# เรื่อง เครื่องควบคุมเครื่องใช้ไฟฟ้าเปิด-ปิดแบบอัตโนมัติ

(AUTOMATIC APPLIANCE CONTROL SYSTEM ON - OFF)

จัดทำโดย

นรจ.เอกกมล ต่อมสุวรรณ์
นรจ.นิติธร พวงบุษบา
นรจ.ธนายุทธ ป้อมแก้ว
นรจ.ชัยวัฒน์ ปานนนท์
นรจ.ชนนันท์ โรจนวิภาต
นรจ.พงศธร อิ่มรอ

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรนักเรียนจ่าทหารเรือ ชั้นปีที่ 2

พรรค พิเศษ เหล่า ทหารช่างยุทธโยธา (อิเล็กทรอนิกส์) ปีการศึกษา 2563

โรงเรียนอิเล็กทรอนิกส์ กองวิทยาการ กรมอิเล็กทรอนิกส์ทหารเรือ

**หัวข้อโครงงาน** เครื่องควบคุมเครื่องใช้ไฟฟ้าเปิด-ปิดแบบอัตโนมัติ

(AUTOMATIC APPLIANCE CONTROL SYSTEM ON - OFF)


**ผู้จัดทำ** นรจ.เอกกมล ต่อมสุวรรณ์

นรจ.นิติธร พวงบุษบา

นรจ.ธนายุทธ ป้อมแก้ว

นรจ.ชัยวัฒน์ ปานนนท์

นรจ.ชนนันท์ โรจนวิภาต

นรจ.พงศธร อิ่มรอ

**ครูที่ปรึกษา** ร.ท.วินัย ศิริโชติ

พ.จ.อ.นเรศ แสงม่วง

พ.จ.ต.นรภัทร แย้มดวง


**ปีการศึกษา ๒๕๖๓**

# บทคัดย่อ

โครงงานนี้นำเสนอระบบควบคุมเครื่องใช้ไฟฟ้าอัตโนมัติเครื่องควบคุมเครื่องใช้ไฟฟ้าเป็นเทคโนโลยีอีกทางที่ใช้ควบคุมเครื่องใช้ไฟฟ้าต่างๆเครื่องใช้ไฟฟ้าเป็นสิ่งจำเป็นสำหรับมนุษย์ที่ใช้ในชีวิตประจำวันเพื่อป้องกันความเสียหายจากการใช้งานเป็นเวลานานหรือไฟฟ้าลัดวงจรดังนั้นจึงได้จัดทำโครงงานสิ่งประดิษฐ์เครื่องควบคุมเครื่องใช้ไฟฟ้าเปิด-ปิดแบบอัตโนมัติเพื่อนำไปใช้ในความปลอดภัยของผู้ใช้โดยเครื่องจะทำการกำหนดเวลาการทำงานของเครื่องใช้ไฟฟ้าและจะปิดเครื่องใช้ไฟฟ้าตามเวลาที่กำหนดเป็นสิ่งที่มีประโยชน์อย่างยิ่งจะนำมาใช้เพื่อความปลอดภัยต่อผู้ใช้เครื่องใช้ไฟฟ้าเป็นเวลานาน

# กิตติกรรมประกาศ

โครงงานสิ่งประดิษฐ์นี้สำเร็จได้ด้วยความกรุณาของผู้อำนวยการโรงเรียนอิเล็กทรอนิกส์ กองวิทยาการกรมอิเล็กทรอนิกส์ทหารเรือและครูที่ปรึกษาซึ่งได้ให้คำที่ปรึกษา ข้อชี้แนะ และความช่วยเหลือต่างๆ ที่เป็นประโยชน์จนกระทั่งโครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้ ขอกราบขอบพระคุณอาจารย์ที่ปรึกษาที่ให้ความกรุณาในการแก้ไขข้อบกพร่องต่างๆ ของโครงงานและให้ความรู้ให้คำแนะนำ และให้กำลังใจ สุดท้ายนี้คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงงานนี้เป็นประโยชน์ต่อการศึกษาและเป็นที่น่าสนใจสำหรับผู้ที่สนใจต่อๆไป

## คณะผู้จัดทำ

นรจ.เอกกมล ต่อมสุวรรณ์

นรจ.นิติธร พวงบุษบา

นรจ.ธนายุทธ ป้อมแก้ว

นรจ.ชัยวัฒน์ ปานนนท์

นรจ.ชนนันท์ โรจนวิภาต

นรจ.พงศธร อิ่มรอ

# สารบัญ


บทคัดย่อ ก

กิตติประกาศ ข

บทที่ 1 บทนำ 7

1.1 ที่มาและความสำคัญ

1.2 วัตถุประสงค์

1.3 ขอบเขตงานและโครงงาน

1.4 ประโยชน์ของโครงงาน 8

1.5 ประโยชน์ของโครงงานที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 9

2.1 ความรู้เกี่ยวกับจอ LCD เบื้องต้น

2.1.1 เทคโนโลยีที่พัฒนามาใช้กับ LCD 2 ประเภท

2.1.2เทคโนโลยี TFT LCD Monitor

2.1.3 ข้อดีของจอภาพแบบ LCD 13

2.1.4 ข้อเสียของจอภาพแบบ LCD

2.2 ความรู้เกี่ยวกับ ESP32 14

2.2.1 เซ็นเซอร์ต่าง ๆ ของ ESP32 15

2.2.2 ขาใช้งานต่าง ๆ ของ ESP32

2.2.3 ฟังก์ชั่นเกี่ยวกับความปลอดภัยต่าง ๆ 16

2.2.4 ในด้านประสิทธิ์ภาพการใช้งาน ตัว ESP32

2.2.5 บอร์ดพัฒนา ESP32

2.3 ความรู้เกี่ยวกับ รีเลย์ (RELAY) 20

2.3.1 ประเภทของรีเลย์

2.3.2 ชนิดของรีเลย์

2.3.3 ความรู้ทั่วไปเกี่ยวกับรีเลย์

2.3.4 ประโยชน์ของรีเลย์ 22

2.3.5 คุณสมบัติที่ดีของรีเลย์

2.4 Blynk Application 23

2.5 Power Supply 26

2.6 Real Time Clock 27

บทที่ 3 วิธีการดำเนินงาน 28

3.1 แผนการดำเนินงาน 29

3.2 ตารางวัสดุอุปกรณ์ 30

3.3BLOCK DIAGRAM การทำงาน 34

3.4 ขั้นตอนการดำเนินงาน 35

3.5 โค้ดการทำงาน 42

บทที่ 4 ผลการทดลอง 56

บทที่ 5 สรุปปัญหาและข้อเสนอแนะ 57

ประวัติคณะผู้จัดทำโครงงาน 58

# บทที่ 1

# บทนำ

## 1.1ที่มาและความสำคัญ

โครงงานสิ่งประดิษฐ์ชิ้นนี้ผู้จัดทำได้นำความรู้ที่ได้รับจากการเรียนการสอนในแต่ละวิชาทั้ง ๒ ชั้นปี รวมทั้งความรู้จากการค้นคว้าเพิ่มเติมมาบูรณาการเพื่อสร้าง ระบบควบคุมเครื่องใช้ไฟฟ้าอัตโนมัติ โดยมีวัตถุประสงค์เพื่อนำความรู้ที่เรียนมาสู่การปฏิบัติงานจริงและเป็นการควบคุมเครื่องใช้ไฟฟ้าที่เวลานานให้อยู่ในเวลาที่กำหนดเพื่อให้ผู้ใช้มีความปลอดภัยต่อเครื่องใช้ไฟฟ้าเป็นเวลานาน

ผู้จัดทำจึงนำเอาการควบคุมการเปิดปิดจากระบบเครือข่ายอินเตอร์เน็ตโดยใช้โทรศัพท์มือถือเพื่อสามารถควบคุมการเปิดปิดได้ในระยะไกลรวมเข้าด้วยกันเพื่อจะทำโครงงานสิ่งประดิษฐ์นี้ขึ้นโดยการใช้ Application blynk โดยการเขียนชุดคำสั่งผ่านบอร์ด ESP32

## 1.2 วัตถุประสงค์

1. เพื่อควบคุมการทำงานของเครื่องใช้ไฟฟ้า
2. เพื่อความปลอดภัยในการใช้เครื่องใช้ไฟฟ้า
3. เพื่อประหยัดเวลา
4. เพื่อลดค่าใช้จ่าย

## 1.3 สมมุติฐาน

ระบบควบคุมเครื่องใช้ไฟฟ้าเปิด-ปิดแบบอัตโนมัติ สามารถรับไฟ 220V AC เข้ามาเพื่อรอการสั่งให้เปิดปิดจากหน้าตู้ Control และผ่าน Mobile Phone โดยจะแบ่งโหมดการทำงานออกเป็น 2 โหมด คือ

1) manual mode

2) Auto mode

1 manual mode จะมีการสั่งใช้งานคือ เปิดปิดที่หน้าจอ Display ของตู้ Control และสามารถเปิดปิดที่ปุ่มกดที่หน้าตู้ Control ได้ในกรณีหน้าจอใช้ไม่ได้

2 Auto mode จะเป็นการเปิดปิดผ่านทาง Application Blynk สามารถสั่งการได้ในระยะไกล ในกรณีที่ไม่มีคนอยู่ในบ้านหรืออาคาร สามารถตั้งเวลาเปิดปิดได้

## 1.4 ขอบเขตโครงงาน

1. ควบคุมเครื่องใช้ไฟฟ้าภายในบ้าน
2. ควบคุมเครื่องใช้ไฟฟ้าที่มีการใช้งานบ่อยๆ
3. ควบคุมเครื่องใช้ไฟฟ้าที่มีกระแสตั้งแต่ 250W – 500W

## 1.5 ประโยชน์ของโครงงานที่คาดว่าจะได้รับ

1. ควบคุมการทำงานของเครื่องใช้ไฟฟ้า
2. ป้องกันการเกิดการลัดวงจร
3. ลดค่าใช้จ่าย
4. พัฒนาความริเริ่มสร้างสรรค์
5. นำความรู้ที่ได้ไปใช้จริง
6. ฝึกวางแผนการทำงาน

# บทที่ 2

# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

## 2.1 ความรู้เกี่ยวกับจอ LCD เบื้องต้น

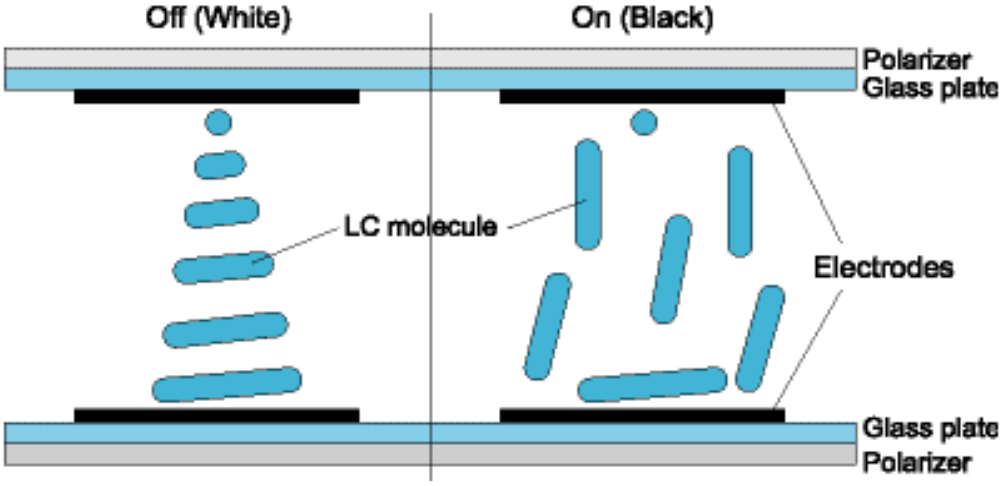
เทคโนโลยีมอนิเตอร์ LCD ย่อมาจาก Liquid Crystal Display ซึ่งเป็นจอแสดงผลแบบ (Digital ) โดยภาพที่ ปรากฏขึ้นเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพ (Black Light) ผ่านชั้นกรองแสง (Polarized filter) แล้ววิ่งไปยัง คริสตัลเหลวที่เรียงตัวด้วยกัน 3 เซลล์คือ แสงสีแดง แสงสีเขียวและแสงสีน้ำเงิน กลายเป็นพิกเซล (Pixel) ที่สว่างสดใสเกิดขึ้น

### 2.1.1 เทคโนโลยีที่พัฒนามาใช้กับ LCD นั้นแบ่งออกเป็น 2 ประเภทคือ

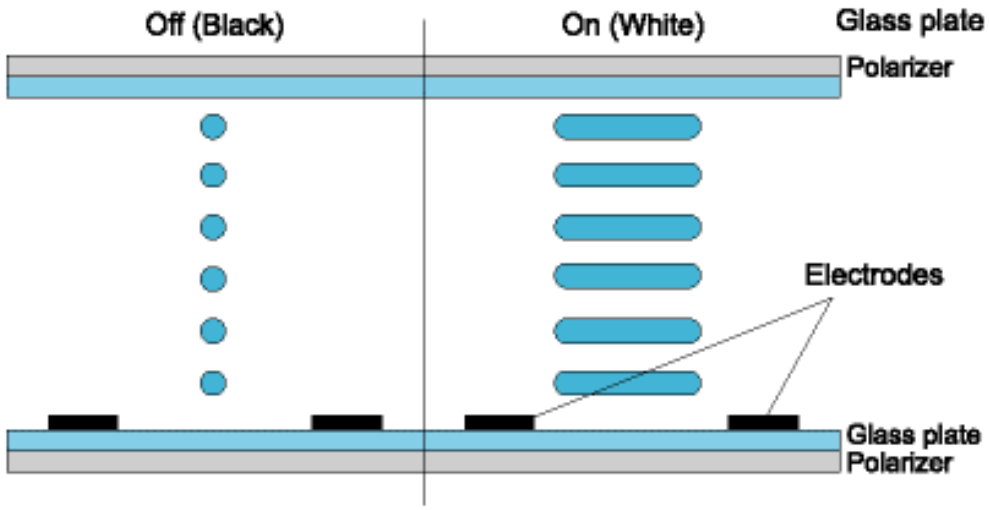
1. Passive Matrix หรือที่เรียกว่า Super-Twisted Nematic (STN) เป็นเทคโนโลยีแบบเก่าที่ให้ความ คมชัดและความสว่างน้อยกว่า ใช้ในจอโทรศัพท์มือถือทั่วไปหรือจอ Palm ขาวดำเป็นส่วนใหญ่
2. Active Matrix หรือที่เรียกว่า Thin Film Transistors (TFT) สามารถแสดงภาพได้คมชัดและสว่าง กว่าแบบแรก ใช้ในจอมอนิเตอร์หรือโน๊ตบุ๊ก

