บทที่ 2  
เอกสารและทฤษฎีที่เกี่ยวข้อง

ในการจัดทำโครงงานการพัฒนาระบบควบคุมห้องเรียนอัจฉริยะ ด้วย Internet of things (IoT) ผู้จัดทำโครงงาน ได้ศึกษาทฤษฎีและเอกสารที่เกี่ยวข้อง ดังต่อไปนี้

2.1 Internet of things (IoT)

2.2 MQTT (Message Queue Telemetry Transport)

2.3 Arduino IDE

2.4 อุปกรณ์ควบคุม IoT

2.4.1 ESP32

2.4.2 ESP-01

2.4.3 Relay

2.4.4 Infrared Module

2.5 อุปกรณ์ ตรวจวัด (Sensor)

2.5.1 AC Current click bundle Current Measurement

2.5.2 DHT11

2.6 การทดลองและออกแบบวงจร

2.6.1 Proteus

2.6.2 EasyEDA

2.1 Internet of things (IoT)

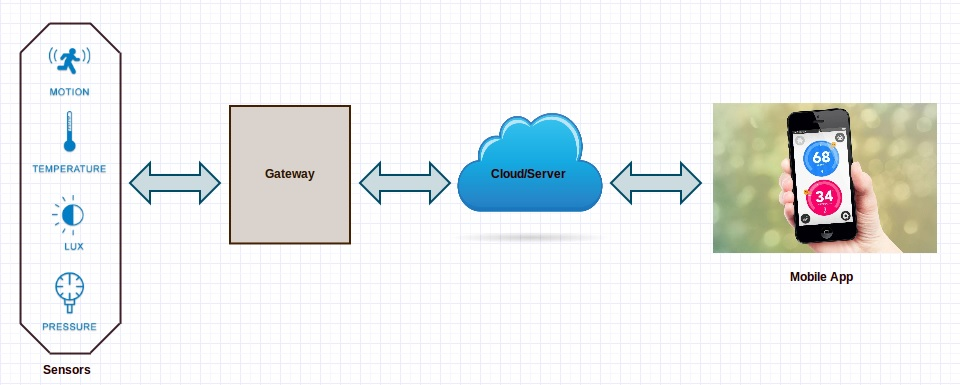
Internet of Things (IoT) อินเทอร์เน็ตในทุกสิ่ง การที่อุปกรณ์ต่าง ๆ สิ่งต่าง ๆ ได้ถูกเชื่อมโยงทุกสิ่งทุกอย่างสู่โลกอินเทอร์เน็ต ทำให้สามารถสั่งการควบคุมการใช้งานอุปกรณ์ต่าง ๆ ผ่านทางเครือข่ายอินเทอร์เน็ต เช่น การเปิด ปิด อุปกรณ์เครื่องใช้ไฟฟ้า การสั่งการเปิดไฟฟ้าภายในบ้านด้วยการเชื่อมต่ออุปกรณ์ควบคุมโทรศัพท์มือถือ เครื่องมือสื่อสาร ผ่านเครือข่ายอินเทอร์เน็ต เป็นต้น

อินเทอร์เน็ตในทุกสิ่ง มีชื่อเรียกอีกอย่างว่า M2M ย่อมาจาก Machine to Machine คือเทคโนโลยีอินเทอร์เน็ตที่เชื่อมต่ออุปกรณ์กับเครื่องมือต่าง ๆ เข้าไว้ด้วยกัน

เทคโนโลยีอินเทอร์เน็ตในทุกสิ่งมีความจำเป็นต้องทำงานร่วมกับอุปกรณ์ประเภท RFID และ Sensors ซึ่งเปรียบเสมือนการเติมสมองให้กับอุปกรณ์ต่าง ๆ ที่ขาดไม่ได้คือการเชื่อมต่ออินเทอร์เน็ต เพื่อให้อุปกรณ์สามารถรับส่งข้อมูลถึงกันได้ เทคโนโลยีอินเทอร์เน็ตในทุกสิ่งมีประโยชน์ในหลายด้าน

แต่ก็มาพร้อมกับความเสี่ยง เพราะหากระบบรักษาความปลอดภัยของอุปกรณ์ และเครือข่ายอินเทอร์เน็ตไม่ดีพอก็อาจทำให้มีผู้ไม่ประสงค์ดีเข้ามาขโมยข้อมูลหรือละเมิดความเป็นส่วนตัวของเราได้ ดังนั้นการพัฒนาอินเทอร์เน็ตในทุกสิ่งจึงจำเป็นต้องพัฒนามาตรการ และระบบรักษาความปลอดภัยไอทีควบคู่กันไปด้วย

ในการทำงานของอินเทอร์เน็ตในทุกสิ่งนั้น จะมีการตรวจจับข้อมูลผ่านระบบเซ็นเซอร์ โดยข้อมูลเหล่านี้จะมีขนาดเล็กเช่น ข้อมูลทั่วไปขนาดเล็ก ตัวอย่างเช่น ค่าอุณหภูมิ ค่าความเข้มของแสง ระดับความชื้น ตำแหน่ง เมื่อถูกส่งข้อมูลผ่านระบบเครือข่ายขึ้นไประบบคลาวด์ ข้อมูลจะถูกประมวลผล เพื่อช่วยอำนวยการในการตัดสินใจในการทำกิจกรรมต่าง ๆ ยกตัวอย่างเช่น ระบบรดน้ำอัตโนมัติสำหรับแปลงผัก จะมีอุปกรณ์เซ็นเซอร์ฝังไว้ที่ดินเพื่อวัดระดับความชื้น อุณหภูมิ แสงและส่งผ่านข้อมูลเหล่านั้นทางอินเทอร์เน็ตเพื่อไปประมวลผลกับข้อมูลที่มาจากการพยากรณ์อากาศประจำวัน เมื่อได้ผลลัพธ์จากการประมวลผลแล้ว ระบบจะส่งข้อมูลย้อนกลับไปที่แปลงผักเพื่อสั่งให้เปิด ปิดวาล์วน้ำ บนพื้นฐานข้อมูลปัจจุบันกับปริมาณน้ำที่เหมาะสม ทำให้ผู้ใช้ประหยัดเวลาและเงินในการรดน้ำแปลงผักในแต่ละวัน



