

รายงานการปฏิบัติงานสหกิจศึกษา

ระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และ
ระบบน้ำ Soft

**IOT SYSTEM NOTIFIES THE NORMAL OPERATION OF THE WATER
CHILLER ,AIR COOLER ,SOFT WATER BELL.**

นายพีรวัส เจียจำรูญ รหัสนักศึกษา 630910483

นายสริต วนาวิเศษศักดิ์ รหัสนักศึกษา 630910590

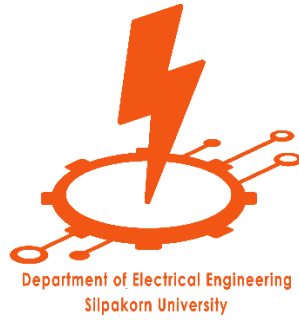
รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา 618493 สหกิจศึกษา

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยศิลปากร

ภาคการศึกษาต้น ปีการศึกษา 2566



รายงานการปฏิบัติงานสหกิจศึกษา

ระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และ
ระบบน้ำ Soft

**IOT SYSTEM NOTIFIES THE NORMAL OPERATION OF THE WATER
CHILLER ,AIR COOLER ,SOFT WATER BELL.**

นายพีรวัส เจียจรรย์ รหัสนักศึกษา 630910483

นายสริต วนาวิเศษศักดิ์ รหัสนักศึกษา 630910590

ปฏิบัติงาน ณ

บริษัท ลักกี้สตาร์วิฟวิ่งพลัส จำกัด

155/18-19 นิคมอุตสาหกรรมราชบุรี หมู่ 4 ต.เจ็ดเสมียน อ.โพธาราม

จ.ราชบุรี

บทคัดย่อ

บริษัท ลักกี้สตาร์วีฟวิ่งพลัส จำกัด เป็นผู้ผลิตผ้าพลาสติกสาน ที่สะอาดและปลอดภัย มีการพัฒนาผลิตภัณฑ์ สร้างสรรค์บริการ และปรับปรุงมาตรฐาน อย่างต่อเนื่องตามที่คุณค่าต้องการและพึงพอใจ จากการได้เข้าไปปฏิบัติงานใน โครงการสหกิจศึกษาในบริษัท ลักกี้สตาร์วีฟวิ่งพลัส จำกัด ในแผนก ซ่อมบำรุง ได้มอบหมายให้จัดทำระบบแจ้งเตือนปัญหาของ ระบบน้ำ Chiller ถังลม และระบบน้ำ Soft อัตโนมัติ พบว่ามีปัจจัยที่ส่งผลต่อการหยุดการทำงานของเครื่องจักร คือ ระดับน้ำ Chiller, อุณหภูมิ น้ำ Chiller, ระดับน้ำ Soft, ระดับแรงดันลม และการทำงานของปั้มน้ำ โดยจะเน้นให้ทางช่างรับรู้ การทำงานผิดปกติของระบบต่างๆ ได้ทันก่อนที่จะส่งผลให้เครื่องจักรหยุดการทำงาน

แผนก ซ่อมบำรุง ได้มอบหมายให้ทำการศึกษาเกี่ยวกับการแจ้งเตือนอัตโนมัติ, การเก็บข้อมูลของระบบที่มีปัญหา และแสดงค่าทางหน้าจอ Touchscreen เพื่อเพิ่มความรวดเร็วในการซ่อมแซม ลดความล่าช้าในกระบวนการผลิต ส่งผลให้บริษัทมีผลผลิตและเพิ่มขึ้น และลดการทำงานของพนักงานคุมเครื่องจักรที่ต้องทำการแจ้งซ่อมมาทาง Line ซ่อมบำรุง โดยส่งผลให้ระยะเวลาในการแจ้งซ่อมลดลงประมาณ 2 – 3 นาที ซึ่งรวดเร็วกว่าการแจ้งซ่อมแบบเดิมที่ใช้พนักงานคุมเครื่องแจ้งซ่อมมาทาง Line ซ่อมบำรุง

Abstract

Lucky Star Weaving Company Limited, in the area of cleaner plastic cloth, has developed creative products and services. Continuous standards that customers demand and positive results from the factory's operations in the Cooperative Education Project in Lucky Star Weaving Company Limited, in the Maintenance department needed a system to automatically warn of problems in the Chiller water system, air dish, and soft water ball. It was found that there were no stopping factors. Control targets are Chiller water level, water cooler, soft water level, wind level, and water pump control. Emphasis will be placed on engineers being able to control various normal things as soon as they are successful. Stop working

The maintenance department can check this by conducting an automatic inspection study. Problematic investigation and display the values on the screen Touch screen to speed up the deceleration control system in a control system that the company has efficient and Notified in advance by the machine operator who must know the office, report the repair via Line Maintenance. The delay time in notifying the repair is reduced by approximately 2 - 3 minutes, which is faster than the original repair notification using the machine operator. Report repairs via Line Maintenance.

กิตติกรรมประกาศ

การที่คณะผู้จัดทำได้มาปฏิบัติงานสหกิจศึกษา ณ บริษัทกักัสตาร์วีฟวิ่งพลัส จำกัด ตั้งแต่วันที่ 19 มิถุนายน พ.ศ. 2566 ถึงวันที่ 6 ตุลาคม พ.ศ. 2566 นั้น ส่งผลให้คณะผู้จัดทำได้รับความรู้ และประสบการณ์ต่างๆ ที่มีค่า สำหรับรายงานสหกิจศึกษาฉบับนี้ มีผลสำเร็จได้ด้วยดีจากความร่วมมือ และสนับสนุนจากหลายฝ่าย ดังนี้

1. คุณมนูญ	พาลาด	ผู้จัดการฝ่ายซ่อมบำรุง
2. น.ส.จุฑามาศ	ทรัพย์ประมูล	พนักงานธุรการ
3. นายชินดนัย	ศรีรุ่ง	ช่างเทคนิคไฟฟ้า
4. นายราเมศ	จันทิษฐ์	ช่างเทคนิคไฟฟ้า
5. นายชยางกูร	เคร่าเครือ	ช่างเทคนิคไฟฟ้า
6. นายพงศกร	สวัสดิรักษา	ช่างเทคนิคไฟฟ้า
7. นายศิวะ	สุพันธ์	ช่างเทคนิคไฟฟ้า
8. นายจิรวัฒน์	จันทร์ชญา	ช่างเทคนิคเครื่องกล
9. นายอนุรักษ์	สนโสภ	ช่างสาธารณูปโภค
10. นายประจักษ์	กรณ์ใหม่	ช่าง Modify

