**บทที่ 1**

**บทนำ**

* 1. **ที่มาและความสำคัญ**

ปัจจุบันคนไทยเริ่มหันมาสนใจการปลูกพืชผักสวนครัวกันมากขึ้น แต่ปัญหาที่พบคือคนส่วนมากไม่มีเวลาดูแลพืชผักสวนครัวของตนเอง จึงทำให้พืชผักเสียหาย พวกเราจึงคิดว่าควรทำอย่างไรจะทำให้การดูแลพืชผักสวนครัวเป็นเรื่องที่ง่ายและสะดวกโดยการพัฒนาเว็บไซต์สำหรับควบคุมการรดน้ำผ่านโทรศัพท์มือถือ ที่แสดงค่าความชื้นปัจจุบันในดิน และสามารถตั้งการทำงานแบบอัตโนมัติโดยวัดค่าความชื้นจากดินได้อีกด้วย

**1.2 วัตถุประสงค์**

1.2.1 เพื่ออำนวยความสะดวกให้แก่ผู้ที่ปลูกพืชหรือทำเกษตรกรแต่ไม่มีเวลาในการรดน้ำ

1.2.2เพื่อเสริมสร้างแนวคิดริเริ่มในการพัฒนาเว็บไซต์ที่สามารถนำมาใช้ประโยชน์ในชีวิตประจำวัน

**1.3 สมมติฐาน**

1.3.1 เพิ่มทักษะและกระบวนการคิดและปฏิบัติให้กลุ่มนักศึกษาในการทำชิ้นงาน

1.3.2 สามารถใช้เป็นต้นแบบในการทำธุรกิจและสามารถมาประยุกต์ใช้เสริมสร้างจนประกอบอาชีพได้

**1.4 ขอบเขตของการศึกษา**

ขอบเขตการทำงานในการทำเครื่องรดน้ำอัตโนมัติ ผ่านเว็บไซต์ คือ การสร้างระบบแจ้งสถานะความชื้นของเครื่องรดน้ำอัตโนมัติ ผ่านเว็บไซต์ ขั้นตอนการดำเนินงาน ดังนี้

1.4**.**1. คัดเลือกหัวข้อโครงงานที่สนใจจะทำ1.4**.**2. ศึกษาค้นคว้าจากเอกสารและแหล่งข้อมูล

1.4**.**3. จัดทำเค้าโครงของโครงงานที่จะทำ

1.4**.**4. การลงมือทำโครงงาน

1.4**.**5. การเขียนรายงานและจัดทำคู่มือการใช้

### 1.5 ผลที่คาดว่าจะได้รับ

### เพื่ออำนวยความสะดวกสบายและประหยัดเวลาในการรดน้ำ และสามารถพัฒนาต่อเพื่อนำไปใช้กับงานทางเกษตรกรรมที่มีขนาดใหญ่ขึ้น

**บทที่ 2**

**เอกสารและงานวิจัยที่เกี่ยวข้อง**

การศึกษาวิจัย เรื่อง แปลงผักรดน้ำอัจฉริยะ ทำงานร่วมกับเว็บไซต์ที่สามารถแสดงค่าความชื้นปัจจุบันในดิน และสามารถกำหนดค่าความชื้นที่ต้องการให้รดน้ำได้  คณะผู้จัดทำได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องแนะนำเสนอตามลำดับต่อไปนี้

**2.1 ความรู้เกี่ยวกับ ESP8266**

**2.2 Soil Moisture Sensor Module**

**2.3 Relay 3V**

**2.4 ความรู้เกี่ยวกับ NETPIE**

**2.5 ความรู้เกี่ยวกับ Surge.sh**

**2.6 ความรู้เกี่ยวกับ Arduino IDE**

**2.1 ความรู้เกี่ยวกับ ESP8266**

ESP8266 เป็นชื่อเรียกของชิฟของโมดูล ESP8266 สำหรับติดต่อสื่อสารบนมาตรฐาน WiFi ทำงานที่แรงดันไฟฟ้า 3.0-3.6V ทำงานโดยใช้กระแสโดยเฉลี่ย 80mA รองรับคำสั่ง deep sleep ในการประหยัดพลังงาน ใช้กระแสน้อยกว่า 10 ไมโครแอมป์ สามารถ wake up กลับมาส่งข้อมูลใช้เวลาน้อยกกว่า 2 มิลลิวินาที ภายในมี Low power MCU 32bit ทำให้เราเขียนโปรแกรมสั่งงานได้ มีวงจร analog

digital converter ทำให้สามารถอ่านค่าจาก analog ได้ความละเอียด 10bit ทำงานได้ที่อุณหภูมิ -40 ถึง 125 องศาเซลเซียส