ภาพที่ 2-1 การทำงานของ Internet of Things (IoT)

ประโยชน์ของอินเทอร์เน็ตในทุกสิ่ง

ช่วยเพิ่มประสิทธิภาพในการทำงาน อินเทอร์เน็ตในทุกสิ่งจะช่วยเพิ่มประสิทธิภาพในการทำงานให้แม่นยำ และรวดเร็วยิ่งขึ้น เนื่องจากความสามารถในการทำงาน และการส่งผ่านข้อมูลของ อินเทอร์เน็ตในทุกสิ่งนั้นสูงกว่าการใช้มนุษย์ทำงาน การทำงานของมนุษย์อาจจะทำให้เกิด Human Error และเกิดข้อจำกัดด้านพลังงาน ,เวลา และสถานที่ได้

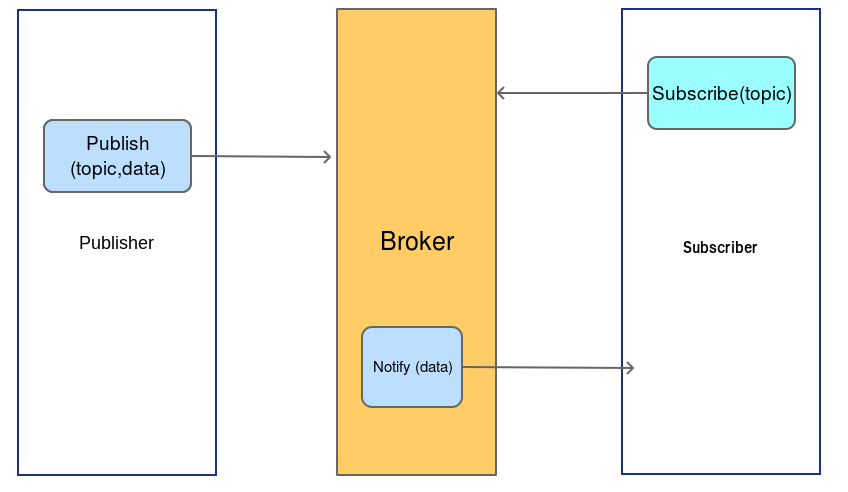
ไร้ข้อจำกัดด้านเวลาและสถานที่ อินเทอร์เน็ตในทุกสิ่งสามารถทำงานได้แบบไร้พรมแดน เพราะขับเคลื่อนด้วยอินเทอร์เน็ต และยังสามารถทำงานได้ตลอดเวลา ต่างจากมนุษย์ที่มีพลังงานจำกัดต้องการการพักผ่อน

ในด้านการศึกษาบ่งบอกถึงอนาคตการทำงานร่วมกันสำหรับระบบการศึกษา. อุปกรณ์ IoT ประกอบด้วยเทคโนโลยีการสร้างอัจฉริยะ เช่น ลำโพง, ล็อค, เทอร์โมสตัท , สมาร์ทไลท์ติ้ง, กริ่งประตู. สามารถเชื่อมต่อถึงกันได้และควบคุมจากส่วนกลางผ่านระบบต่างๆ

ด้าน Smart Home เป็นการใช้เทคโนโลยีควบคุมอุปกรณ์ต่าง ๆ ภายในบ้านให้ทำงานร่วมกัน เช่น การควบคุมอุปกรณ์ไฟฟ้า การตรวจจับสภาพแวดล้อมในบ้านเพื่อควบคุมเครื่องปรับอากาศ การตรวจจับความเคลื่อนไหวเพื่อแจ้งเตือนไปยัง Smart Device

2.2 MQTT (Message Queue Telemetry Transport)

Message Queuing Telemetry Transport (ADMIN, 2559 : Web Site) เป็นProtocol ที่ ออกแบบมาเพื่อการเชื่อมต่อแบบ M2M (machine-to-machine)คืออุปกรณ์กับอุปกรณ์ สนับสนุนเทคโนโลยี iot (Internet of Things) คือเทคโนโลยีที่อินเทอร์เน็ตเชื่อมต่อกับอุปกรณ์ต่าง ๆ เช่น โทรศัพท์มือถือ รถยนต์ โทรทัศน์ ตู้เย็น เข้ากับอินเทอร์เน็ตทำให้สามารถเชื่อมโยงสื่อสารกับอุปกรณ์ต่าง ๆ ได้ โดยผ่านเครือข่าย อินเทอร์เน็ต ซึ่งจะทำให้มนุษย์สามารถ ควบคุมอุปกรณ์ต่าง ๆ จากที่อื่นได้ เช่นการสั่งปิดเปิดไฟในบ้านจากที่ อื่น ๆ เนื่องจากโปรโตคอลตัวนี้มีน้ำหนักเบา ออกแบบมาเพื่อใช้งานกับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก การรับส่งข้อมูลในเครือข่ายที่มีขนาดเล็ก แบนร์วิธต่ำ ใช้หลักการแบบ publisher / subscriber คล้ายกับ หลักการที่ใช้ใน Web Service ที่ต้องใช้ Web Server เป็นตัวกลางระหว่างคอมพิวเตอร์ของผู้ใช้ แต่ MQTT จะใช้ตัวกลางที่เรียกว่า Broker เพื่อทำหน้าที่ จัดการคิว รับ - ส่ง ข้อมูลระหว่างอุปกรณ์ และทั้งในส่วนที่เป็น Publisher และ Subscriber



