



คู่มือ

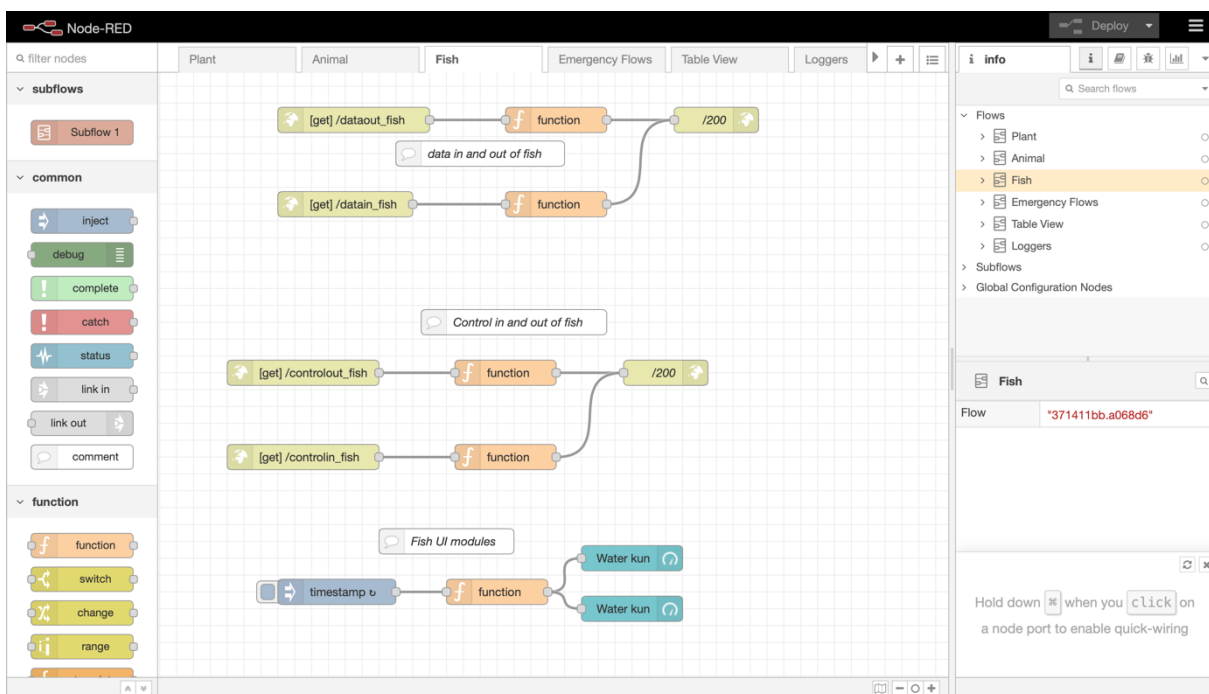
หลักการทำงาน การใช้งาน

และการแก้ไข

Server Node-red

บทนำ

Node-red (อ่านว่า โนด-เร็ด) เป็นเครื่องมือพัฒนาโปรแกรมด้วยภาพ (Visual programming) โดยมีแนวคิดในการทำงานแบบ Flow-base สำหรับ Internet of Things (IoT) สามารถใช้เชื่อมต่ออุปกรณ์ Hardware เข้ากับ APIs (Application Programming Interfaces) โดยมี User Interface (UI) ที่สามารถเข้าถึงได้จาก web browser ซึ่งสามารถเชื่อมข้อมูลจากแต่ละส่วนได้ง่าย

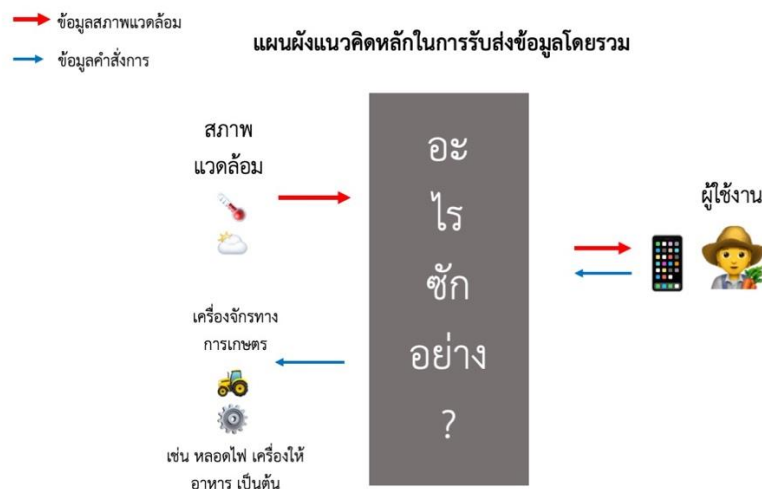


โปรแกรม Node-red

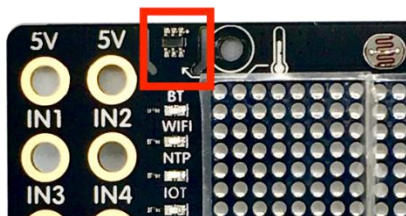
แนวคิด และหลักการทำงานของ Server โดยรวม

Server หรือ เซิร์ฟเวอร์นั้นเป็นคำเรียกสำหรับคอมพิวเตอร์ที่ทำงานเป็นแกนหลัก โดยมีคอมพิวเตอร์หรืออุปกรณ์อื่น ๆ เชื่อมต่ออยู่ด้วย ซึ่ง Server ที่นำมาใช้ในระบบ Smart Farm มีจุดมุ่งหมายเพื่อใช้เป็นระบบในการส่งข้อมูล โดยสามารถส่งข้อมูลสภาพแวดล้อมภายในฟาร์มไปยังหน้าเว็บของผู้ใช้งานได้ และสามารถรับคำสั่งผ่านหน้าเว็บไปยังเครื่องมือ ในที่นี้คือการควบคุมอุปกรณ์หรือเครื่องจักรต่างๆ ภายในฟาร์มของผู้ใช้งาน ซึ่งสามารถส่งการผ่านทางหน้าเว็บได้แบบ Real-time

สรุปได้ตามแผนผังดังต่อไปนี้



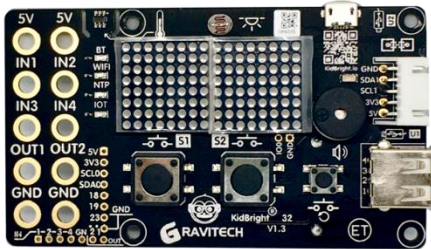
สภาพแวดล้อมภายในฟาร์ม (อุณหภูมิ ความชื้น ในดินและในอากาศ แสง ลม น้ำฝน) สามารถเปลี่ยนไปเป็นข้อมูลได้โดยการใช้อุปกรณ์ เซนเซอร์ (Sensor) ตามประเภท เช่น เซนเซอร์วัดอุณหภูมิ สามารถอ่านอุณหภูมิของอากาศบริเวณโดยรอบ และส่งค่าออกมาเป็นตัวเลข



เซนเซอร์วัดอุณหภูมิบนบอร์ด Kid Bright

โดยการจะอ่านค่าจากเซนเซอร์ต่าง ๆ ได้นั้นจำเป็นต้องเชื่อมต่อเซนเซอร์เข้ากับ **บอร์ด Micro-controller** (ในที่นี้คือ บอร์ด) ซึ่งเป็นคอมพิวเตอร์ขนาดเล็กที่อาศัยการเขียนโปรแกรม หรือชุดคำสั่งเพื่อให้สามารถทำงานได้ตามจุดประสงค์ (ในการที่จะทำให้บอร์ดอ่านค่าจากเซนเซอร์ และส่งค่าไปยัง Server Node-RED ได้อย่างสมบูรณ์ได้นั้น สามารถศึกษารายละเอียดเพิ่มเติมได้จากคู่มือในส่วนของฝ่ายงานเกษตรกรรม ฝ่ายปศุสัตว์ หรือฝ่ายการประมง)

โดยบอร์ดที่ใช้ภายในโครงงานนี้คือบอร์ด **Kid Bright** ซึ่งเป็นบอร์ดที่มี CPU หลักเป็นชิพ ESP32 ซึ่งเป็นชิพที่ใช้พลังงานต่ำ และสามารถเชื่อมต่อกับ Wi-Fi ได้ และยังมีเซนเซอร์อุณหภูมิ และเซนเซอร์แสง รวมไปถึงมีพอร์ตเชื่อมต่อเพิ่มเติมที่สามารถต่อเซนเซอร์ หรือสามารถต่อพ่วงกับมอเตอร์ หลอดไฟ หรืออุปกรณ์อื่น ๆ ได้