เมื่อนำชิฟ ESP8266 มาผลิตเป็นโมดูลหลายรุ่น ก็จะขึ้นต้นด้วย ESP866 แล้วตามด้วยรุ่น เช่น ESP-01 , ESP-03 , ESP-07 , ESP-12E

ESP8266 ติดต่อกับ WI-FI แบบ Serial สามารถเขียนโปรแกรมลงไปในชิฟ โดยใช้ Arduino IDE จึงทำให้การเขียนโปรแกรมและใช้งานเป็นเรื่องง่าย คล้ายกับการใช้ Arduino สามารถติดต่อกับเซนเซอร์ได้ จึงทำให้ผู้ที่มีพื้นฐานการใช้งาน Arduino สามารถทำงานได้สะดวกสะบายมากยิ่งขึ้น

GPIO0 เป็นขาสำหรับเลือกโหมด โดยเมื่อต่อกับ GND จะเข้าโหมดโปรแกรม เมื่อต้องการให้ทำงานปกติก็ไม่ต้องต่อ

-GPIO15 เป็นขาที่ต้องต่อลง GND เพื่อให้โมดูลทำงาน

-CH\_PD หรือ EN เป็นขาที่ต้องต่อไฟ VCC เพื่อ pull up สัญญาณ ให้โมดูลทำงาน โมดูลบางร่นไม่มีขา Reset มาให้ เมื่อต้องการรีเซต ให้ต่อขา CH\_PD กับ GND

-Reset ต่อกับไฟ VCC เพื่อ pull up สัญญาณ โดยเมื่อต้องการรีเซต ให้ต่อกับไฟ GND

-VCC เป็นขาสำหรับจ่ายไฟเลี้ยง ใช้ไฟเลี้ยง 3.0-3.6V

-GND ต่อกับไฟ 0V

-GPIO เป็นขาดิจิตอล INPUT/OUTPUT ทำงานที่ไฟ 3.3V

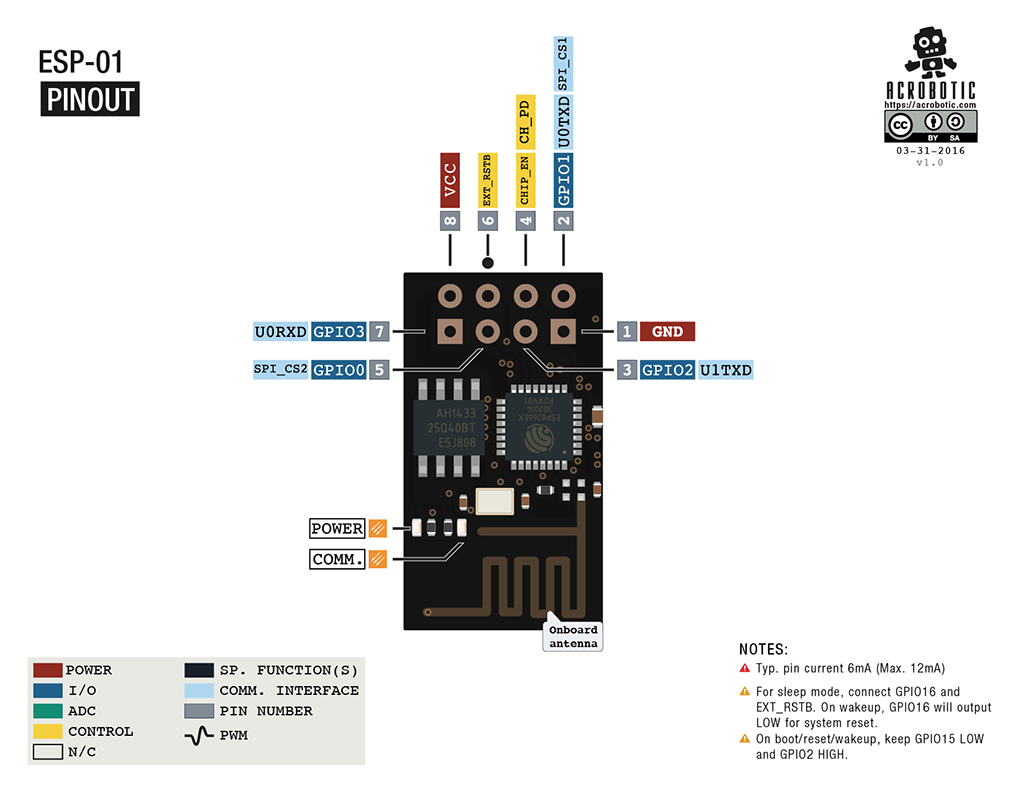
-ADC เป็นขา Analog INPUT รับแรงดันสูงสุด 1V ความละเอียด 10bit หรือ 1024 ค่า

