โครงงานสิ่งประดิษฐ์ กลุ่ม 8
เปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต
(on-off lights with application blynk via internet system)

จัดทำโดย

นรจ.ศุภวุฒิ ขวัญอยู่เย็น
นรจ.เอราวัต นิ่มบุญ
นรจ.ธนกร ปะนามะทัง
นรจ.ธีระวัฒน์ ทรัพย์คง
นรจ.ภานุภัณฑ์ แงชัยภูมิ

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรนักเรียนจ่าทหารเรือ ชั้นปีที่ ๒
พรรค พิเศษ เหล่า ทหารช่างยุทธโยธา (อิเล็กทรอนิกส์) ปีการศึกษา ๒๕๖๑

โรงเรียนอิเล็กทรอนิกส์ กองวิทยาการ กรมอิเล็กทรอนิกส์

หัวข้อโครงงาน เปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต

ผู้จัดทำ

นรจ.ศุภวุฒิ	ขวัญอยู่เย็น
นรจ.เอราวัต	นิ่มบุญ
นรจ.ธนกร	ปะนามะทัง
นรจ.ธีระวัฒน์	ทรัพย์คง
นรจ.ภานุภัณฑ์	แงชัยภูมิ

ครูที่ปรึกษา

น.ต.เสถียร	ตั้งพรประเสริฐ
พ.จ.อ.อนัน	ไฝดีโบ๊ต
จ.อ.ศุภร	สุเมธาวัฒนพงศ์
จ.ต.วิเศรษฐ์	แสงทอง

ปีการศึกษา ๒๕๖๑

บทคัดย่อ

บทความนี้เป็นการนำเสนอ การประยุกต์ใช้เทคโนโลยี ระบบอินเตอร์เน็ต ในการควบคุมอุปกรณ์ไฟฟ้า และสร้างระบบต้นแบบเพื่อแสดงการทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าผ่านระบบอินเตอร์เน็ต เพื่อให้เกิดความสะดวกสบาย และเพิ่มประสิทธิภาพในการใช้งานอุปกรณ์ไฟฟ้า การทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าผ่านโทรศัพท์เคลื่อนที่ สัญญาณควบคุมจะถูกส่งจากโทรศัพท์เคลื่อนที่ไปยัง ไมโครคอนโทรลเลอร์และไมโครคอนโทรลเลอร์ทำการ ประมวลผลเพื่อสั่งการให้อุปกรณ์ไฟฟ้าทำงานด้วย สัญญาณการควบคุมจากไมโครคอนโทรลเลอร์ ระบบการทำงานนั้นพัฒนาจากโปรแกรมต่าง ๆ เช่น ภาษาซี ภาษาไพธอน เป็นต้น และจัดทำเป็นระบบต้นแบบแสดงการทำงานของระบบ การทำงานของระบบควบคุม เป็นไปตามโปรแกรมที่พัฒนาขึ้น และสถานการณ์ทำงานของอุปกรณ์ยังมีการแสดงที่ โทรศัพท์เคลื่อนที่เพื่อให้ผู้ใช้งานหรือผู้ควบคุมระบบทราบ

กิตติกรรมประกาศ

โครงงานสิ่งประดิษฐ์นี้สำเร็จได้ด้วยความกรุณาของผู้อำนวยการโรงเรียนอิเล็กทรอนิกส์ กองวิทยาการ กรมอิเล็กทรอนิกส์ทหารเรือและครูที่ปรึกษาซึ่งได้ให้คำที่ปรึกษา ข้อชี้แนะ และความช่วยเหลือต่าง ๆ ที่ เป็นประโยชน์จนกระทั่งโครงงานสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้ ขอกราบขอบพระคุณอาจารย์ที่ปรึกษาที่ให้ความกรุณาในการแก้ไขข้อบกพร่องต่าง ๆ ของโครงงาน และให้ความรู้ให้คำแนะนำ ทั้งกำลังใจ สุดท้ายนี้คณะผู้จัดทำหวังเป็นอย่างยิ่งว่า โครงงานนี้จะเป็นประโยชน์ต่อการศึกษาและเป็นที่น่าสนใจสำหรับผู้ที่สนใจต่อๆไป

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ค
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
บทที่ 3 ขั้นตอนการดำเนินการ	35
บทที่ 4 ผลการทดลอง	44
บทที่ 5 สรุปผล อภิปรายผล และข้อเสนอแนะ	47
บรรณาณุกรม	48
ภาคผนวก	49

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันอุปกรณ์ควบคุมการทำงานของอุปกรณ์ไฟฟ้าไม่ว่าจะเป็นตามบ้านหรือตามอาคารกองวิทยาการนั้นจะถูกควบคุมโดยผู้ใช้ผ่านทางสวิตช์เป็นหลัก ซึ่งบางครั้งไม่สะดวกและไม่คล่องตัว และอาจจะมีบ่อยครั้งที่อาจจะลืมปิดไฟฟ้าหลังจากออกจากอาคาร ซึ่งทำให้เกิดการสิ้นเปลืองพลังงานโดยไม่จำเป็น หรือในบางครั้งเราอาจไม่ได้อยู่ที่พักอาศัยนาน ๆ เช่น การเดินทางไปต่างจังหวัดนาน ๆ แล้วเปิดไฟทิ้งไว้ซึ่งอาจทำให้เกิดไฟฟ้าลัดวงจร ดังนั้นหากเราสามารถควบคุมไฟฟ้าภายในอาคารผ่านระบบอินเตอร์เน็ตได้จะเป็นสิ่งที่ดีมากเพื่ออำนวยความสะดวกและตอบสนองความต้องการของเรา

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อลดความเสี่ยงการสูญเสียพลังงานที่ไม่จำเป็น

1.2.2 เพื่อสะดวกต่อการเปิด-ปิดไฟภายในอาคาร

1.2.3 เพื่อประยุกต์ใช้เทคโนโลยีมีความสอดคล้องกับชีวิตประจำวันมากที่สุด

1.3 สมสุติฐาน

1.3.1 การควบคุมอุปกรณ์ไฟฟ้าสามารถควบคุมได้โดยตรงผ่านสวิตช์

1.3.2 การควบคุมอุปกรณ์ไฟฟ้าผ่านทางโทรศัพท์เคลื่อนที่ และ การควบคุมโดยตรงผ่าน สวิตช์สามารถเชื่อมโยงกันได้

1.3.3 การควบคุมไฟผ่านระบบอินเตอร์เน็ตสามารถควบคุมจากโทรศัพท์เคลื่อนที่

1.4 รายละเอียดในการดำเนิน

1.4.1 วางแผนจัดทำและค้นคว้า

1.4.2 ดำเนินการทำโครงงาน

1.5 ขั้นตอนการทำงาน

1.5.1 PLAN

- ออกแบบวงจร
- ออกแบบกล่องติดตั้ง
- จัดหาอุปกรณ์

1.5.2 Do

- เขียนคำสั่ง controller
- ต่อวงจร
- อัพโหลดคำสั่ง controller ลงบอร์ด mcu esp 8266

1.5.3 Chdeck

- ทดสอบเปิดไฟโดยการกดปุ่ม On บน Application
- ทดสอบปิดไฟโดยการกดปุ่ม Off บน Application

1.5.4 Action

- แก้ไขปรับปรุงส่วนที่บกพร่อง

1.6 ขอบเขตโครงงาน

1.6.1 สร้างกล่องสำหรับติดตั้งวงจร

1.6.2 เขียนโปรแกรม controller

1.6.3 ติดตั้งอุปกรณ์ตามจุดต่างๆที่ต้องการจะใช้

1.7 ขอบเขตอุปกรณ์

1.7.1 บอร์ด NodeMCU-V3-ESP8266

1.7.2 relay 4 ch for Arduino

1.7.3 สายจั้ม ตัวผู้

1.7.4 หลอดไฟฟลูออเรสเซ็นต์ พร้อมราง

1.7.5 ขั้วหลอดไฟ พร้อมสาย

1.7.6 หลอดตะเกียบ

1.7.7 Switching Power supply 5V 10A

1.7.8 สายไฟ 20 เมตร

1.8 ระยะเวลา

1.8.1 ขั้นเตรียมการ 15 ต.ค. 61 – 20 ม.ค. 62

1.8.2 ขั้นดำเนินการจัดทำโครงงาน 21 ม.ค. 62 - 15 มี.ค. 62

1.9 ประโยชน์ที่คาดว่าจะได้รับ

1.9.1 สมรรถนะของผู้ทำโครงงานสิ่งประดิษฐ์

1.9.1.1 ☑ ความสามารถในการสื่อสาร

1.9.1.2 ☑ ความสามารถในการคิด

1.9.1.3 ☑ ความสามารถในการแก้ปัญหา

1.9.1.4 ☑ ความสามารถในการใช้ทักษะชีวิต

1.9.1.5 ☑ ความสามารถในการใช้เทคโนโลยี

1.9.2 ทักษะในการจัดทำโครงงานสิ่งประดิษฐ์ (3Rs 8Cs)

1.9.2.1 ทักษะในสาระวิชาโครงการวิชาชีพ ผ่านโครงงานสิ่งประดิษฐ์ (3Rs)

1.9.2.1.1 ☑ (R)Eeading (อ่าน)

1.9.2.1.2 ☑ (W)Riting (เขียน)

1.9.2.1.3 ☑ (A)Rithemetics (คิดเลขเป็น)

1.9.2.2 ทักษะการเรียนรู้และการจัดทำโครงงานสิ่งประดิษฐ์ (8Cs)

1.9.2.2.1 ☑ Critical Thinking and Problem Solving (การคิดวิจารณญาณ และแก้ปัญหา)

1.9.2.2.2 ☑ Creativity and Innovation (การสร้างสรรค์ และนวัตกรรม)

1.9.2.2.3 ☑ Cross-cultural Understanding (ความเข้าใจความต่างวัฒนธรรม)

1.9.2.2.4 ☑ Collaboration,Teamwork and Leadership (การทำงานเป็นทีม ภาวะผู้นำ)

1.9.2.2.5 ☑ Communications, Information, and Media Literacy (การสื่อสาร สารสนเทศ)

1.9.2.2.6 ☑ Computing and ICT Literacy (ทักษะด้านคอมพิวเตอร์ และเทคโนโลยี)

1.9.2.2.7 ☑ Career and Learning Skills (ทักษะอาชีพ และการเรียนรู้)

1.9.2.2.8 ☑ Compassion (คุณธรรม เมตตา กรุณา ระเบียบวินัย)

1.9.3 อื่น ๆ (ระบุ)

1.9.3.1 ทำให้ผู้ใช้มีเวลาในการทำงานอื่น ๆ ได้มากขึ้น

1.9.3.2 ผู้ใช้นำไปประยุกต์ใช้งานใน อาคารบ้านเรือน อย่างแพร่หลาย

1.9.3.3 สามารถลดความกังวล ที่จะสูญเสียพลังงานโดยสิ้นเปลือง

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

Arduino ESP8266 (NodeMCU)

ผู้สร้างชิป ESP คือคุณ Teo Swee Ann ชาวสิงคโปร์แห่งบริษัท Espressif System โดยในโมดูลประกอบด้วย ชิป Microcontroller + WiFi Module ราคาถูก ดังนั้นตัวมันสามารถโปรแกรม ลงไปได้ทำ ให้สามารถนำ ไปใช้งานแทนไมโครคอนโทรลเลอร์ได้เลยและมีพื้นที่โปรแกรมที่มากถึง 4MB ทำให้มีพื้นที่เหลือมากในการเขียนโปรแกรมลงไป

ESP8266 เป็นชื่อของชิปไอซีบนบอร์ดของโมดูล ซึ่งไอซี ESP8266ไม่มีพื้นที่โปรแกรม (flash memory) ในตัว ทำให้ต้องใช้ไอซีภายนอก (external flash memory) ในการเก็บโปรแกรม ที่ใช้การเชื่อมต่อผ่าน โปรโตคอล SPI ซึ่งสาเหตุนี้เองทำให้โมดูล ESP8266 มีพื้นที่โปรแกรมมากกว่า ไอซีไมโครคอนโทรลเลอร์เบอร์อื่นๆ

ESP8266 สำหรับติดต่อสื่อสารบนมาตรฐาน WiFi ทำงานที่แรงดันไฟฟ้า 3.3-3.6V ทำงานใช้กระแสโดยเฉลี่ย 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

ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3V -3.6V การนำไปใช้งานร่วมกับ เซ็นเซอร์อื่นๆที่ใช้แรงดัน 5V ต้องใช้ วงจรแบ่งแรงดัน มาช่วยเพื่อไม่ให้โมดูลพังเสียหายกระแสที่โมดูลใช้งานสูงสุดคือ 200mA ความถี่คริสตอล 40MHz ทำให้เมื่อนำไปใช้งานกับอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำให้การแสดงผลข้อมูล รวดเร็วกว่าไมโครคอนโทรลเลอร์ยอดนิยมอย่าง Arduino มาก

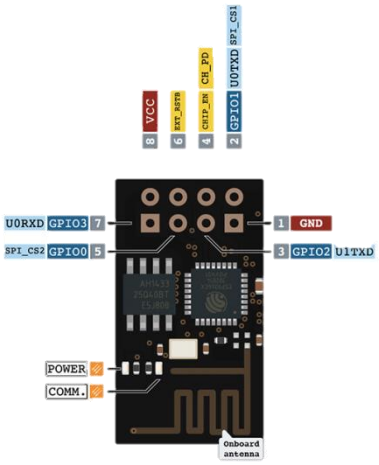
ขาของโมดูล ESP8266แบ่งได้ดังนี้

- VCC เป็นขาสำหรับจ่ายไฟเข้าเพื่อให้โมดูลทำงานได้ซึ่งแรงดันที่ใช้งานได้คือ 3.3 -3.6V
- GND เป็นขาสำหรับต่อกับไฟ 0V
-Reset และ CH_PD (หรือ EN) เป็นขาที่ต้องต่อเข้าไฟ + เพื่อให้โมดูลสามารถทำงานได้ทั้ง 2ขานี้ สามารถนำมาใช้รีเซ็ตโมดูลได้เหมือนกันแตกต่างตรงที่ขา Reset สามารถลอยไว้ได้ แต่ขา CH_PD (หรือ EN) จำเป็นต้องต่อเข้าไฟ + เท่านั้นเมื่อขานี้ไม่ต่อเข้าไฟ + โมดูลจะไม่ทำงานทันที

ESP8266 รุ่นต่าง ๆ ที่นิยมใช้

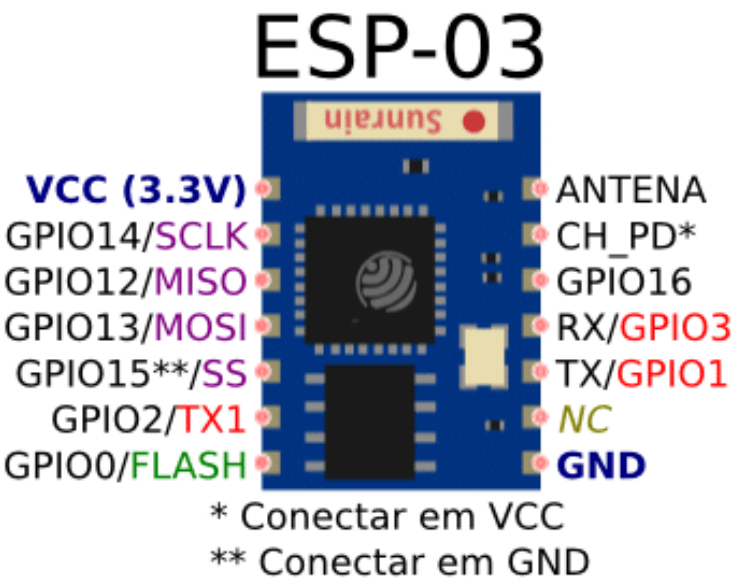
โมดูล ESP8266 รุ่นที่นิยมเช่น ESP-01 , ESP-03 , ESP-07 , ESP-12E นอกจากนี้ยังมีบอร์ด ESP8266 ที่รวมวงจร USB TTL เข้าไปทำให้โปรแกรมกับ Arduino ได้ง่ายขึ้นเช่น NodeMCU , Wemos D1 , Wemosmini

ESP8266 ESP-01



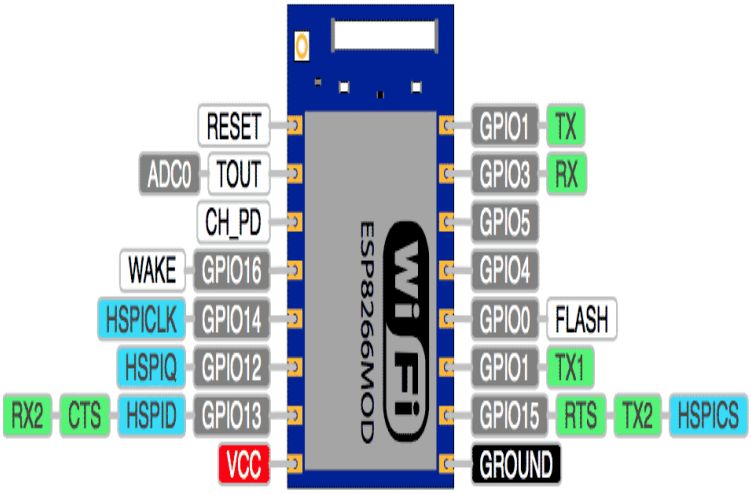
ESP8266 ESP-01 เป็นรุ่นที่มีขาต่อออกมาข้างนอก จึงสามารถต่อกับบอร์ดทดลองได้ง่าย มีขา GPIO 4 ขา คือ GPIO0 , GPIO1, GPIO2 , GPIO3