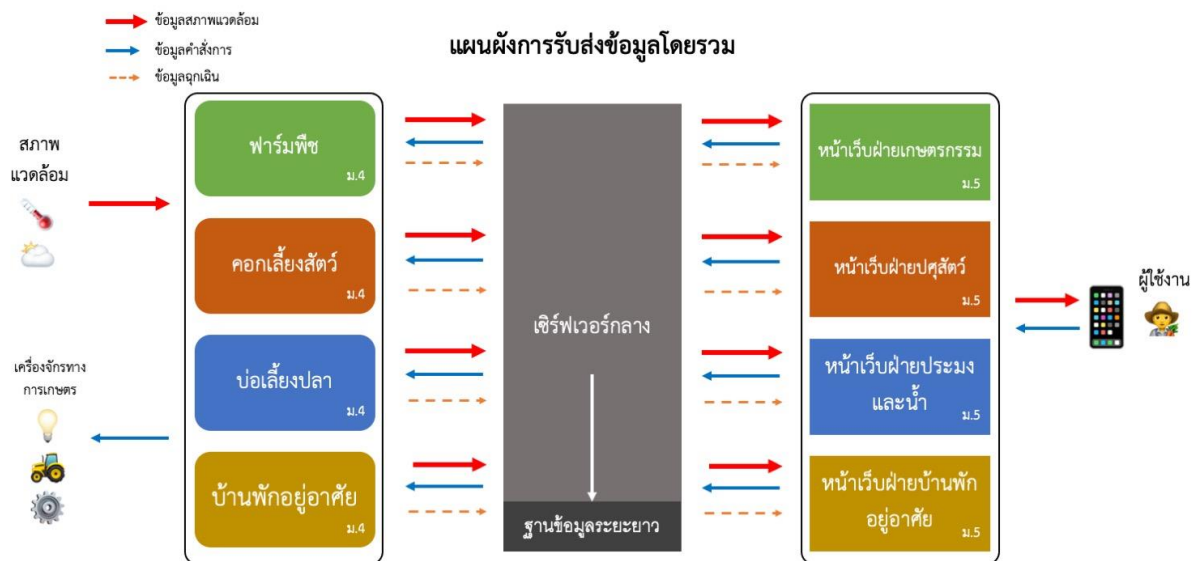
บอร์ด Kid Bright

จากการที่เราได้ทราบถึงหลักการทำงานเบื้องต้นของการทำงานของบอร์ดไปโดยสังเขป เราสามารถสรุปได้ออกมาเป็นแผนภาพดังนี้ได้



โดยข้อมูลสภาพแวดล้อมจะอ่านค่าโดยเซนเซอร์และส่งไปยังบอร์ด Kid Bright จากนั้นบอร์ดจะเก็บค่าข้อมูลเหล่านั้นตามรูปแบบของตัวแปรที่กำหนดไว้ แล้วจึงส่งตัวแปรไปยัง Server ผ่านสัญญาณ Wi-Fi Server จะทำการวิเคราะห์ข้อมูล ก่อนจะส่งไปยังหน้าเว็บผ่านสัญญาณ Wi-Fi และในขณะเดียวกันก็จะนำข้อมูลในรูปแบบตัวแปรเก็บไว้ในฐานข้อมูลระยะยาว (Database)

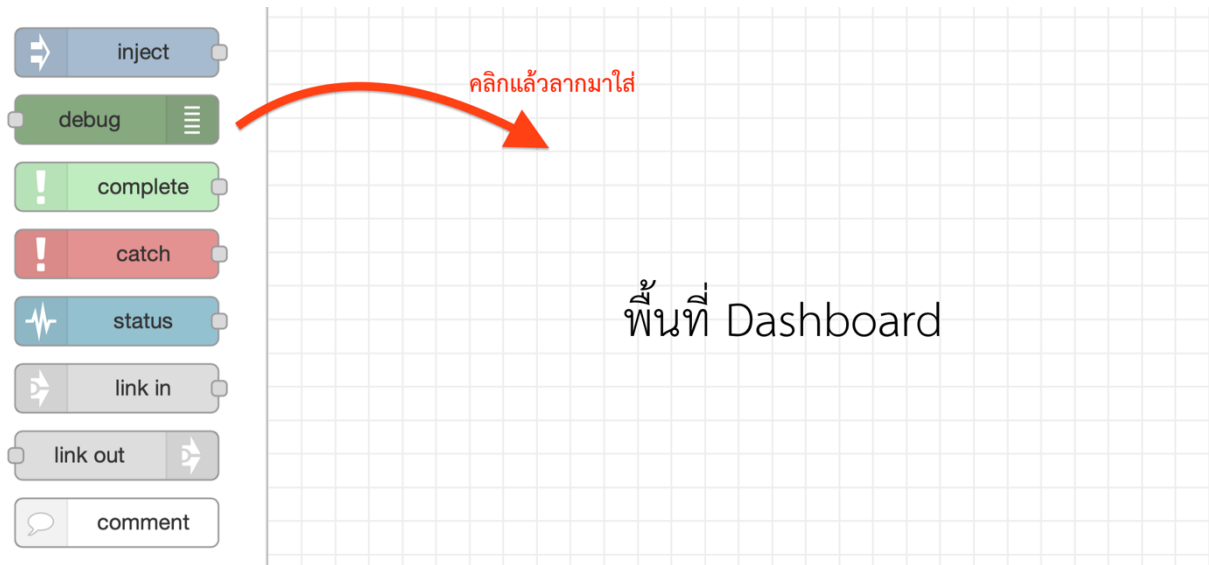
นอกจากนี้ โครงงานการพัฒนาโครงงานแบบประยุกต์ Smart Farm ยังประกอบไปด้วยอีก 4 ฝ่าย ได้แก่ ฝ่ายเกษตรกรรม ฝ่ายปศุสัตว์ ฝ่ายการประมง และฝ่ายที่พักอาศัย ซึ่งในแต่ละฝ่ายมีข้อมูลที่แตกต่างกัน และต้องการรับหรือส่งข้อมูลแยกออกจากกัน เราสามารถเปรียบเทียบ Server เป็นเหมือนไปรษณีย์ เป็นเหมือนตัวกลางในการรับหรือส่งข้อมูลจากอีกแหล่งหนึ่งไปยังอีกแหล่งหนึ่ง ซึ่งสามารถสรุปออกมาได้เป็นแผนภาพดังต่อไปนี้



ลูกศรเส้นประสีเหลืองจากแผนภาพดังกล่าว คือส่วนที่ใช้ในการส่งข้อมูลฉุกเฉิน (Emergency) ที่อยู่นอกเหนือไปจากข้อมูลสภาพแวดล้อม หรือข้อมูลสั่งการ ยกตัวอย่างเช่น หากเซนเซอร์วัดความขุ่นของน้ำในฝ่ายการประมงได้รับความเสียหาย เราสามารถส่งข้อมูลฉุกเฉินผ่านช่องทางนี้ได้ โดยจะแจ้งเตือนไปยังหน้าเว็บทันที ซึ่งผู้ใช้งานสามารถรับรู้และซ่อมแซมเซนเซอร์ได้หลังจากนั้น

หน้าที่ของ node รายการต่าง ๆ ที่พบได้บ่อย

การเลือกใช้งาน node แต่ละรายการจะสามารถคลิกเลือกได้จากเมนูฝั่งด้านซ้าย แล้วลากมาวางบนหน้า dashboard เพื่อเชื่อมต่อกัน



โดย node แต่ละรายการมีหน้าที่ และการทำงานที่แตกต่างกัน ดังนี้

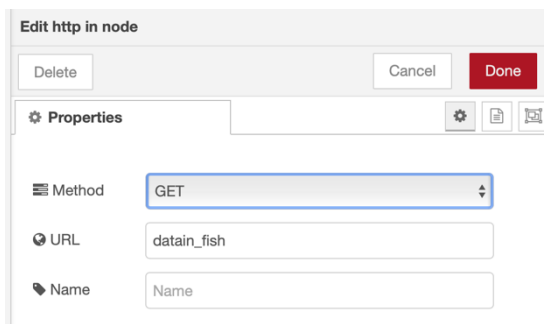
1. “http in”

ใช้สร้าง sub-page สำหรับ input หรือ output เช่น การสร้างหน้า /datain_fish สำหรับเป็นเว็บรับข้อมูลสภาพแวดล้อมของฝ่ายการประมงเข้าสู่ Server หรือ

ใช้สร้าง /dataout_fish สำหรับส่งออกข้อมูลสภาพแวดล้อมของฝ่ายการประมงออกจาก Server ได้