โมดูล ESP8266 รุ่นที่นิยมเช่น ESP-01 , ESP-03 , ESP-07 , ESP-12E นอกจากนี้ยังมีบอร์ด ESP8266 ที่รวมวงจร USB TTL เข้าไปทำให้โปรแกรมกับ Arduino ได้ง่ายขึ้นเช่น NodeMCU

ESP8266 ESP-01



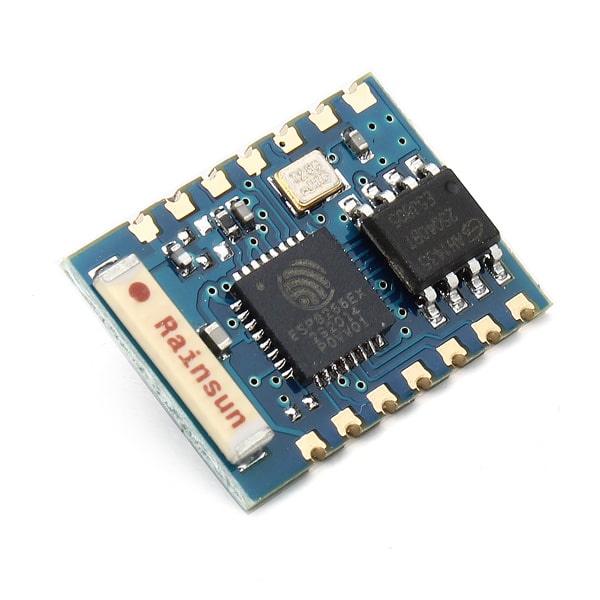
รูปที่ 2.1 แสดงรูป ESP8266 ESP-01



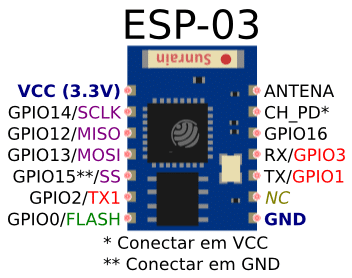
รูปที่ 2.2 แสดงขาPinout ของ ESP8266 ESP-01

ESP8266 ESP-01 เป็นรุ่นที่มีขาต่อออกมาข้างนอก จึงสามารถต่อกับบอร์ดทดลองได้ง่าย มีขา GPIO 4 ขา คือ GPIO0 , GPIO1, GPIO2 , GPIO3

ESP8266 ESP-03



รูปที่ 2.3 แสดงรูป ESP8266 ESP-03



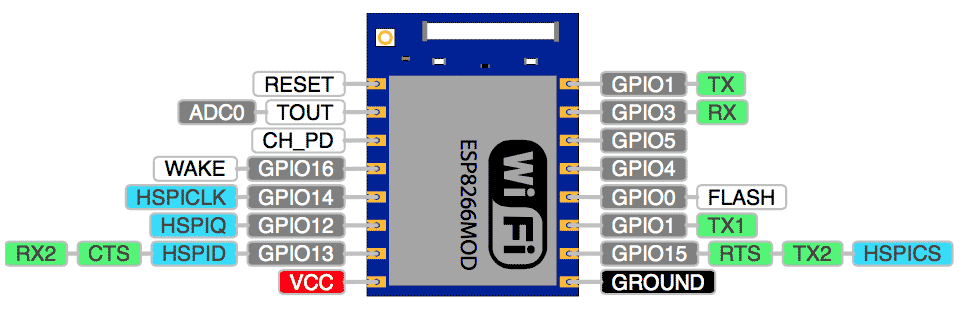
รูปที่ 2.4 แสดงขาPinout ของ ESP8266 ESP-03

ESP8266 ESP-03 จะคล้ายกับ ESP-01 มี package การต่อขาเป็นแบบเซอร์เฟสเมาส์ โดยจะมีขา GPIO มากขึ้น ทำให้เราสั่งงานอุปกรณ์ได้มากกว่า ESP8266 ESP-01 โดย ESP8266-13 มีขาทั้งหมด 14 ขา มีเสาอากาศแบบมาให้ในตัว และยังสามารถต่อสายอากาศเพิ่มเพื่อเพิ่มกำลังการรับส่ง ได้ที่ขา 14

ESP8266 ESP-07



รูปที่ 2.5 แสดงรูป ESP8266 ESP-07



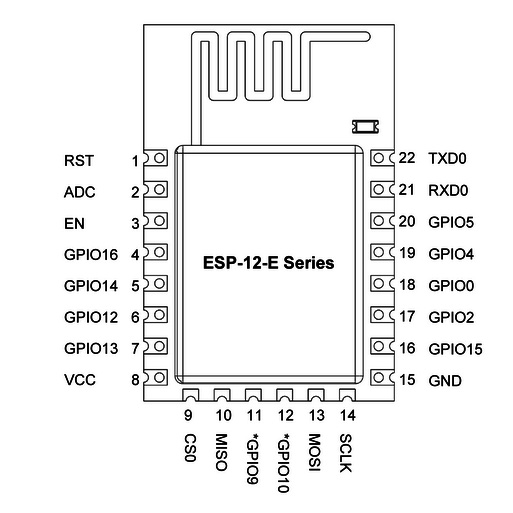
รูปที่ 2.6 แสดงขาPinout ของ ESP8266 ESP-07