และบุคคลท่านอื่นที่ไม่ได้กล่าวนามทุกท่านที่ให้คำแนะนำช่วยเหลือในการจัดทำรายงานฉบับนี้

คณะผู้จัดทำขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในการให้ข้อมูล รวมทั้งเป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแล และให้ความเข้าใจเกี่ยวกับการใช้ชีวิตในการทำงานในโรงงาน และการทำงานร่วมกับผู้อื่น คณะผู้จัดทำขอขอบคุณไว้ ณ ที่นี้

คณะผู้จัดทำรายงาน

2 กันยายน 2566

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูปภาพ	ซ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขต	2
1.4 แผนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 ARDUINO IDE	4
2.2 C - Programming Language หรือ ภาษา C	5
2.3 ภาษาซีพลัสพลัส หรือ C++	6
2.4 Arduino UNO+WiFi R3 ATmega328P+ESP8266	7
2.5 บอร์ด NodeMCU ESP8266 V3	9
2.6 Thermocouple เซ็นเซอร์วัดอุณหภูมิ K-Type 0-800C	11
2.7 Waterproof Ultrasonic Module (JSN-SR04T) เซ็นเซอร์วัดระยะทาง	13
2.8 จอ Hmi Samkoon SK-050HE Touch Screen	14
2.9 Switching Power Supply 24V 1A 25W	15

สารบัญ

	หน้า
2.10 Line Notify	17
บทที่ 3 วิธีการดำเนินงาน	18
3.1 ศึกษาปัญหาของสถานประกอบการ	20
3.2 การสืบค้น เก็บรวบรวมข้อมูลและโครงงานที่เกี่ยวข้อง	20
3.3 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	21
3.4 การออกแบบโครงสร้างฮาร์ดแวร์ของการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	36
3.5 การติดตั้งอุปกรณ์ลงกล่องจัดเก็บวงจร	39
3.6 การออกแบบตารางจัดเก็บข้อมูลเมื่อแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	46
3.7 การออกแบบจอแสดงผล Hmi Samkoon	47
บทที่ 4 ผลการดำเนินการ	48
4.1 ผลการต่อวงจรระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	48
4.2 ผลการติดตั้งวงจรเข้าตู้คอนโทรล	50
4.3 ผลการแสดงค่าบนหน้าจอ Hmi Samkoon Touch Screen	51
4.4 ผลการเก็บข้อมูล	52
4.5 ต้นทุนในการประเมินการสร้างหรือผลิตนวัตกรรม	52
บทที่ 5 สรุปผลและข้อเสนอแนะ	54
5.1 สรุปผล	54
5.2 ปัญหาและข้อเสนอแนะ	54
5.3 แนวทางพัฒนาต่อยอด	54

เอกสารอ้างอิง	55
ประวัติผู้จัดทำรายงาน	56

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน	2
2.1 สวิตช์ DIP เพื่อเชื่อมต่อโมดูล	8
2.2 รายละเอียดจอ Hmi Samkoon SK-050HE Touch Screen	15
4.1 การประเมินราคา	51

สารบัญรูปภาพ

รูปที่	หน้า
2.1 แสดงคำสั่งภาษา C	5
2.2 แสดงคำสั่งภาษา C++	7
2.3 บอร์ด Arduino UNO+WiFi R3 ATmega328P+ESP8266	9
2.4 โครงสร้างของ Arduino UNO+WiFi R3 ATmega328P+ESP8266	9
2.5 บอร์ด NodeMCU ESP8266 V3	10
2.6 โครงสร้างของ NodeMCU ESP8266 V3	11
2.7 Thermocouple เซ็นเซอร์วัดอุณหภูมิ K-Type	11
2.8 MAX6675	12
2.9 โครงสร้างของ MAX6675	12
2.10 โครงสร้างของ Thermocouple ต่อกับ MAX6675	12
2.11 Waterproof Ultrasonic Module (JSN-SR04T) เซ็นเซอร์วัดระยะทาง	13
2.12 โครงสร้างของ Waterproof Ultrasonic Module (JSN-SR04T)	
เซ็นเซอร์วัดระยะทาง	14
2.13 จอ Hmi Samkoon SK-050HE Touch Screen	15
2.14 โครงสร้างของจอ Hmi Samkoon SK-050HE Touch Screen	15
2.15 Switching Power Supply 24V 1A 25W	16
2.16 หลักการทำงานของ Switching Power Supply	17
3.1 แผนขั้นตอนการทำงาน	18

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.2 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด Arduino UNO+WiFi R3 ATmega328P+ESP8266	21
3.3 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด Arduino UNO+WiFi R3 ATmega328P+ESP8266	22
3.4 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266	23
3.5 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266 (Temperature)	24
3.6 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266 (Flowswitch)	25
3.7 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266 (WaterChiller)	26
3.8 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266 (WaterSoft)	27
3.9 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266 (Pressure)	28
3.10 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหา ระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บน App Script	29

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.11 การออกแบบโครงสร้างฮาร์ดแวร์ของการแจ้งเตือนปัญหา ระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	36
3.12 บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์ระบบแจ้งเตือนปัญหา ระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	37
3.13 การเชื่อมต่ออุปกรณ์ด้านฮาร์ดแวร์	38
3.14 บล็อกไดอะแกรมแสดงการทำงานของระบบการแจ้งเตือนปัญหา ระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	39
3.15 การประกอบวงจรแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	40
3.16 การเชื่อมต่อสายไฟของระบบการแจ้งเตือนปัญหา ระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	40
3.17 การยี่ดอุปกรณ์กับวงจรแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	41
3.18 การประกอบจอ Hmi Samkoon SK-050HE Touch Screen Touch screen เข้ากับตู้	41
3.19 การประกอบขาชัฟเฟอร์ติดตั้งตู้	42
3.20 การประกอบเซ็นเซอร์วัดระยะทาง Waterproof Ultrasonic Module (JSN-SR04T) เข้ากับกล่องและท่อ	42
3.21 เซ็นเซอร์วัดระยะทาง (JSN-SR04T) วัดระดับน้ำChiller	43
3.22 ภายในของกล่องใส่เซ็นเซอร์วัดระยะทาง (JSN-SR04T) วัดระดับน้ำChiller	43
3.23 ภายในของกล่องใส่เซ็นเซอร์วัดระยะทาง (JSN-SR04T) วัดระดับน้ำSoft	44