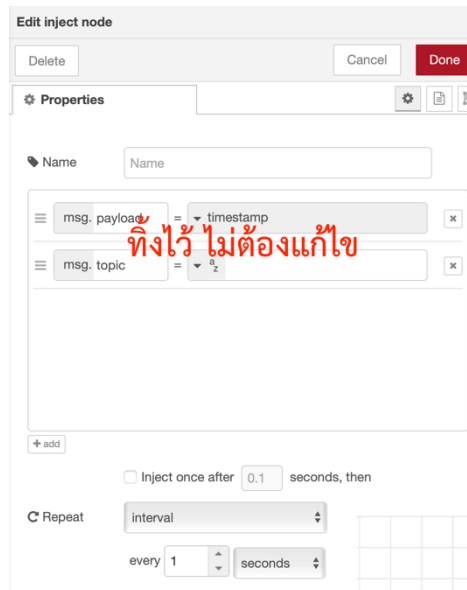
- การตั้งค่า Node



- ให้ตั้ง Method เป็น GET เสมอ
- ตั้งชื่อ URL ได้ตามต้องการ
- Name มีไว้สำหรับตั้งชื่อเพิ่มความสะดวกในการแยกแยะ

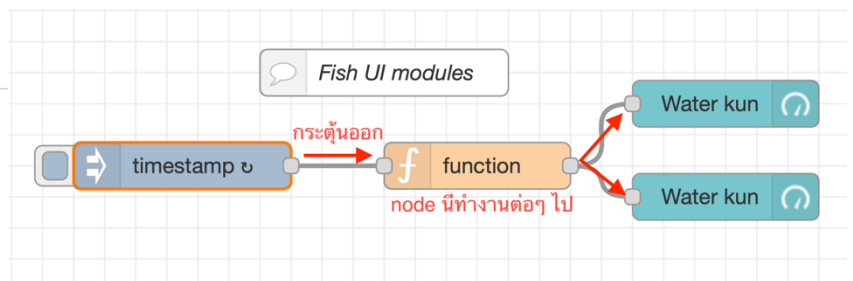
2. “inject”

ใช้ในการอัดฉีดเพื่อกระตุ้นให้เกิดการทำงานของ node บางประเภท สามารถตั้งให้อัดฉีดทุกกี่หน่วยเวลาก็ได้



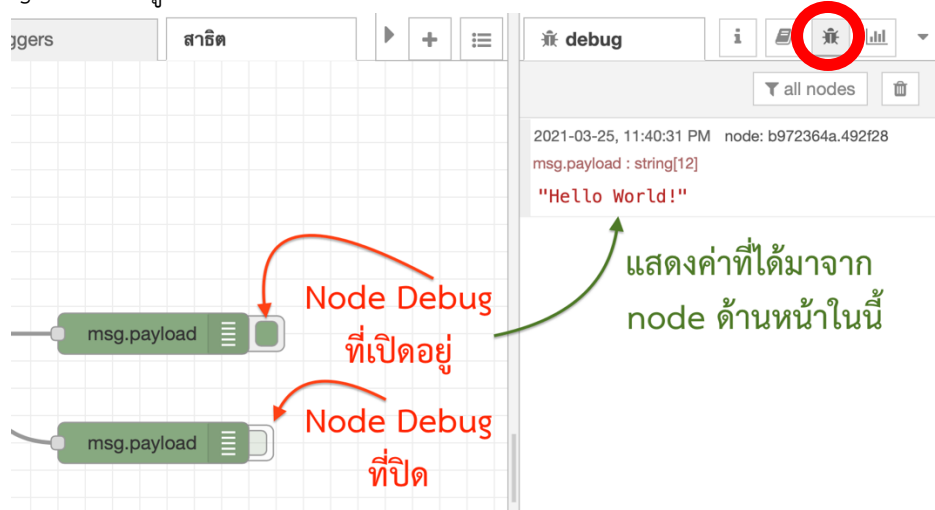
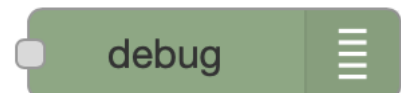
- การตั้งค่า node

- ช่อง Name มีไว้สำหรับตั้งชื่อเพิ่มความสะดวกในการแยกแยะ
- ช่องสื่ห่ล่ยม ไม่ต้องแก้ไข โดยเป็นตัวส่ง timestamp ออกมาเพื่อกระตุ้น node ที่เชื่อมต่อกับ inject ให้ทำงาน
- ช่อง Repeat ด้านล่างไว้สำหรับตั้งให้ node นี้กระตุ้นเป็นเวลาทุก ๆ กี่หน่วยเวลา ในที่นี้ให้เลือกเป็น interval



3. “debug”

ใช้ในการ debug หรือใช้ในการตรวจสอบว่า node ที่เป็นตัวส่งข้อมูลนั้นได้ส่งอะไรมาให้ โดยสามารถเข้าไปดูได้จากหน้า debug ในแถบเมนูด้านขวา และสามารถเปิดปิดการทำงานได้



4. “function”

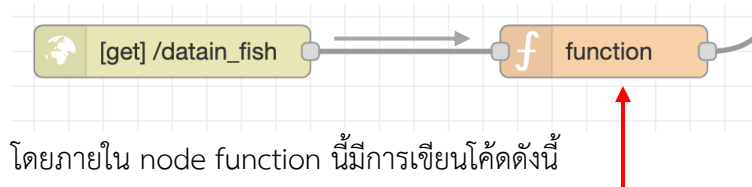
ใช้ในการจัดทำฟังก์ชันต่าง ๆ โดยตั้งค่าโดยใช้ภาษา C รวมไปถึงการประกาศ และบันทึกค่าที่ได้มาลงในตัวแปร class

global (ระดับคลาสของตัวแปรที่ทุกส่วนสามารถเข้าถึงได้) ยังสามารถใช้ในการแตก JSON ออกมาเป็นตัวแปรได้ หรือรวมตัวแปรเป็น JSON ได้เช่นกัน



- ตัวอย่างการใช้งาน node function ที่พบได้บ่อยครั้ง

- การรับค่าจาก http



โดยภายใน node function นี้มีการเขียนโค้ดดังนี้

```
1 global.set("fi_kun",msg.payload.fi_kun);
2 return msg;
3
```

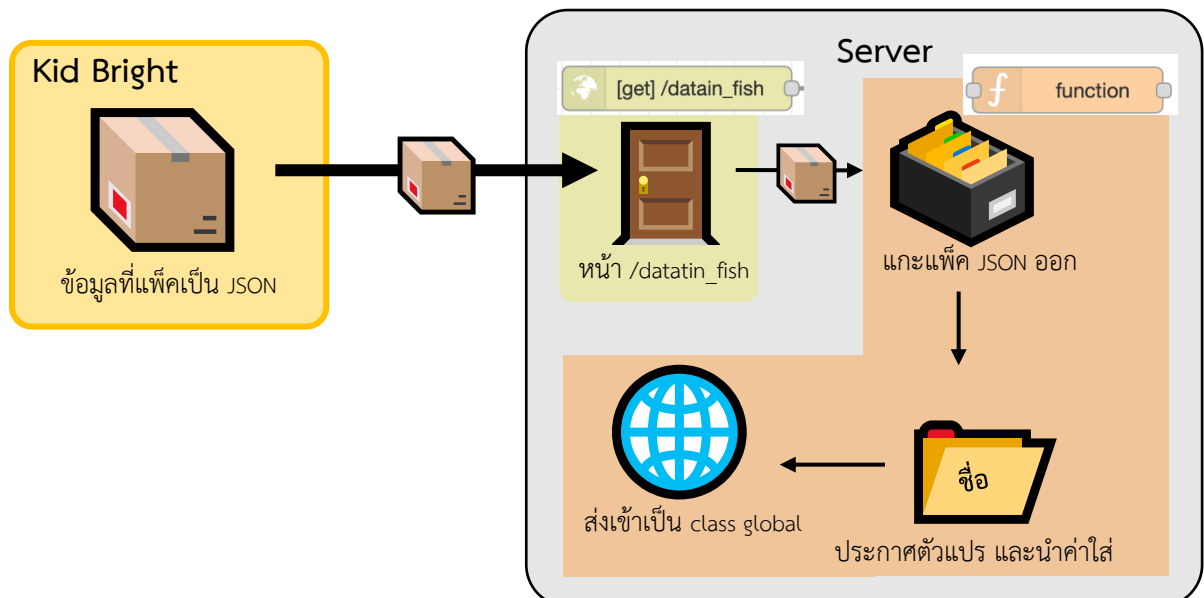
โดยข้อมูลจากหน้า /datain_fish จะรับค่าพร้อมกับชื่อตัวแปรมา ซึ่งออกมาเป็นไฟล์ JSON