ภาพที่ 2-2 แสดงหลังการทำงานของโปรโตคอล MQTT

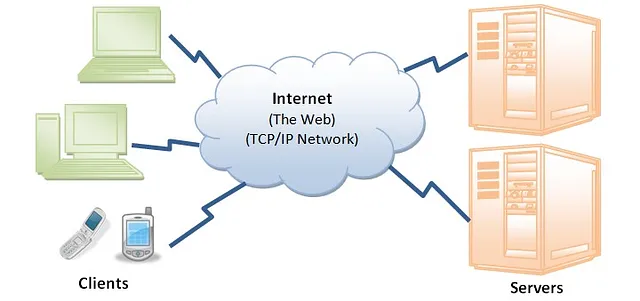
MOTT มีสถาปัตยกรรมแบบ client/server ซึ่งมีหลักการทำงานของ publish/subscribe เหมาะกับการใช้งานที่รองรับอุปกรณ์ที่มีทรัพยากรจำกัด อุปกรณ์ปลายทางจะทำหน้าที่เป็น client ซึ่งทำการเชื่อมต่อบน TCP ไปยังเซิร์ฟเวอร์ หรือเรียกว่า Broker มีหน้าที่เป็นตัวส่งข้อมูลในการรับส่ง message ระหว่าง client ที่เป็นทั้ง publisher และ subscriber จุดเด่นของการสื่อสารแบบ MQTT คือสามารถรองรับ QoS ที่แตกต่างกัน 3 ระดับ โดยคุณภาพข้อมูล QoS แบ่งออกเป็น 3 ระดับดังนี้

QoS0 – ส่งข้อมูลเพียงครั้งเดียว ไม่สนใจว่าผู้รับจะได้รับหรือไม่

QoS1 – ส่งข้อมูลเพียงครั้งเดียว ไม่สนใจว่าผู้รับจะได้รับหรือไม่ แต่ให้จำค่าที่ส่งล่าสุดไว้ เมื่อมีการเชื่อมต่อใหม่จะได้รับข้อมูลครั้งล่าสุดอีกครั้ง

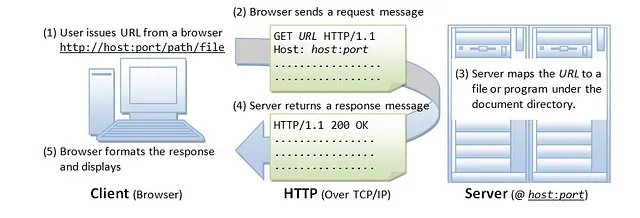
QoS2 – ส่งข้อมูลหลาย ๆ ครั้งจนกว่าปลายทางจะได้รับข้อมูล มีข้อเสียที่สามารถทำงานได้ช้ากว่า QoS0 และ QoS1

TCP/IP (Transmission Control Protocol/Internet Protocol)คือระบบการสื่อสารผ่าอินเตอร์เน็ตระหว่าง 1 อุปกรณ์กับอีก 1 อุปกรณ์ โดยที่จะต้องทำการ connect กันก่อนถึงจะส่งหรือรับข้อมูลหากันได้ อย่างที่ได้บอกไปด้านบนการทำงานของ TCP/IP นั้นจะรับประกันว่าข้อมูลของเราจะไม่หล่นหายระหว่างทาง และรับประกันว่าผู้รับจะได้รับข้อมูลที่เราส่งไปทั้งหมด (ไม่เหมือนกับ UDP ที่ส่งข้อมูลไปแล้วจะไม่สนใจว่าผู้รับจะได้รับข้อมูลครบหรือไม่) แต่อย่างที่ได้กล่าวไปข้างต้น TCP/IP เป็นการสื่อสารแบบ point-to-point หน้าที่ของมันคือการที่ต้องรับประกันว่าข้อมูลนั้นได้ถูกส่งไปครบถ้วนเท่านั้น TCP/IP อาจจะใช้กับระบบ IoT ได้ แต่ลองคิดดูถ้าเรามีหลายๆอุปกรณ์อยู่ในระบบ เราต้องไปไล่ส่งข้อมูลอุปกรณ์ต่ออุปกรณ์ ทีล่ะคู่ ๆ นั้นทำให้เปลืองพลังงานมาก แถมยังต่องมาคอย connect กันอีกนั้นทำให้ TCP/IP ไม่เหมาะสมกับระบบ IoT ใหญ่ ๆ สักเท่าไหร่นัก



**ภาพที่ 2-3 แสดงหลังการทำงานของโปรโตคอล TCP/IP**

HTTP (the Hypertext Transfer Protocol) คือการสื่อสารระหว่าง Web server and Clients (web browsers) โดยที่จะใช้คอนเซปต์ของการ decentralization มันใช้การการสื่อสารแบบ request/response หรือหมายความว่า จะต้องมี response กลับไปหาผู้ที่ส่ง request มาเสมอ (ไม่เหมือน TCP/IP ที่อุปกรณืฝั่งหนึ่งอาจจะแค่รับหรือแค่ส่งอย่างเดียวก็ได้) HTTP พัฒนาต่อมาจาก TCP/IP เหมือนกันดังนั้นข้อมูลข้อเราก็จะไม่หล่นหายเช่นเดียวกัน HTTP นั้นใช้ Web server ดังนั้น Client จะเป็นใครก็ได้ในโลกถ้ามี internet ตัว server นั้นจะรับ request มา และประมวลผลแล้วทำการแจกแจงข้อมูลและส่งกลับไปยังผู้ที่ส่ง request มา ตัว client นั้นหลังจากส่งแล้วก็จะต้องรอจนกว่าจะได้รับข้อมูลกลับมา ภาพข้างล่างอธิบายเกี่ยวกับการทำงานของ HTTP อย่างละเอียดขึ้น