ESP8266 ESP-07 เพิ่มแผ่นเหล็กครอบชิฟ ESP8266 ไว้เพื่อป้องกันสัญญาณรบกวน มีขาทั้งหมด 16 ขา โดยมีขา GPIO ให้เราใช้งาน 11 ขาและขา Analog Read อีก 1 ขา

ESP8266 ESP-12E



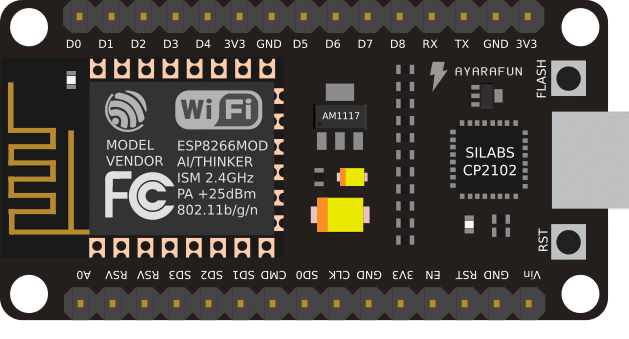
รูปที่ 2.7 แสดงรูป ESP8266 ESP-12E



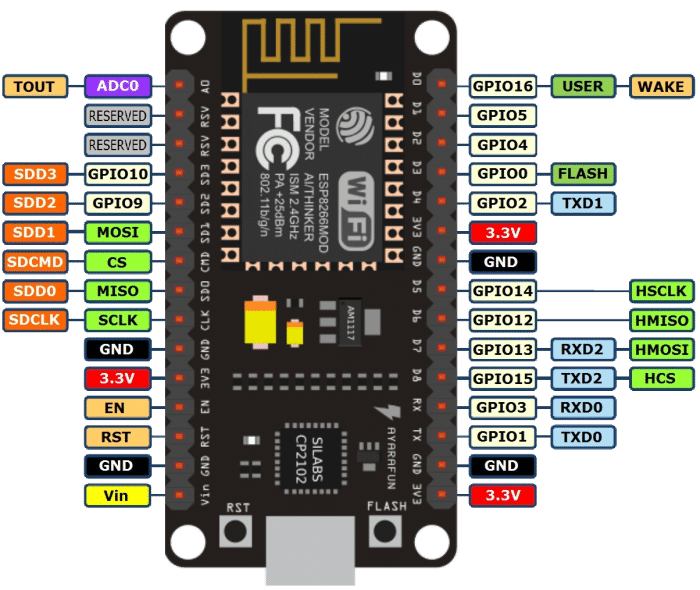
รูปที่ 2.8 แสดงขาPinout ESP8266 ESP-12E

ESP8266 ESP-12E  มีการจัดขาและต่อใช้งานแบบเดียวกับรุ่น ESP-07 โดยเปลี่ยนเสาอากาศมาเป็นแบบเดินวงจรภายใน PCB และเพิ่มขาอีก 6 ขา คือ SCLK MOSI MISO สำหรับติดต่อกับเซนเซอร์อื่น ๆ โปรโตคอล SPI ซึ่งรุ่นนี้ก็ได้รับความนิยมเป็นอย่างมาก จนเกิดการต่อยอดเป็นบอร์ด ESP8266-12 รุ่นต่าง ๆ

บอร์ด ESP8266 NodeMCU V2



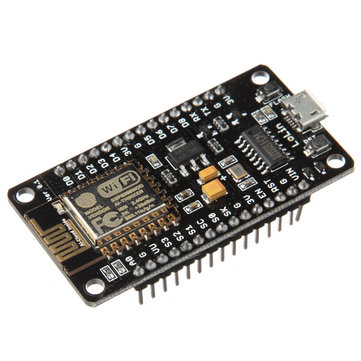
รูปที่ 2.9 แสดงรูปบอร์ด ESP8266 NodeMCU V2



รูปที่ 2.10 แสดงขาPinout ของบอร์ด ESP8266 NodeMCU V2

NodeMCU V2 เป็น ESP8266-12E รวมกับ USB TTL ที่ใช้ชิฟ CP2102 และขยายขาให้สามารถต่อทดลองได้ง่ายขึ้น มีปุ่ม reset และ flash สำหรับใช้โปรแกรม โดยใช้ Arduino IDE หรือโปรแกรมอื่น ๆ ได้อย่างสะดวก