รูปของไฟล์ JSON ที่ส่งออกมาจาก GET /datain_fish คือ;

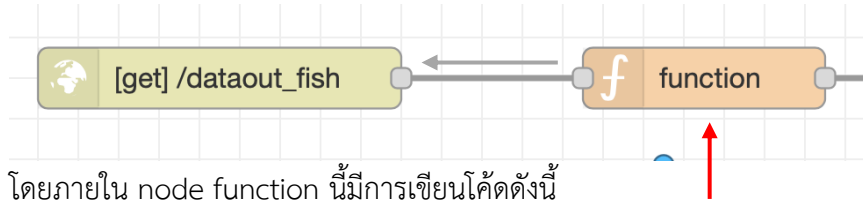
```
{ fi_kun: "123" }
```

ซึ่งตัว node function จะเป็นการดึงค่า หรือในที่นี้คือตัวเลขในรูป string “123” และได้ประกาศตัวแปรใหม่ใน class global ที่สามารถเข้าถึงได้ทั่วทั้ง server ในชื่อของ fi_kun ที่ได้ตั้งไว้ใน node function สามารถสรุปออกมาได้เป็นแผนผังดังนี้

โดยการทำลักษณะนี้ ไม่จำเป็นต้องมีการอัดฉีดกระตุ้น จาก node inject เนื่องจากการที่ข้อมูลถูกส่งเข้าประตุนั้น จะเป็นถือว่าการ inject ไปในตัว



- การส่งค่าออก http

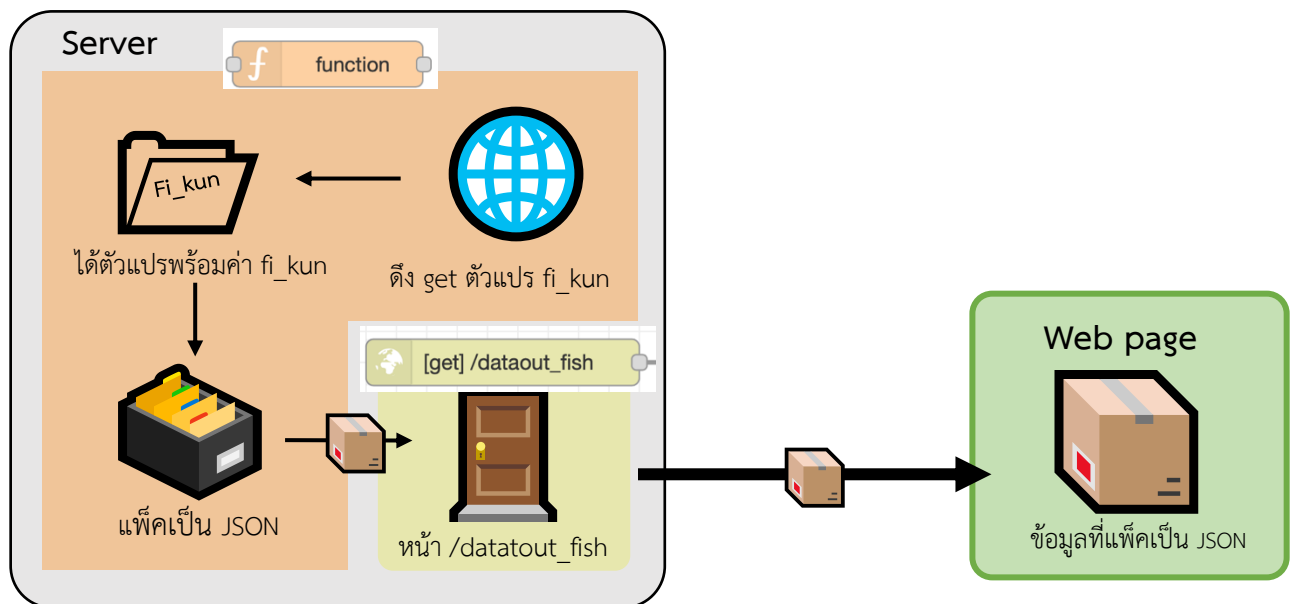


โดยภายใน node function นี้มีการเขียนโค้ดดังนี้

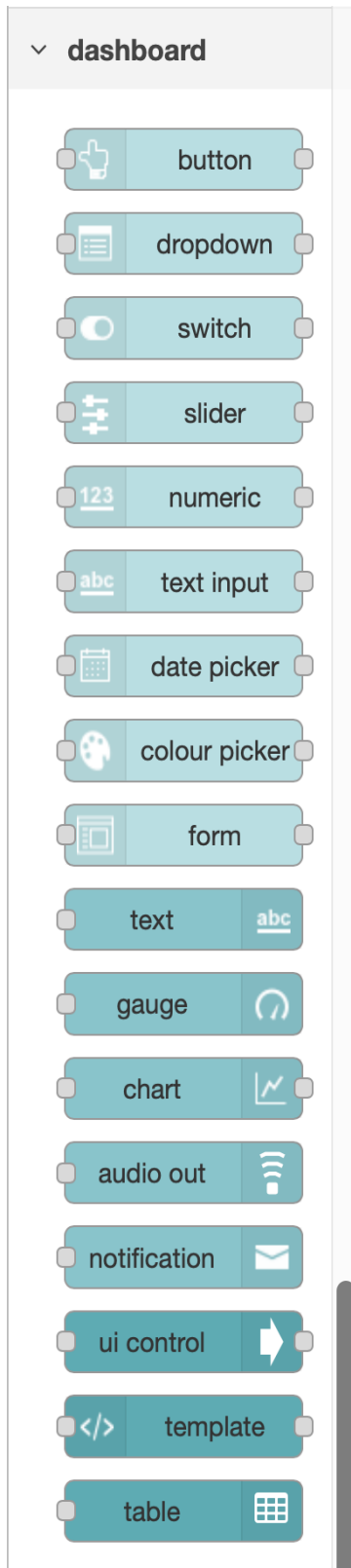
```
1 msg.payload = global.get("fi_pump");
2 return msg;
```

โดยจากโค้ดดังกล่าวนี้จะแสดงให้เห็นว่า ตัว node function นั้นทำการดึง get ตัวแปร fi_pump จาก class global แล้วนำส่งออกไปจาก flow เป็น payload กลับไปยังหน้า /dataout_fish ซึ่งสามารถสรุปออกได้ตามแผนผังดังต่อไปนี้

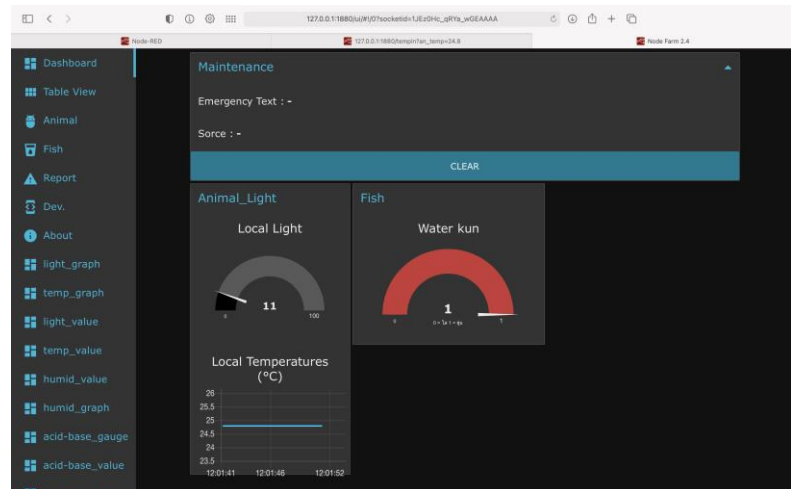
เช่นเดียวกันกับรับค่าจาก http ไม่จำเป็นต้องอัดฉีดกระตุ่นการทำงาน จาก node inject



5. “Dashboards”