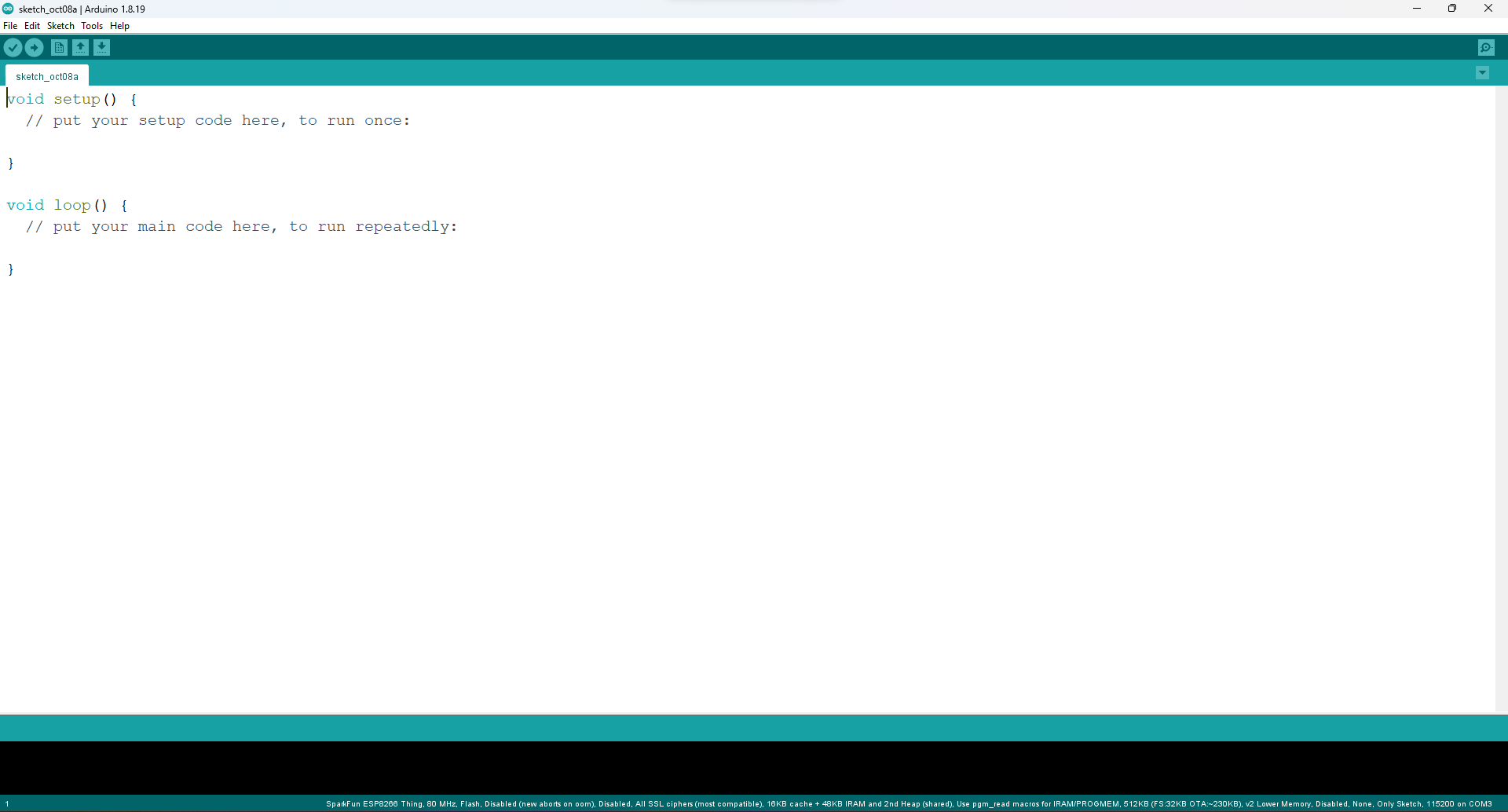
**ภาพที่ 2-4 แสดงหลังการทำงานของโปรโตคอล HTTP**

จากข้อมูลข้างต้นทั้ง TCP/IP และ HTTP จึงไม่เหมาะกับระบบอินเทอร์เน็ตในทุกสิ่งเท่าไหร่นัก เพราะสิ่งที่ต้องการในระบบอินเทอร์เน็ตในทุกสิ่ง คือการส่งข้อมูลแบบ one-to-many ความเร็วในการส่งข้อมูลซึ่งเป็นแค่ข้อมูลขนาดเล็ก และสุดท้ายคือการรับข้อมูลตลอดเวลาเมื่อข้อมูลมีการอัพเดตดังนั้นระบบสื่อสารแบบ publish/subscribe นั้นจึงเหมาะสมกว่า

2.3 Arduino IDE

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

โปรแกรม Arduino IDE เป็นโปรแกรมที่ใช้สำหรับการเขียนโปรแกรมควบคุมและสามารถทำการแปลงไฟล์ดังกล่าวเพื่อนำไปอัพโหลดลงยังบอร์ด Arduino โดยสามารถที่จะเลือกใช้โปรแกรมแบบ online IDE หรือ desktop IDE ได้



**ภาพที่ 2-5 หน้าโปรแกรม Arduino ide**

โปรแกรมแบบ desktop IDE เป็นโปรแกรมแบบ offline นั่นคือ เราจะต้องทำการดาวน์โหลดโปรแกรมที่ https://www.arduino.cc/en/software ซึ่งจะมีให้เลือกตามการใช้งาน OS ของเรา