บอร์ด ESP8266 NodeMCU V3



รูปที่ 2.11 แสดงรูปบอร์ด ESP8266 NodeMCU V3

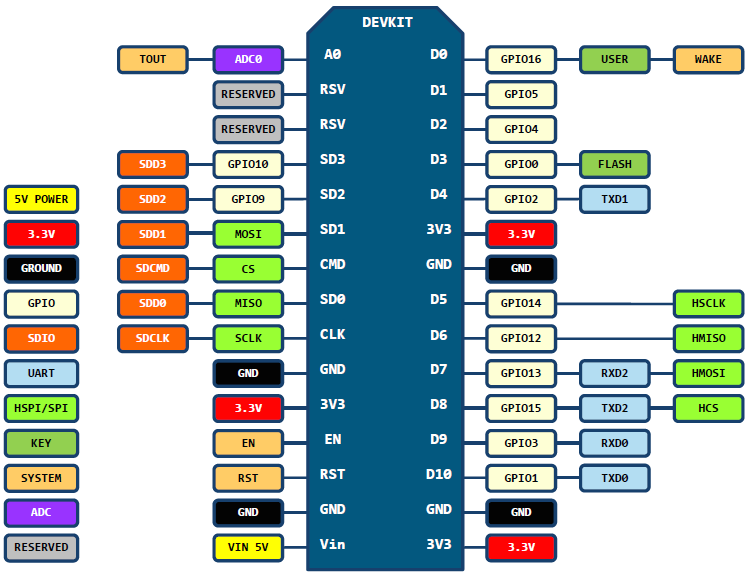
NodeMCU V3 เป็นบอร์ดทีคล้ายกับ NodeMCU V2 ที่ต่างกันคือ NodeMCU V3 จะมีขนาดกว้างกว่า และใช้ชิฟ USB TTL เป็น CH340 ซึ่งการต่อขาใช้งานและโคดโปรแกรมเหมือนกันทุกประการ

ขา GPIO ของ ESP8266 สามารถสั่งงาน ควบคุมเอาต์พุต และอ่านค่าอินพุตแบบดิจิตอลได้ โดยขาที่เป็น Digital จะขึ้นต้นด้วยตัว D หรือเรียกเป็นตัวเลขได้

NodeMCU GPIO มีการ map ขาดังนี้

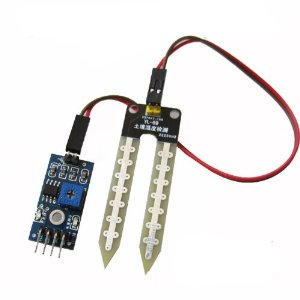
1. static const uint8\_t D0 = 16;
2. static const uint8\_t D1 = 5;
3. static const uint8\_t D2 = 4;
4. static const uint8\_t D3 = 0;
5. static const uint8\_t D4 = 2;
6. static const uint8\_t D5 = 14;
7. static const uint8\_t D6 = 12;
8. static const uint8\_t D7 = 13;
9. static const uint8\_t D8 = 15;
10. static const uint8\_t D9 = 3; บอร์ดเขียนว่า RX
11. static const uint8\_t D10 = 1; บอร์ดเขียนว่า TX