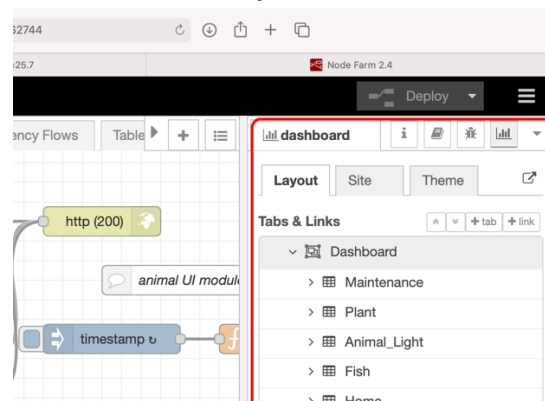


เป็น node ที่ใช้ในการสร้างหน้า dashboard ซึ่งหน้า dashboard จะเป็นหน้าที่แสดงผลข้อมูลต่าง ๆ จากเซิร์ฟเวอร์โดยตรง ซึ่งสามารถเข้าถึงได้จาก <IP address>:1880/ui และ node dashboard ทุกตัวจำเป็นต้องถูกอัดฉีดเพื่อกระตุ้นให้เกิดการทำงานจาก node inject



ซึ่ง node ของ dashboard มีหลายหลายรูปแบบ โดยอาจเป็น input output หรือทั้งคู่ก็ได้ แต่ส่วนที่จะได้ใช้จริง มันจะเป็น output แสดงผลมากกว่า โดยจะยกตัวอย่าง node ที่ได้ใช้บ่อย

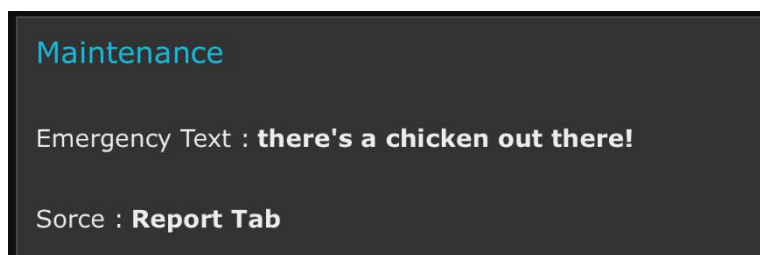
การตั้งค่าหน้า UI จะอยู่ในแถบด้านขวาของหน้าจอ



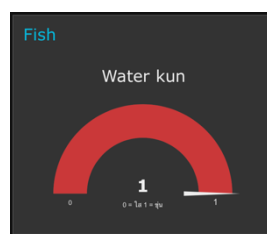
โดยเครื่องมือ UI นั้นจะไม่ได้ถูกติดตั้งมาตั้งแต่แรก จำเป็นต้องทำการดาวน์โหลด node เพิ่มเติม คล้ายกับการเพิ่ม library ให้กับโปรแกรม Arduino โดยสามารถศึกษาเพิ่มเติมได้จาก

<https://flows.nodered.org/node/node-red-dashboard>

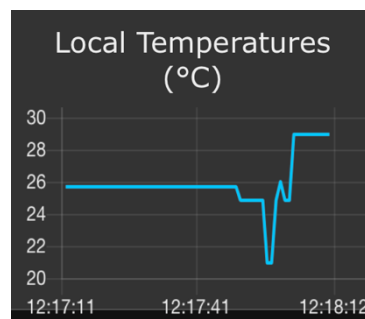
- “text” เป็น output ที่ใช้ในการแสดงผลออกเป็นตัวอักษร หรือตัวเลข



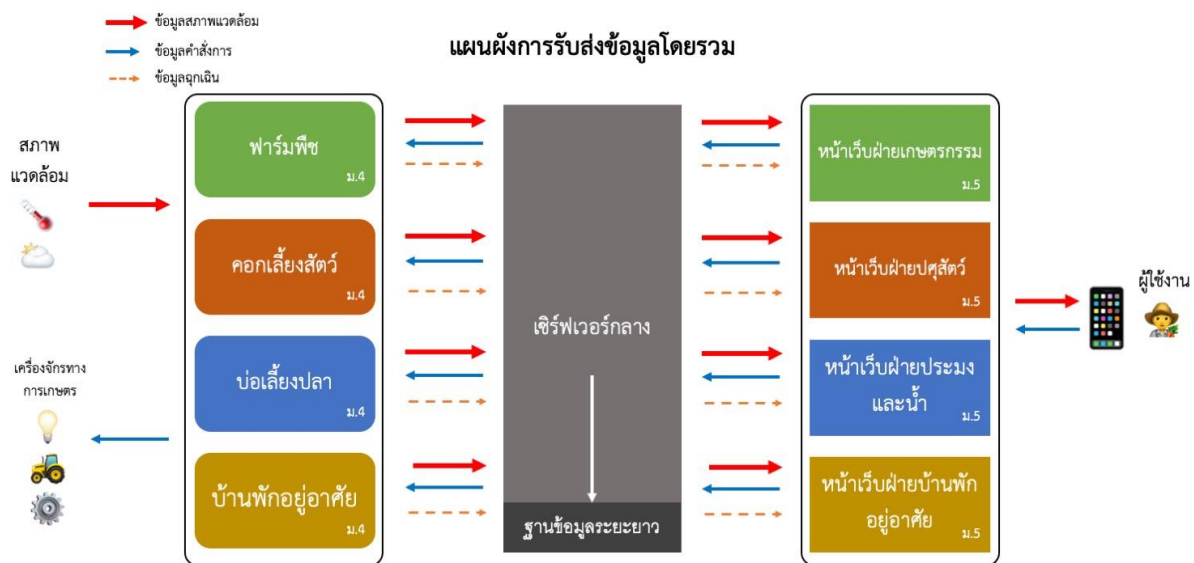
- “gauge” เป็น output ที่ใช้ในการแสดงผลออกเป็นหน้าปัดแสดงขนาดค่า



- “chart” เป็น output ที่ใช้ในการแสดงผลออกเป็นหน้าปัดแสดงขนาดค่า
 - จำเป็นต้องมีการกำหนด topic สำหรับ node นี้
 - สามารถแสดงหลายชุดข้อมูล เป็นกราฟหลายเส้นได้

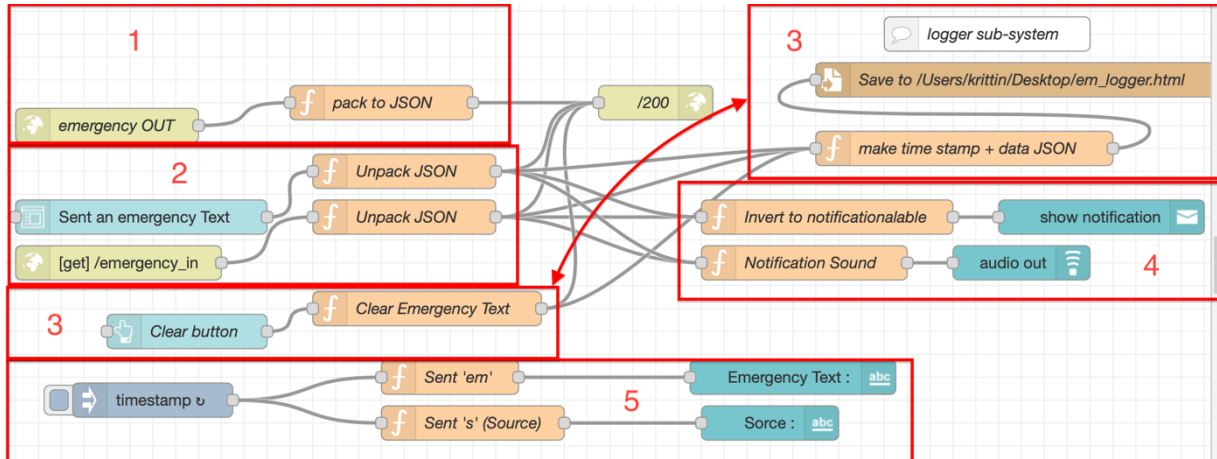


คำอธิบายหลักการทำงานของ Flow ต่าง ๆ



● Flow ระบบการรับส่งข้อมูลฉุกเฉิน

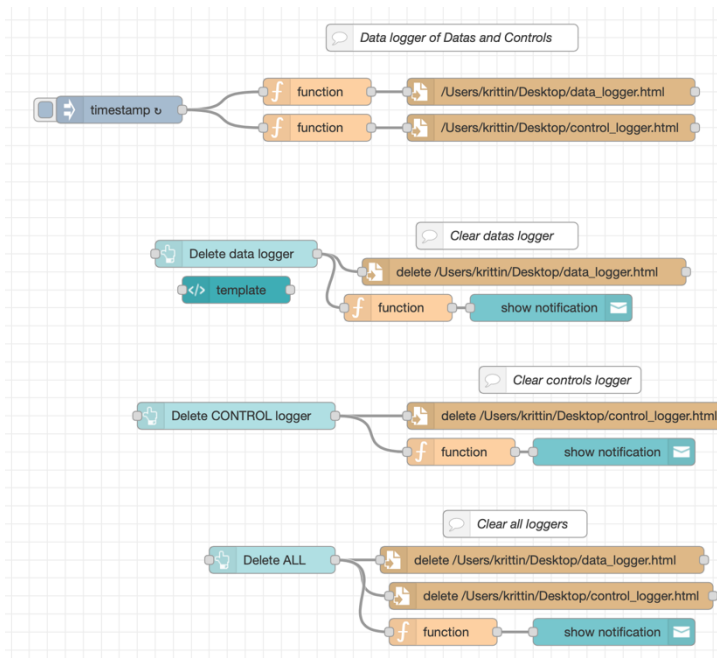
- เป็นระบบการส่งตามลูกศรเส้นประสีเหลืองจากในแผนผัง ซึ่งจะประกอบไปด้วย 5 ส่วนย่อย ดังนี้