ข้อดีของซอฟต์แวร์ Arduino

รองรับการพัฒนาในบอร์ดหลายโมเดล และสามารถเพิ่มบอร์ดอื่น ๆ ที่ไม่ใช่บอร์ด Arduino ให้สามารถพัฒนาโปรแกรม และอัพโหลดโปรแกรมลงบนบอร์ดได้

ภายใน Arduino IDE ประกอบไปด้วยชุดคำสั่งต่าง ๆ ตัวอย่างโปรแกรม และตัวอย่างไลบรารี่ต่าง ๆ ที่ได้ติดตั้งมาพร้อมใช้งานเเล้ว สามารถดาวน์โหลดไลบรารี่เพิ่มเติมและติดตั้งเพิ่มได้เติมได้มี Cloud แบบออนไลน์ ใช้สำหรับเก็บข้อมูล ดึงข้อมูลไปแสดงผลได้

**2.4 อุปกรณ์ควบคุม IoT**

2.4.1 ESP32

ESP32 เป็นชิปไมโครคอนโทรลเลอร์ที่มาพร้อม WiFi มาตรฐาน 802.11 b/g/n และบลูทูธเวอร์ชั่น 4.2 เป็นรุ่นต่อยอดความสำเร็จของ ESP8266 โดยในรุ่นนี้ได้ออกมาแก้ไขข้อเสียของ ESP8266 ทั้งหมด



**ภาพที่ 2-6** ชิปไมโครคอนโทรลเลอร์ ESP32

โดย CPU ใช้สถาปัตยกรรม Tensilica LX6 จำนวน 2 คอร์ สัญญาณนาฬิกา 240MHz สามารถแยกการทำงานระหว่างโปรแกรมจัดการ WiFi และแอพพลิเคชั่นออกจากกันได้ ทำให้มีสเถียรภาพเพิ่มขึ้นมาก มีแรม 520KB มาในตัว  นอกจากนี้ยังมี GPIO เพิ่มขึ้นมาก และมีช่อง ADC เพิ่มขึ้นเป็น 12 ช่อง จากเดิม ESP8266 มีเพียงช่องเดียว ใช้แรงดันไฟฟ้า 3.3V ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA ผลิตโดยบริษัท Espressif จากประเทศจีน

2.4.1.1 ไอซี ESP32 มีสเปคโดยละเอียด ดังนี้

2.4.1.1.1 ซีพียูใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz

2.4.1.1.2 มีแรมในตัว 512KB

2.4.1.1.3 รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB

2.4.1.1.4 มาพร้อมกับ WiFi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ Wi-Fi direct

2.4.1.1.5 มีบลูทูธในตัว รองรับการใช้งานในโหมด 2.0 และโหมด   
4.0 BLE

2.4.1.1.6 ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V

2.4.1.1.7 ทำงานได้ที่อุณหภูมิ -40◦C ถึง 125◦C

2.4.1.2 ESP32 ยังมีเซ็นเซอร์ต่าง ๆ มาในตัวด้วย ดังนี้

2.4.1.2.1 วงจรกรองสัญญาณรบกวนในวงจรขยายสัญญาณ

2.4.1.2.2 เซ็นเซอร์แม่เหล็ก

2.4.1.2.3 เซ็นเซอร์สัมผัส (Capacitive touch) รองรับ 10 ช่อง

2.4.1.2.4 รองรับการเชื่อมต่อคลิสตอล 32.768kHz สำหรับใช้กับส่วนวงจรนับเวลาโดยเฉพาะ