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.24 ต่อเซ็นเซอร์วัดอุณหภูมิกับเซ็นเซอร์วัดระยะทาง (JSN-SR04T) เข้ากับถังChiller	44
3.25 ต่อเซ็นเซอร์วัดระยะทาง (JSN-SR04T) เข้ากับถังSoft	45
3.26 ต่อเซ็นเซอร์วัดระยะทาง (JSN-SR04T) เข้ากับถังSoft	45
3.27 การออกแบบตารางจัดเก็บข้อมูลเมื่อแจ้งเตือนปัญหา ระบบน้ำ Chiller ถังลม และระบบน้ำ Soft	46
3.28 การออกแบบจอแสดงผล HMI Samkoon	46
4.1 การเตรียมอุปกรณ์	47
4.2 การต่อวงจรและติดตั้งบนแผ่น PCB	48
4.3 การติดตั้งจอ Hmi Samkoon Touch Screen	48
4.4 การประกอบอุปกรณ์พร้อมต่อสายวงจร	49
4.5 ด้านหน้าของตู้คอนโทรล	49
4.6 ด้านล่างของตู้คอนโทรล	50
4.7 ผลการแสดงผลค่าบนหน้าจอ Touch Screen	50
4.8 ผลการเก็บข้อมูล	51

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

บริษัท ลักกี้สตาร์วีฟวิ่งพลัส จำกัด ประกอบธุรกิจผลิตผ้าพลาสติกสาน ที่สะอาดและปลอดภัย มีการพัฒนาผลิตภัณฑ์สร้างสรรค์บริการ และปรับปรุงมาตรฐาน อย่างต่อเนื่องตามที่ลูกค้าต้องการและพึงพอใจ

ระบบการทำงานของเครื่องจักร เป็นระบบที่จำเป็นต้องใช้ถังน้ำ Chiller ถังลม ถังน้ำ Soft ซึ่งในระบบทั้งหมดเหล่านี้จำเป็นต้องมีระดับน้ำ Chiller ห้ามต่ำกว่า 40 cm และมีอุณหภูมิไม่สูงเกิน 15 องศาเซลเซียส ระดับน้ำ Soft ห้ามต่ำกว่า 100 cm การทำงานของปั๊มลม และการทำงานของปั๊มน้ำ หากระบบทั้งหมดเหล่านี้มีส่วนใดส่วนหนึ่งมีค่าสูงเกินหรือต่ำกว่าค่าที่กำหนด จะส่งผลให้เครื่องจักรหยุดการทำงานและแสดงผลไปที่หน้าจอของเครื่องจักร เมื่อพนักงานพบเห็นจึงทำการแจ้งปัญหาเครื่องจักรเข้าในโปรแกรมแจ้งซ่อมและแจ้งเตือนเข้า Line group ของบริษัท ซึ่งอาจจะทำให้ฝ่ายซ่อมบำรุงรู้ปัญหาและแก้ไขได้ล่าช้า

จากปัญหาดังกล่าวข้างต้นทางผู้จัดทำได้เล็งเห็นปัญหาจึงได้สร้างระบบการแจ้งเตือนอัตโนมัติ เพื่อใช้แจ้งเตือนปัญหาของ อุณหภูมิใน Chiller ระดับน้ำ Chiller ระดับน้ำ Soft การทำงานของปั๊มน้ำ และแรงดันลม โดยหากพบความผิดปกติ จะแจ้งเตือนอัตโนมัติมาทาง Line group ฝ่ายซ่อมบำรุงและเก็บบันทึกข้อมูลปัญหาใน Google Sheet ทั้งนี้เพื่อนำข้อมูลมาเป็นข้อเปรียบเทียบภายหลัง และมีการแสดงอุณหภูมิ ระดับน้ำที่จอ HMI เพื่อดูค่า ณ ปัจจุบัน

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาการทำงานของ sensor

1.2.2 เพื่อสร้างระบบอัตโนมัติแจ้งเตือนปัญหาผ่านทาง Line Notify ใช้งานได้จริง

1.2.3 เพื่อทราบถึงการทำงานผิดปกติของอุณหภูมิใน Chiller ระดับน้ำ Chiller ระดับน้ำ Soft ปั๊มน้ำ และแรงดันลม

1.2.4 เพื่อไม่ให้มีผลกระทบต่อการทำงานของเครื่องจักร

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้รับความรู้ความเข้าใจเกี่ยวกับการเขียนโปรแกรม Arduino IDE เชื่อมต่อกับบอร์ด Arduino และวงจร

1.5.2 ได้ระบบแจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft อัตโนมัติ

1.5.3 สามารถซ่อมแซมได้ทัน ก่อนเกิดผลกระทบตามมา

1.5.4 สามารถเก็บข้อมูลและแสดงผล

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในการทำโครงงานนี้ผู้จัดทำได้ศึกษาข้อมูลทฤษฎีที่เกี่ยวข้อง ดังหัวข้อต่อไปนี้

2.1 ARDUINO IDE

2.2 ภาษา C (C – Programming Language)

2.3 ภาษา C++

2.4 บอร์ด Arduino UNO+WiFi R3 ATmega328P+ESP8266

2.5 บอร์ด NodeMCU ESP8266 V3

2.6 Thermocouple K-Type เซ็นเซอร์วัดอุณหภูมิ

2.7 Waterproof Ultrasonic Module (JSN-SR04T) เซ็นเซอร์วัดระยะทาง

2.8 จอ HMI SAMKOON SK-050HE

2.9 Switching Power Supply 24V 1A 25W

2.10 Line Notify

2.1 ARDUINO IDE



Arduino IDE เป็นโปรแกรมในการใช้งานลักษณะ Open source ซึ่ง Arduino IDE จะทำหน้าที่ติดต่อระหว่างคอมพิวเตอร์ ไม่ว่าจะเป็นระบบ Windows, Mac OS X หรือ Linux กับบอร์ด Arduino ซึ่งโปรแกรมนี้ออกแบบให้ง่ายต่อการเขียนโค้ดและอัปโหลดโปรแกรมที่เราเขียนเข้าสู่บอร์ด Arduino

Arduino IDE ส่วน IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนา หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น ส่วนในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชันอื่นๆ ของ Arduino