1. ส่วนในการส่งออกข้อมูลฉุกเฉิน
2. ส่วนในการรับข้อมูลฉุกเฉิน ทั้งผ่านหน้า http และ หน้า dashboard
3. ส่วนในการ logger ข้อมูลลงฐานข้อมูลระยะยาว และปุ่มในการลบข้อความ
4. ส่วนในการแสดงผลข้อมูลฉุกเฉินในหน้า dashboard

● Flow ระบบ Data Logger หรือ Database

- เป็นระบบที่ใช้ในการบันทึกข้อมูลทุกตัวแปรลงในฐานข้อมูลระยะยาว



ส่วนที่ 1 เก็บ data กับ control

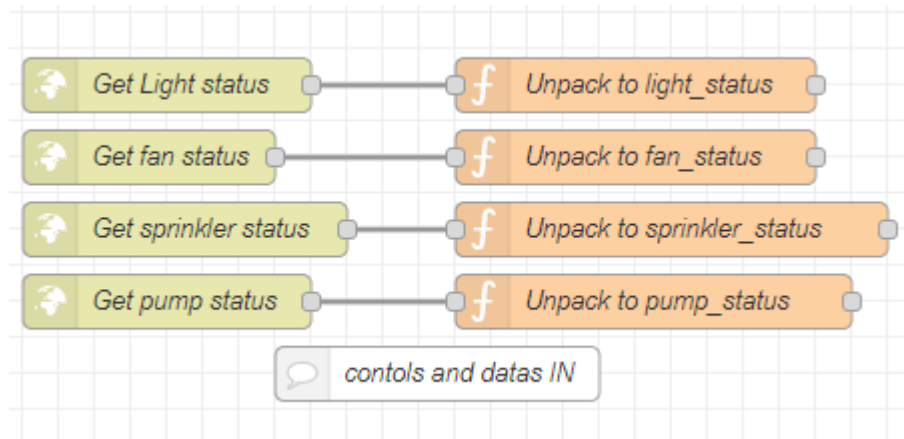
ส่วนที่ 2 มีไว้ลบฐานข้อมูล data

ส่วนที่ 3 มีไว้ลบฐานข้อมูล control

ส่วนที่ 4 มีไว้ลบฐานข้อมูล data และ control \

- Flow ระบบการเกษตรกรรม

การรับค่าคำสั่งการจากหน้าเว็บของผู้ใช้งาน



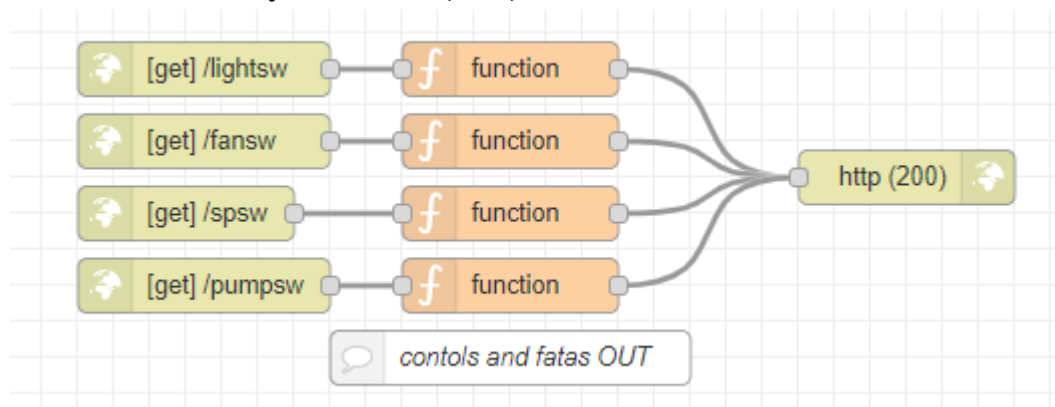
Get Light = การรับคำสั่งการของการเปิด-ปิดแสง ในรูปของตัวแปร light_status

Get Fan = การรับคำสั่งการของการเปิด-ปิดพัดลม ในรูปของตัวแปร fan_status

Get Sprinkler = การรับคำสั่งการของการเปิด-ปิดสปริงเกอร์ ในรูปของตัวแปร sprinkler_status