### 2.1.2เทคโนโลยี TFT LCD Mornitor

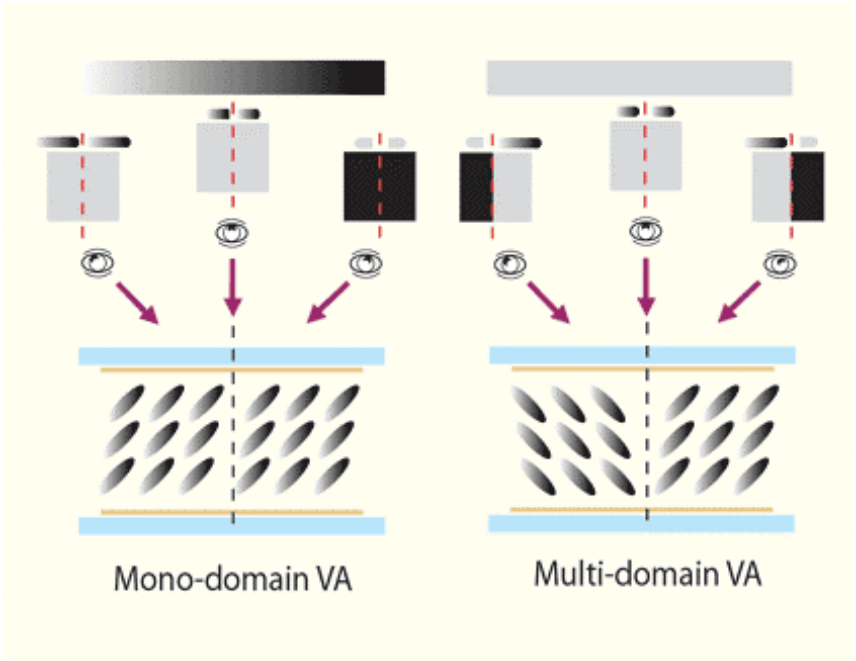
1. TN + Film (Twisted Nematic + Film) Twisted Nematic (TN) คือสารประเภทนี้จะมีการจัดโครงสร้างโมเลกุลเป็นเกลียว แต่ถ้าเราผ่านกระแสไฟฟ้า เข้าไปมันก็จะคลายตัวออกเป็นเส้นตรง เราใช้ปรากฏการณ์นี้เป็นตัวกำหนดว่าจะให้แสงผ่านได้หรือไม่ได้ Twisted Nematic (TN) ผลึกเหลวชนิดนี้จะให้เราสามารถเปลี่ยนทิศทางการสั่นของคลื่นแสงได้ 90? ถึง 150? คือเปลี่ยนจากแนวตั้งให้กลายเป็นแนวนอน หรือเปลี่ยนกลับกันจากแนวนอนให้เป็นแนวตั้งก็ได้ ด้วยจุดนี้เองทำให้ การค่า Response Time (ค่าตอบสนองสัญญาณเทียบกับเวลา) มีค่าสูง

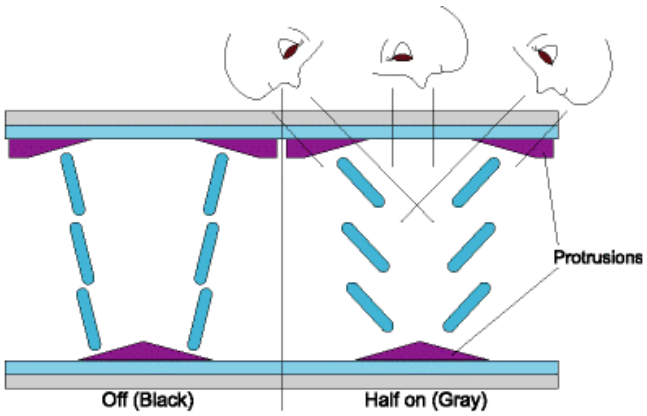


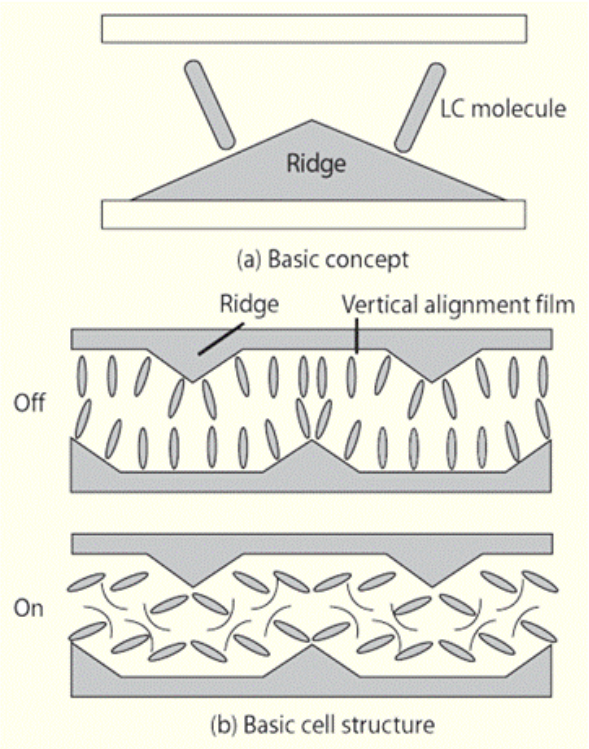
2. IPS (In-Plane Switching or Super-TFT) การจัดโครงสร้างของผลึกจากเดิมที่วางไว้ตามแนวขนานกับแนวตั้ง (เทียบกับระนาบ) เปลี่ยนมาเป็นวางตาม แนวขนานกับระนาบ เรียกจอชนิดนี้ว่า IPS (In-Plane Switching or Super-TFT) จากเดิมขั้วไฟฟ้าจะอยู่คน ละด้านของผลึกเหลวแต่แบบนี้จะอยู่ด้านเดียวกันแปะหัวท้ายเพราะย้ายแนวของผลึกให้ตั้งขึ้น (เมื่อมองจากมุมมอง ของคนดูจอ) เป้าหมายเพื่อออกแบบมาแก้ไขการที่มุมของผลึกเหลวจะเปลี่ยนไปเมื่อมันอยู่ห่างจากขั้วไฟฟ้าออกไป ปัญหานี้ทำให้จอมีมุมมองที่แคบมาก จอชนิด IPS จึงทำให้สามารถมีมุมมองที่กว้างขึ้น แต่ข้อเสียของจอชนิดนี้ก็คือ ต้องใช้ทรานซิสเตอร์สองตัวต่อหนึ่งจุดทำให้เปลืองมาก นอกจานั้นการที่มีทรานซิสเตอร์เยอะกว่าเดิมทำให้แสงจาก ด้านหลังผ่านได้น้อยลง ทำให้ต้องมี Backlite ที่สว่างกว่าเดิม ความสิ้นเปลืองก็มากขึ้นอีกด้วย

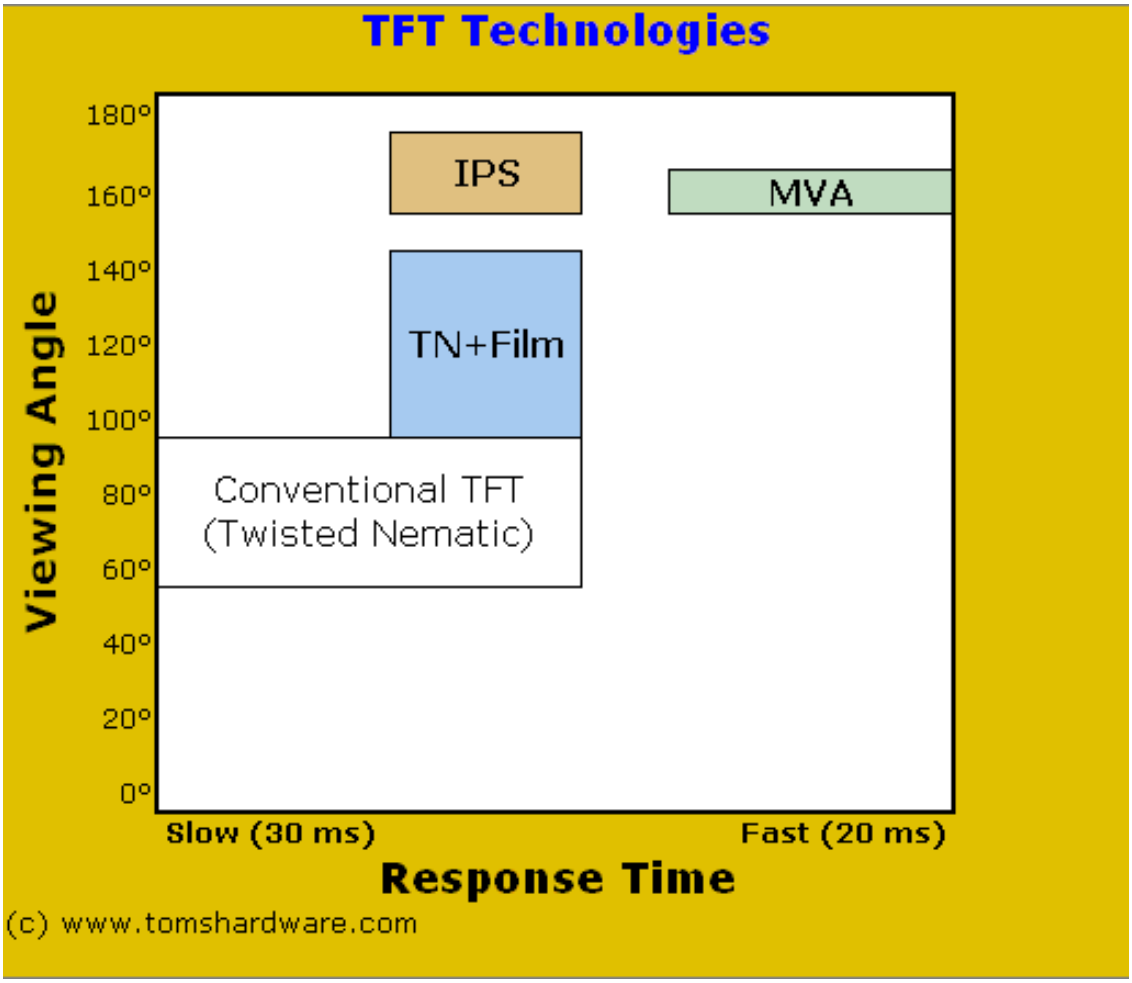



MVA (Multi-Domain Vertical Alignment) บริษัท Fujisu ค้นพบผลึกเหลวชนิดใหม่ที่ให้คุณสมบัติ คือทำงานในแนวระนาบโดยธรรมชาติและต้องการ ทรานซิสเตอร์เพียงตัวเดียวก็สามารถให้ผลลัพธ์เหมือน IPS เลยเรียกว่าว่าชนิด VA (Vertical Align) จอชนิดนี้จะ ไม่ใช้ผลึกเหลวที่ทำงานเป็นเกลียวอีกต่อไป แต่จะมีผลึกเป็นแท่ง ซึ่งปกติถาไม่มีไฟป้อนเข้าไปหาก็จะขวางจอเอาไว้ ทำให้เป็นสีดำ และเมื่อได้รับกระแสไฟฟ้าก็จะตั้งฉากกับจอให้แสงผ่านเป็นสีขาว ทำให้จอชนิดนี้มีความเร็วสูงมาก เพราะไม่ได้คลี่เกลียว แต่ปรับทิศทางของผลึกเท่านั้น จอชนิดนี้จะมีมุมมองได้กว้างราว 160 องศา ปัจจุบันบริษัท Fujisu ได้ออกจอชนิดใหม่คือ MVA (Multi-Domain Vertical Alignment) ออกมาแก้บั๊กตัวเอง คือจากรูจะเห็นว่าด้วยความที่เป็นผลึกแท่ง และองศาของมันใช้กำหนดความสว่างของจุด ดังนั้นเมื่อมองจากมุมมอง อื่นความสว่างของภาพก็จะเปลี่ยนไปเลย เพราะถูกผสมในอีกรูปแบบหนึ่ง จอ Multidomain ก็จะพยายามกระจาย มุมมองให้แต่ละ Pixel นั้นมีผลึกหลายมุมเฉลี่ยกันไป ทำให้ผลกระทบจากการกระมองมุมที่ต่างออกไปหักล้าง กันเอง


Mono-domain VA
Multi-domain VA


Protrusions
Off (Black)
Half on (Gray)


LC molecule
Ridge
(a) Basic concept
Ridge
Vertical alignment film
Off
On
(b) Basic cell structure