2.2 C - Programming Language หรือ ภาษา C

C - Programming Language หรือ ภาษา C เป็นภาษาคอมพิวเตอร์เพื่อวัตถุประสงค์ทั่วไป เป็นภาษาที่มีความจำเป็นมาก ช่วยสนับสนุนการเขียนโปรแกรมที่มีโครงสร้าง การกำหนดขอบเขตของตัวแปร มีรูปแบบการเขียนโปรแกรมเป็นแบบลำดับ (Imperative procedural) ให้ถูกออกแบบให้คอมไพล์อย่างตรงไปตรงมากับคอมพิวเตอร์ที่มีความเกี่ยวข้อง เพื่อให้สามารถเข้าถึงการจัดการหน่วยความจำในระดับต่ำ และทำให้โครงสร้างของภาษาเชื่อมโยงกับคำสั่งการทำงานของคอมพิวเตอร์อย่างมีประสิทธิภาพ

2.2.1 ตัวอย่าง C - Programming Language หรือ ภาษา C

ภาษา C เป็นภาษาโปรแกรมที่มีผู้คนใช้กันอย่างแพร่หลายมากที่สุด ในการเขียนโปรแกรมจะมีคำสั่งต่าง ๆ ฟังก์ชันของโปรแกรม และการสร้างตัวแปรต้องไม่ซ้ำกับคำสั่งในภาษา C และมีตัวแปรต่าง ๆ ที่เราต้องเรียนรู้ก่อนการเขียนโปรแกรม

```

1 //program area of rectangle
2 #include<stdio.h>
3 #include<conio.h>
4 void main()
5 {
6     int width=1000;
7     int length=56;
8     int area;
9     area=width*length;
10    printf("\n");
11    printf("this is width = %d\n",width);
12    printf("this is length = %d\n",length);
13    printf("Area of Rectangle is = %d",area);
14    getch();
15 }
```

รูปที่ 2.1 แสดงคำสั่งภาษา C

2.2.2 คุณสมบัติ

ความสามารถในการใช้งานบนสภาพแวดล้อมที่แตกต่างกัน เป็นลักษณะเด่นที่ถือเป็นจุดเด่นของภาษา C เลยทีเดียว กล่าวคือ ภาษา C สามารถรันอยู่บนคอมพิวเตอร์ได้หลายระดับ ตั้งแต่เมนเฟรมคอมพิวเตอร์ จนถึงไมโครคอมพิวเตอร์ และยังมีประสิทธิภาพที่นำมาใช้วัดกับภาษา C สามารถวัดได้จาก 2 แนวทาง คือ

-ชุดคำสั่งที่มีความกะทัดรัด

-การจัดการหน่วยความจำบนภาษา C ที่มีประสิทธิภาพสูง มีการทำงานที่รวดเร็ว เทียบเท่าภาษาระดับต่ำ

ความสามารถในการโปรแกรมแบบโมดูล ภาษา C อนุญาตให้มีการแบ่งโมดูลเพื่อคอมไพล์ได้ ซึ่งสามารถลิงค์เชื่อมโยงเข้ากันได้ดี รูปแบบโปรแกรมสามารถเขียนขึ้นได้ตามแบบแผนการโปรแกรมเชิงโครงสร้างได้อย่างดี ภาษา C คือภาษาที่ประกอบด้วยฟังก์ชัน ทั้งนี้โมดูลต่างๆจะเขียนอยู่ในรูปของฟังก์ชันทั้งสิ้น และมีความสามารถในการทำงานแบบพอยน์เตอร์ ยากที่จะพบได้ในภาษาระดับสูงทั่วไป โดยพอยน์เตอร์หรือตัวชี้สามารถกำหนดได้จากชนิดข้อมูลหลายชนิด ด้วยกัน เช่นเดียวกับฟังก์ชัน หรือโครงสร้าง รวมถึงตัวแปรแบบอาร์เรย์ ก็สามารถจัดการด้วยการนำ พอยน์เตอร์เข้ามาช่วยก็ยังได้ ถึงแม้ภาษา C จะจัดอยู่ในภาษาคอมพิวเตอร์ระดับสูงก็ตาม แต่ภาษา C ก็ยังสามารถเขียนใช้งานร่วมกับภาษาระดับต่ำอย่างภาษาแอสเซมบลีได้ ดังนั้นจึงมีการกล่าวว่า “ภาษา C เป็นภาษาที่อยู่ กึ่งกลางระหว่างภาษาระดับต่ำและภาษาระดับสูง” ตามปกติภาษาระดับสูงทั่วไป ตัวแปรที่สร้างขึ้นด้วยตัวอักษรพิมพ์เล็กและตัวพิมพ์ใหญ่ สามารถนำ มาใช้ร่วมกันได้ แต่ในภาษา C จะถือว่าแตกต่างกันอย่างสิ้นเชิง

2.3 ภาษาซีพลัสพลัส หรือ C++

ภาษาซีพลัสพลัส หรือ C++ เป็นภาษาโปรแกรมคอมพิวเตอร์อเนกประสงค์ มีโครงสร้างภาษาที่มีการจัดชนิดข้อมูลแบบสแตติก และสนับสนุนรูปแบบการเขียนโปรแกรมที่หลากหลาย ได้แก่ โปรแกรมเชิงกระบวนการ, การนิยามข้อมูล, โปรแกรมเชิงวัตถุ, และโปรแกรมแบบเจเนริก ภาษา C++ เป็นภาษาโปรแกรมเชิงพหุหน้าที่นิยมมากภาษาหนึ่งนับตั้งแต่ช่วงทศวรรษ 1990

เบียเนอ สเตราสตรูป (Bjarne Stroustrup) จากเบลล์แล็บส์ (Bell Labs) เป็นผู้พัฒนาภาษา C++ เดิมใช้ชื่อ "C with classes" ในปี ค.ศ. 1983 เพื่อพัฒนาภาษาซีดั้งเดิม สิ่งที่พัฒนาขึ้นเพิ่มเติมนั้นเริ่มจากการเพิ่มเติมการสร้างคลาสจากนั้นก็เพิ่มคุณสมบัติต่างๆ ตามมา ได้แก่ เวอร์ชวลฟังก์ชัน การโอเวอร์โหลดโอเปอเรเตอร์ การสืบทอดหลายสาย เทมเพลต และการจัดการเอกเซพชัน มาตรฐานของภาษา C++ ได้รับการรับรองในปี ค.ศ. 1998 เป็นมาตรฐาน ISO/IEC 14882:1998 เวอร์ชันล่าสุดคือเวอร์ชันในปี ค.ศ. 2014 ซึ่งเป็นมาตรฐาน ISO/IEC 14882:2014