2.4.1.3 ขาใช้งานต่าง ๆ ของ ESP32 รองรับการเชื่อมต่อบัสต่าง ๆ ดังนี้

2.4.1.3.1 มี GPIO จำนวน 32 ช่อง

2.4.1.3.2 รองรับ UART จำนวน 3 ช่อง

2.4.1.3.3 รองรับ SPI จำนวน 3 ช่อง

2.4.1.3.4 รองรับ I2C จำนวน 2 ช่อง

2.4.1.3.5 รองรับ ADC จำนวน 12 ช่อง

2.4.1.3.6 รองรับ DAC จำนวน 2 ช่อง

2.4.1.3.7 รองรับ I2S จำนวน 2 ช่อง

2.4.1.3.8 รองรับ PWM / Timer ทุกช่อง

2.4.1.3.9 รองรับการเชื่อมต่อกับ SD-Card

2.4.1.4 ESP32 ยังรองรับฟังก์ชั่นเกี่ยวกับความปลอดภัยต่าง ๆ ดังนี้

2.4.1.4.1 รองรับการเข้ารหัส WiFi แบบ WEP และ WPA/WPA2 PSK/Enterprise

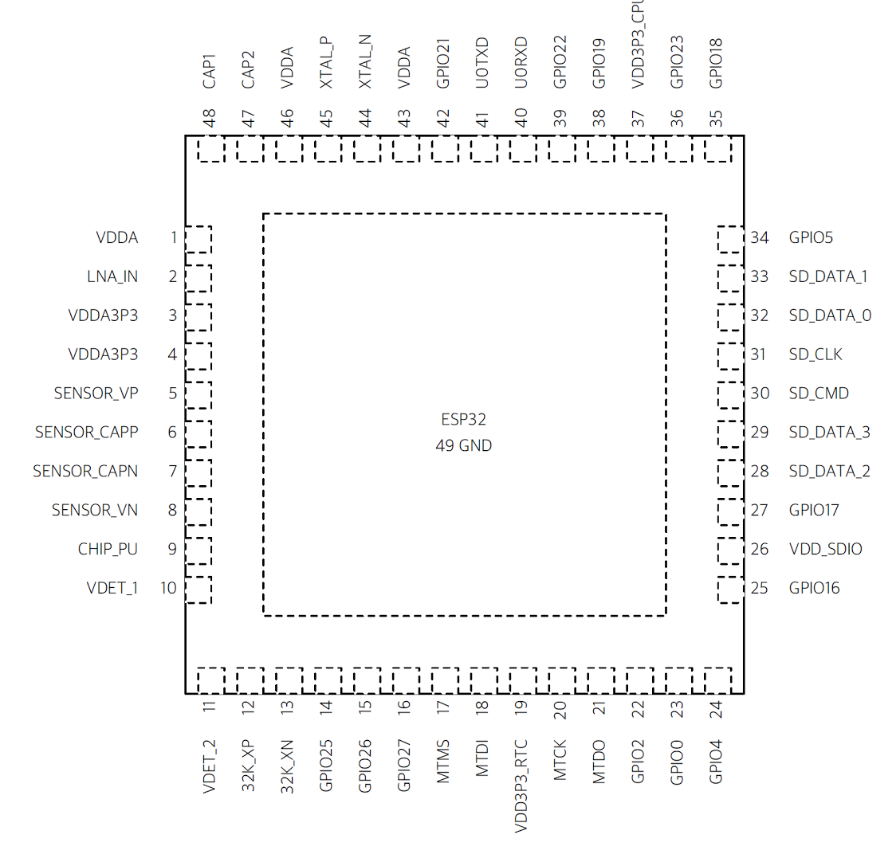
2.4.1.4.2 มีวงจรเข้ารหัส AES / SHA2 / Elliptical Curve Cryptography / RSA-4096 ในตัว

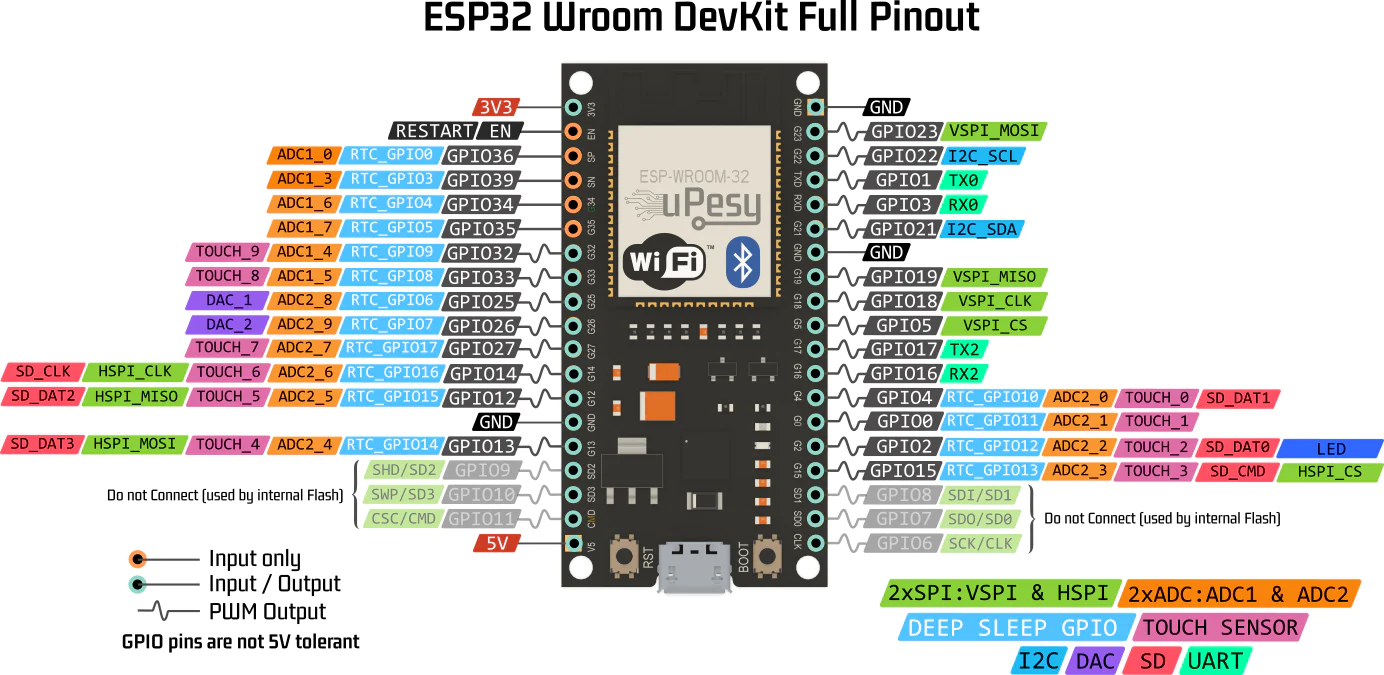
2.4.1.5 ด้านประสิทธิ์ภาพการใช้งาน ESP32 สามารถทำงานได้ดี ดังนี้

2.4.1.5.1 รับ – ส่ง ข้อมูลได้ความเร็วสูงสุดที่ 150Mbps เมื่อเชื่อมต่อแบบ 11n HT40 ได้ความเร็วสูงสุด 72Mbps เมื่อเชื่อมต่อแบบ 11n HT20 ได้ความเร็วสูงสุดที่ 54Mbps เมื่อเชื่อมต่อแบบ 11g และได้ความเร็วสูงสุดที่ 11Mbps เมื่อเชื่อมต่อแบบ 11b

2.4.1.5.2 เมื่อใช้การเชื่อมต่อผ่านโปรโตคอล UDP จะสามารถรับ – ส่งข้อมูลได้ที่ความเร็ว 135Mbps