TFT Technologies
180°
160°
140°
120°
100°
80°
60°
40°
20°
0°
Viewing Angle
IPS
MVA
TN+Film
Conventional TFT
(Twisted Nematic)
Slow (30 ms)
Fast (20 ms)
Response Time
(c) www.tomshardware.com

### 2.1.3 ข้อดีของจอภาพแบบ LCD ก็คือ

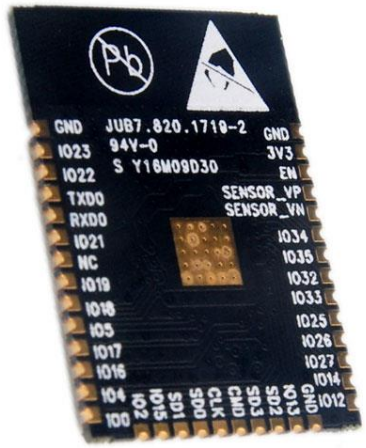
1. อาการภาพค้างติดหน้าจอ (Burn-in) จะไม่มีโอกาสเกิดขึ้นได้เลย
2. จอภาพแบบ LCD นั้นใช้กระแสไฟฟ้าน้อยกว่าจอภาพชนิดอื่น
3. ให้สีที่สว่างสดใส เหมาะกับการแสดงสีกราฟฟิก เช่น การ์ตูน สารคดี ละคร เป็นต้น
4. สามารถนำไปใช้เป็นจอมอนิเตอร์ของคอมพิวเตอร์ได้ดี
5. เหมาะสำหรับใช้ในห้องที่สว่างสูง เช่น บริเวณห้องนั่งเล่นหรือห้องรับแขก

### 2.1.4 ข้อเสียของจอภาพแบบ LCD คือ

1. ไม่สามารถแสดงภาพเคลื่อนไหวเร็ว ๆ ได้ดี เนื่องจากมีความเร็วในการเปลี่ยนสีจอ (Response Time) เร็วที่สุดเพียงแค่ 2 ไมโครวินาที เท่านั้น
2. มีความคลาดเคลื่อนของสีเกิดขึ้น โดยเฉพาะสีแดง โทนสีผิว สีท้องฟ้า ทะเล
3. ไม่สามารถแสดงสีดำสนิทได้ เนื่องจากไฟสว่างหน้าจอจะเปิดตลอดเวลาในขณะที่เครื่องทำงาน ทำให้มีแสงขาวเล็ดลอดออกไปในฉากที่เป็นสีดำจึงทำให้ฉากสีดำนั้นกลายเป็นดำสว่างไม่ใช่ดำมืดอย่างที่มันควรจะเป็น

**2.2ความรู้เกี่ยวกับ ESP32**

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




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 โดยตัวไอซี ESP32 มีสเปคโดยละเอียด ดังนี้

- ซีพียูใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz
- มีแรมในตัว 512KB
- รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB
- มาพร้อมกับ WiFi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ Wi-Fi direct
- มีบลูทูธในตัว รองรับการใช้งานในโหมด 2.0 และโหมด 4.0 BLE
- ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V
- ทำงานได้ที่อุณหภูมิ -40°C ถึง 125°C

### 2.2.1 นอกจากนี้ ESP32 ยังมีเซ็นเซอร์ต่าง ๆ มาในตัวด้วย ดังนี้

- วงจรกรองสัญญาณรบกวนในวงจรขยายสัญญาณ
- เซ็นเซอร์แม่เหล็ก
- เซ็นเซอร์สัมผัส (Capacitive touch) รองรับ 10 ช่อง
- รองรับการเชื่อมต่อคลิสตอล 32.768kHz สำหรับใช้กับส่วนวงจรนับเวลาโดยเฉพาะ

### 2.2.2 ขาใช้งานต่าง ๆ ของ ESP32 รองรับการเชื่อมต่อบัสต่าง ๆ ดังนี้

- มี GPIO จำนวน 32 ช่อง
- รองรับ UART จำนวน 3 ช่อง
- รองรับ SPI จำนวน 3 ช่อง
- รองรับ I2C จำนวน 2 ช่อง
- รองรับ ADC จำนวน 12 ช่อง
- รองรับ DAC จำนวน 2 ช่อง
- รองรับ I2S จำนวน 2 ช่อง
- รองรับ PWM / Timer ทุกช่อง
- รองรับการเชื่อมต่อกับ SD-Card

### 2.2.3นอกจากนี้ ESP32 ยังรองรับฟังก์ชั่นเกี่ยวกับความปลอดภัยต่าง ๆ ดังนี้

- รองรับการเข้ารหัส WiFi แบบ WEP และ WPA/WPA2 PSK/Enterprise
- มีวงจรเข้ารหัส AES / SHA2 / Elliptical Curve Cryptography / RSA-4096 ในตัว

### 2.2.4ในด้านประสิทธิ์ภาพการใช้งาน ตัว ESP32 สามารถทำงานได้ดี โดย

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

เมื่อใช้การเชื่อมต่อผ่านโปรโตคอล UDP จะสามารถรับ – ส่งข้อมูลได้ที่ความเร็ว 135Mbps

ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA

ด้วยเหตุนี้ ESP32 จึงเหมาะสำหรับนำมาใช้งานมาก ด้วยเหตุผลทางด้านราคา และประสิทธิ์ภาพที่ได้

### 2.2.5บอร์ดพัฒนา ESP32

หลังจากชิปไอซี ESP32 เปิดตัวได้ไม่นาน ก็มีผู้ผลิตหลายรายที่ให้การตอบรับโดยการผลิตบอร์ดพัฒนา ESP32 ออกมาช่วยให้ ESP32 สามารถนำมาพัฒนาได้ง่ายมากขึ้น ในแต่ละบอร์ดก็จะมีความแตกต่างกันในเรื่องของฟิเจอร์ ที่เพิ่มเติม เรื่องขนาด และรุ่นของโมดูล ESP32 ที่เลือกใช้ ทำให้บอร์ดพัฒนา ESP32 เกิดขึ้นมาใหม่เสมอ

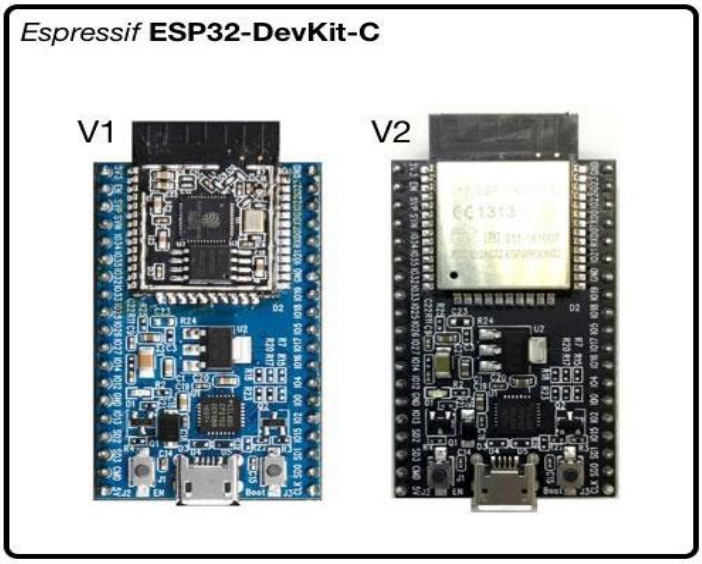
DevKitC ESP32

บอร์ดพัฒนา ESP32 ที่ผลิตโดยบริษัท Espressif ที่เป็นผู้ผลิตชิปไอซี ESP32 (ผู้ผลิตไอซี ผลิตโมดูล และผลิตบอร์ดพัฒนาเอง) เป็นบอร์ดพัฒนา ESP32 บอร์ดแรก ทำให้ตำแหน่งขาต่าง ๆ ของ DevKitC ESP32 ถูกใช้เป็นมาตรฐานของบอร์ดพัฒนา ESP32 ของบริษัท อื่นๆ อีกด้วย

บอร์ดพัฒนา DevKitC ESP32 มีอยู่ด้วยกัน 2 รุ่น คือ

V1 - บอร์ดสีฟ้า เป็นรุ่นแรก

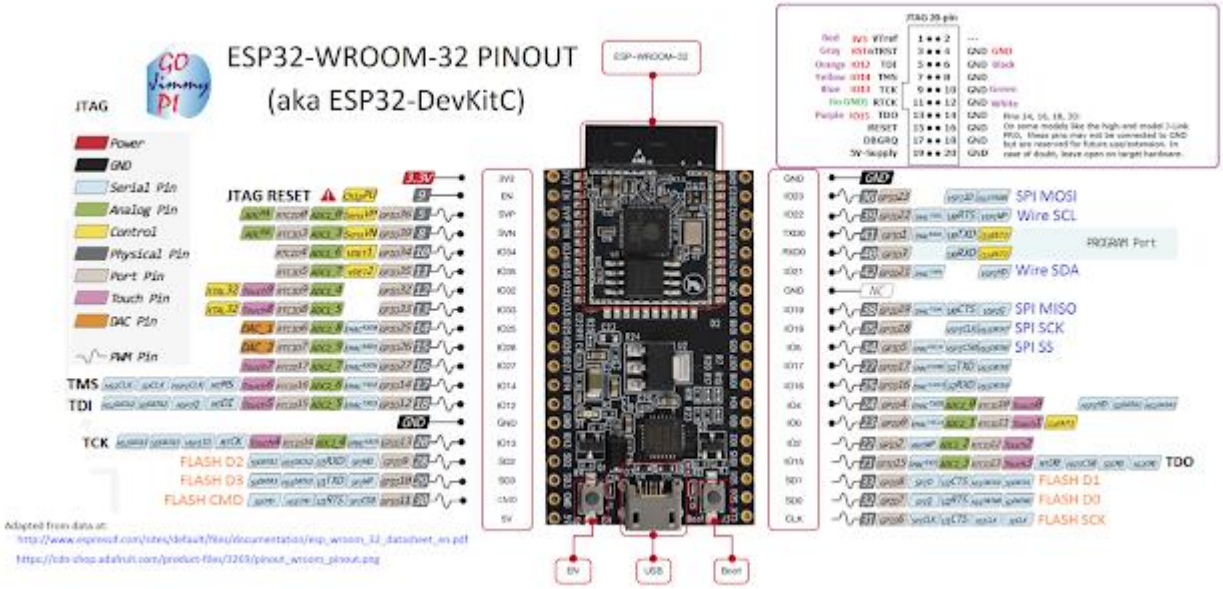
V2 – เปลี่ยนสีของแผ่นวงจรพิมพ์เป็นสีดำ




บอร์ด DevKitC ESP32 ใช้โมดูลรุ่น ESP-WROOM-32 ที่บริษัท Espressif เป็นผู้ผลิตขึ้นเอง และใช้ชิปไอซีแปลง USB เป็น UART เบอร์ CP2102 จาก Silicon Labs สามารถเข้าโหมดอัพโหลดโปรแกรมแบบอัตโนมัติโดยใช้วงจรแบบ nodemcu มีรอม 4MB (หรือ 32Mbit) ใช้ไอซีเรกกูเลเตอร์แบบ LDO เบอร์ AMS1117-3.3 รองรับแรงดันเข้าสูงสุด 7V และจ่ายกระแสได้สูงสุด 700mA

ใช้พลังงานไฟฟ้าและสื่อสารผ่านพอร์ต MicroUSB มีขาต่อใช้งานทั้งหมด 38 ขา เรียงขาแบบเดียวกับบนโมดูล ESP-WROOM-32 เมื่อนำไปเสียบลงโพรโต้บอร์ด จะเหลือช่องให้ใช้งานเพียงด้านเดียว 1 ช่อง

การใช้งานขาต่างๆ ของ DevKitC ESP32



Arduino core for ESP32 WiFi chip

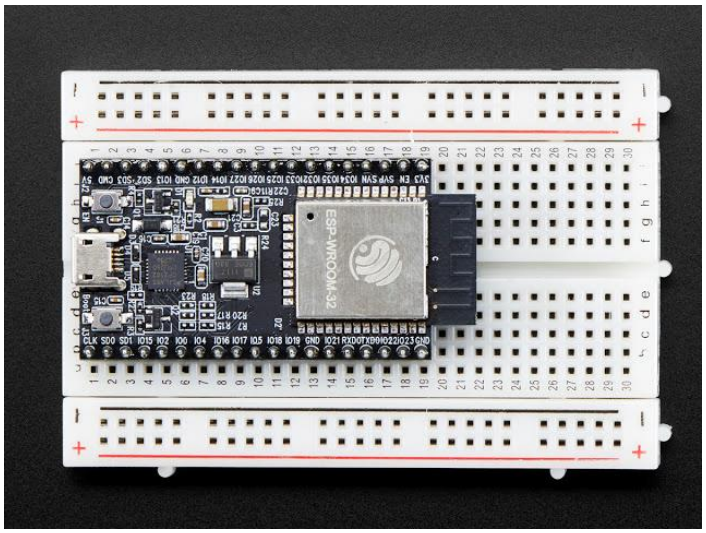
Espressif ได้พัฒนาชุดไลบารี่และคอมไพล์เลอร์สำหรับ Arduino ในชื่อ Arduino core for ESP32 WiFi chip การพัฒนา Arduino core for ESP32 WiFi chip จะทำไปควบคู่กับการพัฒนา ESP-DF โดยที่ ESP-IDF จะเป็นแกนหลัก เมื่อมีการเพิ่มฟิวเจอร์ใหม่ ๆ ให้ ESP-IDF แล้ว จึงจะมีการเพิ่มใน Arduino core for ESP32 WiFi chip