PinOUT ของ ESP8266



รูปที่ 2.12 แสดงขาPinout ของ ESP8266

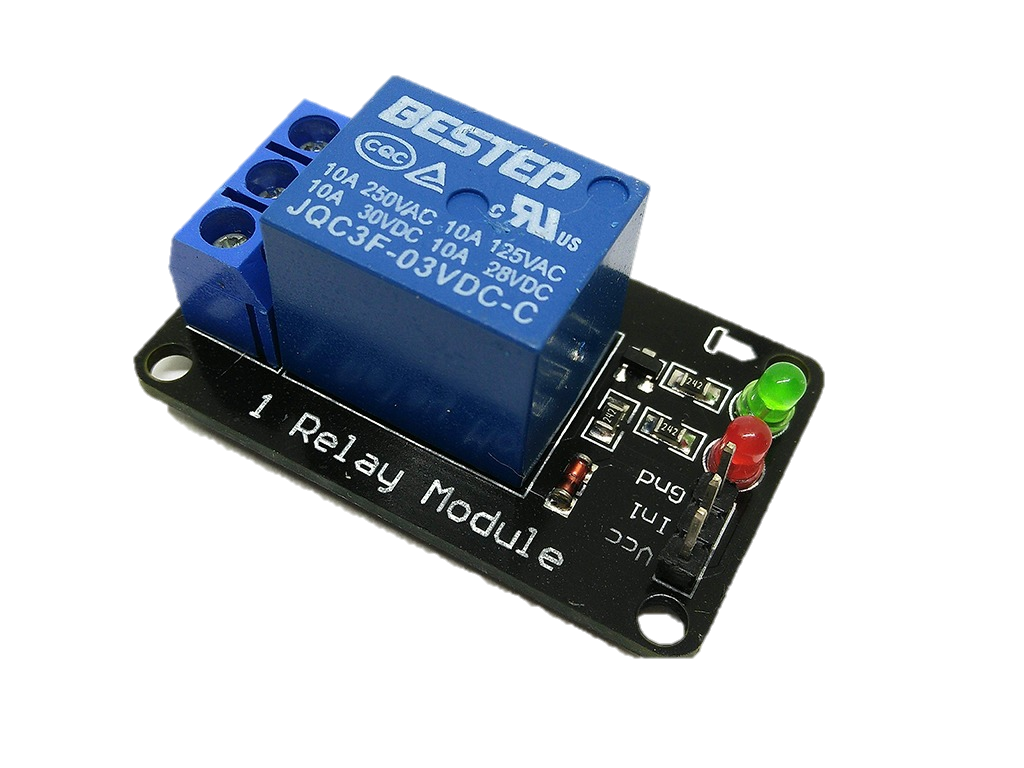
**2.2 Soil Moisture Sensor Module**



รูปที่ 2.13 แสดงรูป เซ็นเซอร์วัดความชื้น

เซนเซอร์วัดความชื้นในดิน ให้เอาต์พุตเป็นอะนาล็อกป้อนให้กับวงจร Arduino เพื่อนำค่าไปเปรียบเทียบตามต้องการ

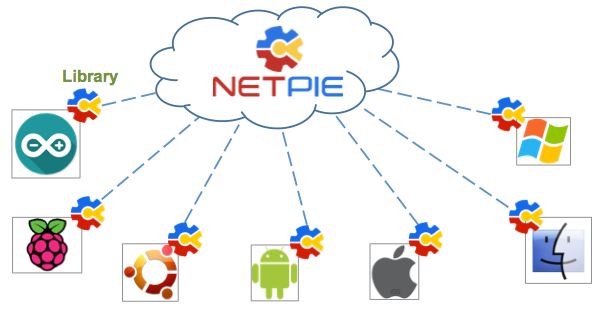
**2.3 Relay 3V module**



รูปที่ 2.14 แสดงรูป โมดูลรีเลย์ 3 โวล

รีเลย์ (Relay) คือ สวิตซ์ตัด-ต่อวงจรแบบอิเล็กทรอนิกส์ โดยใช้หลักการของแม่เหล็กไฟฟ้า จะทำงานเมื่อมีการจ่ายไฟไปตามกำหนดทำให้เกิดวงจรเปิด เมื่อไม่มีการจ่ายไฟจะทำให้เกิดวงจรปิดทำให้อุปกรณ์ไฟฟ้าที่ใช้รีเลย์เป็นสวิตซ์นั้นไม่ทำงาน

**2.4 ความรู้เกี่ยวกับ NETPIE**



รูปที่ 2.15 แสดงรูป NETPIE

NETPIE เป็น IoT (Internet of Things) Cloud Platform ที่พัฒนาขึ้นโดยทีมงานวิจัย และเปิดให้บุคคลทั่วไปใช้งานโดยมี Web Portal ที่ให้สามารถลงทะเบียนและจัดการตัวตนและสิทธิของแอปพลิเคชันและอุปกรณ์ได้ที่เว็บไซต์ https://netpie.io อีกทั้ง NETPIE เป็น Middleware ที่มีหัวใจหลัก (นอกเหนือจากส่วนอื่น ๆ) เป็น Distributed MQTT brokers ซึ่งเป็นเสมือนจุดนัดพบให้สิ่งต่าง ๆ (Things) มาติดต่อสื่อสารและทำงานร่วมกันผ่านวิธีการส่งข้อความแบบ Publish/Subscribe รวมถึง NETPIE มีโครงสร้างสถาปัตยกรรมเป็นคลาวด์อย่างแท้จริงในทุกองค์ประกอบ ทำให้สามารถขยายตัวได้อย่างอัตโนมัติ (Auto-scale) สามารถดูแลและซ่อมแซมตัวเองได้อัตโนมัติเมื่อส่วนหนึ่งส่วนใดในระบบมีปัญหา (Self-healing, Self-recovery) โดยไม่ต้องพึ่งผู้ดูแลระบบ การบริหารจัดการระบบเป็นแบบ Plug and Playไม่ต้อง Configure หรือปรับแต่ง ในฝั่งอุปกรณ์ NETPIE มี Client Library หรือที่เรียกว่า Microgear ซึ่งทำหน้าที่สร้างและดูแลช่องทางสื่อสารระหว่างอุปกรณ์กับ NETPIE รวมไปถึงรักษาความปลอดภัยในการส่งข้อมูล Microgear เป็น Open Source และสามารถดาวน์โหลดได้จาก https://github.com/netpieio โดย ณ ปัจจุบันมี Microgear สำหรับ OS และ Embedded Board หลักๆ ที่เป็นที่นิยมในหมู่นักพัฒนาเกือบทุกชนิด