2.3.1 รูปแบบของการออกแบบภาษา C++

ภาษา C++ ได้ถูกออกแบบมาเพื่อเป็นภาษาสำหรับการเขียนโปรแกรมทั่วไป สามารถรองรับการเขียนโปรแกรมในระดับภาษาเครื่องได้ เช่นเดียวกับภาษา C ในทางทฤษฎีภาษา C++ ควรจะมีความเร็วเทียบเท่าภาษา C แต่ในการเขียนโปรแกรมจริงนั้น ภาษา C++ เป็นภาษาที่มีการเปิดกว้างให้โปรแกรมเมอร์เลือกรูปแบบการเขียนโปรแกรม ซึ่งทำให้มีแนวโน้มที่โปรแกรมเมอร์อาจใช้รูปแบบที่ไม่เหมาะสม ทำให้โปรแกรมที่เขียนมีประสิทธิภาพต่ำกว่าที่ควรจะเป็น และภาษา C++ นั้นเป็นภาษาที่มีความซับซ้อนมากกว่าภาษา C จึงทำให้มีโอกาสเกิด

บ๊วกขณะคอมไพล์มากกว่า ภาษา C++ ได้รับการออกแบบเพื่อเข้ากันได้กับภาษา C++ ในเกือบทุกกรณี มาตรฐานของภาษา C++ ถูกออกแบบมาเพื่อไม่ให้มีการเจาะจงแพลตฟอร์มคอมพิวเตอร์ ภาษา C++ ถูกออกแบบมาให้รองรับรูปแบบการเขียนโปรแกรมที่หลากหลาย

2.3.2 ตัวอย่างภาษา C++

```
1  #include <iostream>
2
3  int main()
4  {
5      std::cout << "Hello World" << endl;
6      return 0;
7  }
```

รูปที่ 2.2 แสดงคำสั่งภาษา C++

2.3.3 ภาษา C++ กับภาษา C

ผู้คนมักจะคิดว่าภาษา C++ เป็นซูเปอร์เซตของภาษา C แต่นี่ไม่เป็นความจริงไปทั้งหมด โค้ดภาษา C ส่วนใหญ่สามารถนำมาคอมไพล์ได้อย่างไม่มีปัญหาโดยคอมไพเลอร์ของภาษา C++ แต่ก็มียกเว้นภาษา C บางอย่างที่ไม่สามารถทำได้ในภาษา C++ ถือว่าผิดหรือไม่ได้ทำงานตามที่เคยเป็น ตัวอย่างเช่น ภาษา C รองรับการแปลงชนิดข้อมูลจาก void* ไปเป็นพอยน์เตอร์ ชนิดอื่นแบบโดยปริยาย ในขณะที่ภาษา C++ ไม่รองรับ (เพื่อไม่ให้เกิดการแปลงชนิดข้อมูลโดยไม่ได้ตั้งใจ) นอกจากนี้ ภาษา C++ ยังเพิ่มคำสั่งวุ่นขึ้นมามากมาย เช่น new และ class ซึ่งคำเหล่านี้อาจถูกใช้เป็นชื่อตัวระบุ (เช่นชื่อตัวแปร) ในภาษา C ส่งผลให้โค้ดภาษา C ดังกล่าวไม่สามารถคอมไพล์ได้

มาตรฐาน C99 ของภาษา C ได้แก้ไขความไม่เข้ากันระหว่างภาษา C และภาษา C++ บางประการ โดยเพิ่มการคอมเมนต์บรรทัด (//) และทำให้สามารถผสมการประกาศตัวแปรเข้าไปในโค้ดได้ (ก่อนหน้านี้ตัวแปรทั้งหมดต้องประกาศ ณ จุดเริ่มต้นของฟังก์ชัน กล่าวคือ ก่อนหน้าที่จะมีการทำงานใด ๆ ในฟังก์ชัน)

2.4 Arduino UNO+WiFi R3 ATmega328P+ESP8266

การผสมการทำงานอย่างเต็มรูปแบบในบอร์ดเดียว: Uno R3 ,ATmega 328P และ WiFi ESP8266 พร้อมหน่วยความจำ 32Mb (MegaByte) โมดูลทั้งหมดสามารถทำงานร่วมกันหรือแยกกันได้ และทุกคนมีฟินเอาร์ท์เฮดเดอร์ของตัวเอง ทางออกที่สะดวกสำหรับการพัฒนาโครงการใหม่ที่ต้องใช้ Uno

และ WiFi ผ่าน USB คุณสามารถอัปเดต สเก็ทช์และเฟิร์มแวร์สำหรับ ATmega328 และสำหรับ ESP8266 สำหรับสิ่งนี้บนเครื่องมีตัวแปลง USB-serial CH340G ใช้อย่างง่ายดายมาก บอร์ดมีสวิตช์ DIP เพื่อเชื่อมต่อโมดูลต่างๆ

ตารางที่ 2.1 สวิตช์ DIP เพื่อเชื่อมต่อโมดูล

Connection	DIP						
	1	2	3	4	5	6	7
ATmega328ESP8266	ON	ON	OFF	OFF	OFF	OFF	OFF
USB ATmega328	OFF	OFF	ON	ON	OFF	OFF	OFF
USBESP8266 (Update firmware or sketch)	OFF	OFF	OFF	OFF	ON	ON	ON
USBESP8266 (communication)	OFF	OFF	OFF	OFF	ON	ON	OFF
All independent	OFF	OFF	OFF	OFF	OFF	OFF	OFF

ภายในบอร์ดของ Arduino UNO+WiFi R3 ATmega328P+ESP8266 ประกอบไปด้วย

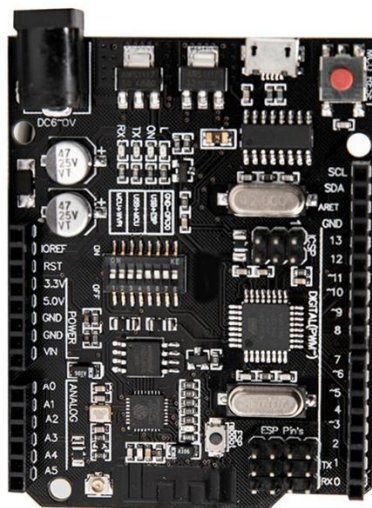
WiFi ESP8266

ไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อ WiFi ได้ พร้อมอุปกรณ์อำนวยความสะดวกต่างๆ เช่น พอร์ต micro USB สำหรับจ่ายไฟ/อัปโหลดโปรแกรม, ชิพสำหรับอัปโหลดโปรแกรมผ่านสาย USB, ชิพแปลงแรงดันไฟฟ้า และขาสำหรับเชื่อมต่ออุปกรณ์ภายนอก เป็นต้น