ESP8266 ESP-03



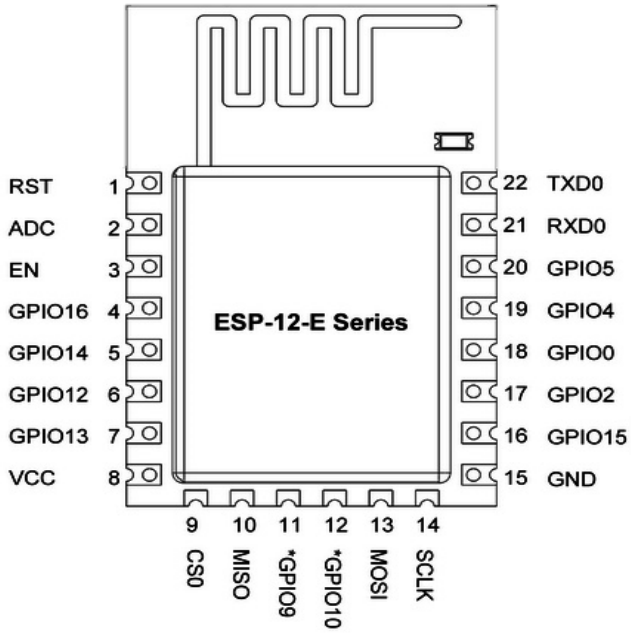
ESP8266 ESP-03 จะคล้ายกับ ESP-01 มี package การต่อขาเป็นแบบเซอร์เฟสเมาส์ โดยจะมีขา GPIO มากขึ้น ทำให้เราสั่งงานอุปกรณ์ได้มากกว่า ESP8266 ESP-01 โดย ESP8266-13 มีขาทั้งหมด 14 ขา มีเสาอากาศแบบมาให้ในตัว และยังสามารถต่อสายอากาศเพิ่มเพื่อเพิ่มกำลังการรับส่ง ได้ที่ขา 1

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

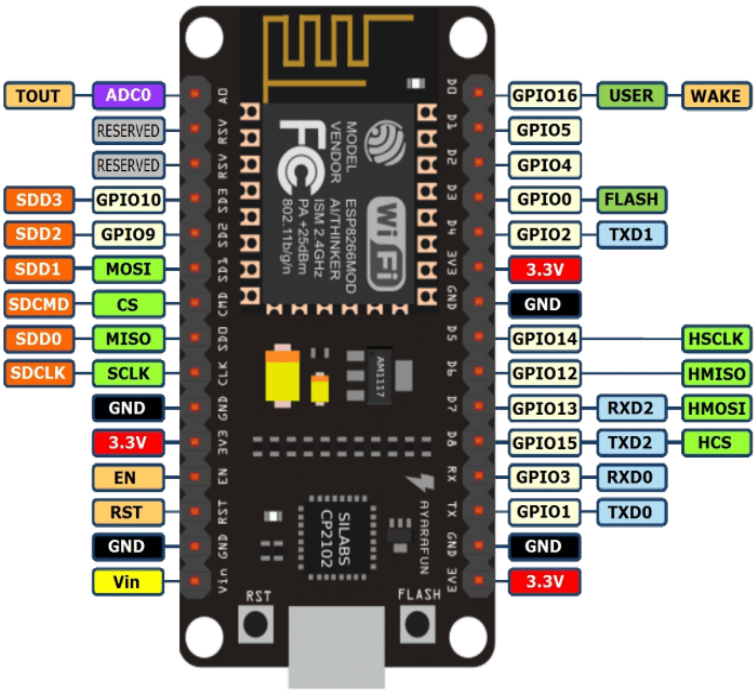


ESP8266 ESP-12E มีการจัดขาและต่อใช้งานแบบเดียวกับรุ่น ESP-07 โดยเปลี่ยนเสาอากาศมาเป็นแบบเดินวงจรภายใน PCB และเพิ่มขาอีก 6 ขา คือ SCLK MOSI MISO สำหรับติดต่อกับเซนเซอร์อื่น ๆ โปรโตคอล SPI ซึ่งรุ่นนี้ก็ได้รับความนิยมเป็นอย่างมาก จนเกิดการต่อยอดเป็นบอร์ด ESP8266-12 รุ่นต่าง ๆ

การเขียนโปรแกรม ESP8266 Arduino IDE

สำหรับการต่อวงจรของ ESP8266 เพื่อเขียนโปรแกรมด้วย Arduino เขียนโคดเหมือนกันทุกรุ่น โดยแต่ละรุ่นจะมีขาไม่เท่ากัน ดังนั้นเราจึงต้องเปรียบเทียบขา GPIO ให้ถูกต้องในการสั่งงานก็ใช้ได้แล้ว การเขียนโปรแกรมอัพโหลดโคดลงบอร์ด ESP8266 เกือบทุกรุ่น จะผ่านทาง Serial ที่ขา rx,tx โดยใช้โมดูล USB TTL ซึ่งต้องเสียเวลาในการต่อวงจรเพื่ออัพโหลดโคด อีกทั้งโมดูล ESP8266 หลาย ๆ รุ่นมีการต่อขาที่เป็นแบบเซอร์เฟสเมาส์ ทำให้ไม่สะดวกกับการต่อทดลองบนบอร์ดทดลอง

ดังนั้นจึงมีการรวม โมดูล USB TTL และต่อวงจรขยายขา ESP8266 ให้เป็นขาระยะห่างขนาด 2.54mm ซึ่งสามารถเสียบลงบอร์ดทดลองได้พอดี กลายเป็น บอร์ด ESP8266 โดยหนึ่งในบอร์ดที่นิยมใช้งานคือ NodeMCU ซึ่งใช้ โมดูล ESP8266 ESP-12E



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

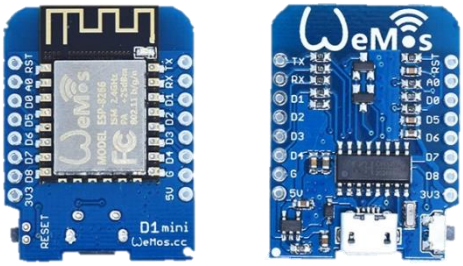
บอร์ด ESP8266 NodeMCU V3

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

บอร์ด WeMos D1

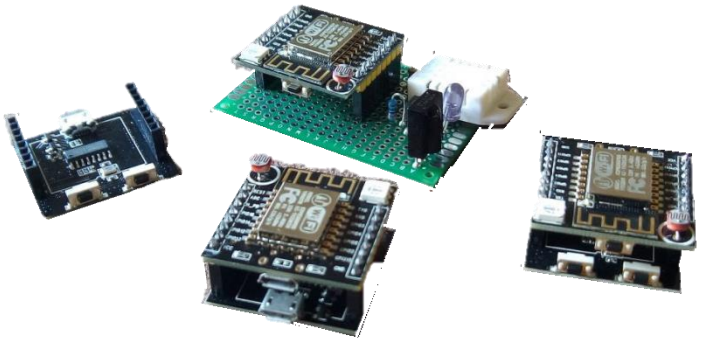
ออกแบบให้คล้ายบอร์ด Arduino Uno ใช้ ESP8266-12E เป็นตัวหลัก ชิฟ CH340 เป็นภาค USB TTL รองรับไฟ 9-12V ที่ช่อง Adapter , รองรับไฟ 5V ที่ช่อง micro usb และ 3.3V ที่ช่อง VIN

บอร์ด WeMos D1 Mini



แปลงบอร์ด Wemos D1 ให้มีขนาดเล็กลง โดยการแยกส่วน usb ttl และ ESP8266-12E ข้อดีคือเวลาใช้งานจริงสามารถถอดเฉพาะส่วน ESP8266-12E ไปใช้งาน ก็จะทำให้ประหยัดไฟได้นิดหน่อย

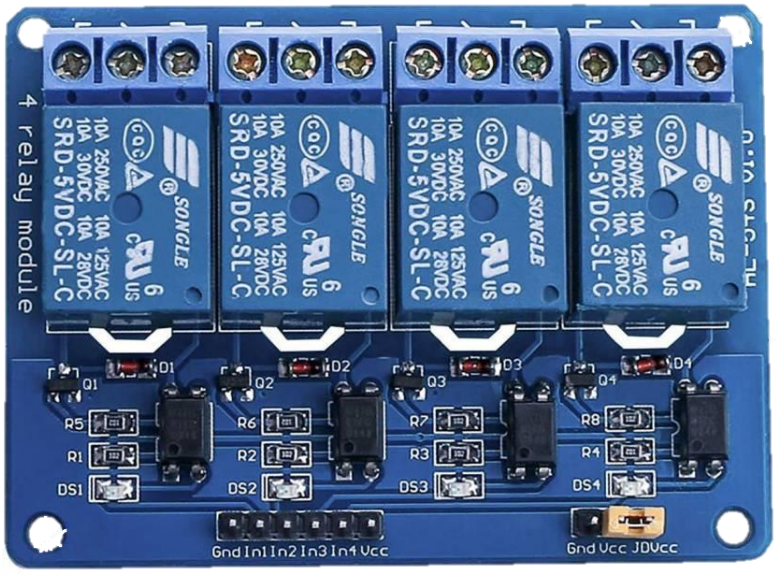
Witty cloud Mini NodeMCU



บอร์ดนี้มีขนาดเล็ก แยกส่วนได้ คล้าย ๆ กับ WeMos D1 Mini โดยใช้ ESP8266-12F โดยมี LDR , สวิตช์ และ LED RGB ต่อมาให้ในบอร์ด ทำให้สะดวกในการต่อทดลอง ข้อดีของรุ่นนี้คือฐานออกแบบมาให้มาขนาดเท่ากับบอร์ดขยายขาของ ESP-07 , ESP-08 , ESP-12 ทำให้สามารถนำไปใช้โปรแกรมกับ ESP รุ่นดังกล่าวได้ด้วย

รีเลย์ (Relay)

รีเลย์ (Relay) เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย



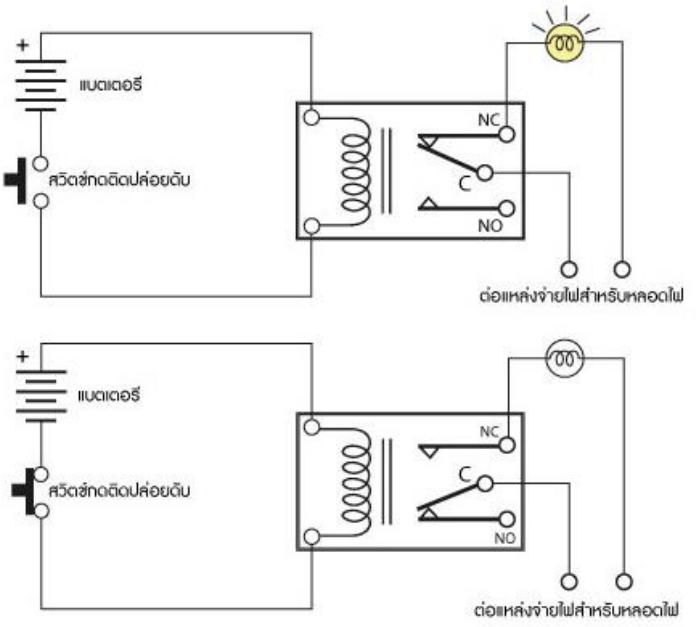
รีเลย์ ประกอบด้วยส่วนสำคัญ 2 ส่วนหลักก็คือ

1. ส่วนของขดลวด (coil) เหนี่ยวนำกระแสต่ำ ทำหน้าที่สร้างสนามแม่เหล็กไฟฟ้าให้แกนโลหะไปกระทุ้งให้หน้าสัมผัสต่อกัน ทำงานโดยการรับแรงดันจากภายนอกต่อคร่อมที่ขดลวดเหนี่ยวนำนี้ เมื่อขดลวดได้รับแรงดัน (ค่าแรงดันที่รีเลย์ต้องการขึ้นกับชนิดและรุ่นตามที่ผู้ผลิตกำหนด) จะเกิดสนามแม่เหล็กไฟฟ้าทำให้แกนโลหะด้านในไปกระทุ้งให้แผ่นหน้าสัมผัสต่อกัน

2. ส่วนของหน้าสัมผัส (contact) ทำหน้าที่เหมือนสวิตช์จ่ายกระแสไฟให้กับอุปกรณ์ที่เราต้องการ

จุดต่อใช้งานมาตรฐาน ประกอบด้วย

จุดต่อ NC ย่อมาจาก normal close หมายความว่าปกติดปิด หรือ หากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา เช่น จุดต่อ NO ย่อมาจาก normal open หมายความว่าปกติเปิด หรือหากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิดเช่นโคมไฟสนามหนือหน้าบ้าน จุดต่อ C ย่อมากจาก common คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ



ข้อคำถึงในการใช้งานรีเลย์ทั่วไป

1. แรงดันใช้งาน หรือแรงดันที่ทำให้รีเลย์ทำงานได้ หากเราดูที่ตัวรีเลย์จะระบุค่า แรงดันใช้งานไว้ (หากใช้ในงานอิเล็กทรอนิกส์ ส่วนมากจะใช้แรงดันกระแสตรงในการใช้งาน) เช่น 12VDC คือต้องใช้แรงดันที่ 12 VDC เท่านั้นหากใช้มากกว่านี้ ขดลวดภายใน ตัวรีเลย์อาจจะขาดได้ หรือหากใช้แรงดันต่ำกว่ามาก รีเลย์จะไม่ทำงาน ส่วนในการต่อวงจรนั้นสามารถต่อขั้วใดก็ได้เพราะตัวรีเลย์จะไม่ระบุขั้วต่อไว้(นอกจากชนิดพิเศษ)

2. การใช้งานกระแสผ่านหน้าสัมผัส ซึ่งที่ตัวรีเลย์จะระบุไว้ เช่น 10A 220AC คือ หน้าสัมผัสของรีเลย์นั้นสามารถทนกระแสได้ 10 แอมแปร์ที่ 220VAC แต่การใช้ก็ควรจะใช้งานที่ระดับกระแสต่ำกว่านี้จะเป็นการดีกว่าเพราะถ้ากระแสมากหน้าสัมผัสของรีเลย์จะละลายเสียหายได้

3. จำนานหน้าสัมผัสการใช้งาน ควรดูว่ารีเลย์นั้นมีหน้าสัมผัสให้ใช้งานกี่อัน และมีขั้วคอมมอนด้วยหรือไม่

ประเภทของรีเลย์

แบ่งออกตามลักษณะการใช้งานได้เป็น2ประเภทคือ

1. รีเลย์กำลัง (power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor)ใช้ในการควบคุมไฟฟ้ากำลังมีขนาดใหญ่กว่ารีเลย์ธรรมดา

2. รีเลย์ควบคุม (controlRelay) มีขนาดเล็กกำลังไฟฟ้าต่ำใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนักหรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุม บางทีเรียกกันง่าย ๆ ว่า "รีเลย์"

แบ่งตามลักษณะของคอยล์

1. รีเลย์กระแส (Current relay) คือ รีเลย์ที่ทำงานโดยใช้กระแสมีทั้งชนิดกระแสขาด (Under- current) และกระแสเกิน(Overcurrent)

2. รีเลย์แรงดัน (Voltage relay) คือ รีเลย์ ที่ทำงานโดยใช้แรงดันมีทั้งชนิดแรงดันขาด (Under-voltage) และแรงดันเกิน(Overvoltage)

3. รีเลย์ช่วย (Auxiliary relay) คือ รีเลย์ที่เวลาใช้งานจะต้องประกอบเข้ากับรีเลย์ชนิดอื่น จึงจะทำงานได้

4.รีเลย์กำลัง (Power relay) คือ รีเลย์ที่รวมเอาคุณสมบัติของรีเลย์กระแส และรีเลย์แรงดันเข้าด้วยกัน

5. รีเลย์เวลา (Time relay) คือ รีเลย์ที่ทำงานโดยมีเวลาเข้ามาเกี่ยวข้องด้วยซึ่งมีอยู่ด้วยกัน 4 แบบ คือ
   5.1 รีเลย์กระแสเกินชนิดเวลาผกผันกับกระแส (Inverse time over current relay) คือ รีเลย์ ที่มีเวลาทำงานเป็นส่วนกลับกับกระแส
   5.2 รีเลย์กระแสเกินชนิดทำงานทันที (Instantaneous over current relay) คือรีเลย์ที่ทำงานทันทีทันใดเมื่อกระแสไหลผ่านเกินกว่าที่กำหนดที่ตั้งไว้
   5.3 รีเลย์แบบดิฟฟินิตไทม์เล็ก (Definite time lag relay) คือ รีเลย์ ที่มีเวลาการทำงานไม่ขึ้นอยู่กับความมากน้อยของกระแสหรือค่าไฟฟ้าอื่นๆที่ทำให้เกิดงานขึ้น
   5.4 รีเลย์แบบอินเวอสดิฟฟินิตมินิมั่มไทม์เล็ก (Inverse definite time lag relay) คือ รีเลย์ ที่ทำงานโดยรวมเอาคุณสมบัติของเวลาผกผันกับกระแส (Inverse time) และ แบบดิฟฟินิตไทม์แล็ก (Definite time lag relay)เข้าด้วยกัน

6. รีเลย์กระแสต่าง (Differential relay) คือ รีเลย์ที่ทำงานโดยอาศัยผลต่างของกระแส

7. รีเลย์มีทิศ (Directional relay) คือรีเลย์ที่ทำงานเมื่อมีกระแสไหลผิดทิศทาง มีแบบรีเลย์กำลังมีทิศ (Directional power relay) และรีเลย์กระแสมีทิศ (Directional current relay)