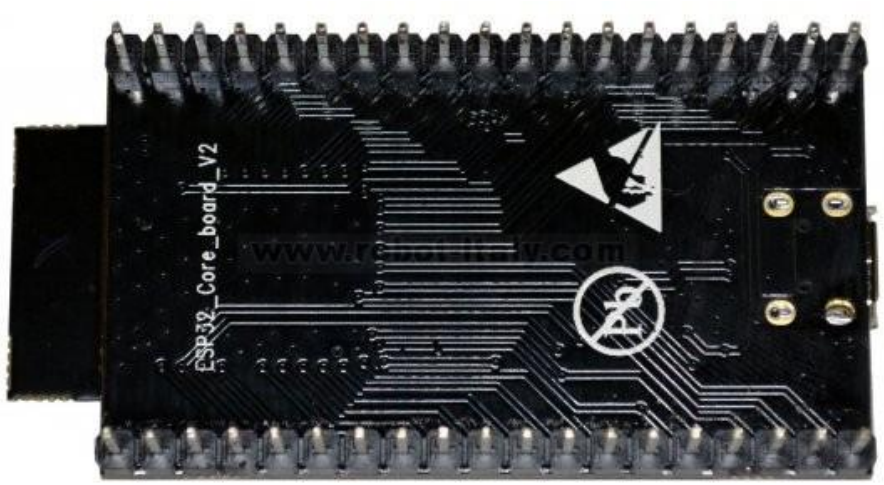
LuaNode

เป็นชื่อของชุดพัฒนา ESP32 ที่นำ Rumtime ของภาษา ภาษา Lua (pronounced 'LOO-ah' หรือออกเสียงว่า 'ลัว' หรือ 'ลูอ่า' ในภาษาโปรตุเกส) มาลงใน ESP32 ทำให้ ESP32 ใช้ภาษา Lua ได้ พัฒนาโดยบริษัท DOIT ที่ทำบอร์ดพัฒนา ESP32 ในชื่อ DOIT ESP32 Development Board มีขาต่อใช้งานทั้งหมด 30 ขา โดยความสามารถของ LuaNode คือรองรับคำสั่งที่ใช้บน Lua จริง ๆ แทบทุกคำสั่ง และรองรับการควบคุม WiFi เต็มรูปแบบ

การเลือกใช้งาน

บทความ โปรเจคหุ่นยนต์ ของ โรบอทสยาม นั้นจะใช้ Arduino core for ESP32 WiFi chip กับ บอร์ด DevKitC ESP32 มีขาต่อใช้งานทั้งหมด 38 ขา หรือ บอร์ด ที่มีตำแหน่งขาต่าง ๆ อ้างอิงจาก DevKitC ESP32 ในการการพัฒนาเป็นหลัก



## 2.3ความรู้เกี่ยวกับ รีเลย์ (RELAY)

### 2.3.1ประเภทของรีเลย์

เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (Solenid) รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าอย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้าแบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power Relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา
2. 2.รีเลย์ควบคุม (control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุม บางทีเรียกกันง่าย ๆ ว่า "รีเลย์"

### 2.3.2ชนิดของรีเลย์

การแบ่งชนิดของรีเลย์สามารถแบ่งได้ 11 แบบ คือ

ชนิดของรีเลย์แบ่งตามลักษณะของคอยล์ หรือ แบ่งตามลักษณะการใช้งาน (Application) ได้แก่รีเลย์ดังต่อไปนี้

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

และกระแสเกิน (Over current)

2. รีเลย์แรงดัน (Voltage relay) คือ รีเลย์ ที่ทำงานโดยใช้แรงดันมีทั้งชนิดแรงดันขาด (Under-voltage) และแรงดันเกิน (Over voltage)
3. รีเลย์ช่วย (Auxiliary relay) คือ รีเลย์ที่เวลาใช้งานจะต้องประกอบเข้ากับรีเลย์ชนิดอื่น จึงจะทำงานได้
4. รีเลย์กำลัง (Power relay) คือ รีเลย์ที่รวมเอาคุณสมบัติของรีเลย์กระแส และรีเลย์แรงดันเข้าด้วยกัน5.รีเลย์เวลา (Time relay) คือ รีเลย์ที่ทำงานโดยมีเวลาเข้ามาเกี่ยวข้องด้วย ซึ่งมีอยู่ด้วยกัน 4 แบบ คือ

5.1 รีเลย์กระแสเกินชนิดเวลาผกผันกับกระแส (Inverse time over current relay) คือ รีเลย์ ที่มีเวลาทำงานเป็นส่วนกลับกับกระแส

5.2 รีเลย์กระแสเกินชนิดทำงานทันที (Instantaneous over current relay) คือรีเลย์ที่ทำงานทันทีทันใดเมื่อมีกระแสไหลผ่านเกินกว่าที่กำหนดที่ตั้งไว้

5.3 รีเลย์แบบดิฟฟินิตไทม์เล็ก (Definite time lag relay) คือ รีเลย์ ที่มีเวลาการทำงานไม่ขึ้นอยู่กับความมากน้อยของกระแสหรือค่าไฟฟ้าอื่นๆ ที่ทำให้เกิดงานขึ้น

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

8.1 รีแอกแตนซ์รีเลย์ (Reactance relay)

8.2 อิมพีแดนซ์รีเลย์ (Impedance relay)

8.3 โมห์รีเลย์ (Mho relay)

8.4 โอห์มรีเลย์ (Ohm relay)

8.5 โพลาไรซ์โมห์รีเลย์ (Polaized mho relay)

8.6 ออฟเซทโมห์รีเลย์ (Off set mho relay)

9. รีเลย์อุณหภูมิ (Temperature relay) คือ รีเลย์ที่ทำงานตามอุณหภูมิที่ตั้งไว
10. รีเลย์ความถี่ (Frequency relay) คือ รีเลย์ที่ทำงานเมื่อความถี่ของระบบต่ำกว่าหรือมากกว่าที่ตั้งไว้
11. บูคโฮลซ์รีเลย์ (Buchholz ‘s relay) คือรีเลย์ที่ทำงานด้วยก๊าซ ใช้กับหม้อแปลงที่แช่อยู่ในน้ำมันเมื่อเกิดฟอลต์ ขึ้นภายในหม้อแปลง จะทำให้น้ำมันแตกตัวและเกิดก๊าซขึ้นภายในไปดันหน้าสัมผัส ให้รีเลย์ทำงาน

## 2.3.3ความรู้ทั่วไปเกี่ยวกับรีเลย์

1.1 หน้าที่ของรีเลย์ คือ เป็นอุปกรณ์ไฟฟ้าที่ใช้ตรวจสอบสภาพการณ์ของทุกส่วน ในระบบกำลังไฟฟ้าอยู่ตลอดเวลาหากระบบมีการทำงานที่ผิดปกติ รีเลย์จะเป็นตัวสั่งการให้ตัดส่วนที่ลัดวงจรหรือส่วนที่ทำงานผิดปกติออกจากระบบทันทีโดยเซอร์กิตเบรกเกอร์จะเป็นตัวที่ตัดส่วนที่เกิดฟอลต์ออกจากระบบจริงๆ

**2.3.4 ประโยชน์ของรีเลย์**

1.ทำให้ระบบส่งกำลังมีเสถียรภาพ (Stability) สูงโดยรีเลย์จะตัดวงจรเฉพาะส่วนที่เกิดผิดปกติ ออกเท่านั้น ซึ่งจะเป็นการลดความเสียหายให้แก่ระบบน้อยที่สุด

2.ลดค่าใช้จ่ายในการซ่อมแซมส่วนที่เกิดผิดปกติ

3.ลดความเสียหายไม่เกิดลุกลามไปยังอุปกรณ์อื่นๆ

4.ทำให้ระบบไฟฟ้าไม่ดับทั้งระบบเมื่อเกิดฟอลต์ขึ้นในระบบ

**2.3.5 คุณสมบัติที่ดีของรีเลย์**

1.ต้องมีความไว (Sensitivity) คือมีความสามารถในการตรวจพบสิ่งที่ผิดปกติเพียงเล็กน้อยได้

2.มีความเร็วในการทำงาน (Speed) คือความสามารถทำงานได้รวดเร็วทันใจ ไม่ทำให้เกิดความเสียหายแก่อุปกรณ์และไม่กระทบกระเทือนต่อระบบ โดยทั่วไปแล้วเวลา ที่ใช้ในการตัดวงจรจะขึ้นอยู่กับระดับของแรงดันของระบบด้วย

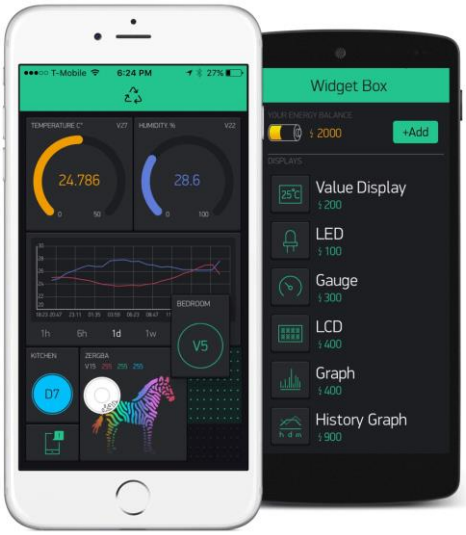
ระบบ 6-10 เควิน จะต้องตัดวงจรภายในเวลา 1.5-3.0 วินาที

ระบบ 100-220 เควิน จะต้องตัดวงจรภายในเวลา 0.15-0.3 วินาที

ระบบ 300-500 เควิน จะต้องตัดวงจรภายในเวลา 0.1-0.12 วินาที

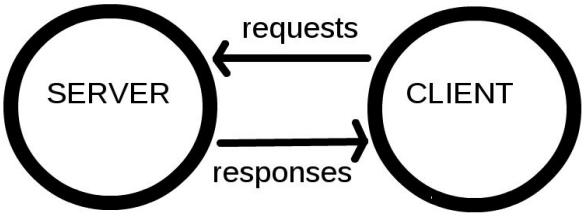
## 2.4 Blynk Application

Blynk คือ Application สำเร็จรูปสำหรับงาน IOT มีความน่าสนใจคือการเขียนโปรแกรมที่ง่าย ไม่ต้องเขียน App เองสามารถใช้งานได้อย่าง Real time สามารถเชื่อมต่อ Device ต่างๆเข้ากับ Internet ได้อย่างง่ายดาย ไม่ว่าจะเป็น Arduino, Esp8266, Esp32, Nodemcu,Rasberry pi นำมาแสดงบน Application ได้อย่างง่ายดาย แล้วที่สำคัญ Application Blynk ยังฟรี และ รองรับในระบบ IOS และ Android อีกด้วย เป็นอะไรที่น่าสนใจมากๆใช่ไหมครับ คราวนี้เรามาเริ่มกันเลย



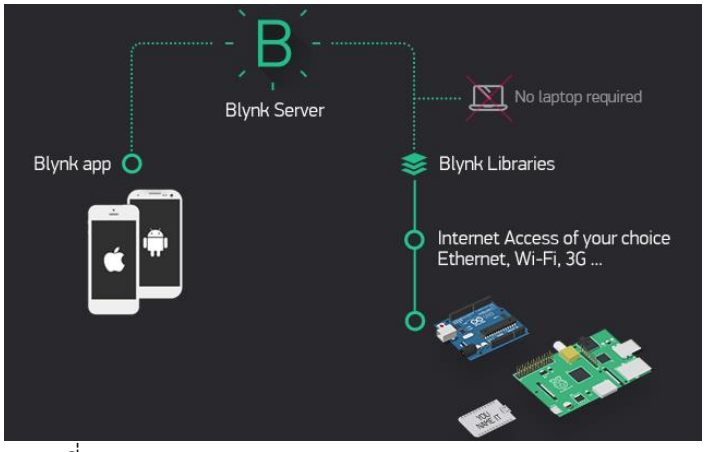

ตัวอย่าง App Blynk ที่มา : //www.ab.in.th/

ในยุคสมัยก่อน การเขียนโปรแกรมเชื่อมต่อกันระหว่าง อุปกรณ์ 2 ชิ้นเข้าด้วยกันมักจะใช้งานในลักษณะของ Server >>> Client ทำให้เกิดข้อจำกัดต่างๆมากมาย ยกตัวอย่าง เราต้องการเปิดปิดไฟ ผ่านหน้าเว็บ เราก็จะให้ Arduino เป็น Server และ เครื่องคอมพิวเตอร์ (Client) เป็นเครื่องลูก ข้อจำกัดที่เกิดขึ้นคือทรัพยากร เช่น CPU RAM ROM ของเราอาจจะไม่พอ มักจะเจอปัญหาเอ๋อบ่อย ค้างไปดื้อๆ ก็มี ทำให้การเขียนโปรแกรมเป็นไปได้ยาก ต้องประหยัดทรัพยากรให้ได้มากที่สุดเพื่อจะให้สามารถทำงานได้ และการเซ็ต Netword เป็นไปได้ยาก ส่วนใหญ่มักจะใช้ในวง Lan หรือถ้าต้องการ ควบคุมผ่าน Wan จะต้อง Forword Set ระบบ Network จนปวดหัว




ภาพที่ 2 การเชื่อมต่อแบบ Server to Client : //www.ab.in.th/

ต่อมาเป็นยุคของ Cloud เกิดขึ้น บวกกับมี Chip Wifi ราคาถูก Esp8266 ถูกผลิตขึ้นมา แต่ด้วยข้อจำกัดทางด้านทรัพยากร จึงมีวิธีการคิดว่า ถ้านำข้อมูลไปใส่ลงใน Server เลยละแล้วให้ Device ของเราเรียกเข้าไปแก้ไข หรืออ่านข้อมูลโดยตรง ทำให้ความฉลาดของตัวอุปกรณ์ของเราไม่มีวันสิ้นสุดหมดข้อจำกัดหลายอย่าง Device กลายเป็นแค่ตัวรับ Data และส่ง Data มาแสดงเท่านั้น ทำให้ Chip Esp8266 จึงได้รับความนิยมในปัจจุบัน