**ข้อดี ของ Netpie**

1. ช่วยลดการใช้ทรัพยากรของการเชื่อมต่อ

NETPIE ช่วยให้อุปกรณ์สามารถสื่อสารกันได้โดยผู้ใช้ไม่ต้องกังวลว่า อุปกรณ์นั้นจะอยู่ที่ใด เพียงแค่นำ Microgear Library ไปติดตั้งในอุปกรณ์ NETPIE จะรับหน้าที่ดูแลเชื่อมต่อให้ทั้งหมด ไม่ว่าอุปกรณ์นั้นจะอยู่ในเครือข่ายชนิดใด ลักษณะใด หรือแม้กระทั่งเคลื่อนย้ายไปอยู่ที่ใด ผู้ใช้สามารถตัดปัญหาในการเข้าถึงอุปกรณ์จากระยะไกล (Remote Access) ด้วยวิธีการแบบเดิมๆ เช่น การใช้ Fixed Public IP Address หรือการตั้ง Port Forwarding ในเราท์เตอร์และการต้องไปลงทะเบียนกับผู้ให้บริการ Dynamic DNS ซึ่งทั้งหมดล้วนมีความยุ่งยาก ลดความยืดหยุ่นของระบบ ไม่เพียงเท่านั้น NETPIE ยังช่วยให้การเริ่มต้นใช้งานเป็นไปโดยง่าย โดยออกแบบให้อุปกรณ์ถูกค้นพบและเข้าสู่บริการโดยอัตโนมัติ (AutomaticDiscovery, Plug-and-Play)

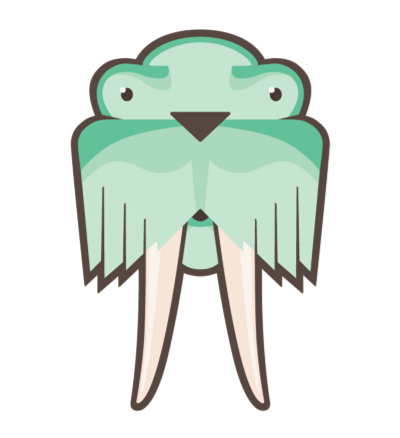
2. ช่วยลดภาระด้านความปลอดภัยของข้อมูล

NETPIE ถูกออกแบบให้มีระดับและสิทธิในการเข้าถึงในระดับ Fine Grain กล่าวคือผู้ใช้สามารถออกแบบได้เองทั้งหมดว่า สิ่งใดมีสิทธิคุยกับสิ่งใด สิ่งใดมีสิทธิหรือไม่-เพียงใดในการอ่านหรือเขียนข้อมูลและสิทธิเหล่านี้จะมีอายุการใช้งานนานเท่าใด หรือจะถูกเพิกถอนภายใต้เงื่อนไขใด เป็นต้น

3. ยืดหยุ่นต่อการขยายระบบ

NETPIE มีสถาปัตยกรรมเป็นคลาวด์เซิร์ฟเวอร์อย่างแท้จริงในทุกองค์ประกอบของระบบ ทำให้เกิดความยืดหยุ่นและคล่องตัวสูงในการขยายตัว นอกจากนี้โมดูลต่างๆ ยังถูกออกแบบให้ทำงานแยกจากกันเพื่อให้เกิดสภาวะ Loose Coupling และสื่อสารกันด้วยวิธี Asynchronous Messaging ช่วยให้แพลตฟอร์มมีความน่าเชื่อถือได้สูง นำไปใช้ซ้ำและพัฒนาต่อได้ง่าย ดังนั้นผู้พัฒนาไม่จำเป็นต้องกังวลกับการขยายตัวเพื่อรับโหลดที่เพิ่มขึ้นในระบบอีกต่อไป