8. รีเลย์ระยะทาง (Distance relay) คือ รีเลย์ระยะทางมีแบบต่างๆ ดังนี้
- รีแอกแตนซ์รีเลย์ (Reactance relay)
- อิมพีแดนซ์รีเลย์ (Impedance relay)
- โมห์รีเลย์ (Mho relay)
- โอห์มรีเลย์ (Ohm relay)
- โพลาไรซ์โมห์รีเลย์ (Polaized mho relay)
- ออฟเซทโมห์รีเลย์ (Off set mho relay)

9. รีเลย์อุณหภูมิ(Temperature relay) คือรีเลย์ที่ทำงานตามอุณหภูมิที่ตั้งไว

10. รีเลย์ความถี่ (Frequency relay) คือ รีเลย์ที่ทำงานเมื่อความถี่ของระบบต่ำกว่าหรือมากกว่าที่ตั้งไว้

11. บูคโฮลซ์รีเลย์ (Buchholz ‘s relay) คือรีเลย์ที่ทำงานด้วยก๊าซ ใช้กับหม้อแปลงที่แช่อยู่ในน้ำมันเมื่อเกิดฟอลต์ขึ้นภายในหม้อแปลงจะทำให้น้ำมันแตกตัวและเกิดก๊าซขึ้นภายในไปดันหน้าสัมผัส ให้รีเลย์ทำงาน

ความรู้ทั่วไปเกี่ยวกับรีเลย์

1.1 หน้าที่ของรีเลย์ คือเป็นอุปกรณ์ไฟฟ้าที่ใช้ตรวจสอบสภาพการณ์ของทุกส่วนในระบบกำลังไฟฟ้าอยู่ตลอดเวลาหากระบบมีการทำงานที่ผิดปกติ รีเลย์จะเป็นตัวสั่งการให้ตัดส่วนที่ลัดวงจรหรือส่วนที่ทำงานผิดปกติ ออกจากระบบทันทีโดยเซอร์กิตเบรกเกอร์จะเป็นตัวที่ตัดส่วนที่เกิดฟอลต์ออกจากระบบจริงๆ

1.2 ประโยชน์ของรีเลย์
- 1.2.1 ทำให้ระบบส่งกำลังมีเสถียรภาพ (Stability) สูงโดยรีเลย์จะตัดวงจรเฉพาะส่วนที่เกิดผิดปกติ ออกเท่านั้น ซึ่งจะเป็นการลดความเสียหายให้แก่ระบบน้อยที่สุด
- 1.2.2 ลดค่าใช้จ่ายในการซ่อมแซมส่วนที่เกิดผิดปกติ
- 1.2.3 ลดความเสียหายไม่เกิดลุกลามไปยังอุปกรณ์อื่นๆ
- 1.2.4 ทำให้ระบบไฟฟ้าไม่ดับทั้งระบบเมื่อเกิดฟอลต์ขึ้นในระบบ

1.3 คุณสมบัติที่ดีของรีเลย์
- 1.3.1 ต้องมีความไว(Sensitivity)คือมีความสามารถในการตรวจพบสิ่งที่ผิดปกติเพียงเล็กน้อยได้
- 1.3.2 มีความเร็วในการทำงาน (Speed) คือความสามารถทำงานได้รวดเร็วทันใจ ไม่ทำให้เกิดความเสียหายแก่อุปกรณ์และไม่กระทบกระเทือนต่อระบบ โดยทั่วไปแล้วเวลา ที่ใช้ในการตัดวงจรจะขึ้นอยู่กับระดับของแรงดันของระบบด้วย

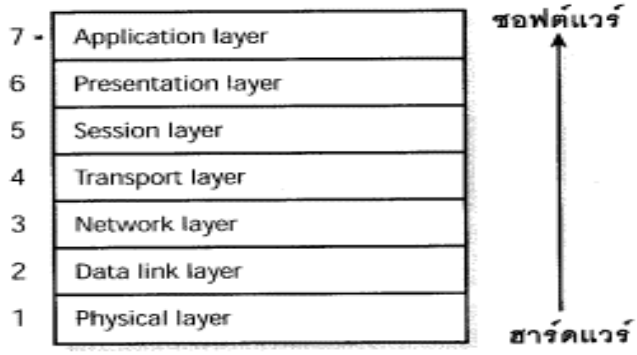
ระบบ6-10เควีจะต้องตัดวงจรภายในเวลา1.5-3.0วินาที
ระบบ100-220เควีจะต้องตัดวงจรภายในเวลา0.15-0.3วินาที
ระบบ 300-500 เควี จะต้องตัดวงจรภายในเวลา 0.1-0.12 วินาที

มาตรฐานการสื่อสารข้อมูล

การกำหนดมาตรฐานของการสื่อสารข้อมูลนั้น นับว่ามีความจำเป็นอย่างมากสำหรับระบบเครือข่ายที่มีองค์ประกอบของอุปกรณ์ต่างๆ หลากหลายผู้ผลิต ซึ่งอุปกรณ์ทั้งหมดเหล่านั้นจะต้องทำงานเข้ากันได้อย่างราบรื่น การกำหนดมาตรฐานต่างๆ นั้นจะเริ่มตั้งแต่โครงสร้างพื้นฐานของฮาร์ดแวร์ระบบเครือข่าย ได้แก่ ระบบสายเคเบิล อุปกรณ์ในการส่งสัญญาณข้อมูล ตลอดจนถึง เครื่องเซิร์ฟเวอร์ และซอฟต์แวร์ในการสื่อสารบนระบบเครือข่าย เพื่อเป็นการรับประกันว่าส่วนประกอบต่างๆ จะสามารถทำงานร่วมกันได้ ผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์ระบบเครือข่าย จะต้องทำตามคำแนะนำตามมาตรฐานการออกแบบและสร้างผลิตภัณฑ์ ซึ่งกำหนดขึ้นโดย องค์กรมาตรฐานสากล (International Organization for Standardization - ISO)โดยมาตรฐานที่กำหนดขึ้นและได้ประกาศใช้ตั้งแต่ปี คศ.1984 เรียกว่า Open Systems Interconnection Reference Model เรียกสั้นๆ ว่า OSI Reference Model หรือ ISO/OSI Model

แบบจำลอง OSI

OSI Reference Model เป็นการกำหนดชุดของคุณลักษณะเฉพาะที่ใช้อธิบายโครงสร้างของระบบเครือข่าย โดยมีวัตถุประสงค์ เพื่อให้ผู้ผลิตฮาร์ดแวร์หรือซอฟต์แวร์ใดๆ ใช้เป็นโครงสร้างอ้างอิงในการสร้างอุปกรณ์ให้สามารถทำงานร่วมกันได้อย่างดีบนระบบเครือข่าย โดยมีการจัดแบ่งเลเยอร์ของ OSI ออกเป็น 7 เลเยอร์ แต่ละเลเยอร์จะมีการโต้ตอบหรือรับส่งข้อมูลกับเลเยอร์ที่อยู่ข้างเคียงเท่านั้น โดยเลเยอร์ที่อยู่ชั้นล่างจะกำหนดลักษณะของอินเตอร์เฟซ เพื่อให้บริการกับเลเยอร์ที่อยู่เหนือขึ้นไปตามลำดับขั้น เริ่มตั้งแต่ส่วนล่างสุดซึ่งเป็นการจัดการลักษณะทางกายภาพของฮาร์ดแวร์และการส่งกระแสของข้อมูลในระดับบิต ไปสิ้นสุดที่แอพพลิเคชั่นเลเยอร์ในส่วนบนสุด



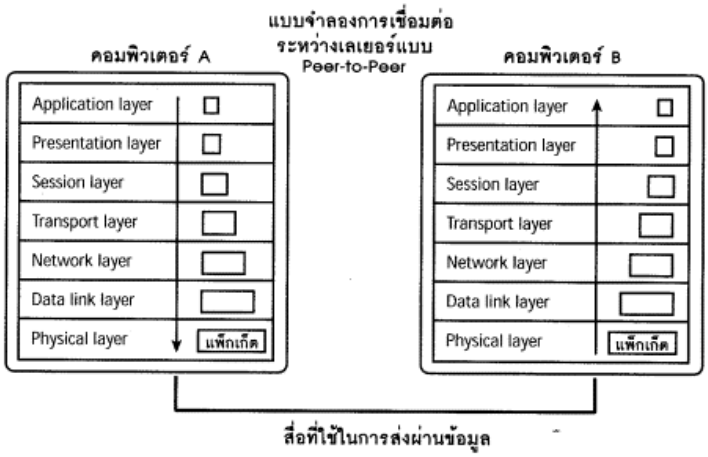
รูปที่ 1 OSI Reference Model

หลักการออกแบบเลเยอร์

-แต่ละเลเยอร์จะมีการกำหนดการทำงานอย่างละเอียดโดยมีการทำงานเป็นอิสระไม่ขึ้นต่อกัน
-ฟังก์ชันภายในเลเยอร์จะพยายามมุ่งไปสู่ข้อกำหนดมาตรฐาน (standard protocol)
-ขอบเขตของเลเยอร์จะถูกเลือกและจำกัดให้มีปริมาณการเชื่อมต่อระหว่างเลเยอร์ให้น้อยที่สุด
-จำนวนของเลเยอร์ต้องมากพอที่จะแยกฟังก์ชั่นที่จำเป็นและแตกต่างกันไม่ให้อยู่ในเลเยอร์เดียวกัน

การทำงานของ OSI Reference Model

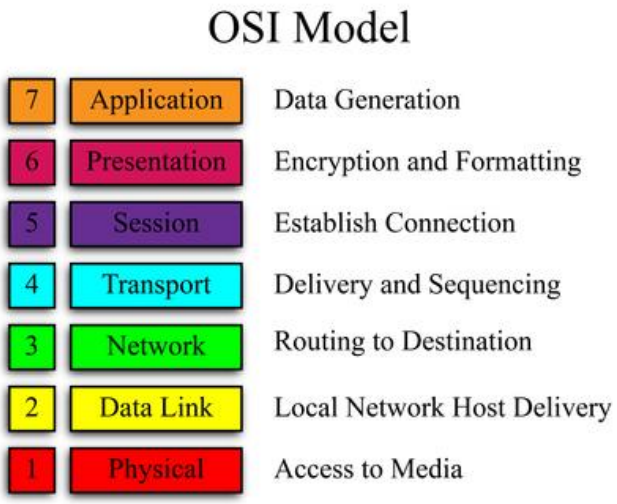
การที่แพ็กเก็ตข้อมูลเดินทางจากเครื่องคอมพิวเตอร์ A ไปยังเครื่องคอมพิวเตอร์ B นั้น มีกระบวนการทำงานดังนี้



รูปที่ 2 การส่งแพ็กเก็ตใน OSI Reference

จากแผนผัง คอมพิวเตอร์ A และคอมพิวเตอร์ B มีโครงสร้างเป็น OSI ซึ่งมี 7 เลเยอร์ เมื่อเครื่องคอมพิวเตอร์ A พร้อมที่จะส่งสัญญาณข้อมูลไปยังเครื่องคอมพิวเตอร์ B นั้น แต่ละเลเยอร์ในเครื่องคอมพิวเตอร์ A จะเสมือนกับว่ามีการสื่อสารกับเลเยอร์ในระดับเดียวกันบนเครื่องคอมพิวเตอร์ B ถึงแม้ว่าจะไม่มีการสื่อสารระหว่าง เลเยอร์เหล่านี้เกิดขึ้นจริง แต่เลเยอร์ในระดับต่างๆ บนเครื่องคอมพิวเตอร์ทั้งคู่นั้นจะทำตามกฎเกณฑ์หรือโปรโตคอล (protocol) อย่างเดียวกัน เพื่อให้มั่นใจได้ว่าแต่ละเลเยอร์บนเครื่องคอมพิวเตอร์ฝ่ายผู้รับจะได้รับแพ็กเก็ตข้อมูล แบบเดียวกันกับแพ็กเก็ตข้อมูลที่รวบรวม โดยแต่ละเลเยอร์บนเครื่องคอมพิวเตอร์ฝ่ายผู้ส่ง โดยแพ็กเก็ตข้อมูลจะเริ่มที่ระดับสูงสุดคือ Application Layer บนเครื่องคอมพิวเตอร์ A และเคลื่อนลงมาทีละระดับชั้นจนมาถึงชั้นล่างสุดคือ Physical Layer การที่แพ็กเก็ตเคลื่อนผ่านจากระดับหนึ่งไปยังระดับถัดไปนั้น จะมีการกำหนดที่อยู่ การจัดรูปแบบของข้อมูลและอื่นๆ ซึ่งแต่ละเลเยอร์จะเป็นตัวจัดการและมีกระบวนการของตนเอง เมื่อแพ็กเก็ตเคลื่อนตัวลงมาถึง Physical Layer ก็จะถูกแปลงให้เป็นกระแสข้อมูลแบบอนุกรมและส่งผ่านสื่อกลางคือสายสัญญาณ ซึ่งเป็นเลเยอร์เดียวที่เครื่องคอมพิวเตอร์ A สื่อสารกับเครื่องคอมพิวเตอร์ B และเมื่อ

สัญญาณข้อมูลมาถึงเครื่องคอมพิวเตอร์ B กระบวนการก็จะเริ่มทำในทางตรงข้าม คือจะทำการแยกแพ็กเก็ตออกผ่าน OSI ทั้ง 7 เลเยอร์ ส่งย้อนกลับขึ้นไปยัง Application Layer ของเครื่องคอมพิวเตอร์ B เมื่อแพ็กเก็ตเดินทางผ่านเลเยอร์ระดับต่างๆ แต่ละเลเยอร์จะแยก ข้อมูลข่าวสารตามกำหนดที่อยู่ และการจัดรูปแบบของแพ็กเก็ต จนเมื่อมาถึงเลเยอร์ระดับสูงสุดคือ Application Layer ก็จะเหลือเฉพาะข้อมูลที่เหมือนกับบน Application Layer ของเครื่องคอมพิวเตอร์ A



1. Physical Layer

Physical Layer หรือชั้นกายภาพ ทำหน้าที่ จัดการเชื่อมต่อและการส่งสัญญาณทางไฟฟ้าจากผู้ส่งไปยังผู้รับ โดยผ่านสื่อกลาง เช่น สายทองแดง คลื่นวิทยุ สายคู่ตีเกลียว และใยแก้วนำแสงเป็นต้น โดยสัญญาณที่ผ่านอาจเป็นสัญญาณไฟฟ้า สัญญาณคลื่นวิทยุ หรือสัญญาณแสง โดยมาตรฐานที่ใช้กันมากที่สุดใน Physical Layer คือ RS-232C มาตรฐานของสัญญาณ และสายที่กำหนด ว่าสัญญาณไหนทำอะไร และระดับแรงดันไฟฟ้าเท่าใดแทน 0 หรือ 1

หลักการทำงาน เป็นการรับส่งข้อมูลทีละบิตเรียงต่อกันไป

2. Data Link Layer

ทำหน้าที่ ควบคุมสายข้อมูล ระหว่างระบบกับปลายทางอีกด้านหนึ่ง โดยการรวมตัวอักขระเข้าด้วยกันเป็นข่าวสาร แล้วตรวจสอบ ก่อนที่จะส่งลงไปในสายสัญญาณ

หลักการทำงาน การส่งข้อมูลใน Layer นี้ จะส่งข้อมูลกระจาย (Broadcast) ไปบนเครือข่าย แล้วเครื่องที่ได้รับ ก็ดูว่าข้อมูลที่ได้นั้นระบุ Destination Address ตรงกับ Hardware Address ของตัวมันเองหรือเปล่า ถ้าตรงก็รับข้อมูลไว้ ถ้าไม่ตรงก็ทิ้งข้อมูลไป

3. Network Layer

ทำหน้าที่ เลือกเส้นทางการส่งผ่านข้อมูล โดย เป็นผู้ตัดสินใจว่าเส้นทางใด ที่ควรส่งข้อมูลไป ทั้งนี้ขึ้นอยู่กับ สภาพของเครือข่าย ลำดับความสำคัญของบริการ และปัจจัยอื่น ๆ ซอฟต์แวร์ใน Network Layer อยู่ในส่วนของชุมสาย ในเครือข่าย

หลักการทำงาน การส่งข้อมูลใน Layer นี้ จะใช้หลักการที่เรียกว่า Hop by Hop โดยจะส่งข้อมูลจากจุด ๆ นึงไปยังจุดต่อไป (Next Hop) โดยอาศัยจากตาราง Routing ของ Router ซึ่งอาจจะต้อง ส่งผ่านหลายจุด กว่าจะไปถึงจุดหมายปลายทางได้

4.Transport Layer

ทำหน้าที่ หลายอย่างเหมือน Network Layer ควบคุมคุณภาพ ของข้อมูลที่ได้รับ ให้ถูกต้องทั้งรูปแบบและลำดับ ถ้ามีความเสียหายเกิดขึ้น ในระบบเครือข่าย และ ถ้าเครือข่ายล้มเหลว ซอฟต์แวร์ใน Transport Layer จะมองหาเส้นทางอื่น ที่จะสามารถไปยังปลายทาง หรืออาจจัดเก็บข้อมูลที่ส่งไว้จนกระทั่งการเชื่อมต่อของเครือข่าย ถูกสร้างขึ้นใหม่