จุดเด่น สามารถเชื่อมต่อกับ WiFi ได้โดยไม่ต้องติดตั้งโมดูล WiFi เพิ่มเติม สามารถเขียน และ อัปโหลดโปรแกรมลงบอร์ดด้วยโปรแกรม Arduino IDE ผ่านสาย USB แบบเดียวกับที่ใช้ชาร์จโทรศัพท์ได้ สามารถอัปโหลดโปรแกรมผ่าน WiFi ได้ เรียกว่า Over the Air (OTA)

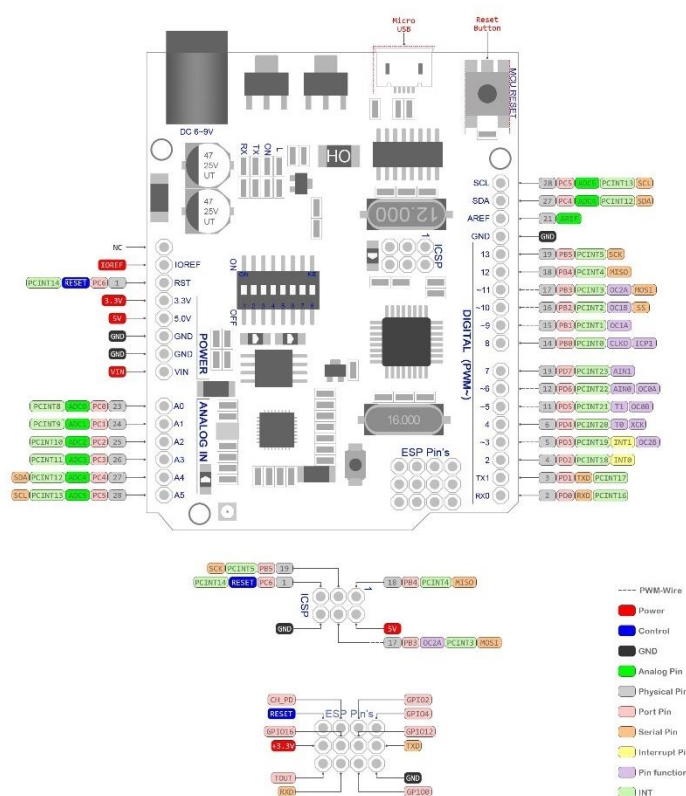
Arduino UNO R3

Arduino Uno R3 คือบอร์ดทดลอง Arduino รุ่นหนึ่ง ที่ได้รับความนิยมมากที่สุด เป็นรุ่นที่แนะนำสำหรับผู้เริ่มต้น ขอเรียกสั้น ๆ ว่าบอร์ดรุ่น Uno ใช้ชิพ ATmega328P มีส่วนประกอบหลักในการใช้งานครบถ้วน ขา Input/Output สักควบคุมอุปกรณ์เพียงพอกับการใช้งาน UNO R3 รองรับการสื่อสารหลัก ๆ ครบถ้วน ใช้งานได้กับโมดูล เซนเซอร์ เกือบทุกชนิดที่มีจำหน่าย บอร์ด uno ราคาถูก มีขาต่อใช้งานง่าย มีบอร์ดโมดูล Shield ที่ออกแบบมาสำหรับรุ่น UNO แค่เสียบก็พร้อมใช้งาน และยังมีโค้ดตัวอย่างการใช้งานให้ศึกษาเป็นจำนวนมาก



รูปที่ 2.3 บอร์ด Arduino UNO+WiFi R3 ATmega328P+ESP8266

2.4.1 โครงสร้างของ Arduino UNO+WiFi R3 ATmega328P+ESP8266

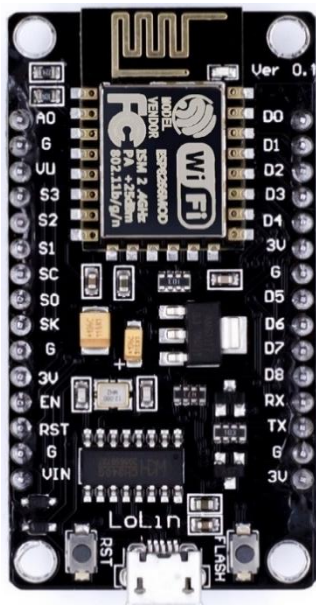


รูปที่ 2.4 โครงสร้างของ Arduino UNO+WiFi R3 ATmega328P+ESP8266

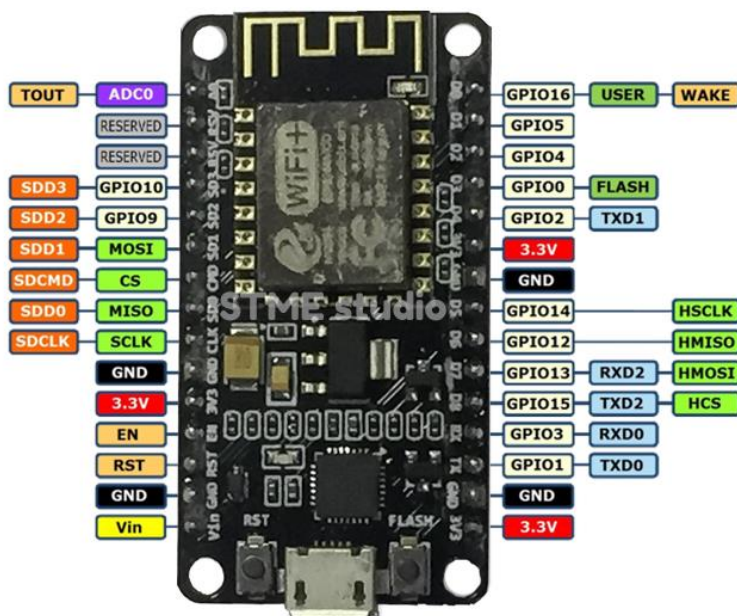
2.5 บอร์ด NodeMCU ESP8266 V3

สามารถใช้โปรแกรม Sketch ของ Arduino (Arduino IDE) ในการเขียนโปรแกรมควบคุมการทำงานของบอร์ด NodeMCU ESP8266 (ซึ่งในบางครั้ง อาจเรียกบอร์ดลักษณะนี้ว่า WiFi controller)