Get Pump = การรับคำสั่งการของการเปิด-ปิดปั้มน้ำ ในรูปแบบของตัวแปร pump_status

การส่งออกค่าคำสั่งการไปสู่บอร์ดเพื่อควบคุมให้อุปกรณ์ทำงาน

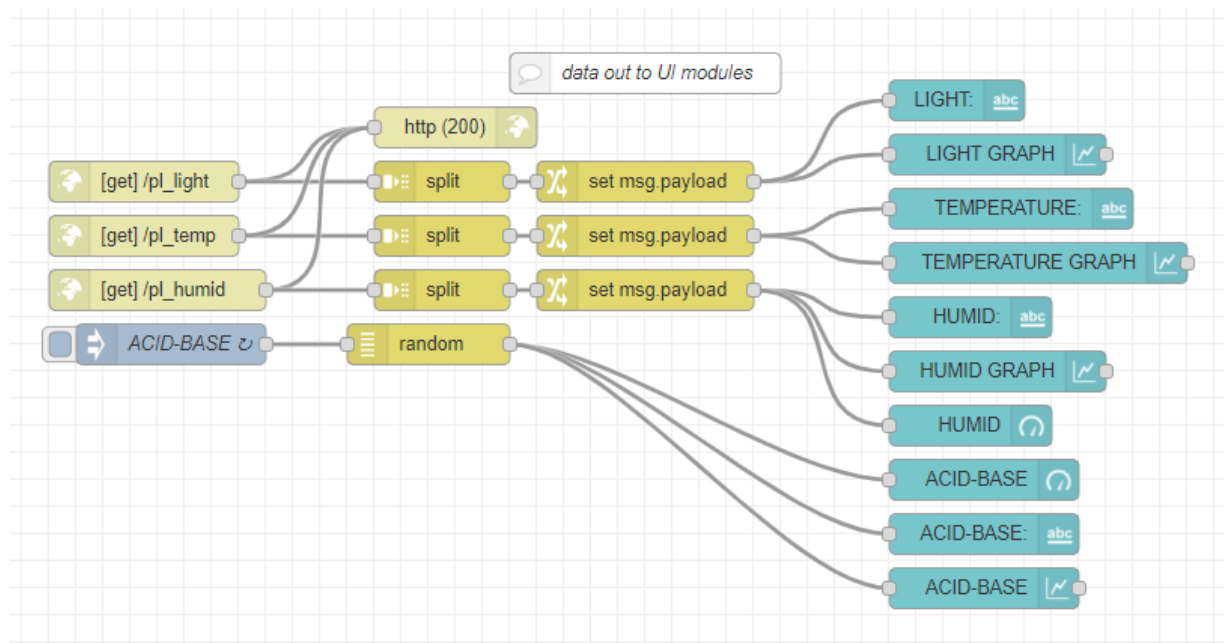


lightsw = การส่งออกคำสั่งการของการเปิด-ปิดแสง

fanse = การส่งออกคำสั่งการของการเปิด-ปิดพัดลม

spsw = การส่งออกคำสั่งการของการเปิด-ปิดสปริงเกอร์

pumpsw = การส่งออกคำสั่งการของการเปิด-ปิดปั้มน้ำ

การดึงค่าข้อมูลจากสภาพแวดล้อมไปแสดงเป็น UI

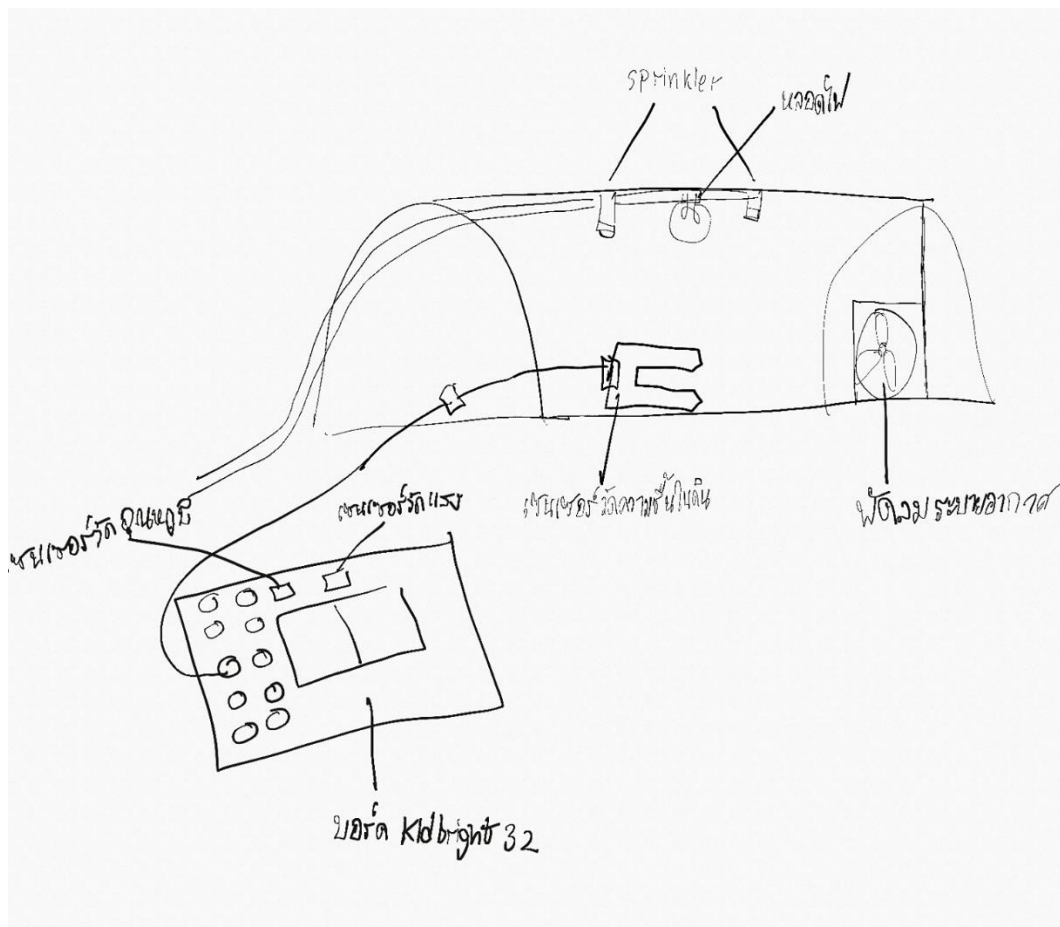
split = การแกะค่าจากไฟล์ JSON แล้ว set ค่าที่ได้ให้เป็น msg.payload เพื่อนำไปแสดงค่าเป็น UI

pl_light = การรับค่าข้อมูลของแสงไฟภายในเรือนเพาะชำ จากนั้นส่งออกไปแสดงเป็น UI ได้แก่ ตัวหนังสือ และกราฟ

pl_temp = การรับค่าข้อมูลของอุณหภูมิภายในเรือนเพาะชำ จากนั้นส่งออกไปแสดงเป็น UI ได้แก่ ตัวหนังสือ และกราฟ

pl_humid = การรับค่าข้อมูลของความชื้นของดินภายในเรือนเพาะชำ จากนั้นส่งออกไปแสดงเป็น UI ได้แก่ ตัวหนังสือ กราฟ และหน้าปัด

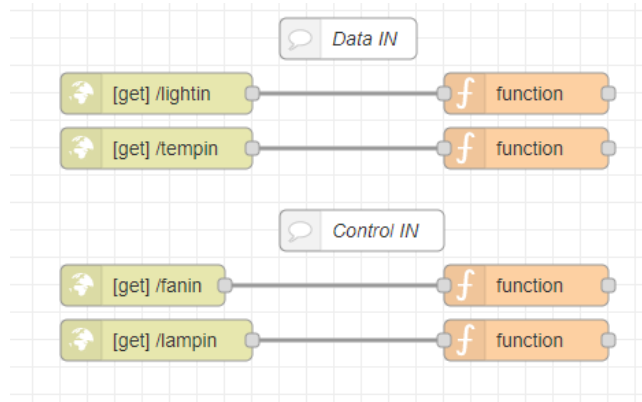
ACID-BASE = จะเห็นว่าเป็น inject ที่เชื่อมกับ random ซึ่งหมายถึงการสมมติค่าของค่าความเป็นกรด-เบสในดิน จากนั้นส่งออกไปแสดงเป็น UI ได้แก่ ตัวหนังสือ กราฟ และหน้าปัด



แผนภาพการทำงานของระบบฝ่ายเกษตรกรรม

- Flow ของระบบปศุสัตว์

การรับค่าของข้อมูลสภาพแวดล้อมและคำสั่งการ



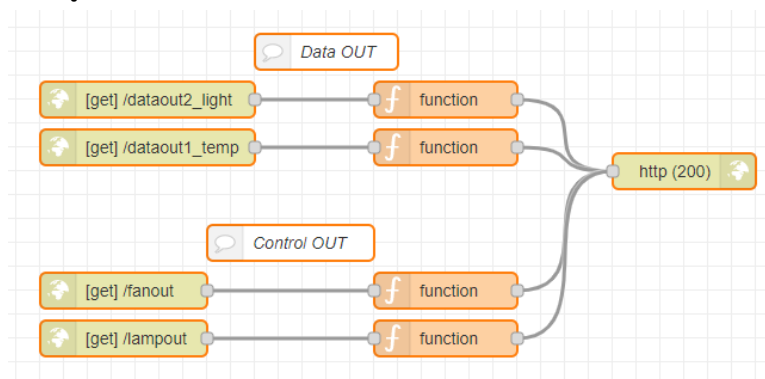
lightin = การรับค่าของข้อมูลแสงภายในโรงปศุสัตว์

tempin = การรับค่าของข้อมูลอุณหภูมิภายในโรงปศุสัตว์

fanin = การรับคำสั่งการเปิด-ปิดพัดลมภายในโรงปศุสัตว์

lampin = การรับคำสั่งการเปิด-ปิดหลอดไฟภายในโรงปศุสัตว์

การส่งออกค่าของข้อมูลสภาพแวดล้อมและคำสั่งการ

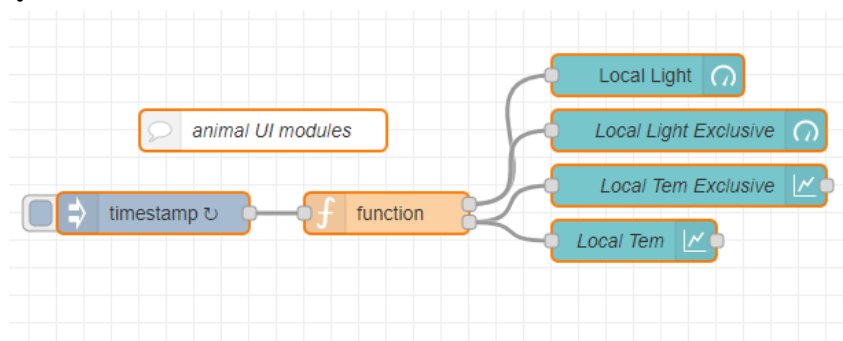


dataout2_light = การส่งออกค่าของข้อมูลแสงภายในโรงปศุสัตว์

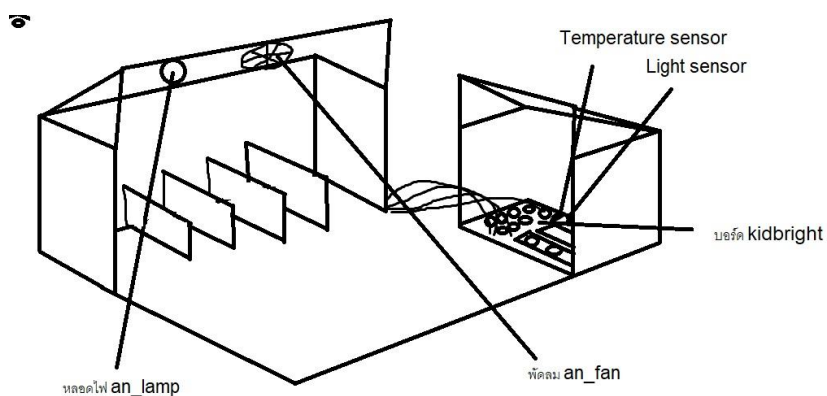
dataout1_temp = การส่งออกค่าของข้อมูลอุณหภูมิภายในโรงปศุสัตว์