หลักการทำงาน การทำงานที่ชั้นของ Host - to - Host Layer นี้จะมีบทบาทในการจัดการต่อจากชั้นของ Process Layer บางครั้งเรามักเรียกชั้น Host - To - Host Layer ว่าเป็น Transport Layer ซึ่งไม่ใช่ชั้นของ Transport Layer ในมาตรฐาน OSI - Reference Model การทำงานของ Host - To - Host Layer นี้จะมีการสร้าง Connection หรือการเชื่อมต่อกันระหว่างแอพพลิเคชันกับ Host - To - Host Layer โดยจุดที่เชื่อมกันเพื่อรับส่งข้อมูลนี้เรียกว่า Port หรือ Socket และในแต่ละแอพพลิเคชัน ก็จะสร้างการเชื่อมต่อ ผ่าน Port ได้พร้อมกันหลายแอพพลิเคชัน ซึ่งการใช้งาน Port ของแต่ละแอพพลิเคชันที่อยู่ในชั้น Process Layer จะแตกต่างกันตามหมายเลขที่กำหนดไว้ และแต่ละโปรโตคอลจะมีการใช้งาน Port หมายเลขต่าง ๆ ไม่ซ้ำกัน

5.Session Layer

เป็นระดับที่ผู้ใช้ทำการติดต่อกับข่ายสื่อสาร โดยผู้ใช้ต้องการจะติดต่อกับเครื่องอื่น ๆ แล้วการติดต่อจะเกิดขึ้น เมื่อผู้ใช้ใช้คำสั่ง หรือข้อความ ที่กำหนดไว้ ป้อนให้แก่ระบบ จากนั้น ผู้ใช้ก็จะกลายเป็นผู้ใช้ระบบทางไกล ซึ่งจะสามารถ ส่งผ่านข้อความ หรือแลกเปลี่ยนแฟ้มข้อมูล กับระบบได้ ในการสร้างการโต้ตอบระหว่างกันได้นี้ ผู้ใช้จะต้องกำหนด รหัสตำแหน่งจุดปลายทาง ที่ต้องการจะติดต่อด้วย
Session Layer มีความสำคัญมากในระบบที่ทำงานบนพีซี มีความสามารถ ทำให้ Application สองตัว (หรือ Application เดียวกัน แต่มีสองส่วน) ในระบบที่ทำงานบนพีซี สามารถสื่อสารข้ามเครือข่ายได้อย่างถูกต้อง เช่นการ dial-up การทำ synchronization หรือการ login นอกจากนี้ ยังทำหน้าที่บริหารและจัดการ เช่น การบันทึกข้อมูลการใช้งาน รวมถึงการ รักษาความปลอดภัย และการแยกแยะชื่อผู้ใช้ เป็นต้น

6. Presentation Layer

เป็นชั้นที่จะแสดงผลออกมาในรูปของ ภาพต่าง ๆที่เรามองเห็น เช่น รูปภาพ ที่ปรากฏบนจอคอมพิวเตอร์ และอาจจะรวมไปถึง การส่งผ่านข้อมูลต่าง ๆในรูปแบบของตัวโปรแกรม ที่มีการเข้ารหัส ว่ามีผลเป็นอย่างไร protocol ที่ใช้งานในชั้นนี้คือ JPEG, ASCII, Binary,EBCDICTIFF, GIF,MPEG, Encryptions เป็นต้นต่อจาก SessionLayer ยกตัวอย่าง msnmessenger ช่วงที่ connecting ถ้าnetwork ปกติ user และ password ถูกต้องจะสามารถเข้าสู่msnmessenger ได้จะมีหน้าตาของ Application ขึ้นมา ซึ่งก็คือ file ภาพต่าง ๆนั่นเอง อาจจะเป็น JPEG , BMP เป็นต้น มีหน้าที่ในการจัดการเข้ารหัส และการจัดรูปแบบการป้อนข้อมูลพิเศษ อักษรกระพริบ ตัวอักษรกลับสี มีการจัดการ เข้ารหัสการจัดรูปแบบไฟล์พิเศษ ควบคุม Syntax ของข้อมูลที่ส่งเข้า ส่งออก เช่นการเปลี่ยนจากรหัส EBCDIC เป็น ASCII เป็นต้น โดยจัดรูปแบบของจอภาพ และไฟล์ เพื่อให้ผลขั้นสุดท้าย มีลักษณะตรงตามความตั้งใจ ของนักเขียนโปรแกรม

7.Application Layer

ใน Layer นี้ จะครอบคลุมถึงระดับผู้ใช้ ที่จะประยุกต์ใช้งานของแต่ละคน เช่น เมื่อมีผู้ใช้สองคนจัดทำโปรแกรมบนเครื่องที่แตกต่างกัน แต่ทั้งสองฝ่าย ก็สามารถที่จะ กำหนดส่วนของข้อความ ที่จะมีผลต่อการกระทำระหว่างกันและกัน ได้โดยไม่ต้องคำนึงถึงว่า ผู้ใช้อีกฝ่ายหนึ่ง ที่จะติดต่อด้วย ใช้เครื่องชนิดอะไร นั่นก็คือความแตกต่าง ระหว่างชนิดของเครื่อง หรือความสัมพันธ์อื่น ๆ จะต้องถูกเปลี่ยนโดย Protocol ในระดับที่ต่ำกว่า ให้เป็นที่เรียบร้อย ก่อนจะถึงระดับผู้ใช้ทั้งสองฝ่าย ในกรณีเช่นนี้ จะเกิดประโยชน์มาก ในระบบฐานข้อมูล เพราะฐานข้อมูลแห่งหนึ่งๆ จะต้องถูกเชื่อมโยง และเรียกใช้ข้อมูลภายในฐานได้ โดยเครื่องคอมพิวเตอร์ที่ต่างชนิดกันออกไป เช่น ในระบบธนาคาร หรือการสำรองที่นั่งของสายการบินเป็นต้น โดยทำหน้าที่ให้บริการแก่ผู้ใช้ทุกสิ่ง ตั้งแต่การแบ่งปันไฟล์ Print Spooling ไปรษณีย์อิเล็กทรอนิกส์ จนถึงการจัดการฐานข้อมูลและบัญชี โดยรวบรวม Protocol ต่าง ๆ มาใช้ร่วมกันเพื่องานๆ นั้น ซึ่ง Layer นี้เป็นชั้นที่สำคัญที่สุด เนื่องจากผู้ใช้ควบคุมได้โดยตรง แต่หน้าที่บางอย่าง เช่น Protocol ถ่ายโอนไฟล์ FTP) ทำงานจาก Application Layer แต่ก็มอบหมายงานให้กับ Layer ที่ต่ำกว่าด้วย

หลอดฟลูออเรสเซนต์

หลอดฟลูออเรสเซนต์ (หรือที่เรียกว่าหลอดนีออน) เป็นอุปกรณ์ไฟฟ้าที่ให้แสงสว่าง มีลักษณะเป็นหลอดแก้วรูปทรงกระบอก เป็นหลอดตรง หรือดัดโค้งเป็นรูปอื่นๆ เปล่งแสงออกมาจากสารฟลูออเรสเซนต์ที่เคลือบอยู่บนผิวภายในของหลอด เนื่องจากถูกกระตุ้น โดยรังสีอุลตราไวโอเลตไปกระทบกับสารเรืองแสงที่เคลือบไว้ที่ผิวด้านในของหลอด ทำให้เปล่งแสงสว่างออกมาได้

การต่อวงจร

หลังจากการนำหลอดฟลูออเรสเซนต์ สตาร์ตเตอร์และแบลลัสต์ ต่อเป็นวงจรแล้วเมื่อกด สวิตซ์ให้วงจรเปิด กระแสไฟฟ้าไหลผ่านไส้หลอดด้านซ้ายมือ ผ่านสตาร์ตเตอร์และไส้หลอดด้าน ขวามือ เข้าสู่แบลลัสต์ ผ่านสวิตซ์ออกไป ยังแผนควบคุมครบวงจร ขณะที่กระแสไฟฟ้าไหลผ่านไส้หลอดทั้งสอง ไส้หลอดจะสว่างพอมองเห็น ณ จุดนั้นก็เกิดความร้อนทำให้ไอปรอทที่บรรจุอยู่ในหลอดฟุ้งกระจายพร้อมที่จะนำอิเล็กตรอน อิสระเคลื่อนที่ภายในหลอดจากซ้ายมือไปยังขวามือ แต่ยังไม่สามารถทำงานได้เพราะแรงดันไฟฟ้าที่ขั้วทั้งสองยังมีโวลต์ต่ำอยู่ ขณะเดียวกันสตาร์ตเตอร์ก็จะทำตัวตัดวงจรหลักที่เกิดจากประกายไฟขึ้นแล้วจึงในขณะนี้เอง แบลลัสต์จะเหนี่ยวนำให้เกิดความต่างศักย์สูงมากขึ้นระหว่างไส้หลอดทั้งสอง จึงมีผลทำให้ไอปรอท แตกตัวเป็นอิออน (ประจุไฟฟ้าบวก) นำอิเล็กตรอนอิสระเคลื่อนที่จากปลายซ้ายมือไปยังปลายหลอดขวามือโดยไม่จำเป็นต้องผ่านสตาร์ตเตอร์อีก เมื่ออิเลคตรอนเคลื่อนที่ผ่านไปนั้นจะชนกับไอปรอทและก๊าซอาร์กอน ทำให้แตกตัวแปล่งรังสีอุลตร้าไวโอเลตขึ้น รังสีนี้ตาของเราไม่สามารถมองเห็น และรังสีนี้จะไปกระทบกับสารเรืองแสงที่อยู่ที่ผิวภายในหลอด สารนี้จะดูดกลืนรังสีอุลตร้าไวโอเลตทำให้เกิดแสงสว่างปรากฏแก่สายตาของเรา เมื่อหลอดฟลูออเรสเซนต์สว่างขึ้น สตาร์ตเตอร์จะหมดหน้าที่ สำหรับแบลลัสต์จะทำหน้าที่ ควบคุมกระแสไฟฟ้าในหลอด มีความงามคงตัว โดยจะเหนี่ยวนำกระแสไฟฟ้ามาต้านหรือเสริม เมื่อกระแสไฟฟ้าที่ขั้วหลอดทั้งสองเกิดการเปลี่ยนแปลงการเกิดสีของหลอดฟลูออเรสเซนต์

สารเคมีที่ใช้ฉาบภายในหลอดฟลูออเรสเซนต์มีอยู่หลายอย่าง ทำให้สีที่เกิดขึ้นมีหลายสีดังนี้

สารที่ฉาบ	แสงที่ได้
แคลเซียมทังสเตน	สีน้ำเงิน
แมกนีเซียมทังสเตน	สีน้ำเงินอ่อน
แคลเซียมซิลิเกต	สีส้ม
ซิงค์ซิลิเกต	สีเขียว
แคลเซียมฮโลฟอสเฟต	สีเขียว
ซิงค์เอบร์เลียมซิลิเกต	สีเหลืองนวล
แคดเมียมซิลิเกต	สีชมพูแก่
แคดเมียมโบเรต	สีชมพูแก่

ปัจจุบันมีการพัฒนาหลอดฟลูออเรสเซนต์เพื่อขึ้นอีก 2 แบบ คือ

1. หลอดฟลูออเรสเซนต์ประสิทธิภาพสูง หลอดนี้ประหยัดพลังงานไฟฟ้ามากกว่าหลอดธรรมดา โดยให้ความสว่างเท่าเดิมหลอดนี้มีเส้นผ่านศูนย์กลางสั้นกว่าหลอดฟลูออเรสเซนต์ธรรมดาเล็กน้อยแต่ความยังคงเดิม บางทีเรียกว่าหลอดผอม

2. หลอดฟลูออเรสเซนต์แบบคอมแพค หลอดแบบนี้เป็นหลอดที่มีเกลียวคล้ายกับหลอดไฟฟ้าแบบธรรมดามีไส้จึงสามารถใช้เปลี่ยนแทนกันได้ทันที มีหลายขนาดกำลังไฟฟ้าให้เลือกตามความเหมาะสมแต่มีราคาค่อนข้างแพง แต่เมื่อพิจารณาแล้วในระยะยาวจุคุ้มค่ากว่าเพราะประหยัดพลังงานไฟฟ้าได้มากและมีอายุการใช้งานมากเมื่อเปรียบเทียบหลอดไฟฟ้าธรรมดาและหลอดเรืองแสงที่ใช้เพลังงานไฟฟ้าเท่ากับพบว่า หลอดเรืองแสงจะให้พลังงานมากกว่า และมีประสิทธิภาพสูงกว่าหลอดไฟธรรมดาประมาณ 4 เท่า เพราะหลอดเรืองแสงพลังงานไฟฟ้าเปลี่ยนไปเป็นพลังงานความร้อนน้อยมาก และยังเปล่งแสงสว่างออกมาได้ทั่วทุกจุดของหลอด ส่วนหลอดไฟฟ้าธรรมดานั้นพลังงานไฟฟ้าจะเปลี่ยนแปลงพลังงานความร้อนก่อนแล้งจึงเปลี่ยนเป็นพลังงานแสง ทำให้มีการสูญเสียพลังงานความร้อนไปบ้าง ทั้งนี้เราสามารถเปรียบเทียบผลการใช้ของหลอดไฟฟ้าทั้งสองได้จากตารางดังต่อไปนี้

หลอดเรืองแสง	หลอดไฟฟ้าแบบธรรมดาชนิดไส้
1.ให้ความสวางมากกว่า (เมื่อเปรียบเทียบการใช้พลังงานไฟฟ้าเท่ากัน) 2.มีอายุการใช้งานนานกว่า 3.เสียเงินค่าพลังงานไฟฟ้าน้อยกว่า (มื่อใช้แสงสว่างเท่ากัน) 4.ราคาอุปกรณ์แพงกว่า 5.การให้แสงสว่างเพื่อการอ่านหนังสือไม่ดีเท่ากับหลอดไฟแบบชนิดมีไส้ 6.แสงที่ปล่อยออกมาให้ความร้อนน้อย 7.ถ้าแรงดันไฟฟ้าลดลงหรือไฟตกประมาณ 30 % หลอดไฟจะดับ	1.ให้ความสว่างน้อย 2.มีอายุการใช้งานสั้น 3.เสียเงินค่าพลังงานไฟฟ้ามากกว่า 4.ราคาอุปกรณ์ถูกกว่า 5.การให้แสงสว่างในการอ่านหนังสือดีกว่าหลอดเรืองแสง 6.แสงที่เปล่งออกมาให้ความร้อนมาก 7.เมื่อแรงดันไฟฟ้าลดลง หรือเกิดไฟตกหลอดจะไม่ดับ

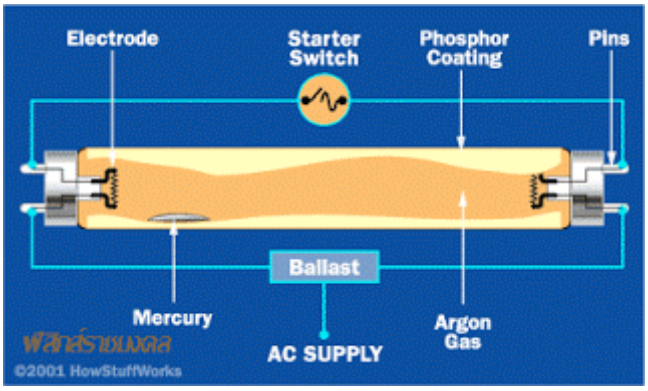
สตาร์ทเตอร์(starter)

สตาร์ทเตอร์ ทำงานโดยการปิด-เปิด วงจร อย่างรวดเร็ว ทำให้เกิดแรงดันสูง พอหลอดติดแล้วหน้าที่ของสตาร์ทเตอร์ก็หมดไป และมีบัลลาสต์ทำหน้าที่ต่อแทน โดยมีการใช้ก๊าซเฉื่อยใส่เข้าไปในหลอดด้วยเพื่อช่วยในการกระตุ้นอิเล็กตรอนไอปรอท

อาจจะกล่าวได้ง่าย ๆ ว่า หลอดฟลูออเรสเซนต์หรือหลอดนีออนนั้น ใช้สตาร์ทเตอร์ช่วยจุดหลอดไฟนั่นเอง เมื่อกดสวิทซ์ไฟ กระแสไฟฟ้าจะไหลผ่านสวิทซ์สตาร์ทเตอร์ครบวงจร ทำให้ไส้หลอดตรงขั้วหลอดร้อนขึ้น และปลดปล่อยอิเล็กตรอนออกมาในหลอด

สวิทช์สตาร์ทเตอร์นั้นทำจากหลอดก๊าซขนาดเล็ก บรรจุด้วยก๊าซซีนอน เมื่อเรากดสวิทช์ กระแสไฟฟ้าก็จะกระโดดข้ามช่องว่างในหลอด ขั้วไฟฟ้าข้างหนึ่งของสตาร์ทเตอร์ทำด้วยโลหะติดกัน 2 ชนิดเรียกว่าไบเมทาลิค (Bimetallic) มันจะบิดตัว เมื่อกระแสไหลผ่านและเกิดความร้อน หลังจากที่หลอดฟลูออเรสเซนต์ติดแล้ว กระแสไฟฟ้าจะไม่ไหลผ่านสตาร์ทเตอร์อีก ทำให้โลหะไบเมทาลิคเย็นลง และแยกออกจากกัน