ภาพที่ 3 ภาพรวมของระบบ Network Blynk : //www.ab.in.th/

วิธีการทำงานของ Blynk เริ่มจาก อุปกรณ์ เช่น Arduino esp8266 Esp32 Rasberry Pi เชื่อมต่อไปยัง Server ของ Blynk โดยตรง สามารถรับส่งข้อมูลหากันได้

คอมพิวเตอร์ Smartphone ก็จะเชื่อมต่อกับ Server ของ Blynk โดยตรง กลายเป็นว่า มี Server เป็นสะพานให้เชื่อต่อหากันจึงหมดปัญหาและข้อจำกัดทุกอย่างทำให้อุปกรณ์ของเรามีความฉลาดมากขึ้น

การออกแบบในลักษณะ ภาพที่ 3 เป็นที่นิยมมากในปัจจุบัน เพราะไม่จำเป็นต้อง Set อุปกรณ์ Network ต่างๆ ให้ปวดหัว

พูดถึงระบบไปแล้วคราวนี้เราจะมาดูความสามารถของ Application Blynk ดูบ้างว่าสามารถทำอะไรได้บ้าง

ภาพที่ 4 ข้อมูลหน้าจอที่จะมาแสดงใน App ของเรา: //www.ab.in.th/

จากภาพที่ 4 เราสามารถเลือก หน้าจอของภาพ คำอธิบาย เกจ์วัดต่างๆ ก็สามารถออกแบบได้เองได้อย่างอิสระอีกด้วย ต้องการอะไรไม่ต้องการอะไรเราสามารถเลือกได้ตาทความของเราได้เลย

ภาพที่ 5 หน้าจอของ App ที่เราได้ทำการสร้างขึ้นมา: //www.ab.in.th/

จากภาพที่ 5 เราสามารถจับลากจัดเลียงปรับขนาดให้เหมาะสมตรงตามความต้องการของเรา

## 2.5 Power Supply

Power Supply คือ อุปกรณ์จ่ายไฟให้กับคอมพิวเตอร์ มีหน้าที่แปลงกระแสไฟฟ้าจากกระแสสลับขนาด 220V เป็นกระแสสตรง 5V,12V ให้กับอุปกรณ์ต่างๆ โดยผ่านทางเมนบอร์ด เพื่อเลี้ยงไฟให้กับอุปกรณ์ต่างๆ

เช่น VGA Card Harddisk , ช่อง DVD รวมไปถึงพัดลมระบายความร้อนตัวใหญ่ๆ เป็นต้น ดังนั้น Power Supply จึงมีความจำเป็นและมีหน้าที่สำคัญเอามากๆ ในเครื่องคอมพิวเตอร์นั้นเอง

หาก Power Supply ให้กำลังไฟไม่พอจะเกิดอะไรขึ้น ?

หลายคนก็เคยเจอว่าเสียบ Flash Drive แล้วเครื่องมองไม่เห็น หรือเสียบอุปกรณ์ USB เยอะๆแล้วบางอย่างใช้งานไม่ได้ สาเหตุจากไฟเลี้ยงจาก Power Supply ไม่เพียงพอ หรือหากแยกว่านั้นคอมพิวเตอร์ก็อาจจะมีอาการผิดปกติเช่นชอบรีสตาร์ตัวเองบ่อยๆเป็นต้น

## 2.6 Real Time Clock

ว่าไปแล้วโมดูล Real Time Clock (RTC) ก็คือ อุปกรณ์ที่ให้ค่าเวลาตามจริง ซึ่งทำงานโดยการจับสัญญานนาฬิกาที่ได้มาจาก Crystal นั่นเองครับ บางรุ่นก็จะมีถ่านสำรองมาให้ด้วย ทำหน้าที่ในการบันทึกเวลาอย่างต่อเนื่องถึงแม้ว่าจะไม่มีไฟเลี้ยงมาที่ตัวบอร์ด ตัวเวลาก็ยังคงนับได้ต่อ ทำให้ไม่ต้องเสียเวลามาตั้งเวลาใหม่หลังจากที่หยุดจ่ายไฟเลี้ยงครับ โมดูล RTC นี้จำเป็นอย่างยิ่งกับการใช้งานที่ต้องมีการบันทึกเวลา (Time Stamp) เช่น อุปกรณ์ Data logger ครับ

หลายคนก็อาจจะสงสัยว่าทำไมเราต้องการ RTC module นี้ในเมื่อ Arduino Board ของเราก็มีตัวจับเวลา เช่น millis() อยู่แล้ว คำตอบก็คือ ไมโครโปรเซสเซอร์ที่เป็นหัวใจในการทำงานของ Arduino board ทั้งหลายนั้น ต้องทำงานสารพัดอย่างครับ ไม่ว่าจะเป็นคำสั่งพื้นฐาน เช่น บวก ลบ คูณ หาร ไปจนถึงการติดต่อกับอปุกรณ์ภายนอก คำสั่งที่มากมายซึ่งเราเป็นผู้เขียนลงใน Sketch นั้นจะทำงานแบบ อนุกรม (Serial) ว่าง่ายๆ ก็คือ ทำทีละบรรทัดครับ ทำให้การทำงานของคำสั่งจับเวลาก็จะถูกรบกวนไปด้วย จับได้บ้าง ไม่ได้บ้าง เดี๋ยวโดนสั่งไปทำโน่น โดยแทรก (Interrupt) ไปทำนี่ เวลาที่ได้จากการใช้คำสั่งนี้ ก็เลยไม่สามารถนำมาเป็นเวลาตามจริงที่ต้องการบันทึกไปพร้อมกับค่าอื่นๆ ที่ต้องการวัดได้ครับ

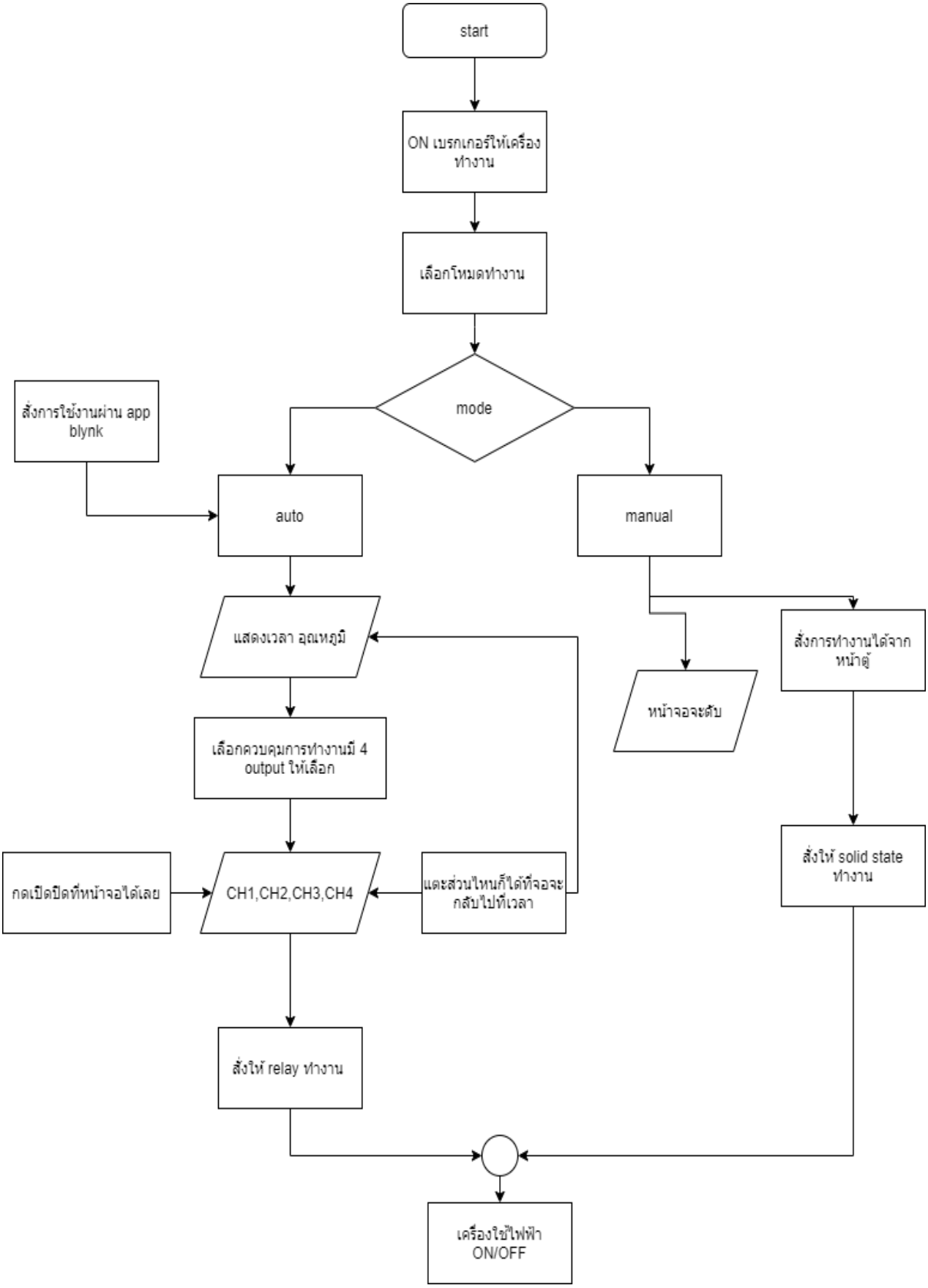
ดังนั้น ในการประยุกต์ใช้กับงานที่ต้องการเวลาที่แม่นยำ และเป็นเวลาตามนาฬิกา ที่บอก วันที่ เดือน ปี ชั่วโมง นาที วินาที ก็เลยจำเป็นต้องใช้อุปกรณ์ที่ทำหน้าที่จับเวลาแยก ซึ่งก็ทำให้ต้องมีสัญญาณนาฬิกาจาก Crystal แยกต่างหากด้วยเช่นกัน ฟังดูแล้วอาจจะคิดว่ามันช่างยุ่งยากเหลือเกิน แต่ไม่เลยครับ มีคนที่ออกแบบ Chip หลายแบบ ที่ทำหน้าที่นี้มาให้เราเรียบร้อยแล้ว วิธีใช้งานก็ง่ายๆ ติดต่อผ่านไปที่บอร์ดโดยใช้การสื่อสารแบบ I2C หรือ Inter - Integrated Circuit ที่ใช้ SDA SCL VCC และ GND เท่านั้นครับ

ทีนี้การที่มันมีหลายรุ่นก็ทำให้เกิดความงงกันได้ โมดูล RTC นั้นมีอยู่หลายแบบครับ เอาเฉพาะที่ร้าน Arduitronics ขายก็มี 3 รุ่นแล้วครับ ได้แก่ DS3231, DS1302, Tiny RTC I2C 24C32 DS1307 แต่ละรุ่นก็จะมีความต่างกันในเรื่องของความละเอียดในการจับเวลา การมีแบตเตอรี่สำรอง ขนาด และแน่นอนก็คือราคาครับ

ตัวที่ผมจะใช้ในการสาธิตการทำงานของโมดูล RTC วันนี้ก็คือ DS3231 ครับ ตัวนี้จะมีราคาสูงกว่าตัวอื่นนิดหน่อย (แต่ก็ไม่กี่สิบบาทหรอกครับ) แต่ข้อดีของมันก็คือ มีการชดเชยการเปลี่ยนแปลงของสัญญาณนาฬิกา เนื่องจากการเปลี่ยนแปลงของอุณหภูมิแวดล้อมครับ ว่าง่ายๆ ก็คือ เวลาอุณหภูมิเปลี่ยน สัญญาณนาฬิกาจาก Crystal ก็เปลี่ยน ทำให้เวลาก็เพี้ยนไปด้วย แต่โมดูลนี้ได้ทำการวัดค่าอุณหภูมิพร้อมทั้งชดเชยความเปลี่ยนแปลงนี้ไปด้วยแล้ว ทำให้เวลาที่ได้มีความแม่นยำสูงมาก

# บทที่ 3

## วิธีการดำเนินงาน



**3.1 แผนการดำเนินงาน**

แผนงานโครงงานสิ่งประดิบษฐ์

หลักสูตร นรจ. พรรค พศ. เหล่า ยย. (อิเล็กทรอนิกส์-ไฟฟ้า) ชั้นปีที่ 2 ประจำปีการศึกษา 2563

โรงเรียนอิเล็กทรอนิกส์ กองวิทยาการ กรมอิเล็กทรอนิกส์ทหารเรือ

ตารางแผนการดำเนินงาน

| รายการปฏิบัติ | พ.ย. | | | | ธ.ค. | | | | ม.ค. | | | | ก.พ. | | | | มี.ค. | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| นักเรียนเสนอชื่องโครงงาน | | | | | | | | | | | | | | | | | | | | |
| กลั่นกรองโครงงาน | | | | | | | | | | | | | | | | | | | | |
| เสนอรายการวัสดุ จำนวนและราคา | | | | | | | | | | | | | | | | | | | | |
| ค้นขว้าข้อมูล | | | | | | | | | | | | | | | | | | | | |
| นักเรียนจัดทำเอกสารเสนอขออนุมัติจัดทำโครงงาน | | | | | | | | | | | | | | | | | | | | |
| เสนอ รร.อล. ขออนุมัติจัดทำโครงงาน | | | | | | | | | | | | | | | | | | | | |
| ดำเนินการจัดทำโครงงาน | | | | | | | | | | | | | / | / | / | / | / | / | | |
| ฝึกนำเสนอโครงงาน | | | | | | | | | | | | | | | | | | | | |
| ส่งชิ้นงานและเอกสารโครงงาน | | | | | | | | | | | | | | | | | | | | |
| จัดทำบอร์ดนิทรรศการโครงงาน | | | | | | | | | | | | | | | | | | | | |
| จัดนิทรรศการโครคงงาน | | | | | | | | | | | | | | | | | | | | |