2.4.1.5.3 ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA

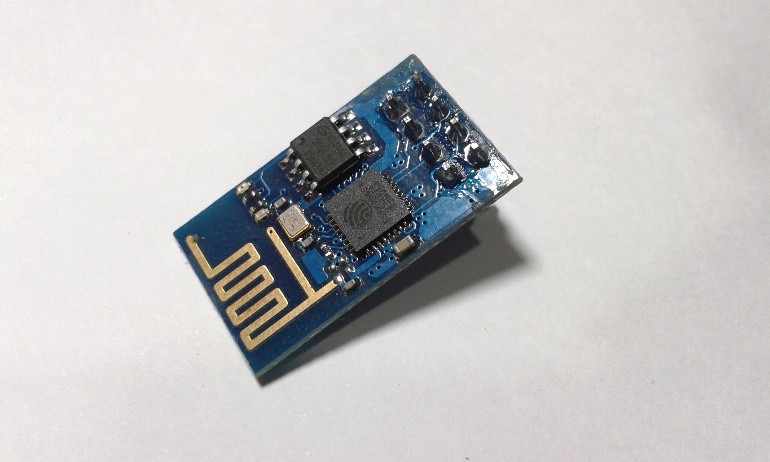
**  
ภาพที่ 2-7 ชนิดขาแต่ละขาของ** ESP32

**ภาพที่ 2-8 วิธีการใช้งานขาแต่ละชนิดของ** ESP32

**2.4.2 ESP-01**

ESP-01 เป็นชิปไมโครคอนโทรลเลอร์ขนิดเดียวกันกับ EPS8266 แต่นำเจ้าตัว ESP8266 นำมาลดขนาดลง และเหมาะสำหรับนำไปใช้งานงานที่โปรแกรมเล็กๆ มีขาทั้งหมด 8 ขา ได้แก่ VCC CH\_PD Reset Rx Tx GPIO0 GPIO2 และ GND โมดูลนี้ทำงานได้ค่อนข้างที่จะช้ามาก หากมีการเขียนโปรแกรมที่ไม่รัดกุมพอ หรือมีคำสั่งทำงานมากๆ โดยเจ้า โมดูล ESP-01 นั้นจะ ถูกออกแบบมาเพื่อเน้นการเชื่อมต่อระบบเครือข่ายไร้สายให้กับไมโครคอนโทรลเลอร์ตัวอื่น ๆ ผ่านทาง RS232 ด้วยชุดคำสั่ง AT โดยเชื่อมต่อสาย RX / TX / CH\_PD / Vcc และ GND เพื่อใช้งาน แต่สามารถเขียนโปรแกรมลงชิพ ESP8266 ได้ด้วยการกำหนดให้ขา GPIO0 เป็น GND