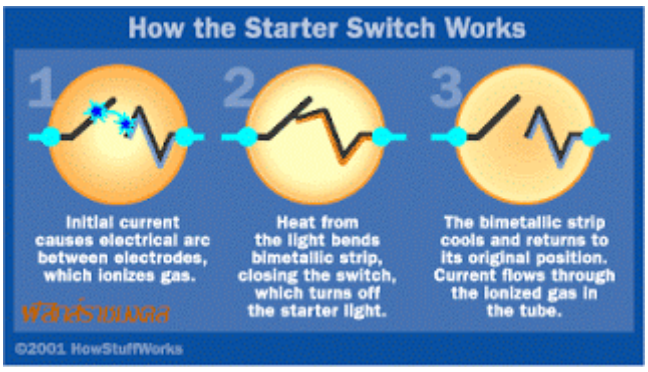
หลอดฟลูออเรสเซนต์ ใช้สตาร์ทเตอร์ ช่วยจุดหลอดไฟ



(รูปวงจรไฟฟ้าของหลอด)

เมื่อกดสวิทช์ไฟ กระแสไฟฟ้าจะไหลผ่านสวิทช์สตาร์ทเตอร์ ครบวงจร ทำให้ไส้หลอดตรงขั้วหลอดร้อนขึ้น และปลดปล่อยอิเล็กตรอนออกมาในหลอด

สวิทช์สตาร์ทเตอร์ทำจากหลอดก๊าซขนาดเล็ก บรรจุด้วยก๊าซซีนอน เมื่อเรากดสวิทช์ กระแสไฟฟ้ากระโดดข้ามช่องว่างในหลอดดังรูป



ขั้วไฟฟ้าข้างหนึ่งของสตาร์ทเตอร์ทำด้วยโลหะติดกัน 2 ชนิดเรียกว่า ไบเมทาลิค (Bimetallic) มันจะบิดตัว เมื่อกระแสไหลผ่านและเกิดความร้อน หลังจากที่หลอดฟลูออเรสเซนต์ติดแล้ว กระแสไฟฟ้าจะไม่ไหลผ่านสตาร์ทเตอร์อีก ทำให้โลหะไบเมทาลิคเย็นลง และแยกออกจากกัน

บัลลาสต์

บัลลาสต์ ทำหน้าที่สร้างแรงดันค่าสูงขึ้นมาในขณะที่หลอดฟลูออเรสเซนต์เริ่มทำงาน เมื่อหลอดฟลูออเรสเซนต์ทำงานแล้ว จำทำหน้าที่ลดแรงดันที่ตกคร่อมให้ต่ำลง และช่วยจำกัดกระแสไม่ให้ไหลผ่านหลอดมากเกินไปในขณะที่หลอดเปล่งแสงสว่างออกมา บัลลาสต์ที่ผลิตมาใช้งานแบ่งออกได้เป็น 2 ชนิด คือ ชนิดบัลลาสต์แม่เหล็กไฟฟ้า (Electromagnetic Ballast) และ ชนิดบัลลาสต์อิเล็กทรอนิกส์ (Electronic Ballast)

บัลลาสต์อิเล็กทรอนิกส์ที่จำหน่ายในเมืองไทยจะใช้แรงดันไฟฟ้า 220V ความถี่ 50Hz มีข้อมูลที่สำคัญอื่นๆที่เกี่ยวข้องกับบัลลาสต์อิเล็กทรอนิกส์ที่ควรทราบดังนี้

1. ปริมาณของกระแสขณะทำงาน (Irms และ Ipeak) ปริมาณของกระแสยิ่งน้อยเท่าใดความสูญเสียเนื่องจากความร้อนที่เกิดบนสายไฟก็ยิ่งน้อยลง

2. ค่าประกอบกำลัง (Power Factor) เป็นค่าที่บอกว่าบัลลาสต์อิเล็กทรอนิกส์ที่ใช้อยู่สามารถสร้างความสัมพันธ์ระหว่างกระแสและแรงดันให้สอดคล้องกันได้มากน้อยเพียงไร บัลลาสต์อิเล็กทรอนิกส์ที่มีคุณภาพสูงจะมีค่า Power Factorใกล้เคียง 1.0 ในขณะที่บัลลาสต์อิเล็กทรอนิกส์คุณภาพต่ำจะมีค่า Power Factor ต่ำ เมื่อติดตั้งบัลลาสต์อิเล็กทรอนิกส์ที่มีค่าPower Factor ต่ำจำนวนมาก จะทำให้สูญเสียพลังงานไฟฟ้าในระบบไฟฟ้าและในตัวบัลลาสต์อิเล็กทรอนิกส์เอง ซึ่งพลังงานที่สูญเสียนี้จะเป็นสัดส่วนผกผันกับค่า Power Factor

3. ค่า THD (Total Harmonics Distortion) เป็นค่าที่บอกว่าบัลลาสต์อิเล็กทรอนิกส์ที่ใช้อยู่มีสัญญาณรบกวนความถี่สูงมากน้อยเพียงไร สัญญาณรบกวนความถี่สูงที่เกิดจากบัลลาสต์อิเล็กทรอนิกส์จะไหลเข้าไปในระบบไฟฟ้า และจะไหลผ่านหม้อแปลงไฟฟ้าที่มีอยู่ในเครื่องใช้ไฟฟ้าต่างๆในบ้านทำให้เกิดความร้อนขึ้น มีผลให้อายุการใช้งานของเครื่องใช้ไฟฟ้าในบ้านลดลง บัลลาสต์อิเล็กทรอนิกส์ที่มีคุณภาพสูงจะมีค่า THD ต่ำ

ภาษา C

ภาษา C เป็นภาษาคอมพิวเตอร์เพื่อวัตถุประสงค์ทั่วไป เป็นภาษาที่มีความจำเป็นมาก มันสนับสนุนการเขียนโปรแกรมที่มีโครงสร้าง การกำหนดขอบเขตของตัวแปร และการเรียกใช้ตัวเอง (Recusion) และมันเป็นภาษาที่อยู่ในระดับต่ำ (Low level) นั่นคือ มันเป็นภาษาที่สามารถทำงานได้ดีในระดับของฮาร์ดแวร์ ภาษา C เป็นสามารถที่ออกแบบมาให้สามารถที่จะทำงานกับคำสั่งพื้นฐานของคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ เพราะฉะนั้นมันจึงถูกพบบ่อยในการใช้สร้างแอพพลิเคชันในสมัยก่อนที่เขียนโดยภาษาแอสเซมบลี รวมถึงระบบประฏิบัติการ เช่นเดียวกันกับซอฟต์แวร์ประยุกต์สำหรับคอมพิวเตอร์ ซุปเปอร์คอมพิวเตอร์ และระบบฝังตัว ภาษา C นั้นถูกพัฒนาครั้งแรกโดย Dennis Ritchie ในระหว่างปี 1969 และ 1973 ที่ Bell Labs และใช้สำหรับพัฒนาและปรับปรุงระบบปฏิการ Unix ใหม่ ตั้งแต่นั้นมันได้มาเป็นภาษาที่มีการใช้งานอย่างกว้างขวางมากที่สุดตลอดเวลา ที่มากับ C คอมไพเลอร์จากบริษัทพัฒนาต่างๆ สำหรับพัฒนาในสถาปัตยกรรมคอมพิวเตอร์และระบบปฏิบัติการที่ได้รับความนิยมเป็นจำนวนมาก ภาษา C ได้ถูกกำหนดมาตฐานโดย American National Standards Institute (ANSI) ตั้งแต่ปี 1989 และ International Organization for Standardization (ISO) ในเวลาต่อมา

ภาษา C เป็นภาษาที่มีรูปแบบการเขียนโปรแกรมเป็นแบบลำดับ (Imperative procedural) ให้ถูกออกแบบให้คอมไพล์อย่างตรงไปตรงมากับคอมไพเลอร์ที่มีความเกี่ยวข้อง เพื่อให้สามารถเข้าถึงการจัดการหน่วยความจำในระดับต่ำ และทำให้โครงสร้างของภาษาเชื่อมโยงกับคำสั่งการทำงานของคอมพิวเตอร์อย่างมีประสิทธิภาพ ดังนั้น ภาษา C จึงมีประโยชน์กับการพัฒนาแอพพลิเคชันที่เคยเขียนโดยภาษา Assembly ยกตัวอย่าง เช่น โปรแกรมระบบ

ถึงแม้ว่าภาษา C มีความสามารถใน Low-level แต่มันยังถูกออกแบบเพื่อช่วยให้สามารถเขียนโปรแกรมแบบ Cross-platform โค้ดของโปรแกรมที่เขียนขึ้นจากมาตรฐานของภาษา C นั้นสามารถนำไปคอมไพล์ได้ในคอมพิวเตอร์ในแพลตฟอร์มและระบบปฏิบัติการที่หลากหลายโดยเพียงแค่เปลี่ยนแปลงโค้ดเพียงเล็กน้อย ภาษา C นั้นสามารถใช้ได้อย่างกว้างขวางในแพลตฟอร์มขนาดต่างๆ ตั้งแต่ Embedded microcontrollers ไปจนถึง Supercomputer

หลังจากคุณเรียนจบบทเรียนนี้ คุณจะเข้าใจพื้นฐานและโครงสร้างของภาษา C ได้ดีขึ้นอ รวมถึงแนวคิดและวิธีในการเขียนโปรแกรม และสามารถสร้างโปรแกรมอย่างง่ายไปจนถึงโปรแกรมที่มีความซับซ้อนได้ โดยคุ ณสามารถที่จะนำไปประยุกต์ใช้ในการเขียนโปรแกรมภาษาอื่นอีกหลายๆ ภาษาได้ เพราะว่าภาษาส่วนมากนั้นสร้างมากจากภาษา C เช่น ภาษา C++ ภาษา Java และภาษา PHP ดังนั้น ในการที่คุณเริ่มต้นเรียนรู้จากภาษา C คุณจะได้เปรียบมากกว่า และมันจะง่ายสำหรับคุณในการเขียนรู้การเขียนโปรแกรมในภาษาอื่นต่อไป

โครงสร้างของโปรแกรม

Blocks

บล็อค คือสิ่งที่กำหนดขอบเขตและควบคุมการทำงานของโปรแกรม ซึ่งจะใช้เครื่องหมาย { และสิ้นสุดด้วย } ในภาษา C บล็อคนั้นมีหลายรูปแบบ เช่น บล็อคของฟังก์ชัน บล็อคของคำสั่งควบคุม หรือบล็อคย่อยในโปรแกรม และนอกจากนี้บล็อคยังสามารถซ้อนกันได้ นี่เป็นตัวอย่างของบล็อคในภาษา C

```c
#include <stdio.h>

int sum(int a, int b)
{
    return a + b;
}

int main()
{
    int x = 3, y = 4;

    if (x < y)
    {
        printf("%d\n", sum(x, y));
    }

    printf("Summation program\n", sum(x, y));
    return 0;
}
```

ในตัวอย่าง เป็นโปรแกรมหาผลรวมของตัวเลขสองจำนวนในภาษา C ในโปรแกรมนั้นประกอบไปด้วย 3 บล็อค คือบล็อคของฟังก์ชัน main ซึ่งเป็นฟังก์ชันหลักในการทำงานของโปรแกรมซึ่งโปรแกรมจะเริ่มการทำงานที่นี่ บล็อคที่สองเป็นของฟังก์ชัน sum() ซึ่งเป็นฟังก์ชันสำหรับหาผลรวมของตัวเลขที่เรียกโดยฟังก์ชันหลัก และภายในฟังก์ชัน main มีบล็อคคำสั่งเงื่อนไข if เพื่อควบคุมการทำงานของโปรแกรมอยู่ภายใน

```c
#include <stdio.h>
```

นภาษา C คำสั่ง #include ใช้สำหรับนำเข้าโค้ดจากไฟล์อื่นเข้ามาใช้งานในโปรแกรม โดยปกติแล้วเราได้นำเข้าชุดไลบรารี่ stdio.h ของภาษา C ซึ่งประกอบไปด้วยฟังก์ชันต่างๆ ที่จำเป็นในการเขียนโปรแกรม เช่น ฟังก์ชันสำหรับแสดงผลออกทางหน้าจอ หรือรับค่าจากคีย์บอร์ด เป็นต้น คุณสามารถสร้างไฟล์ .h ของคุณเองได้เพื่อนำมาใช้ในโปรแกรม

Comment

Comment เป็นส่วนของโค้ดที่ไม่มีผลต่อการทำงานของโปรแกรม มันถูกใช้เพื่ออธิบายโปรแกรมสำหรับมนุษย์เข้าใจ ในภาษา C เราสามารถคอมเม้นท์ได้สองวิธี คือ การคอมเมนต์แบบหนึ่งบรรทัดและแบบหลายบรรทัด

```c
// Single line comment example
#include <stdio.h>

int main()
{
    // This is my comment
    printf("Hello C language\n");
    return 0;
}
```

ในตัวอย่างเป็นการคอมเมนต์แบบหนึ่งบรรทัด เราจะใช้เครื่องหมาย Double slash (//) และตามด้วยสิ่งที่เราต้องการคอมเมนต์ ในตัวอย่างเราได้สร้างคอมเม้นต์สองอันในโปรแกรม

```c
/* Multiple lines comment example */
#include <stdio.h>

int main()
{
    /* This is my comment
    This another line comment */
    printf("Hello C language\n");
    return 0;
}
```

คอมเม้นท์อีกแบบหนึ่งคือการคอมเมนต์หลายบรรทัด มันจะละเว้นทุกอย่างหลังจากการปรากฏครั้งแรกของเครื่องหมาย Slash star (/*) และสิ้นสุดที่ Star slash (*/)

Semicolon

เซมิโคลอน (Semicolon) (;) เป็นสัญลักษณ์ที่ใช้เพื่อแบ่งแยกคำสั่งภายในโปรแกรม ซึ่งมันหมายถึงการจบคำสั่งนั้นๆ เซมิโคลอนใช้ในภาษาต่างๆ และมันเป็นสิ่งที่บังคับ เพื่อให้ตัวคอมไพเลอร์ของภาษาสามารถแยกแยะคำสั่งในการทำงานได้

```
int a;
int b = 5; b = 2;
printf("%d", a + b);
```

ในตัวอย่างเรามีสี่คำสั่ง บรรทัดแรกเป็นการประกาศตัวแปร บรรทัดที่สองคุณจะเห็นสองคำสังอยู่ในบรรทัดเดียวกันและมันสิ้นสุดด้วยเซมิโคลอน บรรทัดที่สามจะแสดงค่าผลรวมของตัวแปร a และ b

Whitespace

Whitespace คือตัวอักษรหรือเครื่องหมายที่ใช้แบ่งแยกคำสั่งและ Token ในโค้ดของโปรแกรม ในภาษา C นั้น Whitespace จะประกอบไปด้วย การเว้นวรรค Tab และการขึ้นบรรทัดใหม่ Whitespace ที่เรียงต่อกันเป็นจำนวนมากนั้นไม่มีผลต่อการทำงานของโปรแกรมและคอมไพเลอร์ แต่มันช่วยให้โปรแกรมเมอร์สามารถทำโค้ดให้เป็นระเบียบและสามารถอ่านเข้าใจได้ง่ายขึ้นโดยคนอื่นๆ แต่ Whitespace ยังคงต้องใช้ในบางที่ เช่น ระหว่างคำสั่งของภาษา C และชื่อของตัวแปร เป็นต้น

```
int a = 1;
int b=2;
int c   =  3;
```

ในตัวอย่าง เป็นการใช้ Whitespace ในรูปแบบต่างๆ ซึ่งถือว่าเป็นรูปแบบการเขียนโปรแกรมของแต่ละคน แต่โดยปกติแล้วเรานิยมใช้ Whitespace ในรูปแบบของการประกาศตัวแปร a

Keywords

Keyword เป็นกลุ่มคำที่ถูกสงวนไว้โดยเราไม่สามารถใช้คำเหล่านี้ในการประกาศเป็นชื่อตัวแปร ฟังก์ชัน ซึ่งในโปรแกรมทุกภาษาต่างก็มี Keyword

นี่เป็น Keyword มาตราฐานในภาษา C

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

ตัวแปรใน Arduino

ตัวแปร หมายถึง กลุ่มของ ตัวอักษร ตัวเลข และ เครื่องหมายใดๆ ที่รวมกันเป็นชื่อ เพื่อใช้กำหนด เป็นตัวแทนของค่าข้อมูลที่เราต้องการจะอ้างถึงในโปรแกรม ทั้งนี้ก็เนื่องจากว่าในการทำงานของโปรแกรม จริงๆ นั้นจะใช้ค่าตัวเลขที่ผู้ใช้กำหนดให้ มาทำการประมวลผล ซึ่งในการเขียนโปรแกรมถ้าเราต้องเขียน โปรแกรมโดยกำหนดเป็นค่าตัวเลขให้กับโปรแกรมตรงๆเลย ก็จะทำให้โปรแกรมที่เราเขียนขึ้นเต็มไปด้วยค่า ตัวเลขต่างๆ มากมาย ซึ่งยากต่อการอ่าน ยากต่อการทำความเข้าใจ และยากต่อการตรวจสอบความถูกต้อง และอาจทำให้เกิดความผิดพลาดได้ง่ายด้วย ดังนั้นทุกภาษา จึงยอมให้มีการกำหนดชื่อ ขึ้นมาใช้งานแทน ค่าตัวเลข เพื่อให้เขียนโปรแกรมได้สะดวกและง่ายต่อการอ่าน ทำความเข้าใจ ได้มากยิ่งขึ้น

ซึ่งลักษณะของข้อมูล อาจมีทั้งแบบที่เป็นค่าซึ่งสามารถเปลี่ยนแปลงได้ (variable) หรือ อาจเป็น แบบที่มีค่าคงที่ไม่สามารถเปลี่ยนแปลงได้ (constant) ในการประกาศใช้งานตัวแปร จำเป็นต้องประกาศ ชนิดของตัวแปร หรือบางครั้งอาจมีการกำหนดค่าเริ่มต้นให้กับตัวแปรด้วยก็ได้