## 3.2ตารางวัสดุ

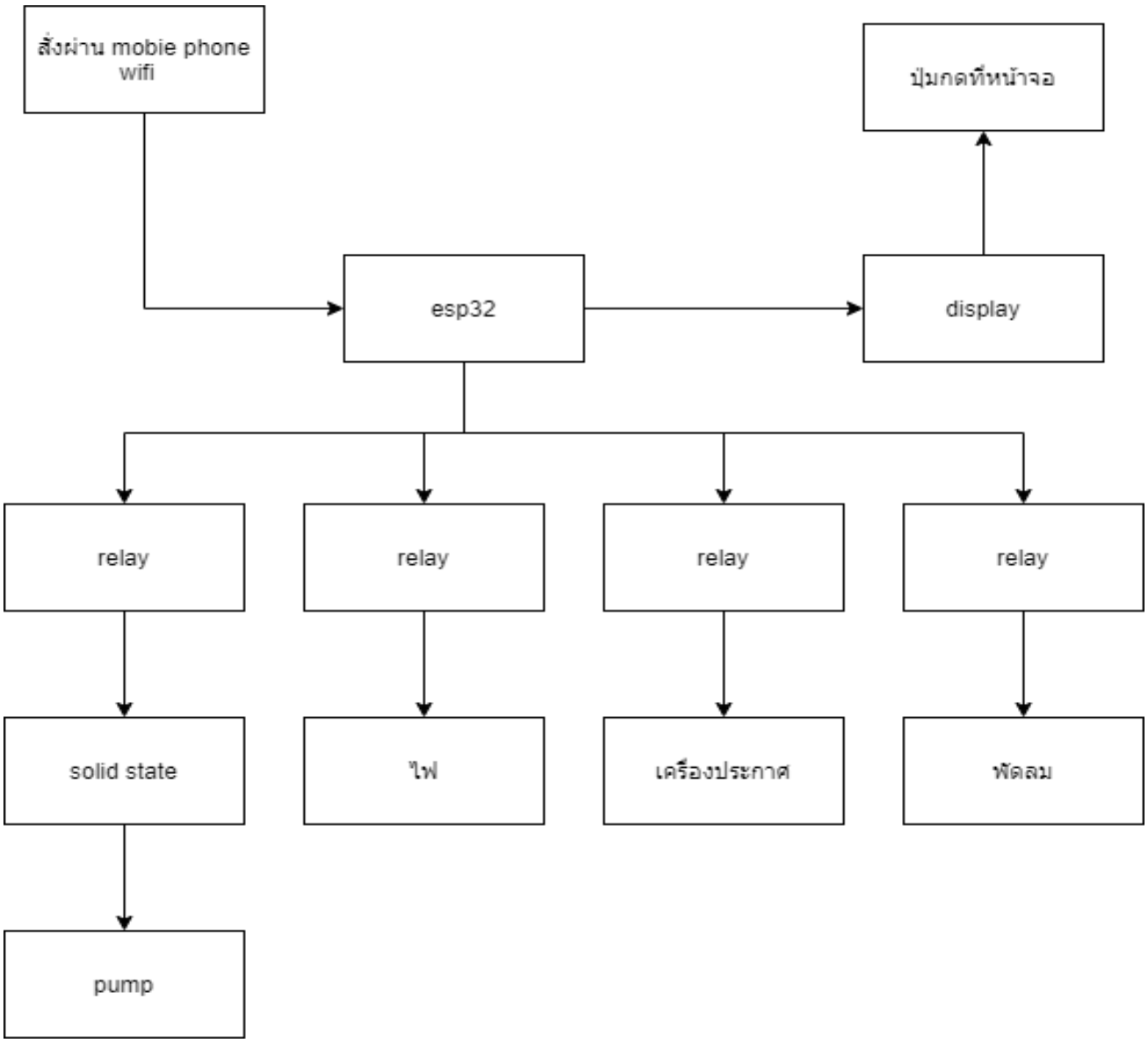
| ลำดับ | รายการ | จำนวน | ราคาต่อหน่วย | ราคารวม | รูปภาพประกอบ |
|---|---|---|---|---|---|
| 1 | ไม้ใช้ | 2 | 40 | 80 | |
| 2 | เคเบิ้ลทาย 8N | 1 | 65 | 65 | |
| 3 | หางปลาแฉก | 1 | 120 | 120 | |
| 4 | สายต่อวงจร ผู้เมีย<br>สายต่อวงจร ผู้ผู้<br>สายต่อวงจร เมียเมีย | 1<br>1<br>1 | 65 | 65 | <br> |
| 5 | TB-2512<br>12CH | 2 | 55 | 110 | |
| 6 | ESP32 | 1 | 150 | 150 | |
| 7 | ใส้ไก่ | 1 | 45 | 45 | |

| | | | | | |
|---|---|---|---|---|---|
| 8 | Ammeter | 1 | 250 | 250 | |
| 9 | โวล์มิเตอร์ | 1 | 250 | 250 | |
| 10 | emergency switch | 1 | 59 | 59 | |
| 11 | Pilot lamp | 2 | 9 | 18 | |
| 12 | Pilot lamp | 2 | 9 | 18 | |
| 13 | จอแสดงผล TFT LCD<br>ขนาด 4 นิ้ว | 1 | 495 | 495 | |
| 14 | ตู้ควบคุม | 1 | 250 | 250 | |

| | | | | | |
|---|---|---|---|---|---|
| 15 | พัดลม ขนาด 12 v | 2 | 35 | 70 | |
| 16 | สาย USB | 1 | 19 | 19 | |
| 17 | ซีเล็คเตอร์สวิทช์ | 1 | 125 | 125 | |
| 18 | Push button switch | 1 | 30 | 30 | |
| 19 | Push button switch | 1 | 30 | 30 | |
| 20 | โมดูลดีเล | 1 | 395 | 395 | |
| 21 | สายไฟ | 30 m | 320 | 320 | |

| | | | | | |
|---|---|---|---|---|---|
| 22 | Solid State | 1 | 80 | 80 | |
| 23 | รางเหล็ก | 1 | 30 | 30 | |
| 24 | Delay | 1 | 150 | 150 | |
| 25 | บอด | 1 | 150 | 150 | |
| 26 | รางเก็บสายไฟ | 1 | 45 | 45 | |

# 3.3BLOCK DIAGRAM การทำงาน

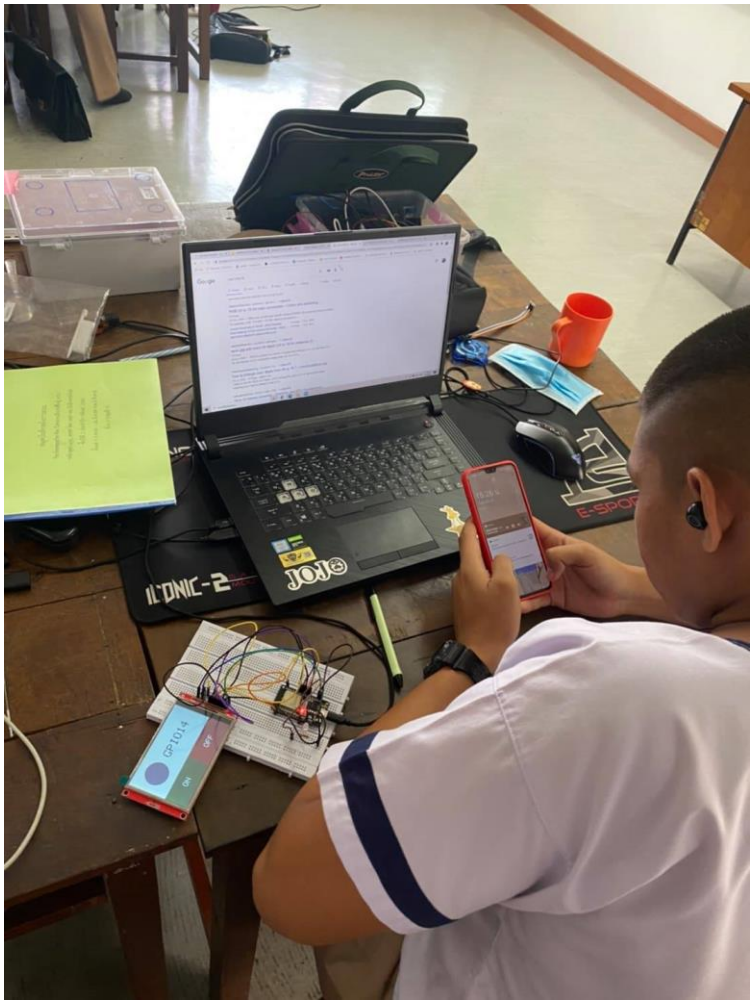


**3.4 ขั้นตอนการดำเนินงาน**

1. **วางแผนออกแบบวงจรควบคุมเครื่องใช้ไฟฟ้าเปิด-ปิดอัตโนมัติ และ จัดเตรียมอุปกรณ์**