ในตอนบูตระบบ ทำให้โมดูลนี้ทำงานได้หลากหลายมากขึ้น



**ภาพที่ 2-9 ชิปไมโครคอนโทรลเลอร์ ESP-01**

**2.4.2.1 คุณสมบัติของ ESP-01 มีดังนี้**

2.4.2.1.1 หน่วยประมวลผล Tensilica L106 เป็นหน่วยประมวลผลแบบ 32 บิต

2.4.2.1.2 ความถี่สัญญาณนาฬิกา 160MHz

2.4.2.1.3 SRAM สำหรับใช้งานน้อยกว่า 50KB เพราะต้องใช้ร่วมกับบัฟเฟอร์ของการสื่อสาร

2.4.2.1.4 ROM ของ ESP-01 มีขนาด 512KB

2.4.2.1.5 แรงดันไฟฟ้าในการทำงาน 2.5V ถึง 3.6v

2.4.2.1.6 อุณหภูมิ -40 ถึง 125 องศาเซลเซียส

2.4.2.1.7 ความต้องการของกระแสในการทำงานปกติประมาณ 80mA

2.4.2.1.8 ความต้องการกระแสในโหมดประหยัดพลังงาน (Sleep Mode) 20uA

2.4.2.1.9 รองรับมาตรฐาน IEEE 802.11 b/g/n

2.4.2.1.10 รองรับ WPA/WPA2

2.4.2.1.11 รองรับการเข้ารหัส WEP/TKIP และ AES

2.4.2.1.12 รองรับโพรโทคอล IPv4, TCP/UDP/HTTP

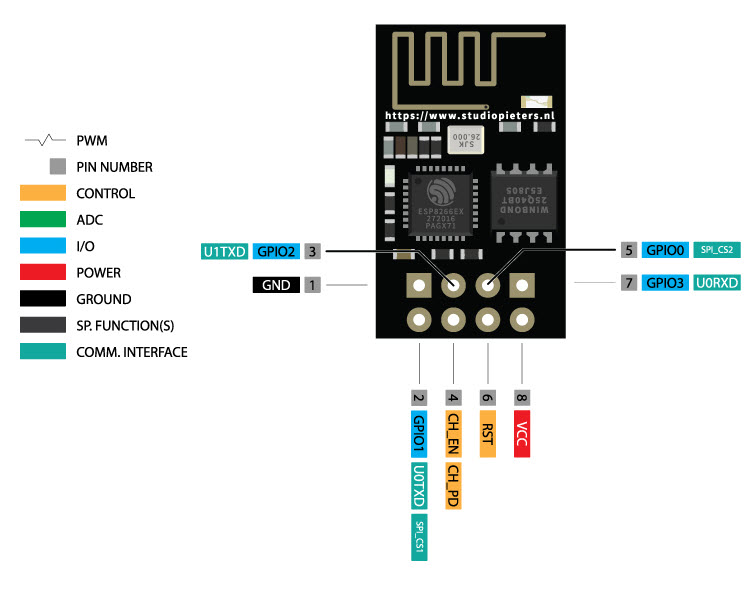
**2.4.2.2 ขาเชื่อมต่อ**

2.4.2.2.1 RX

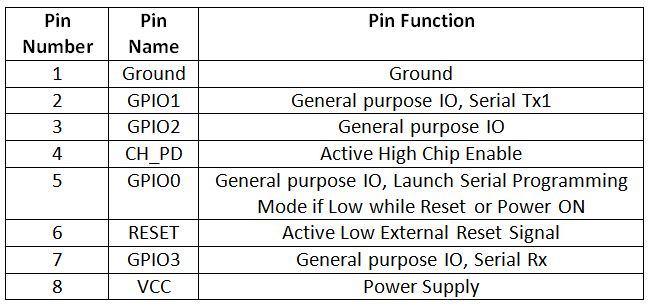
2.4.2.2.2 TX

2.4.2.2.3 GPIO0

2.4.2.2.4 GPIO2



**ภาพที่ 2-10 วิธีการใช้งานขาแต่ละชนิดของ** ESP-01



**ภาพที่ 2-11 วิธีการใช้งานขาแต่ละชนิดของ** ESP-01

**2.4.3 Relay (รีเลย์)**

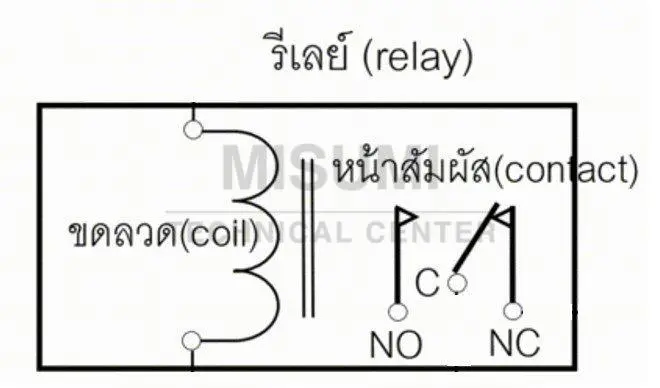
(รีเลย์) คืออุปกรณ์ที่ทำหน้าที่ตัดต่อวงจรให้ระบบไฟฟ้าคอนโทรลและเป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้มาเป็นพลังงานแม่เหล็ก เพื่อใช้ในการเหนี่ยวนำคอนแทคให้เปลี่ยนสถานะการทำงาน โดยการป้อนกระแสไฟฟ้าให้กับคอยล์รีเลย์ เพื่อทำการเปิดปิดคอนแทคคล้ายกับสวิตช์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ เช่น การทำงานของเครื่องจักร งานประกอบตู้ไฟฟ้าคอนโทรล เป็นต้น

**2.4.3.1 หลักการทำงานของรีเลย์**

รีเลย์จะทำงานโดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อเปลี่ยนแรงดันไฟฟ้าให้เป็นพลังงานแม่เหล็ก สำหรับใช้ดึงดูดหน้าสัมผัส(contact)ให้เปลี่ยนทิศทางการไหลของไฟฟ้า เพื่อควบคุมการจ่ายไฟให้กับอุปกรณ์ต่างๆคล้ายกับสวิตซ์

**2.4.3.2 ส่วนประกอบสำคัญของรีเลย์**

**2.4.3.2.1 ขดลวด(coil)**ทำหน้าที่รับแรงดันไฟฟ้าจากวงจรตัวควบคุมหรือ controller เพื่อเหนี่ยวนำกระแสไฟฟ้าให้เปลี่ยนเป็นพลังงานแม่เหล็กในการทำให้ดึงดูดหน้าสัมผัส(contact) ให้เปลี่ยนตำแหน่ง  
 **2.4.3.2.2 หน้าสัมผัส(contact)** ทำหน้าที่เหมือนสวิตช์ ที่กำหนด ทิศทางการจ่ายกระแสไฟฟ้าให้กับอุปกรณ์ที่เราต้องการ



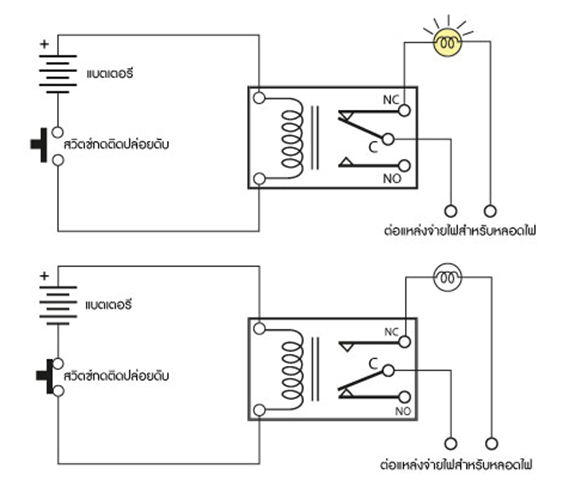
**ภาพที่ 2-12 โครงสร้างภายในของ Relay (รีเลย์)**

2.4.3.3 จุดต่อใช้งานมาตรฐาน ประกอบด้วย

2.4.2.3.1 จุดต่อ NC ย่อมาจาก normal close หมายความว่า ปกติปิด คือ หากยังไม่มีการจ่ายไฟให้ขดลวด(coil) หน้าสัมผัสนี้จะเชื่อมต่อกับจุดต่อ C โดยทั่วไปแล้วเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา

2.4.2.3.2 จุดต่อ NO ย่อมาจาก normal open หมายความว่า ปกติเปิด คือ หากยังไม่มีการจ่ายไฟให้ขดลวด(coil) หน้าสัมผัสจะยังไม่เชื่อมต่อกับจุดต่อ C โดยทั่วไปแล้วเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานในช่วงเวลาจำกัดเท่านั้น

2.4.2.3.3 จุดต่อ C ย่อมาจาก common หมายถึง จุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ [8]



ภาพที่ 2-13 การทำงานของรีเลย์