ชนิดและประเภทของตัวแปร

ถ้าหากว่าเราจะเปรียบเทียบว่า ตัวแปร คือ ภาชนะสำหรับบรรจุสิ่งของ และ ข้อมูล คือ สิ่งของที่เราต้องการจะเก็บ จะเห็นได้ว่า สิ่งของต่างๆรอบๆตัวเรานั้น จะมีคุณสมบัติที่แตกต่างกันไป ดังนั้นในการเลือกภาชนะสำหรับใช้บรรจุสิ่งของ เราก็จำเป็นต้องเลือกชนิดของภาชนะให้มีความเหมาะสมที่จะใช้เก็บสิ่งของ ด้วยซึ่งสิ่งแรกที่ต้องพิจารณาคือ เราจะต้องรู้จักคุณสมบัติของสิ่งของที่ต้องการจะจัดเก็บ และ จุดประสงค์ การใช้งาน ก่อน จากนั้นจึงจัดหาภาชนะที่มี ขนาด และ รูปทรงของภาชนะ เหมาะสมที่จะใช้เก็บสิ่งของ เพื่อให้สามารถจัดเก็บ และ นำสิ่งของออกมาใช้งานได้อย่างเขประหยัดมากที่สุด

ในภาษาซีนั้น มีการกำหนด และ จำแนก ชนิดของตัวแปร ไว้เป็น 5 ชนิดด้วยกัน โดยแต่ละชนิดจะมี คุณสมบัติการใช้งานที่ต่างกัน เพื่อใช้ในการเก็บข้อมูลที่มีรูปแบบแตกต่างกัน คือ

-char ใช้เก็บข้อมูลที่เป็นตัวอักษร (character) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 256 ค่า
-int ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็ม (integer) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 65536 ค่า
-float ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Single Precision
-double ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Double Precision ซึ่ง สามารถเก็บค่าตัวเลขทศนิยมที่มีความละเอียดและถูกต้องของทศนิยมมากว่าแบบ float ถึง 2 เท่า
-void ใช้เก็บตัวแปรที่ไม่มีค่า

คุณสมบัติเฉพาะของตัวแปร

สำหรับตัวแปรชนิดที่ใช้เก็บค่าเลขจำนวนเต็ม (char และ int) นั้น ในภาษาซี ไม่ได้มีการจำแนก ชนิดของตัวแปรเพื่อใช้เก็บค่าตัวเลขที่เป็น ค่าบวก หรือ ค่าลบ เป็นการเฉพาะ แต่ภาษาซี จะใช้วิธีการเพิ่ม คำสั่งสำหรับกำหนดคุณสมบัติเฉาพะให้กับตัวแปรไว้อีก 4 คำสั่ง สำหรับใช้กำหนดคุณสมบัติของตัวแปรแบบนี้ ให้มีคุณสมบัติที่เฉพาะเจาะจงลงไปอีกเป็นต้นว่า จะใช้ตัวแปรในการเก็บค่าตัวเลขที่เป็นค่าบวกอย่างเดียว หรือต้องการเก็บค่าตัวเลขแบบคิดเครื่องหมายด้วย เพื่อให้ผู้ใช้สามารถปรับแก้คุณสมบัติในการใช้งานของ ตัวแปรให้มีคุณสมบัติใกล้เคียงกับความต้องการใช้งานมากขึ้นไปอีก และเพื่อจำกัดขอบเขตการใช้งานของ ตัวแปรให้ตรงกับจุดประสงค์มากยิ่งขึ้น และยังเป็นการช่วยให้ประหยัดจำนวนของหน่วยความจำที่ใช้สร้าง ตัวแปรด้วยและทำให้โปรแกรมทำงานได้เร็วขึ้นกว่าเดิมอีกด้วย โดยในภาษาซีจะมีคำสั่ง ที่ใช้สำหรับระบุ คุณสมบัติเฉพาะของตัวแปรที่ใช้เก็บค่าเลขจำนวนเต็ม 4 คำสั่ง คือ

-unsigned ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรเฉพาะค่าที่เป็นบวกเท่านั้น
-signed ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรทั้งค่า บวก และ ลบ
-short ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรที่มีค่าน้อยกว่า int
-long ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรที่มีค่ามากกว่า int เป็น 2 เท่า

ชนิดข้อมูล	การเก็บข้อมูล	ขนาด
boolean	จริง (True) หรือ เท็จ (False)	1 บิต
char	ตัวเลข หรือตัวอักษร	1 ไบต์ ใส่ค่าได้ตั้งแต่ -128 ถึง 127
unsigned char	ตัวเลข หรือตัวอักษร	1 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 255
byte	ไบต์	1 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 255
int	ตัวเลขจำนวนเต็ม	2 ไบต์ ใส่ค่าได้ตั้งแต่ -32,768 ถึง 32,767
unsigned int	ตัวเลขจำนวนเต็ม	2 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 65,535 (2^16) - 1)
long	ตัวเลขจำนวนเต็มที่มีความยาว	4 ไบต์ ใส่ค่าได้ตั้งแต่ -2,147,483,648 ถึง 2,147,483,647
unsigned long	ตัวเลขจำนวนเต็มที่มีความยาว	4 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 4,294,967,295 (2^32 - 1)
float	ตัวเลขทศนิยมใช้ในการคำนวน	4 ไบต์ ใส่ค่าได้ตั้งแต่ 3.4028235E+38 ถึง -3.4028235E+38 มีทศนิยมได้ 6 ถึง 7 ตำแหน่ง
double (เฉพาะบอร์ด Arduino Due)	ตัวเลขทศนิยมที่มีความยาวและต้องการความแม่นยำ	8 ไบต์ ใช้ในการคำนวนที่ต้องการประสิทธิ์ภาพสูง
String	ข้อความ	ไม่ระบุ

การประกาศตัวแปร เซ็ตค่าตัวแปร

การประกาศตัวแปรจะเหมือนกับภาษา C โดยปกติ คือ
TYPE KEY;
โดย TYPE เป็นชนิดของข้อมูล ส่วน KEY เป็นชื่อตัวแปร การประกาศตัวแปรข้างต้นคือการประกาศตัวแปรแบบไม่กำหนดค่า ดังนั้นค่าปกติที่อ่านจากตัวแปรจะเป็น 0

TYPE KEY = VAL;
จากตัวอย่างด้านบน เป็นลักษณะของการประกาศตัวแปรแบบกำหนดค่า เมื่ออ่านค่าของตัวแปรออกมา จะได้เป็นค่าที่กำหนดไว้ตอนประกาศ

ตัวอย่างการประกาศตัวแปร

```
int i;
int a = 10, b = 20;
```

จากตัวอย่าง จะเห็นว่าเราสามารถกำหนดชนิดของข้อมูลให้ทีเดียวหลายๆตัวแปรก็ได้ โดยใช้เครื่องหมาย,คั่นไว้

```
int i;
int a = 10, b = 20;
i = a + b;
```

จากตัวอย่างด้านบน เราสามารถกำหนดค่าให้กับตัวแปรเมื่อไรก็ได้ โดยใช้เครื่องหมายเท่ากับ = เป็นตัวเชื่อม ชื่อตัวแปรจะอยู่ทางซ้าย และจะกำหนดค่าเป็นอะไร ให้อยู่ทางขวา ค่าที่อยู่ทางขวาจะถูกนำมาใส่ในค่าที่อยู่ทางซ้ายเสมอ

```
int i = 10, a;
a = i;
```

จากตัวอย่างด้านบน จะเห็นว่า a ไม่ได้กำหนดค่าไว้ตอนประกาศ ทำให้ค่าที่อ่านได้จาก a คือ 0 แต่บรรทัดถัดมา มีการกำหนดค่าให้ a เท่ากับ i ซึ่งตอนประกาศ i ได้ประกาศไว้ว่าค่าเท่ากับ 10 เมื่อนำมาใส่ a ค่าที่อ่านได้จาก a จึงเป็น 10 ด้วยเช่นกัน

```
boolean is = false;
is = !is;
```

จากตัวอย่างด้านบน มีการประกาศตัวแปร boolean ซึ่งเป็นตัวแปรทางลอจิก มีค่าเป็น True (1) หรือ False (0) ได้เท่านั้น ในบรรทัดแรกได้ประกาศว่าตัวแปร is เป็นตัวแปรชนิด boolean และมีค่าเป็น false หรือลอจิก 0 บรรทัดต่อมา ได้มีการกำหนดให้ is เท่ากับ !is การที่เครื่องหมายนิเสธไปอยู่หน้าตัวแปร หมายถึงการกลับเป็นค่าตรงข้าม จากบรรทัดแรก ตัวแปร is มีค่าเป็น false เมื่อเจอ !is ค่าจึงถูกกลับเป็น true และถูกนำไปเซ็ตในตัวแปร is ทำให้สุดท้ายแล้วตัวแปร is มีค่าเป็น true

```
String text = "Myarduino";
```

จากตัวอย่างด้านบน ได้มีการประกาศตัวแปรชื่อ text เป็นชนิด String เมื่ออ่านค่าที่ได้จาก text จึงได้ค่าออกมาเป็น "Myarduino" เลย

* การกำหนดค่าแบบข้อความให้กับตัวแปร จะต้องอยู่ภายใต้เครื่องหมาย " " เท่านั้น มิฉนั้นโปรแกรมจะแสดงข้อความผิดพลาดออกมา

สรุปชนิดของตัวแปรใน Arduino ที่ใช้บ่อยๆ

- **boolean** ใช้เก็บค่าข้อมูล เพียง 2 จำนวน คือ TRUE (จริง) และ FALSE (เท็จ)
- **char** ใช้เก็บค่าข้อมูลขนาด 8 บิต ใช้สำหรับเก็บค่ารหัสของตัวอักษร ซึ่งสามารถกำหนดเป็นค่า หรือเขียนตัวอักษรไว้ภายใต้เครื่องหมาย ฟันเดียวก็ได้ เช่น 'A' หรือ 0x41 หรือ 65
- **byte** ใช้เก็บค่าข้อมูลขนาด 8 บิตที่เป็นค่าจำนวนเต็มแบบไม่คิดเครื่องหมาย เหมือนกันกับunsigned char ในภาษาซี) ซึ่งสามารถเก็บค่าข้อมูลได้ 256 ค่า คือ 0-255
- **int หรือ Integer** ใช้เก็บค่าข้อมูลขนาด 16บิต ที่เป็นค่าจำนวนเต็ม แบบคิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 65536 ค่า คือ -32768 ถึง +32767
- **unsigned int** ใช้เก็บค่าข้อมูลขนาด 16บิต ที่เป็นค่าจำนวนเต็ม แบบไม่คิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 65536 ค่า คือ 0-65535
- **long** ใช้เก็บค่าข้อมูลขนาด 32บิต ที่เป็นค่าเลขจำนวนเต็มแบบคิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 4294967296 ค่า คือ -2,147,483,648 ถึง 2,147,483,647
- **unsigned long** ใช้เก็บค่าข้อมูลขนาด 32 บิต ที่เป็นค่าเลขจำนวนเต็มแบบไม่คิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 4294967296 ค่า คือ 0 ถึง 4,294,967,295
- **float** ใช้เก็บค่าข้อมูลที่เป็นเลขทศนิยมแบบคิดเครื่องหมายขนาด 32 บิต โดยสามารถเก็บค่าได้ระหว่าง 3.4E-38 ถึง 3.4E+38 (-3.4028235E+38 ถึง 3.4028235E+38)
- **double** ใช้เก็บค่า เลขทศนิยมเช่นเดียวกันกับ float แต่มีค่าความละเอียดกว่า float ถึง 2 เท่า สามารถเก็บค่าได้มากถึง 1.7E+308
- **void** เป็นตัวแปรแบบที่ไม่มีการเก็บค่าใดๆ คือ ไม่มีค่านั้นเอง
- **arrays** เป็นตัวแปรที่ใช้เก็บข้อมูลหลายๆค่าไว้ในตัวแปรตัวเพียงชื่อเดียว แต่มีตัวเลขสำหรับชี้ตำแหน่งการเก็บข้อมูลต่างกัน โดยตัวเลขที่ใช้ทำหน้าที่เป็นตัวชี้ตำแหน่งของข้อมูล เรียกว่า Index Number โดยค่าลำดับของข้อมูลในตัวแปร array ตำแหน่งแรกจะมีค่าเป็น ศูนย์เสมอ
- **string** เป็นตัวแปรใช้เก็บข้อความ หรือ ตัวอักษรหลายๆตัว ซึ่ง string ก็คือ array ของตัวแปร แบบ char นั่นเอง
- **pointer** เป็นตัวแปรที่ไม่ได้ใช้เก็บข้อมูล แต่ใช้เก็บค่าตำแหน่งแอดเดรสของหน่วยความจำที่ใช้ สร้างเป็นตัวแปรสำหรับเก็บข้อมูล ซึ่งตัวแปรแบบนี้จะใช้ทำหน้าที่เป็นตัวชี้ไปยังตำแหน่งแอดเดรส ของตัวแปรอื่นๆอีกที่หนึ่ง

บทที่ 3
ขั้นตอนการดำเนินการ

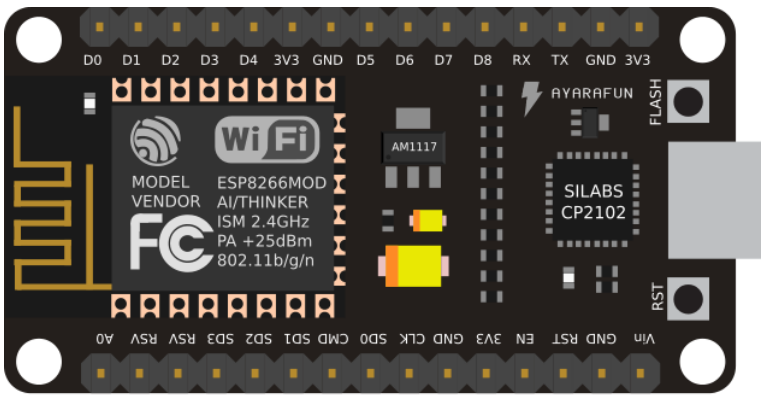
ในบทนี้ได้อธิบายวิธีการออกแบบแต่ละส่วนที่ใช้ในโครงงานนี้ประกอบไปด้วย ชิ้นส่วนอุปกรณ์ทำโครงงาน แบ่งเป็น ชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์ และ การสร้างบล็อกไดอะแกรม การทำกล่องวงจร

3.1 ชิ้นส่วนอุปกรณ์ทำโครงงาน

ชิ้นส่วนและอุปกรณ์ในการทำโครงงานเซนเซอร์กันขโมยแจ้งเตือนผ่านโทรศัพท์มือถือ ประกอบด้วย

3.1.1 อุปกรณ์อิเล็กทรอนิกส์

- NodeMCU คือ บอร์ดคล้าย Arduino ที่สามารถเชื่อมต่อกับ WiFi ได้, สามารถเขียนโปรแกรมด้วย Arduino IDE ได้เช่นเดียวกับ Arduino และบอร์ดก็มีราคาถูกมากๆ เหมาะแก่ผู้ที่คิดจะเริ่มต้นศึกษา หรือทดลองใช้งานเกี่ยวกับ Arduino , IoT , อิเล็กทรอนิกส์ หรือแม้แต่การนำไปใช้จริงในโปรเจคต่างๆ ก็ตาม เพราะราคาไม่แพง ภายในบอร์ดของ NodeMCU ประกอบไปด้วย ESP8266 (ไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อ WiFi ได้) พร้อมอุปกรณ์อำนวยความสะดวกต่างๆ เช่น พอร์ต micro USB สำหรับจ่ายไฟ/อัปโหลดโปรแกรม, ชิพสำหรับอัปโหลดโปรแกรมผ่านสาย USB, ชิพแปลงแรงดันไฟฟ้าและขาสำหรับเชื่อมต่ออุปกรณ์ภายนอกเป็นต้น



-รีเลย์ (Relay)คือ สวิตช์ตัด-ต่อวงจรแบบอิเลคทรอนิค โดยใช้หลักการของแม่เหล็กไฟฟ้า จะทำงานเมื่อมีการจ่ายไฟไปตามกำหนดทำให้เกิดวงจรเปิด เมื่อไม่มีการจ่ายไฟจะทำให้เกิดวงจรปิดทำให้อุปกรณ์ไฟฟ้าที่ใช้รีเลย์เป็นสวิสต์นั้นไม่ทำงาน

-adapter คือ หม้อแปลงไฟฟ้า จากไฟฟ้ากระแสสลับ(ไฟบ้าน)ที่มีค่าความต่างศักย์ 220 โวลต์ ให้เป็นไฟฟ้ากระแสตรง (DC) ที่มีค่าความต่างศักย์ต่ำลง เพื่อให้สามารถจ่ายกระแสไฟฟ้าให้กับเครื่องใช้ไฟฟ้าได้

