมีเปิดขึ้นมาใหม่เป็นสีเหลือง หลอดไฟเปิดปิดอัตโนมัติพร้อมตลาด ซึ่งแน่นอนว่าวันหยุดจะเปิดไม่ติด 😊

ดูแล้วน่าจะเหมาะกับคนที่ไม่ชอบอะไรจำเจ เพราะมีให้ลุ้นทุกวัน ว่าวันนี้ไฟโต๊ะทำงานจะเป็นสีอะไร รวมทั้งนั่งลงทุนที่... ดูเพิ่มเติม

เพิ่มสมาชิก
+ ป้อนชื่อหรือที่อยู่อีเมล

สมาชิก 3,080 คน (ใหม่ 33)

สมาชิกที่แนะนำ ช้อน

Tananun Orawiwattanakul เพิ่มสมาชิก

Prasopsri Panart เพิ่มสมาชิก

Arnon Noe Thongtem เพิ่มสมาชิก

ดูเพิ่มเติม

คำอธิบาย แก้ไข
A community of makers and developers who would like to use or contribute to NETPIE

แท็ก แก้ไข
Internet of things · NECTEC

ตำแหน่งที่ตั้ง
เพิ่มตำแหน่งที่ตั้ง

Keep in touch

Facebook Group "NETPIE"

<https://www.facebook.com/groups/netpie/>

For more questions, contact

support@netpie.io

Q&A