fanout = การส่งออกคำสั่งการเปิด-ปิดพัดลมภายในโรงปศุสัตว์

lampout = การส่งออกคำสั่งการเปิด-ปิดหลอดไฟภายในโรงปศุสัตว์

การดึงค่าข้อมูลจากสภาพแวดล้อมไปแสดงเป็น UI

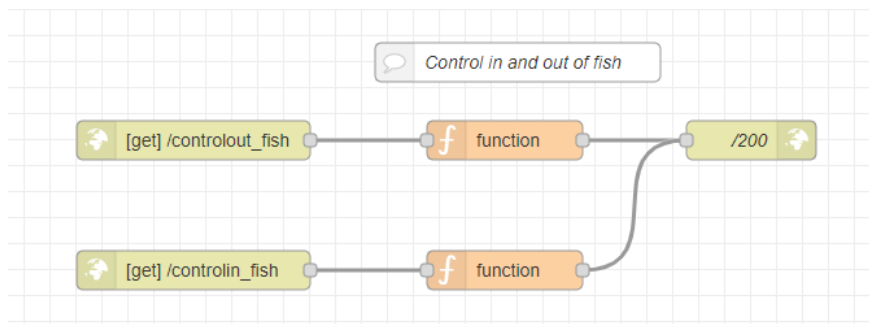
การส่งออกค่าข้อมูลของสภาพแวดล้อม คือ ค่าแสงและค่าอุณหภูมิไปแสดงเป็น UI อันได้แก่ หน้าปัด (แสง) และกราฟ(อุณหภูมิ)



แปลนภาพการทำงานของระบบฝ่ายปศุสัตว์

- Flow ของระบบการประมง

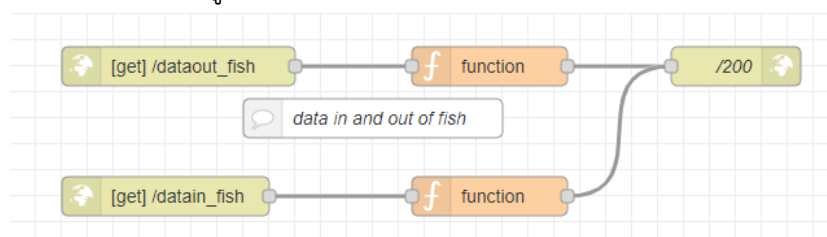
การรับค่าและส่งออกค่าการสั่งการ



controlin_fish = การรับค่าการสั่งการการเปิด-ปิดปั้มน้ำ

controlout_fish = การส่งออกค่าการสั่งการการเปิด-ปิดปั้มน้ำ

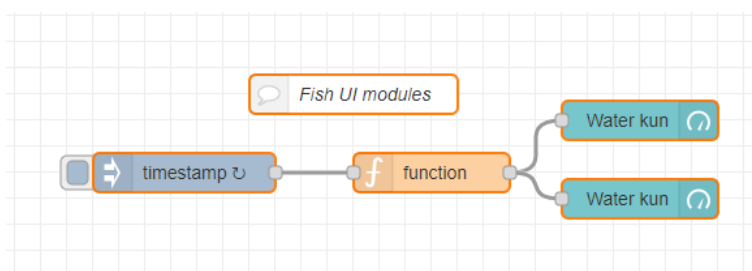
การรับค่าและส่งออกค่าของข้อมูลจากสภาพแวดล้อม



datain_fish = การรับค่าของข้อมูลความชุ่มชื้นของน้ำ

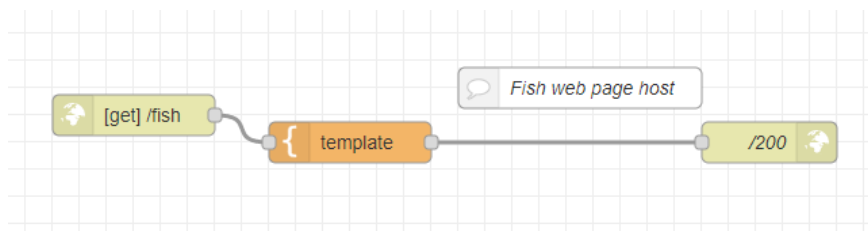
dataout_fish = การส่งออกค่าของข้อมูลความชุ่มชื้นของน้ำ

การดึงค่าข้อมูลจากสภาพแวดล้อมไปแสดงเป็น UI

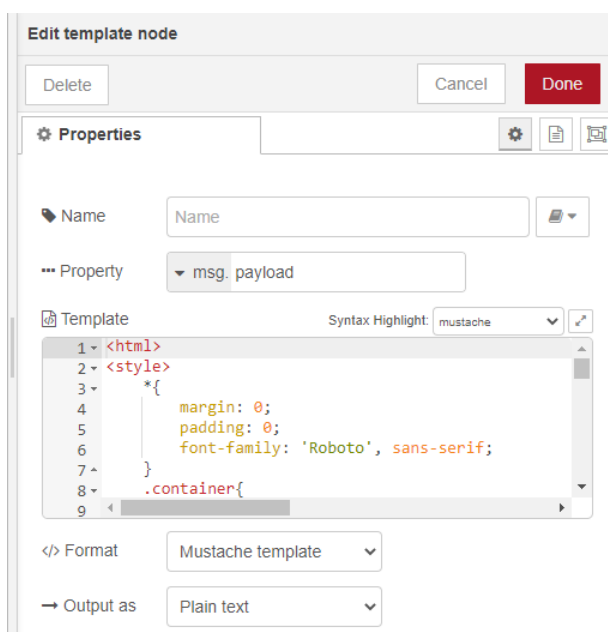


การส่งออกค่าข้อมูลของสภาพแวดล้อม คือ ค่าความชุ่มชื้นของน้ำไปแสดงเป็น UI อันได้แก่ หน้าปัด

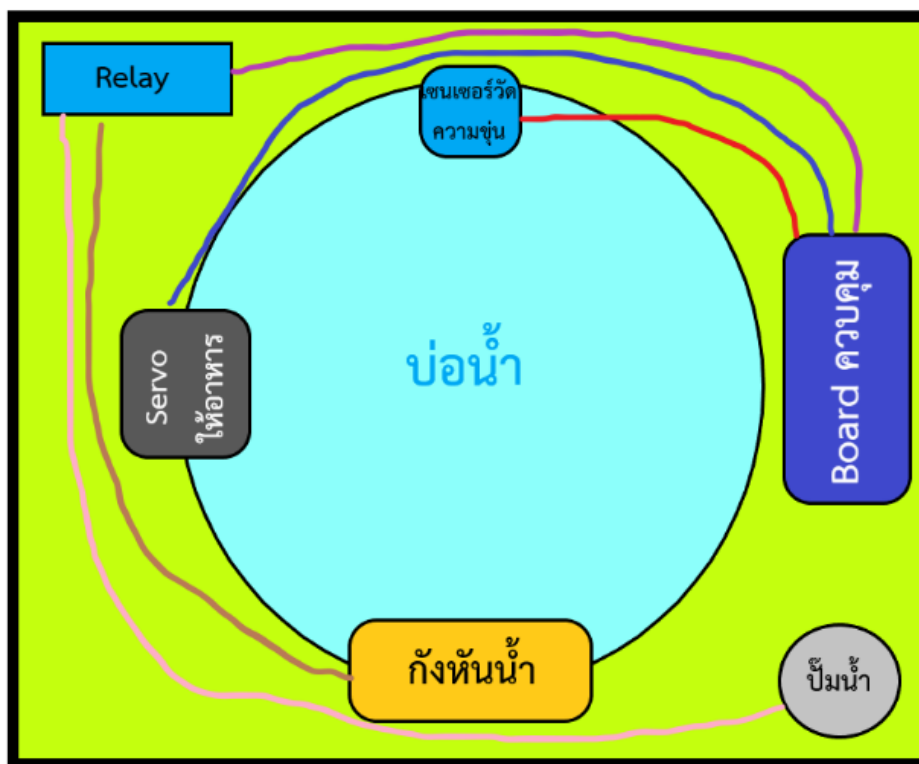
การสร้างหน้าเว็บจาก Node-RED



ฝ่ายประมงได้ทำการสร้างหน้าเว็บเพจจากตัว Node-RED เองซึ่งได้ใช้ภาษา html และ css



ภาษา html และ css ซึ่งเขียนไว้ใน template



แปลนภาพการทำงานของระบบฝ่ายการประมง