-สวิตซ์ 3 ทาง ใช้เลือกการทำงานของวงจรว่าจะใช้ระบบออโต้หรือแมนนวล

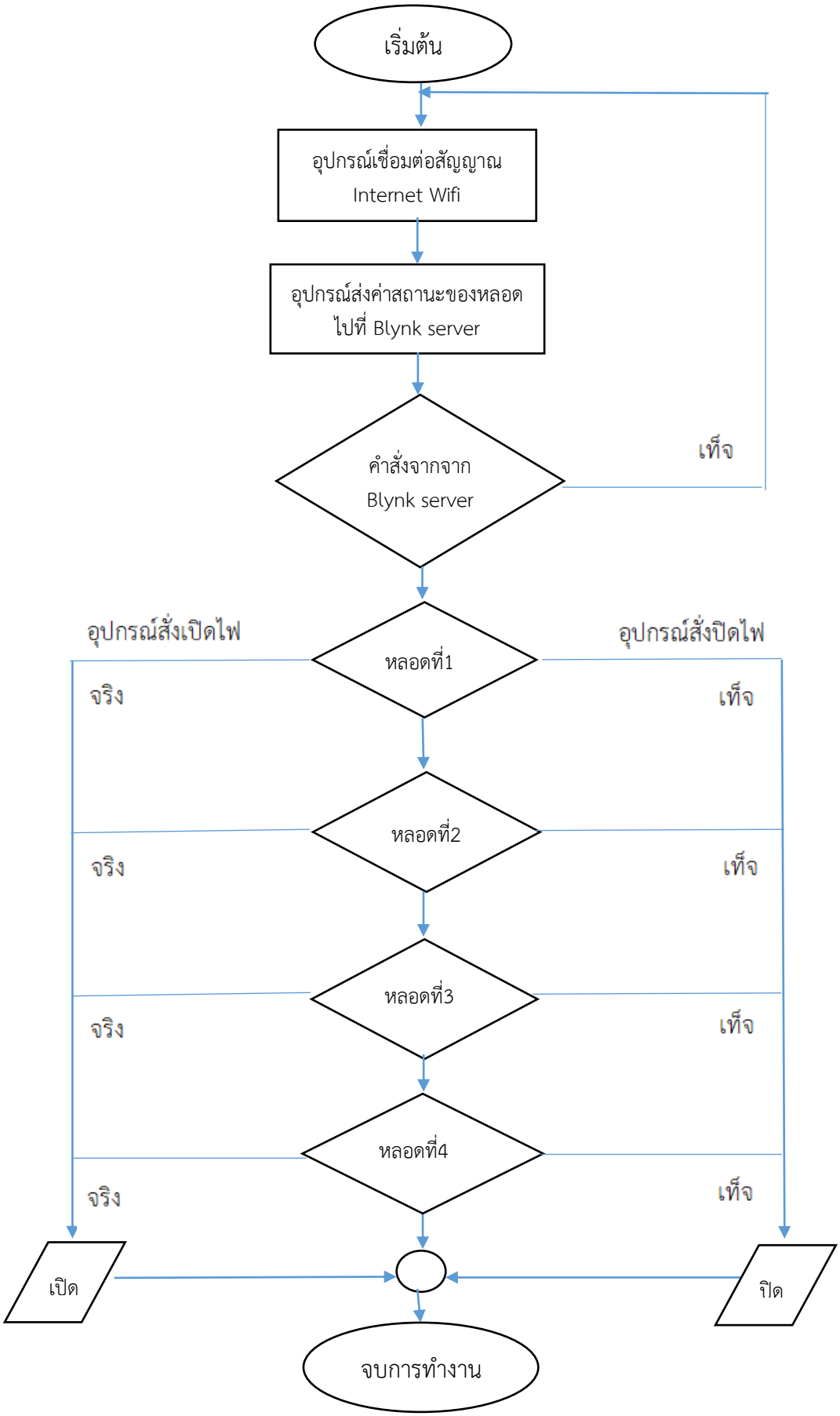
3.1.2 อุปกรณ์ทำกล่องวงจร

-แผ่นอะคริลิค (Acrylic)

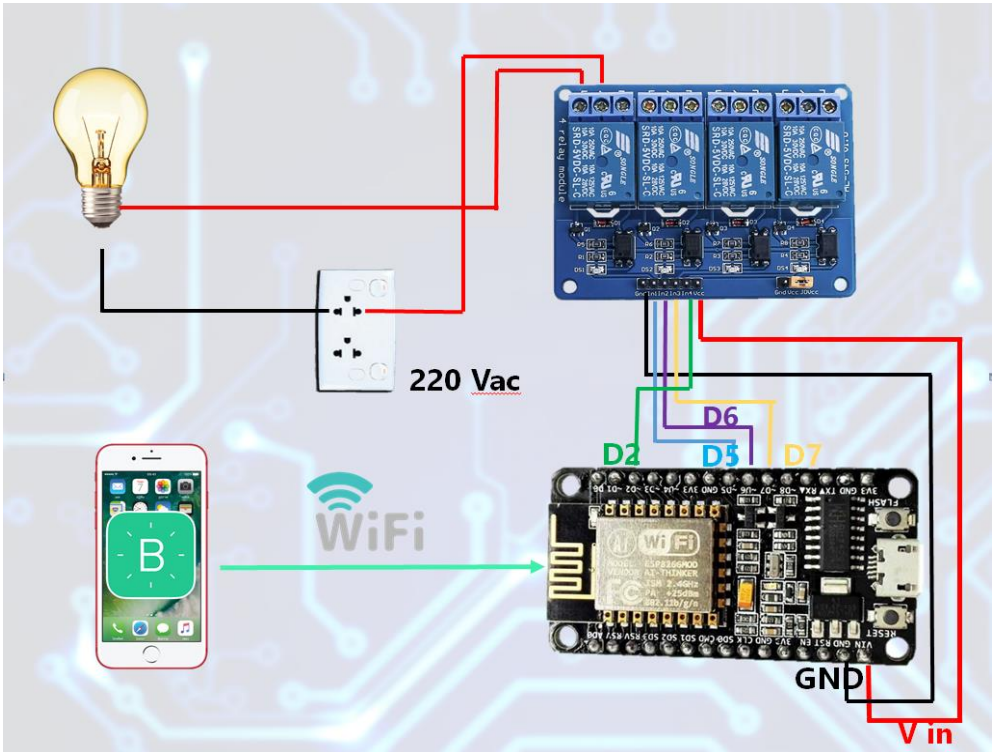
-กระเป๋าเครื่องมืองานอิเล็กทรอนิกส์ (Electronic Tool Pouch)

-กล่องวงจร

3.2Flow chart การทำงานของโครงงาน



3.3 ภาพรวมการเชื่อมต่อของระบบ (Wiring Diagram)



ภาพรวมเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต

3.4 วิธีการดำเนินโครงงาน

3.4.1 ประชุมโครงงานทำการประชุมกันภายในกลุ่มโครงงานโดยมีครูที่ปรึกษาคอยให้คำแนะนำ และ แบ่งมอบหน้าที่ให้แต่ละคนที่จะต้องรับผิดชอบ

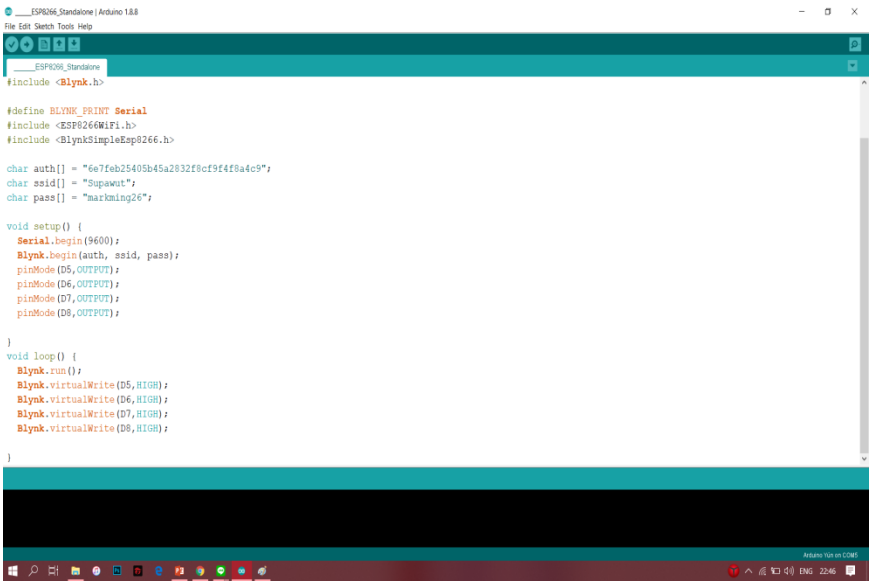
3.4.2 สืบค้นข้อมูล ค้นหาข้อมูลที่เกี่ยวข้องในการทำโครงงาน หาข้อมูล ของ Application Blynk จากแหล่งข้อมูลต่างๆ

3.4.3 ขั้นตอนการประกอบวงจรเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต

- ติดตั้งบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 ลงบอร์ด

-ตัดแผ่นอะคริลิคทำแผ่นยึดวงจร

-เขียนโปรแกรมลงใส่บอร์ด



-ติดตั้งวรจรลงในกล่องใส่วงจร

-นำโครงงาน เปิด-ปิดไฟ ด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต ไปติดตั้ง

บทที่ 4
ผลการทดลอง

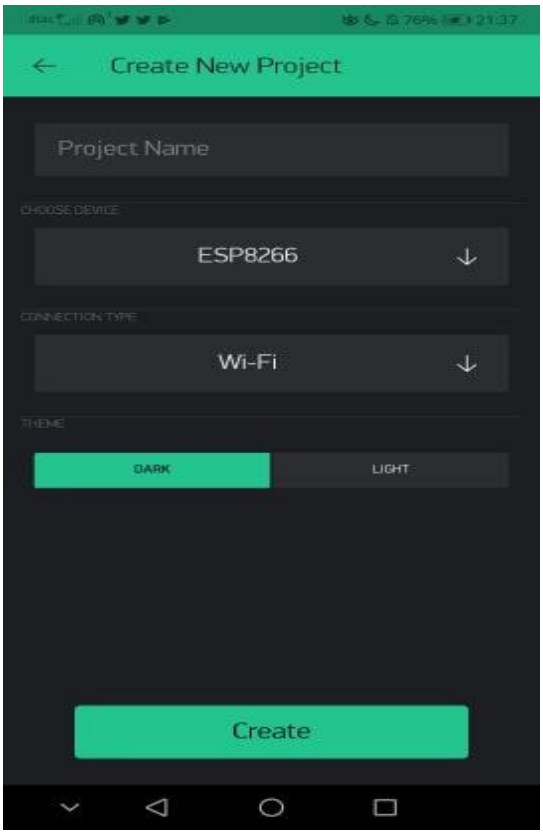
การสร้างและพัฒนาระบบ เปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต เรียบร้อย แล้ว ลำต่อไปคือ การทดสอบการใช้งานระบบ เปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต โดยทางผู้จัดทำได้ดำเนินการทดสอบการใช้งานโดยมีรายละเอียดซึ่งจะได้กล่าวในบทนี้

4.1 วิธีการทดสอบ

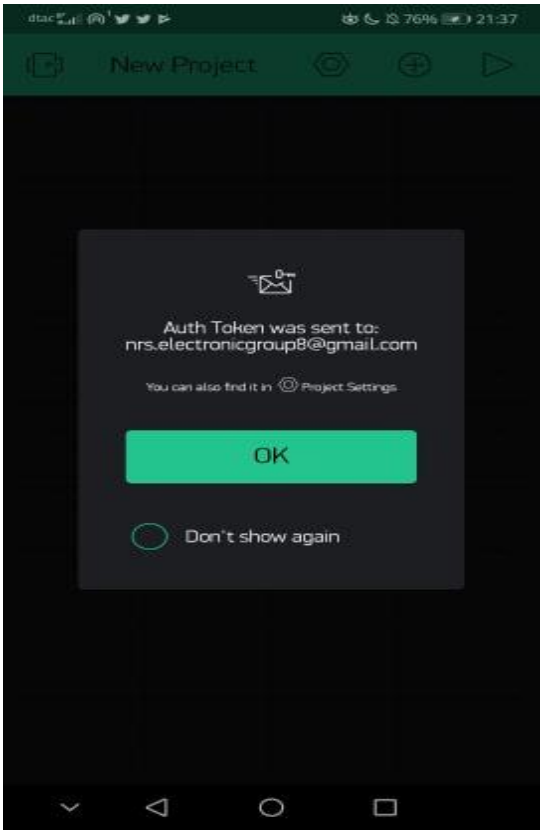
เนื่องจากระบบเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต ที่สร้างขึ้นนี้จัดว่าเป็นเครื่องต้นแบบซึ่งใช้ระยะเวลานานพอสมควรด้วยข้อจำกัดทางด้านเวลาและทุนที่ใช้ในการทดสอบทางผู้จัดท าโครงงาน จึงได้ทดสอบระบบแจ้งเตือนความปลอดภัยโดยแบ่งการทดลองเป็น 2 ส่วน มีดังนี้

4.1.1 การทดลองคำสั่งด้วยแอปพลิเคชัน

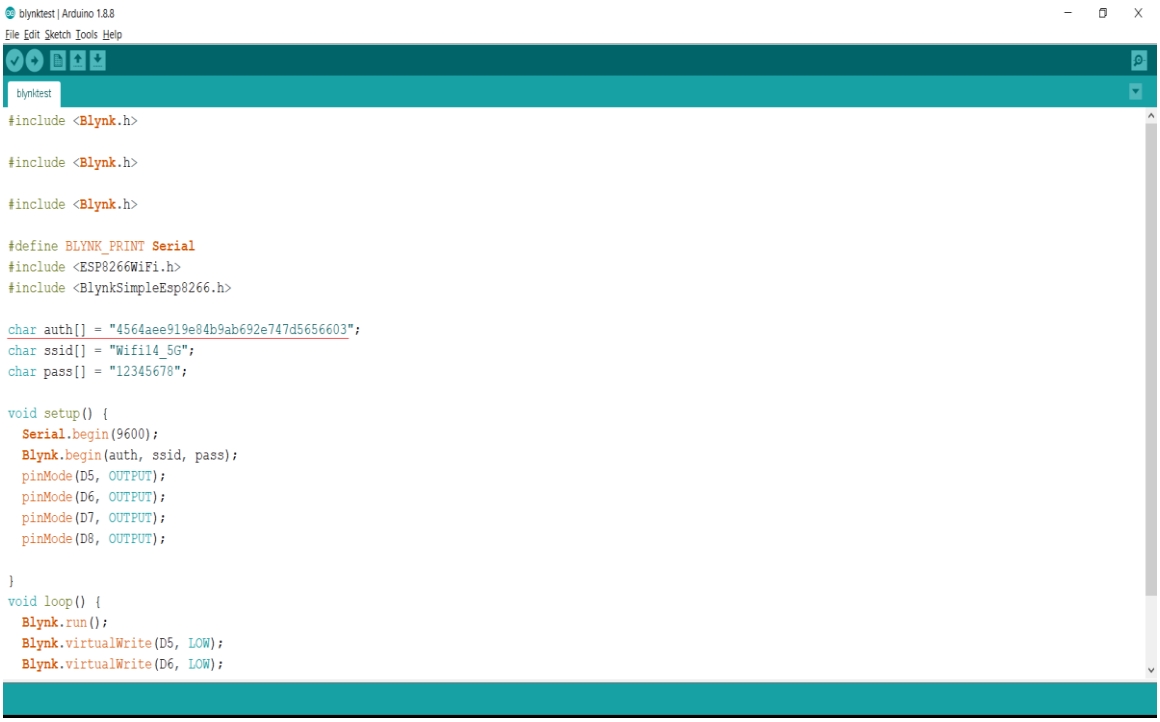
ทำการเปิดแอปพลิเคชัน ขึ้นมาโดยกดที่ไอคอน new project จากนั้น เลือกชนิดบอร์ดที่เราต้องการใช้ และ เลือก วิธีเชื่อมต่อสัญญาณกับบอร์ดเข้าด้วยกัน หน้าจอจะเป็นตาม รูปที่ 4.1 จากนั้นกดสร้างแอปพลิเคชันจะส่งรหัสโทเค่นไปยัง e-mail และนำโทเค่นที่ได้มาใส่ใน ในขั้นตอนการเขียนโปรแกรมที่เราเขียนไว้ ดังรูปที่ 4.2 ให้ทำการรอสักพักเพื่อให้แอปพลิเคชันและบอร์ด เชื่อมต่อกับ wifi จะมีข้อความว่า Connected ขึ้นดัง รูปที่ 4.3 แสดงว่าแอปพลิเคชัน เชื่อมต่อกับบอร์ดแล้วพร้อมทำการสั่ง ON/OF ต่อไป



รูปที่ 4.1 หน้าจอแอปพลิเคชัน



รูปที่ 4.2 แอปพลิเคชันจะทำการส่ง token



รูปที่ 4.3 นำ token ที่ได้รับใส่ในโปรแกรม arduino

4.2 ขั้นตอนการทดสอบประสิทธิภาพ ระบบ เปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต

คณะผู้จัดทำได้ดำเนินการทดลองโปรแกรมและบันทึกผลโดยใช้เกณฑ์จากการใช้การจับเวลาเปิด-ปิด และใช้การวัดระยะการสั่งงาน แล้วนำผลการทดลองที่ได้มาบันทึก

ตารางบันทึกผลจากการ ใช้การจับเวลาเปิด และ ปิด

หลอดไฟ	**เปิดครั้งที่1**	**ปิดครั้งที่1**	**เปิดครั้งที่2**	**ปิดครั้งที่2**	**เปิดครั้งที่3**	**ปิดครั้งที่3**
หลอดที่1	1 วินาที	1 วินาที	1 วินาที	1 วินาที	1 วินาที	1 วินาที
หลอดที่2	1 วินาที	1 วินาที	1 วินาที	300 วินาที	1 วินาที	250 วินาที
หลอดที่3	1 วินาที	1 วินาที	1 วินาที	1 วินาที	1 วินาที	1 วินาที
หลอดที่4	5 วินาที	1 วินาที	5 วินาที	1 วินาที	5 วินาที	1 วินาที