**ภาพที่ การวางแผนการออกแบบวงจร**

## 2. ขั้นตอนการเขียนโปรแกรมการควบคุม

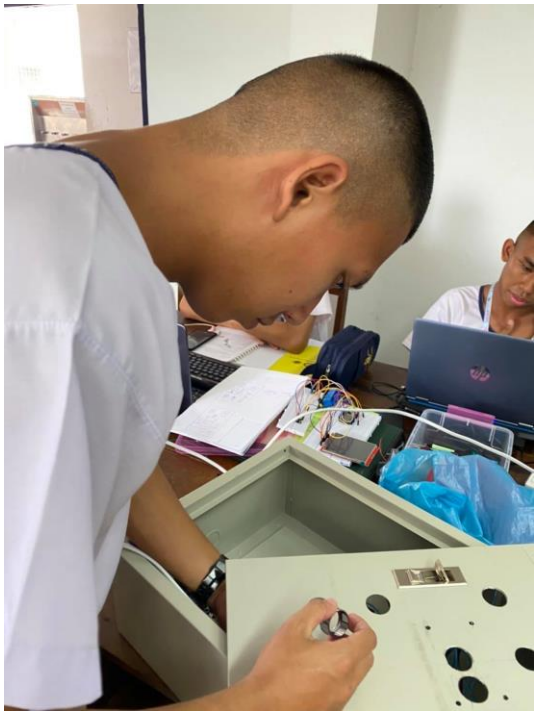


ภาพที่ การเขียนโปรแกรม

3. การเจาะเหล็กชิ้นงาน

ภาพที่ การเจาะรูโดยใช้สว่าน

**4.** การเจาะรูโดยใช้เครื่อง

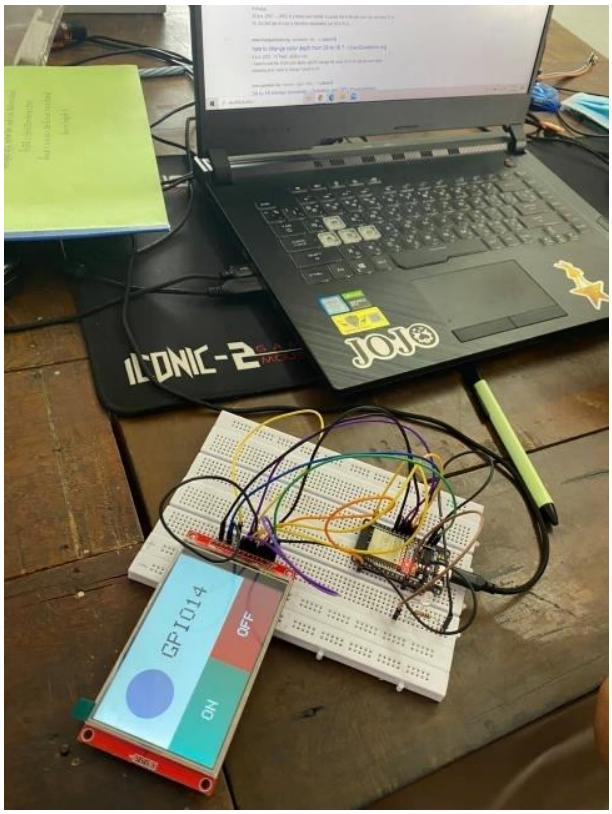


ภาพที่ แสดงการเจาะรูเพื่อใส่อุปกรณ์

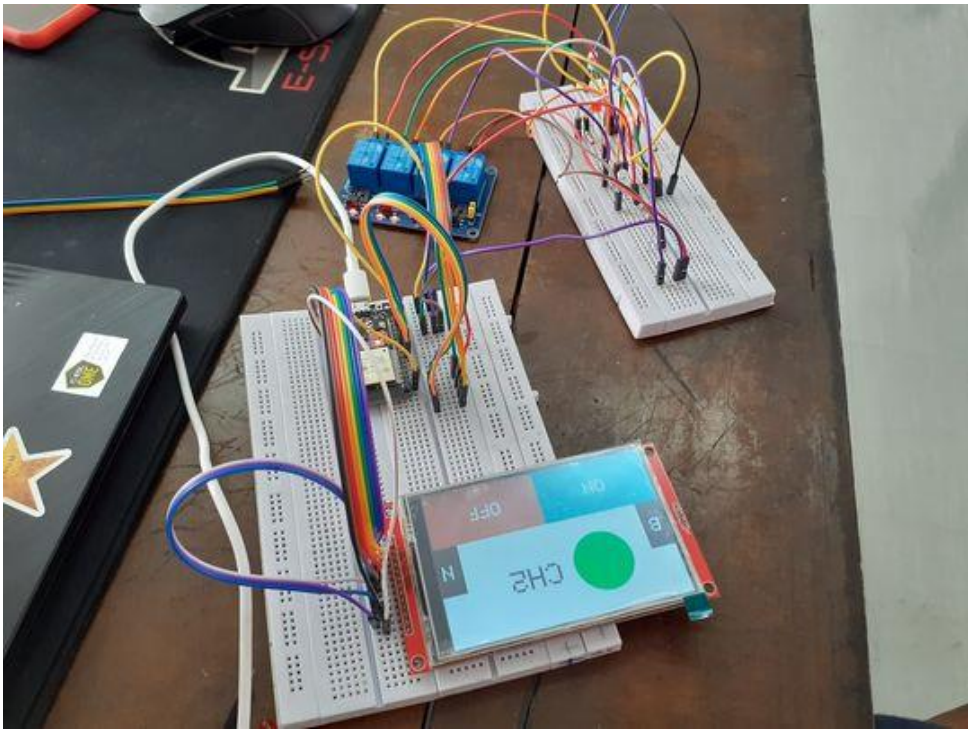
**5.** การประกอบชิ้นโครงงาน

ภาพที่ การประกอบชิ้นโครงงาน

**6.** การทดลองการต่อวงจรครั้งที่ 1



**7.** การทดลองการต่อวงจรครั้งที่ 2



**8.** ใส่อุปกรณ์ในโครงงาน

# โค้ดการทำงาน

```
//////////////////////คำสั่งเพิ่ม Library มาใช้ใน code

#define BLYNK_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <TFT_eSPI.h>

#include <SPI.h>

#include "RTClib.h"




char auth[] = "hclBWHuwCgaqYmvRQ7q3uiig6E4w0qC7"; //เก็บค่า token ของ Blynk Library

char ssid[] = "vivo"; //เก็บค่า ชื่อของ Acess Point ของ Blynk Library

char pass[] = "123456789"; //เก็บค่า รหัส ของ ของ Acess Point ของ Blynk Library

int x1; // ตัวแปรเก็บค่าที่ได้รับมาจาก แอพ Blynk

int x2;

int x3;

int x4;

int i = 0; //กำหนดค่าเพื่อใช้สำหรับนับ loop switch case

char LED1='b';//เก็บค่าสถานะของ status บนจอ TFT LCD

char LED2='b';

char LED3='b';

char LED4='b';

int checkstate = 0; //กำหนดค่าเพื่อใช้อ้างอิงในการใช้งานฟังก์ชันภายใน void loop


TFT_eSPI tft = TFT_eSPI(); //กำหนดตัวแปรเพื่อใช้กำหนดการสืบทอดคลาส

RTC_DS3231 rtc;
```

```
void setup() { //สั่งการทำงานเพียง 1 ครั้ง
  Serial.begin(115200); //กำหนดการใช้งาน Serial Monitor
  //rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); //กำหนดเพื่อใช้ในการตั้งค่าเวลาของ RTC โดยจะใช้เวลาในการ compile เป็นตัว set เวลา
  Blynk.begin(auth, ssid, pass); //เริ่มการใช้งานติดต่อกับ APP Blynk
  pinMode(14, OUTPUT); //กำหนดขา Output
  pinMode(25, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(27, OUTPUT);
  digitalWrite(14, HIGH);
  digitalWrite(27, HIGH);
  digitalWrite(26, HIGH);
  digitalWrite(25, HIGH);

  tft.init(); //setให้จอ LCD เริ่มทำงาน
  tft.setRotation(1); //set ให้หน้าจอเป็นแนวนอน(1)
  tft.fillScreen(TFT_BLACK);//เติมสีดำให้จอ
  tft.setTextColor(TFT_WHITE, TFT_BLACK);//set ตัวอักษรสีขาว พื้นหลังดำ


  uint16_t calData[5] = { 268, 3640, 249, 3520, 7 };//ค่ากำหนดของทัชสกรีน
  tft.setTouch(calData);


  rtc.begin(); //เริ่ม RTC


}
```

```
void loop() { //คำสั่งวนซ้ำ

 if (checkstate == 0){ // กำหนดให้เข้าเงื่อนไขเมื่อ checkstate=0 ก่อน
  Blynk.run(); // สั่งใช้งานฟังก์ชัน blynk
 timedis();//สั่งใช้งานฟังก์ชันก์ timedis

 }

 if(checkstate==1){// กำหนดให้เข้าเงื่อนไขเมื่อ checkstate=1
 Blynk.run();    // สั่งใช้งานฟังก์ชัน blynk
    switch(i){//คำสั่งเลือกทำเมื่อ i เท่ากับ 1 2 3 หรือ 4
     case 1://เข้าทำงานเมื่อ i = 1

          dis1();//เรียกใช้งานฟังก์ชัน dis1
          break;
     case 2:

          dis2();
          break;
     case 3:

          dis3();
          break;
     case 4:

          dis4();
          break;
   }
```

```
 }

}
void dis1(){ //ฟังก์ชัน dis1 โดยไม่มีรีเทิร์นค่ากลับไป

 tft.drawChar( 240, 100,'C',TFT_WHITE,TFT_BLACK, 6);//สั่งให้วาดตัวอักษรชนิด char บนหน้าจอ(แกนx, แกนy, ตัวอักษร, สีตัวอักษร, สีพื้นหลัง, ขนาดตัวอักษร)
 tft.drawChar( 280, 100,'H',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 320, 100,'1',TFT_WHITE,TFT_BLACK, 6);
 if(LED1 == 'a'){tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E));} //สั่งวาดวงกลมโดยจุดกึ่งกลาง (แกนx, แกนy, ขนาดเส้นผ่าน ศก, สีของวงกลม)
 if(LED1 == 'b'){tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3));}

while (true){ //กำหนดให้วนซ้ำในเงื่อนไขจนกว่าเงื่อนไขจะไม่เป็นจริง
  Blynk.run();
  uint16_t x, y;
  if (tft.getTouch(&x, &y)) { //เช็คสถานะการกดทัชสกรีน
    Serial.printf("Touched at (%d, %d)\n", x, y);
    if ((x > 0 && x < 240) && (y > 200 && y < 320)) { //ตรวจสอบตำแหน่งการกดทัชสกรีน
      digitalWrite(14, LOW); // สั่งให้ DIGITAL PIN มี State เป็น LOW
      tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E)); //สั่งให้วาดวงกลม
      LED1='a'; //กำหนดค่าในการตรวจสอบ output
      Blynk.virtualWrite(V1, 1); //ให้สถานะปุ่มกด CH1 ของแอพ Blynk เป็น ON
      }
    else if ((x > 240 && x < 480) && (y > 200 && y < 320)){
      digitalWrite(14, HIGH);
      tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3));
      LED1='b';
      Blynk.virtualWrite(V1, 0);
```

```
    }
   else if ((x > 430 && x < 480) && (y > 100 && y < 200)){ //กำหนดให้เมื่อกดปุ่มหน้าถัดไปจากจอ LCD จะทำการ
break ฟังก์ชัน และเปลี่ยนค่าขอ i เพื่อให้เข้าเงื่อนไขต่อไปใน switch case
    i++;
    if(i>4){i=1;}
    if(i<=0){i=4;}
    delay(200);
    break;
  }else if ((x > 0 && x < 50) && (y > 100 && y < 200)){
    i--;
    if(i>4){i=1;}
    if(i<=0){i=4;}
    delay(200);
    break;
    } else if(!((x > 240 && x < 480) && (y > 200 && y < 320)&&(x > 430 && x < 480)
        && (y > 100 && y < 200)&&(x > 430 && x < 480) && (y > 100 && y < 200)
        &&(x > 0 && x < 50) && (y > 100 && y < 200))){ //กำหนดว่าหากสัมผัสหน้าจอนอกเหนือจากพื้นที่ข้างต้น
ให้ใส่สีดำลงบนหน้าจอแล้วกลับไปทำงานที่ checkstate = 0
      tft.fillScreen(TFT_BLACK);
      checkstate = 0;
      delay(200);
      break;
      }
   }
 }
}
```

```
void dis2(){

tft.drawChar( 240, 100,'C',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 280, 100,'H',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 320, 100,'2',TFT_WHITE,TFT_BLACK, 6);
 if(LED2 == 'a'){tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E));}
 if(LED2 == 'b'){tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3));}

 while (true){
  Blynk.run();
  uint16_t x, y;
  if (tft.getTouch(&x, &y)) {
    Serial.printf("Touched at (%d, %d)\n", x, y);
    if ((x > 0 && x < 240) && (y > 200 && y < 320)) {
      digitalWrite(27, LOW);
      tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E));
      LED2='a';
      Blynk.virtualWrite(V2, 1);
      }
    else if ((x > 240 && x < 480) && (y > 200 && y < 320)){
      digitalWrite(27, HIGH);
      tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3));
      LED2='b';
      Blynk.virtualWrite(V2, 0);
      }
      else if ((x > 430 && x < 480) && (y > 100 && y < 200)){
      i++;
      if(i>4){i=1;}
      if(i<=0){i=4;}
      delay(200);
      break;
```

```
   }else if ((x > 0 && x < 50) && (y > 100 && y < 200)){
     i--;
     if(i>4){i=1;}
     if(i<=0){i=4;}
     delay(200);
     break;
   }else if(!((x > 240 && x < 480) && (y > 200 && y < 320)&&(x > 430 && x < 480)
          && (y > 100 && y < 200)&&(x > 430 && x < 480) && (y > 100 && y < 200)
          &&(x > 0 && x < 50) && (y > 100 && y < 200))){

       tft.fillScreen(TFT_BLACK);
       checkstate = 0;
       delay(200);
       break;
       }

     }

   }

}

void dis3(){
 tft.drawChar( 240, 100,'C',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 280, 100,'H',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 320, 100,'3',TFT_WHITE,TFT_BLACK, 6);
 if(LED3 == 'a'){tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E));}
 if(LED3 == 'b'){tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3));}

 while (true){
  Blynk.run();
```

```
uint16_t x, y;
if (tft.getTouch(&x, &y)) {
  Serial.printf("Touched at (%d, %d)\n", x, y);
  if ((x > 0 && x < 240) && (y > 200 && y < 320)) {
    digitalWrite(26, LOW);
    tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E));
    LED3='a';
    Blynk.virtualWrite(V3, 1);
    }
  else if ((x > 240 && x < 480) && (y > 200 && y < 320)){
    digitalWrite(26, HIGH);
    tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3));
    LED3='b';
    Blynk.virtualWrite(V3, 0);
    }
    else if ((x > 430 && x < 480) && (y > 100 && y < 200)){
    i++;
    if(i>4){i=1;}
    if(i<=0){i=4;}
    delay(200);
    break;
  }else if ((x > 0 && x < 50) && (y > 100 && y < 200)){
    i--;
    if(i>4){i=1;}
    if(i<=0){i=4;}
    delay(200);
    break;
    }else if(!((x > 240 && x < 480) && (y > 200 && y < 320)&&(x > 430 && x < 480)
        && (y > 100 && y < 200)&&(x > 430 && x < 480) && (y > 100 && y < 200)
        &&(x > 0 && x < 50) && (y > 100 && y < 200))){
```

```
      tft.fillScreen(TFT_BLACK);
      checkstate = 0;
      delay(200);
      break;
      }
   }
  }
}

void dis4(){
tft.drawChar( 240, 100,'C',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 280, 100,'H',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 320, 100,'4',TFT_WHITE,TFT_BLACK, 6);
 if(LED4 == 'a'){tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E));}
 if(LED4 == 'b'){tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3));}
 while (true){
  Blynk.run();
  uint16_t x, y;
  if (tft.getTouch(&x, &y)) {
   Serial.printf("Touched at (%d, %d)\n", x, y);
   if ((x > 0 && x < 240) && (y > 200 && y < 320)) {
     digitalWrite(25, LOW);
     tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E));
     LED4='a';
     Blynk.virtualWrite(V4, 1);
     }
   else if ((x > 240 && x < 480) && (y > 200 && y < 320)){
     digitalWrite(25, HIGH);
     tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3));
     LED4='b';
     Blynk.virtualWrite(V4, 0);
```

```
    }
   else if ((x > 430 && x < 480) && (y > 100 && y < 200)){
    i++;
    if(i>4){i=1;}
    if(i<=0){i=4;}
    delay(200);
    break;
  }else if ((x > 0 && x < 50) && (y > 100 && y < 200)){
    i--;
    if(i>4){i=1;}
    if(i<=0){i=4;}
    delay(200);
    break;
    }else if(!((x > 240 && x < 480) && (y > 200 && y < 320)&&(x > 430 && x < 480)
        && (y > 100 && y < 200)&&(x > 430 && x < 480) && (y > 100 && y < 200)
        &&(x > 0 && x < 50) && (y > 100 && y < 200))){

      tft.fillScreen(TFT_BLACK);
      checkstate = 0;
      delay(200);
      break;
      }
   }
 }
}
void timedis(){//ฟังก์ชันแสดงเวลาบนหน้าจอ

DateTime now = rtc.now(); //กำหนดการสืบทอดคลาส

if(now.hour()<=9){//now.hour คือค่าของชั่วโมง  บันทัดนี้แสดงให้เห็นว่าถ้าค่าน้อยกว่าหรือเท่ากับ 9
```

```
    tft.drawNumber(0, 100, 100, 7); // ให้แสดงเลข 0 ด้านหน้า
    tft.drawNumber(now.hour(), 132, 100, 7);// ตามด้วยชั่วโมง //เพราะจะแสดงแค่เลขเพียงตัวเดียวในตำแหน่งด้านหน้า
    }
if(now.hour()>9){tft.drawNumber(now.hour(), 100, 100, 7);}//ถ้ามากกว่า 9 เป็นต้นไป ให้แสดงตำแหน่งเดียวกับเลข 0 ในด้านบน  เนื่องจากแสดงเป็นตัวเลข 2 หลัก

tft.drawString(":", 175, 100, 7); //ใส่ : ขั้นกลาง

if(now.minute()<=9){
    tft.drawNumber(0, 200, 100, 7);
    tft.drawNumber(now.minute(), 232, 100, 7);
    }
if(now.minute()>9){tft.drawNumber(now.minute(), 200, 100, 7);}

tft.drawString(":", 275, 100, 7);

if(now.second()<=9){
    tft.drawNumber(0, 300, 100, 7);
    tft.drawNumber(now.second(), 332, 100, 7);
    }
if(now.second()>9){tft.drawNumber(now.second(), 300, 100, 7);}

tft.setTextColor(TFT_WHITE, TFT_BLACK);
tft.drawString("Hardware Temp.", 50, 200, 4);
tft.drawFloat(rtc.getTemperature(),2, 275, 200, 4); // แสดงค่าของอุณหภูมิ
tft.drawString(".C", 375, 200, 4);

  uint16_t x, y;
  if (tft.getTouch(&x, &y)){   //ตรวจสอบการสัมผัสหน้าจอ
```

```
 tft.fillScreen(tft.color24to16(0x000000)); //วาดรูปให้กับหน้าจอเพื่อเป็น interface มนการใช้งานเปิดปิดสวิซ
 tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3)); //วาดวงกลม
 tft.fillRect(0, 200, 240, 120, tft.color24to16(0x1ABC9C)); //วาดสี่เหลี่มผืนผ้า
 tft.fillRect(240, 200, 240, 120, tft.color24to16(0xE74C3C));
 tft.fillRect(430, 100, 50, 100, tft.color24to16(0x000000));
 tft.fillRect(0, 100, 50, 100, tft.color24to16(0x000000));

 tft.drawChar( 0,120,'<',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 445,120,'>',TFT_WHITE,TFT_BLACK, 6);
 tft.drawChar( 90, 250,'O',TFT_WHITE,tft.color24to16(0x1ABC9C), 4);
 tft.drawChar( 120, 250,'N',TFT_WHITE,tft.color24to16(0x1ABC9C), 4);
 tft.drawChar( 320, 250,'O',TFT_WHITE,tft.color24to16(0xE74C3C), 4);
 tft.drawChar( 350, 250,'F',TFT_WHITE,tft.color24to16(0xE74C3C), 4);
 tft.drawChar( 380, 250,'F',TFT_WHITE,tft.color24to16(0xE74C3C), 4);
  i=1;
  checkstate = 1;
  delay(100);
  }
 }


BLYNK_WRITE(V1) //ฟังก์ชันการรับค่าปุ่มกดของแอพ Blynk คือ V1
{
x1=param.asInt();//กำหนดตัวแปรให้กับ state ที่รับมากจากแอพ Blynk
if(x1==0){LED1='b';digitalWrite(14,1); //ถ้า state มาเป็น 0 ให้สั่งpinout เป็น 1
 if(checkstate==1&& i==1){ tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3)); }//เช็คการโชว์ state ที่หน้าจอ
}
if(x1==1){LED1='a';digitalWrite(14,0);
 if(checkstate==1&& i==1){ tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E)); }
```

```
}

}

BLYNK_WRITE(V2)
{

x2=param.asInt();
if(x2==0){LED2='b';digitalWrite(27,1);
  if(checkstate==1&& i==2){ tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3)); }
}
if(x2==1){LED2='a';digitalWrite(27,0);
  if(checkstate==1&& i==2){ tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E)); }
}
}
BLYNK_WRITE(V3)
{

x3=param.asInt();
if(x3==0){LED3='b';digitalWrite(26,1);
  if(checkstate==1&& i==3){ tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3)); }
}
if(x3==1){LED3='a';digitalWrite(26,0);
  if(checkstate==1&& i==3){ tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E)); }
}
}
BLYNK_WRITE(V4)
{

x4=param.asInt();
if(x4==0){LED4='b';digitalWrite(25,1);
```

```
  if(checkstate==1&& i==4){ tft.fillCircle(150, 100, 60, tft.color24to16(0xC3C3C3)); }
}
if(x4==1){LED4='a';digitalWrite(25,0);
if(checkstate==1&& i==4){ tft.fillCircle(150, 100, 60, tft.color24to16(0x00FF1E)); }
}
}
```