**2.5 ความรู้เกี่ยวกับ surge.sh**



รูปที่ 2.16 แสดงรูป surge.sh

เป็นเครื่องมือที่ช่วยการอัพโหลด Front-end website หรือ Static website

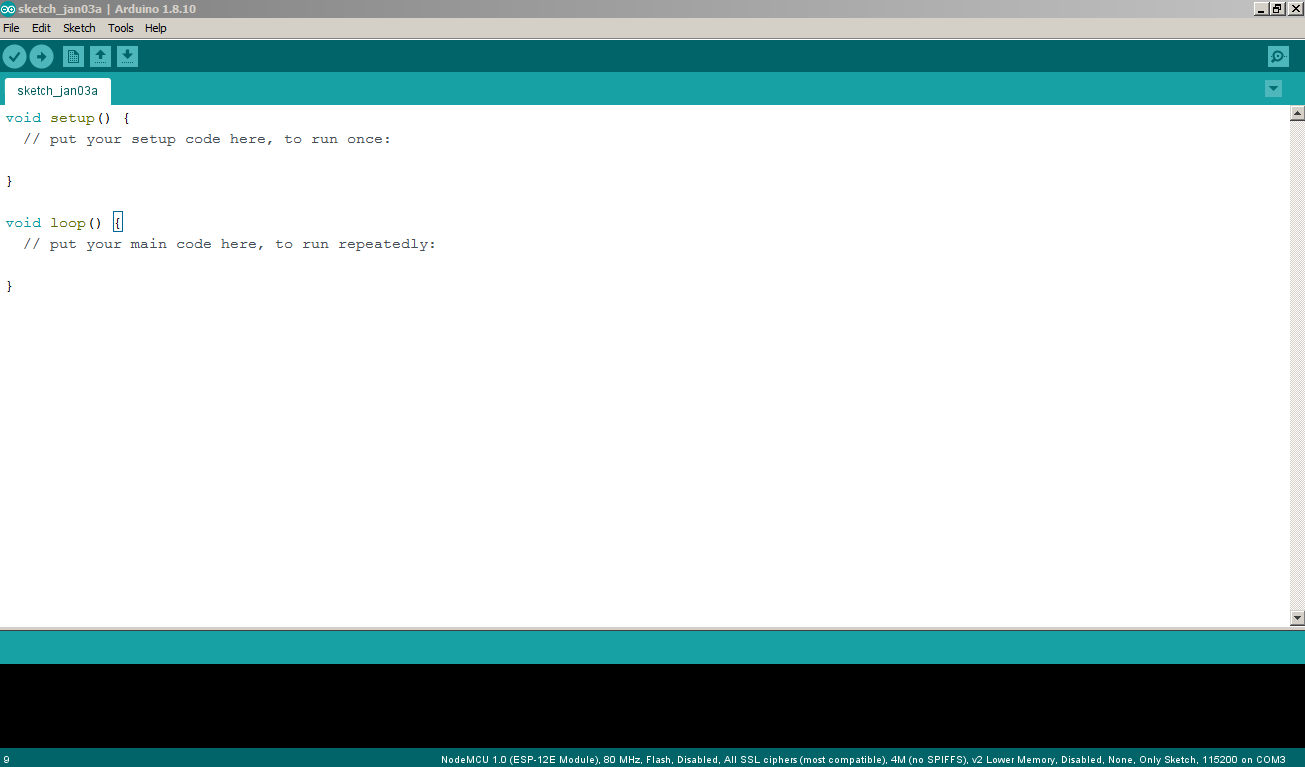
- มีบริการsubdomainฟรีให้ใช้บริการ สามารถเลือกใช้ชื่ออะไรก็ได้ภายใต้โดเมน [name].surge.sh โดยห้ามชื่อห้ามซ้ำกัน

- สามารถใช้ custom domain ได้ฟรี เพียง[ชี้ DNS ไปที่เซิร์ฟเวอร์ของ surge.sh](https://surge.sh/help/adding-a-custom-domain)

- มี SSL ให้หากใช้ subdomain ใต้ surge.sh หากต้องการใช้กับ custom domain ก็สามารถซื้อได้ในเว็บไซต์

- หากต้องหารใช้ชื่อโดเมนเดียวกันโดยไม่ต้องพิมพ์ซ้ำ ๆ สามารถสร้างไฟล์ชื่อ CNAME ที่มีชื่อ domain ข้างในไฟล์ ภายในไดเร็กทอรี่ของโปรเจกต์ได้เหมือนกัน ต่อไปเวลาใช้คำสั่ง surge . ก็จะอัพโหลดให้เลยไม่ต้องกรอกชื่อโดเมนเองทุกครั้ง

**2.6 ความรู้เกี่ยวกับ Arduino IDE**



รูปที่ 2.17 แสดงโปรแกรม Arduino IDE

Arduino IDE คือซอฟต์แวร์ที่ใช้ในการพัฒนางานสำหรับบอร์ด Arduino นั่นคือโปรแกรมที่เรียกว่า Arduino IDE ในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนาหรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น

**แนวคิดการใช้งานโปรแกรม Arduino IDE**

1. เขียนโปรแกรมด้วยภาษา C/C++ สำหรับ Arduino
2. คอมไพล์หรือแปลโปรแกรมภาษา C/C++ ให้เป็นภาษาสำหรับไมโครคอนโทรลเลอร์และบันทึกเป็น [Intel Hex File](https://en.wikipedia.org/wiki/Intel_HEX)
3. อัปโหลด Intel Hex File ลงบนไมโครคอนโทรลเลอร์ซึ่งอยู่บนบอร์ด Arduino ผ่านสาย USB หรือผ่าน [Programmer](https://en.wikipedia.org/wiki/Programmer_(hardware))