เพราะเป็นโปรแกรมที่ใช้งานง่าย เขียนด้วยภาษา C และมีโมดูล Wifi สามารถเชื่อมต่อเพื่อส่งข้อมูลหรือสั่งงานผ่านทางอินเทอร์เน็ตได้โดยไม่ต้องหาซื้ออุปกรณ์มาต่อเพิ่ม และความเป็น Open Source ได้รับความนิยมสูง และในส่วนของบอร์ด NodeMCU ESP8266 เองนั้น เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีจำนวนขาพอร์ตอินพุตและเอาต์พุตมากพอสำหรับการนำไปใช้งานจริง สามารถต่อกับเซ็นเซอร์ได้ทั้งแบบดิจิตอลและแอนะล็อก และยังต่อเพื่อขับอุปกรณ์เอาต์พุตให้ทำงาน โดยที่เราจะต้องเขียนโปรแกรมเพื่อสั่งงานให้บอร์ด NodeMCU ESP8266 สามารถควบคุมอุปกรณ์อิเล็กทรอนิกส์ต่างๆ นอกจากนั้นยังมีราคาถูก



รูปที่ 2.5 บอร์ด NodeMCU ESP8266 V3



รูปที่ 2.6 โครงสร้างของ NodeMCU ESP8266 V3

2.6 Thermocouple เซ็นเซอร์วัดอุณหภูมิ K-Type 0-800C

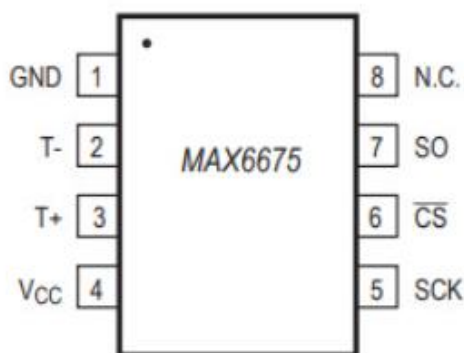
Temperature Sensor Thermocouple เซ็นเซอร์วัดอุณหภูมิแบบ Thermocouple K-Type วัดอุณหภูมิได้ระหว่าง 0-800 องศาเซลเซียส สายเป็นโลหะถักเสริมความแข็งแรงและทนต่ออุณหภูมิที่สูงๆ ได้ นิยมใช้กับ **MAX6675** ที่มีการชดเชยการเชื่อมต่อเย็น, การแก้ไขเชิงเส้น, การตรวจจับการแตกของเทอร์โมคัปเปิล ADC เทอร์โมคัปเปิลชนิด K แบบอนุกรม, ความสามารถในการแก้ปัญหา



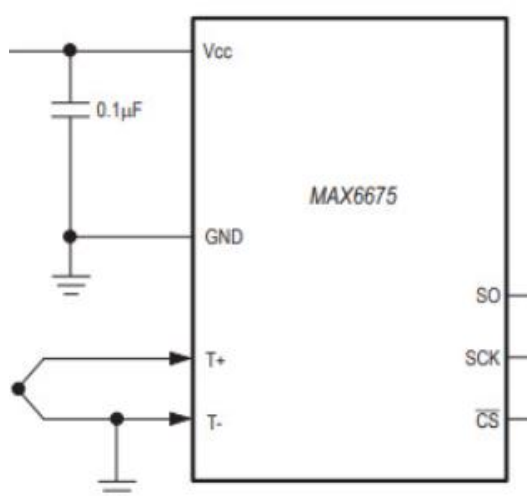
รูปที่ 2.7 Thermocouple เซ็นเซอร์วัดอุณหภูมิ K-Type



รูปที่ 2.8 MAX6675



รูปที่ 2.9 โคลงสร้างของ MAX6675



รูปที่ 2.10 โคลงสร้างของ Thermocouple ต่อกับ MAX6675

2.6.1 คุณสมบัติ Thermocouple K-Type

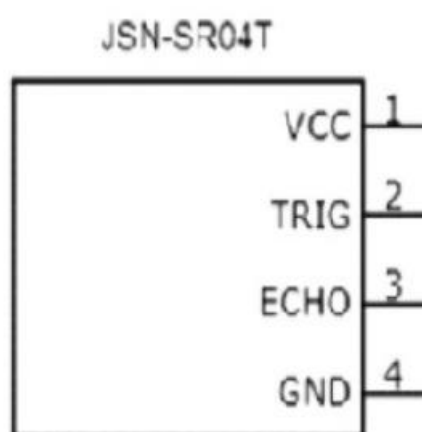
- แรงดันไฟฟ้าที่ใช้งาน: 3.0 ~ 5.5V
- วงจรชดเชยจุดแยกความเย็นภายในในตัว
- ด้วยอินเทอร์เฟซแบบอนุกรมสามแบบที่เรียบง่าย
- สัญญาณอุณหภูมิสามารถแปลงเป็นดิจิตอล 12 บิต
- ความละเอียดของอุณหภูมิ: 0.25 °C
- ช่วงการชดเชยความเย็น: -20 ~ +80 องศา,
- วงจรตรวจจับการแตกหักของเทอร์โมคัปเปิลในตัว
- ใช้การสื่อสารผ่านสาย SPI 3
- หัววัดอุณหภูมิแบบ K
- Type K ช่วงอุณหภูมิ 0-800 °C

2.7 Waterproof Ultrasonic Module (JSN-SR04T) เซ็นเซอร์วัดระยะทาง

JSN-SR04T คือ โมดูล Ultrasonic ที่ป้องกันความชื้นและละอองน้ำได้ โดยสามารถตรวจจับวัตถุและหาระยะห่างระหว่างตัว Sensor และวัตถุได้ในระยะ 25 cm ถึง 4.5 m



รูปที่ 2.11 Waterproof Ultrasonic Module (JSN-SR04T) เซ็นเซอร์วัดระยะทาง



รูปที่ 2.12 โครงสร้างของ Waterproof Ultrasonic Module (JSN-SR04T) เซ็นเซอร์วัดระยะทาง

2.7.1 คุณสมบัติ Waterproof Ultrasonic Module (JSN-SR04T)

- แรงดันใช้งาน 5VDC 5mA - 30mA
- ความถี่เสียงที่เปล่งออกมา 40 kHz
- ระยะทางไกลสุด: 4.5 ม.
- Blind: 25 ซม.
- ขนาดโมดูล: 41 มม. x 28.5 มม.