# บทที่ 4

## ผลการทดลอง

หลักการทำงานของเครื่องควบคุมเครื่องใช้ไฟฟ้าเปิด-ปิดแบบอัตโนมัติ คือแบ่งเป็น 2 โหมด คือ Manual และ Autoโหมด Manual Mode คือจะทำการโดยกด Push Button Switch ที่อยู่ที่หน้าตู้จะสั่งการทำงานไปที่Relay โดยจะใช้ Relay ทริก Solid State Relay เพื่อจะสั่งให้ Pump ทำงาน โหมดที่ 2 คือ Auto Mode จะเป็นการสั่งทำงานผ่านหน้าจอสัมผัสที่หน้าตู้โดยแบ่ง Chanel ออกเป็น 4 Chanel สามารถสั่งผ่าน Mobie Phone ผ่านเครือข่ายWifi สั่งการทำงานได้ในระยะไกลหรือเมื่อไม่ได้อยู่ที่หน้าตู้

# บทที่ 5

# สรุปปัญหาและข้อเสนอแนะ

## 5.1 สรุปปัญหา

1. การเขียนโปรแกรมที่ซับซ้อน ทำให้ใช้เวลาในการเขียนโค้ดเป็นเวลานาน
2. การต่อวงจรที่ซับซ้อน ทำให้เกิดการทดลองหลายครั้ง เกิดความยุ่งยากในการทำโครงงาน
3. การวางอุปกรณ์ต้องวางแผนให้ดี ถ้าหากไม่วางแผนอาจจะทำให้เกิดการต่อสายไฟได้ยากหรือสายไฟไม่เป็นที่เป็นทางไม่เป็นระเบียบ

## 5.2 ข้อเสนอแนะ

1. ขั้นตอนการต่อสายไฟควรสังเกตว่าการต่อสายไฟตรงขั้วหรือไม่
2. ควรศึกษาโครงสร้างของอุปกรณ์อิเล็กทรอนิกส์ที่จำนำมาใช้งานก่อนเพื่อป้องกันเมื่อต่ออุปกรณ์แล้วถูกต้องไม่ผิดพลาด ไม่เกิดอุบัติเหตุในการทดลอง

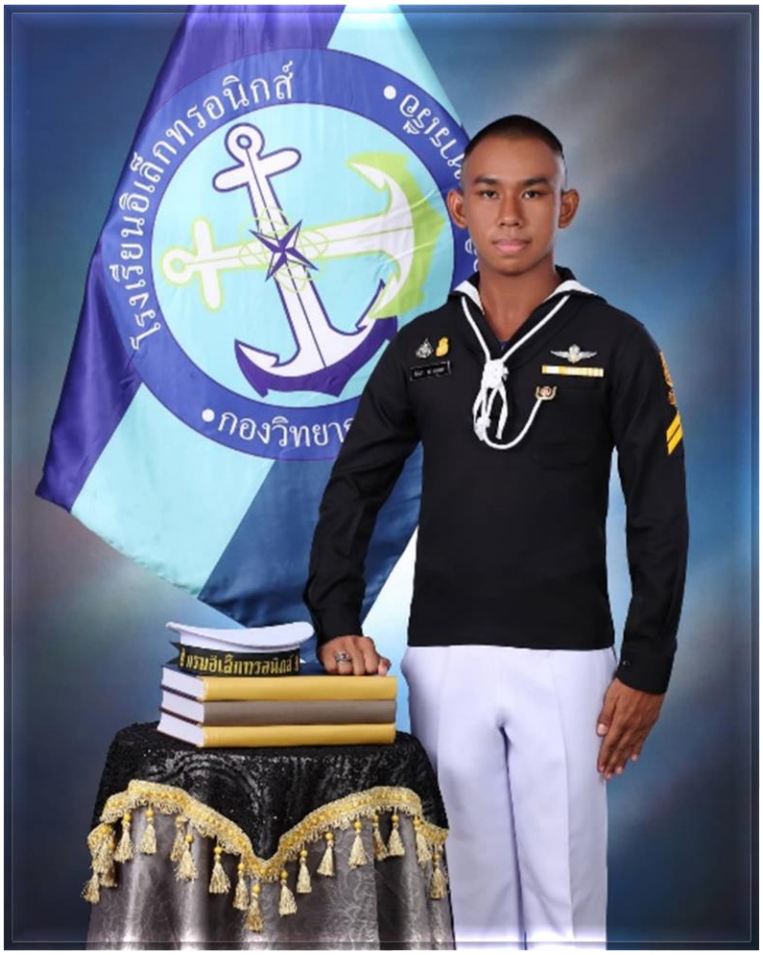
**ประวัติคณะผู้จัดทำโครงงาน**

นรจ.เอกกมล  ต่อมสุวรรณ์ เหล่า อิเล็กทรอนิกส์

ที่อยู่ 28/1 ม.2 ต.เกาะขนุน อ.พนมสารคาม

จ.ฉะเชิงเทรา

จบจาก โรงเรียน กาญจนาภิเษกวิยาลัย ฉะเชิงเทรา




นรจ.นิติธร  พวงบุษา เหล่า อิเล็กทรอนิกส์

ที่อยู่ 2 หมู่ 4 ต.บางคนที อ.บางคนที

จ.สมุทรสงคราม

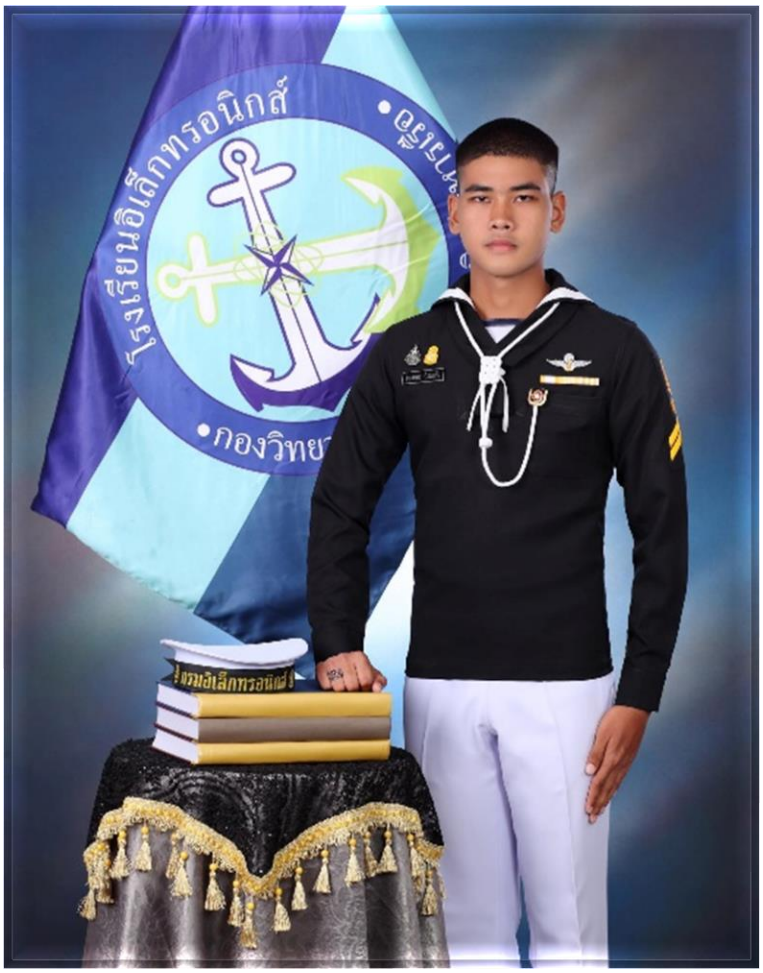
จบจาก โรงเรียน ศรัทธาสมุทร

นรจ.ชนนันท์ โรจนวิภาต เหล่า อิเล็กทรอนิกส์

ที่อยู่ 179 หมู่ 1 ถ.จรัญยานนท์

ต.บางวัว อ.บางปะกง จ.ฉะเชิงเทรา

จบจาก โรงเรียนกีฬาจังหวัดชลบุรี




นรจ.ธนายุทธ ป้อมแก้ว เหล่า อิเล็กทรอนิกส์

ที่อยู่ 13/2 หมู่ 5 ต.โพนางดำตก

อ.สรรพยา จ.ชัยนาท

จบจาก โรงเรียนสรรพยาวิทยา

นรจ.ชัยวัฒน์ ปานนนท์ เหล่า อิเล็กทรอนิกส์

ที่อยู่ 963/33 ถนนสุขสวัสดิ์ ซอยสุขสวัสดิ์26

แขวงบางปะกอก เขตราษฎร์บูรณะ กทม. 10140

จบจาก โรงเรียนชิโนรสวิทยาลัย

นรจ.พงศธร อิ่มรอ เหล่า อิเล็กทรอนิกส์

ที่อยู่ 183/2073 ถ.สรงประภา แขวงสีกัน

เขตดอนเมือง กรุงเทพมหานคร

จบจาก โรงเรียนมัธยม6อุดรธานี