ตารางบันทึกผลจากการ ใช้การวัดระยะการสั่งงาน

หลอดไฟ	**เปิดครั้งที่1 10เมตร**	**ปิดครั้งที่1 10เมตร**	**เปิดครั้งที่2 50เมตร**	**ปิดครั้งที่2 50เมตร**	**เปิดครั้งที่3 100เมตร**	**ปิดครั้งที่3 100เมตร**
หลอดที่1	1 วินาที	1 วินาที	2 วินาที	2 วินาที	5 วินาที	7 วินาที
หลอดที่2	1 วินาที	1 วินาที	2 วินาที	2 วินาที	5 วินาที	6 วินาที
หลอดที่3	1 วินาที	1 วินาที	2 วินาที	30 วินาที	5 วินาที	40 วินาที
หลอดที่4	5 วินาที	1 วินาที	6 วินาที	2 วินาที	8 วินาที	6 วินาที

บทที่ 5
สรุปผล อภิปรายผล และข้อเสนอแนะ

โครงงาน ระบบเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต สรุปผลการดำเนินการตามลำดับ ดังนี้

5.1 สรุปผลการดำเนินการ
5.2 อภิปรายผล
5.3 ข้อเสนอแนะ

5.1 สรุปผลการศึกษา

5.1.1 การประดิษฐ์โครงงานเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต ได้เครื่องเปิด-ปิดไฟ ที่มีลักษณะเด่น คือ

1. มีคุณลักษณะการใช้งานที่ง่าย
2. สามารถนำไปติดตั้งใช้งานได้จริง

5.1.2 การทดสอบประสิทธิภาพโครงงานเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต โดยการทดสอบ ประสิทธิภาพพบว่าการทำงานของอุปกรณ์ต่างๆในโครงงานเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต มีดังนี้

5.1.2.1 บอร์ด Node mcu esp 8266 ไม่สามารถเชื่อมต่ออินเตอร์เน็ตภายในอาคารได้ เราจึงทำการตรวจหาสาเหตุเพื่อที่จะได้แก้ไข จึงคาดว่าสาเหตุอาจจะมาจากการใช้อินเตอร์เน็ตที่ต้องใช้ค่าพร็อกซี่

5.1.2.2 หลังจากแก้ไขปัญหาโดยการเปลี่ยนเครือข่ายอินเตอร์เน็ต ปรากฏว่าโครงงานเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต สามารถใช้งานได้ปกติ

5.2 อภิปรายผล

โครงงานเปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ตที่ศึกษาเพื่อที่ต้องการจะทำให้สามารถควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าผ่านอินเตอร์เน็ตได้ โดยผลลัพธ์ของสิ่งประดิษฐ์นี้ เราสามารถเปิดปิดอุปกรณ์ไฟฟ้าด้วยโทรศัพท์มือถือผ่านระบบอินเตอร์เน็ตได้ จากระยะทางที่ไกลโดยที่ไม่ต้องเดินไปที่ตัวสวิตซ์ไฟฟ้า

5.3 ข้อเสนอแนะ

1. ความแรงของสัญญาณอินเตอร์เน็ตมีผลต่อการส่งข้อมูลจากโทรศัพท์มือถือไปยังบอร์ด Node MCU8266
2. ควรติดตั้งกล่องวงจรไว้ภายในอาคาร

บรรณานุกรม

1. https://www.arduinospro.com/b/25

2. https://spidyhero.wordpress.com/2015/08/01/esp8266-esp-01-blynk-led-controller-thai-version/?fbclid=IwAR0YEFZwSo_cDAxiMvrpjf48p8EbMwbwT0TxbH_ea3iNlMHoHBs7mKHK2c8

3. http://nodemcutom.blogspot.com/2018/07/7-internet-blynk.html

4. http://dtecesp8266arduino.blogspot.com/2017/12/esp8266-esp8266-esp8266-wifi-3.html

ภาคผนวก ก.

ระบบ เปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต

```
#include <Blynk.h>
#include <Blynk.h>
#include <Blynk.h>

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "0a09c210e8d440e0a6154ffeb7436066";
char ssid[] = "Wifi14_5G";   //wifiname
char pass[] = "12345678";  //wifipassword

void setup() {
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(D5,OUTPUT);
  pinMode(D6,OUTPUT);
  pinMode(D7,OUTPUT);
  pinMode(D8,OUTPUT);
  digitalWrite(D5,1);
  digitalWrite(D6,1);
  digitalWrite(D7,1);
  digitalWrite(D8,1);
}
void loop() {
  Blynk.run();
  Blynk.virtualWrite(D5,HIGH);
  Blynk.virtualWrite(D6,HIGH);
  Blynk.virtualWrite(D7,HIGH);
  Blynk.virtualWrite(D8,HIGH);

}
```

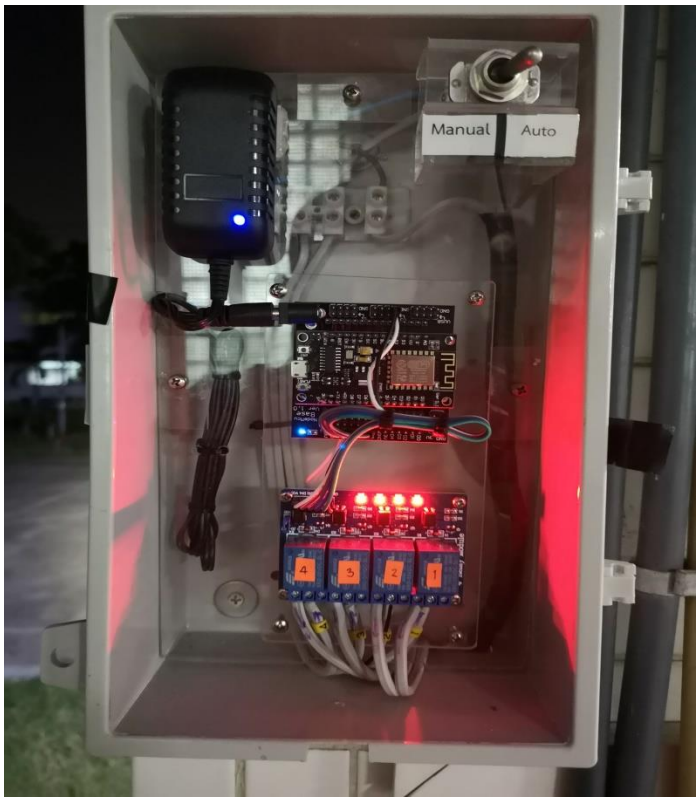
ภาคผนวก ข.

วิธีการประกล่องวงจร เปิด-ปิดไฟด้วย Application Blynk ผ่านระบบอินเตอร์เน็ต

1.ประกอบกล่องวงจร

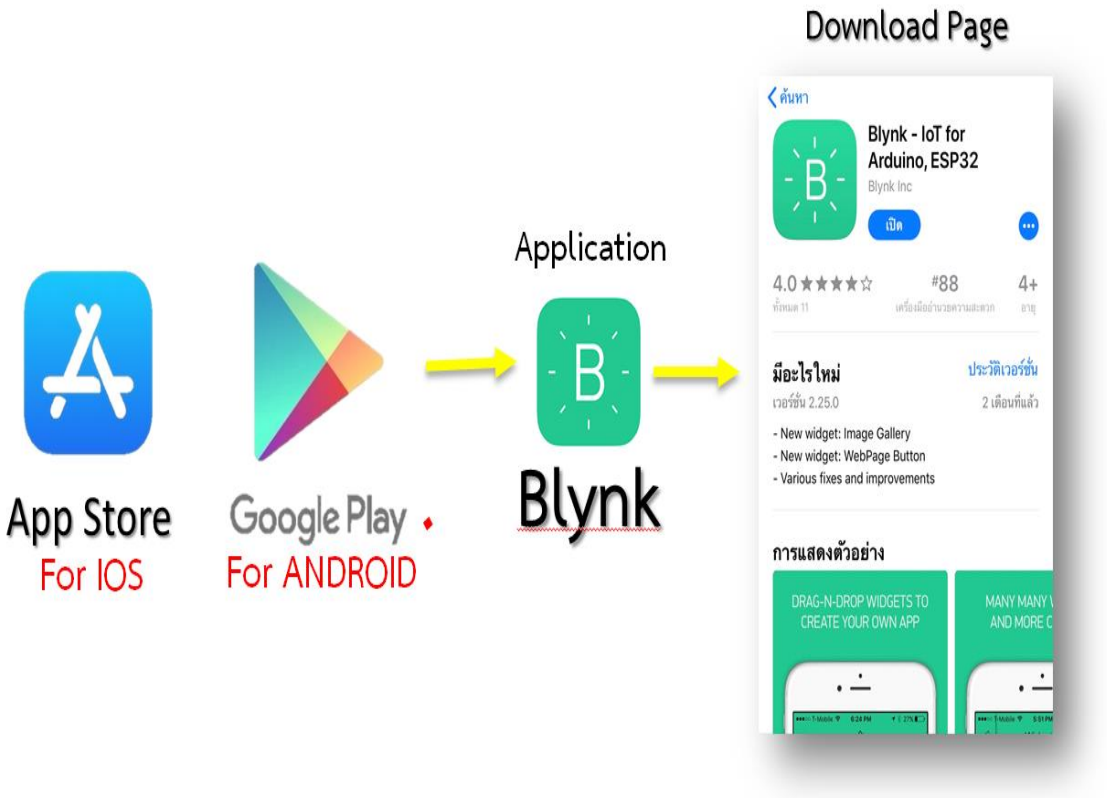
4.ติดตั้งวงจรลงในกล่อง

5.ประกอบเสร็จสมบรูณ์

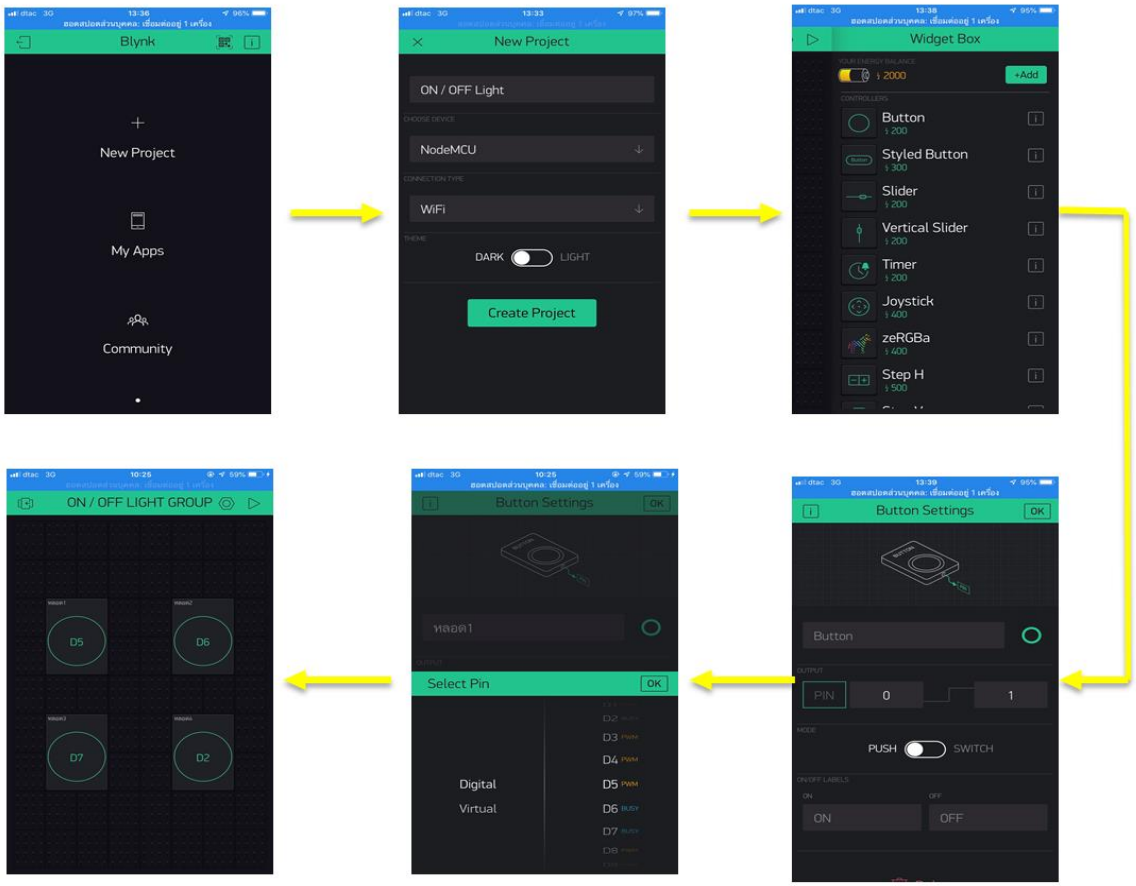


ภาคผนวก ค
คู่มือการใช้งาน

ช่องทางการดาวน์โหลด



สมัครใช้งาน Application Blynk

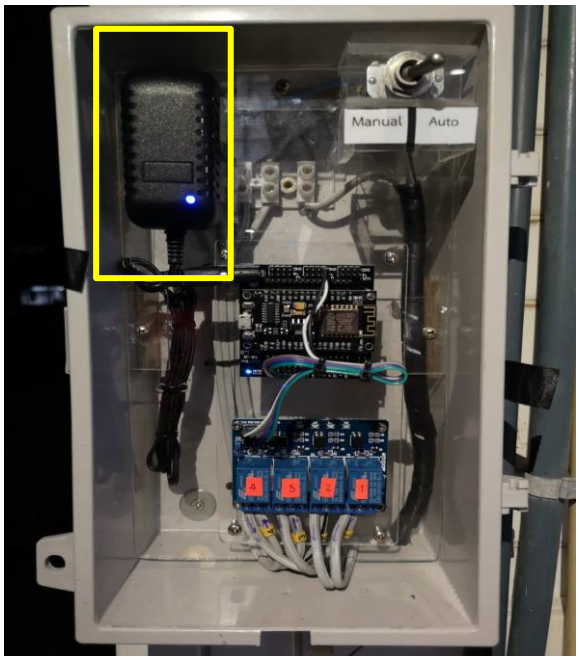


เข้าระบบ Application Blynk

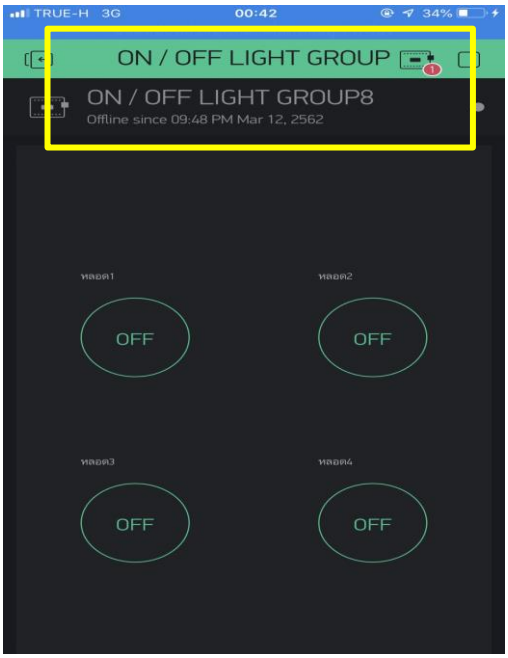
1. ใช้รหัสบัญชีผู้ใช้ nrs.electronicgroup8@gmail.com
2. รหัสผ่าน 123456789group8

QR-Code ของ Application Blynk

1.เสียบปลั๊กในกล่องทดลอง



2. เช็คสถานการณ์เชื่อมต่อ internet ของตัวบอร์ดใน Application Blynk.



3.สับสวิตช์ข้างบนขวามือในกล่องมาที่ Auto

ภาคผนวก จ.
ประวัติคณะผู้จัดทำโครงงาน

1.นรจ.ศุภวุฒิ ขวัญอยู่เย็น เหล่า อิเล็กทรอนิกส์
ที่อยู่ 22/53 หมู่6 ต.สัตหีบ อ.สัตหีบ จ.ชลบุรี 20180
จบจาก โรงเรียนสิงห์สมุทร

2.นรจ.เอราวัต นิ่มบุญ เหล่า อิเล็กทรอนิกส์
ที่อยู่ 41/12 หมู่6 ต.สัตหีบ อ.สัตหีบ จ.ชลบุรี 20180
จบจาก โรงเรียนสิงห์สมุทร

3.นรจ.ธนกร ปะนามะทัง เหล่า อิเล็กทรอนิกส์
ที่อยู่ 991050 หมู่6 ต.บางเมือง อ.เมือง จ.สมุทรปราการ 10270
จบจาก โรงเรียนเทพศิรินทร์ สมุทรปราการ

3.นรจ.นรจ.ธีระวัฒน์ ทรัพย์คง เหล่า อิเล็กทรอนิกส์
ที่อยู่ 55/2 หมู่1 ต.เขาขลุง อ.บ้านโป่ง 70110
จบจาก โรงเรียนโรงเรียนรัตนราษฎร์บำรุง

3.นรจ.ภานุภัณฑ์ แงชัยภูมิ เหล่า อิเล็กทรอนิกส์
ที่อยู่ 141/2 หมู่2 ต.นาเสียว อ.เมือง 36000
จบจาก โรงเรียนชัยภูมิภัคดีชุมพล