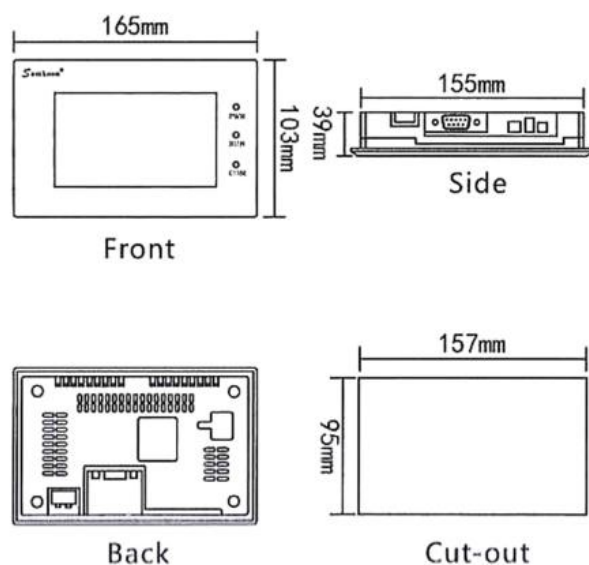
- ความละเอียด: ประมาณ 0.5 ซม.
- มุม: น้อยกว่า 50 องศา
- อุณหภูมิในการทำงาน: $-10 \sim 70^{\circ}\text{C}$
- อุณหภูมิในการจัดเก็บ: $-20 \sim 80^{\circ}\text{C}$

2.8 จอ Hmi Samkoon SK-050HE Touch Screen

Hmi Samkoon SK-050HE Touch Screen คือ หน้าจอแสดงผลและนำเข้าข้อมูลจากอุปกรณ์ ผู้ใช้สามารถทำการป้อนข้อมูลผ่านการสัมผัสที่สะดวกและง่ายต่อการใช้งาน โดยจอจะแสดงผลเป็น ตัวอักษร, ตัวเลข, รูปภาพและกราฟ ซึ่งสามารถเชื่อมต่อกับบอร์ด Arduino, PLC นิยมใช้ในการควบคุม เครื่องจักรในอุตสาหกรรม



รูปที่ 2.13 จอ Hmi Samkoon SK-050HE Touch Screen



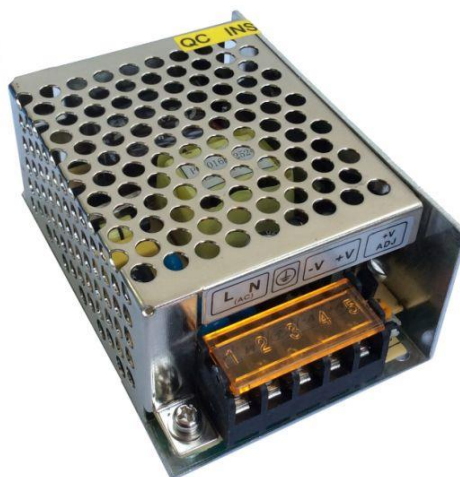
รูปที่ 2.14 โคลงสร้างของจอ Hmi Samkoon SK-050HE Touch Screen

ตารางที่ 2.2 รายละเอียดจอ Hmi Samkoon SK-050HE Touch Screen

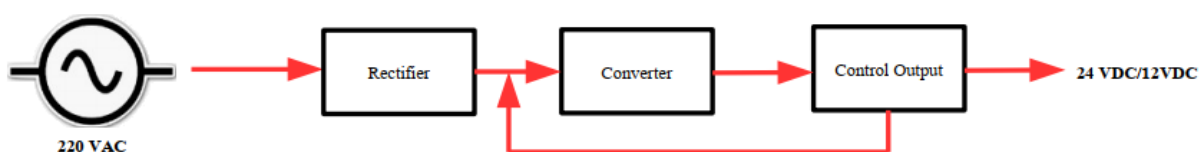
Cut-out Size	157 × 95
Display Size	5" (16:9)
Resolution (Px)	800 × 480
Touch Screen	Four-wire screen
Memory	128M FLASH + 128M DDR2
Protection Grade	IP65(Front panel)
Power Supply	DC24V (+/-15%)
Power Consumption	3W
Series Port	COM : RS232/422/485
USB-A	YES
USB-B	YES
Ethernet Port	NO
Operation Temperature	-20 ~ 65°C
Operation Humidity	10 ~ 90%RH
PC Software	SKTOOL

2.9 Switching Power Supply 24V 1A 25W

Switching Power Supply (สวิตชิงเพาเวอร์ซัพพลาย) คือ อุปกรณ์แปลงแรงดันไฟฟ้ากระแสสลับที่มีแรงดันสูง เช่น 220VAC ไปเป็นแรงดันไฟฟ้าที่มีแรงดันต่ำ โดย Switching Power Supply จะทำงานในลักษณะเดียวกันกับหม้อแปลงแรงดันทั่วไป แต่มีประสิทธิภาพที่ดีกว่าและมีขนาดเล็กกว่า โดยหลักการทั่วไปของ Switching Power Supply จะประกอบด้วย เรกติไฟเออร์ (Rectifier) ทำหน้าที่ แปลงแรงดันไฟฟ้ากระแสสลับให้เป็นแรงดันไฟฟ้ากระแสตรง, คอนเวอร์เตอร์ (Converter) ทำหน้าที่ แปลงความถี่แรงดันไฟฟ้ากระแสตรงให้เป็นแรงดันไฟฟ้ากระแสสลับที่มีความถี่สูง และแปลงแรงดันไฟฟ้ากระแสสลับให้เป็นแรงดันไฟฟ้ากระแสตรง โดยมีความต้านทานทางด้านเอาต์พุตของแรงดันไฟฟ้ากระแสตรงให้ได้ตามความต้องการอีกครั้ง



รูปที่ 2.15 Switching Power Supply 24V 1A 25W



รูปที่ 2.16 หลักการทำงานของ Switching Power Supply

ในปัจจุบันมีหลากหลายรุ่นที่ใช้กันแพร่หลายในงานอุตสาหกรรม เช่น Switching Power Supply แปลงแรงดันไฟฟ้ากระแสสลับจาก 220VAC เป็นแรงดันไฟฟ้ากระแสตรง 24VDC (1.2A, 2.5A, 5A, 10A) หรือ 12VDC (2A, 5A, 10A, 20A) เป็นต้น

2.9.1 คุณสมบัติ Switching Power Supply 24V 1A 25W

- Switching Power supply แหล่งจ่ายไฟ 24V 1A
- แหล่งจ่ายไฟแบบสวิตชิง 24V 1A
- Switching power supply AC 100-240V to DC 24V 1A 25W module
- แรงดันอินพุต :100-240VAC
- แรงดันเอาต์พุต: 24Vdc
- กระแสเอาต์พุต: 1A
- กำลังเอาต์พุต : 25W
- ระบบตัดไฟอัตโนมัติ เมื่อมีการช้อตวงจร
- ขนาด : 84x58x38mm

2.10 Line Notify



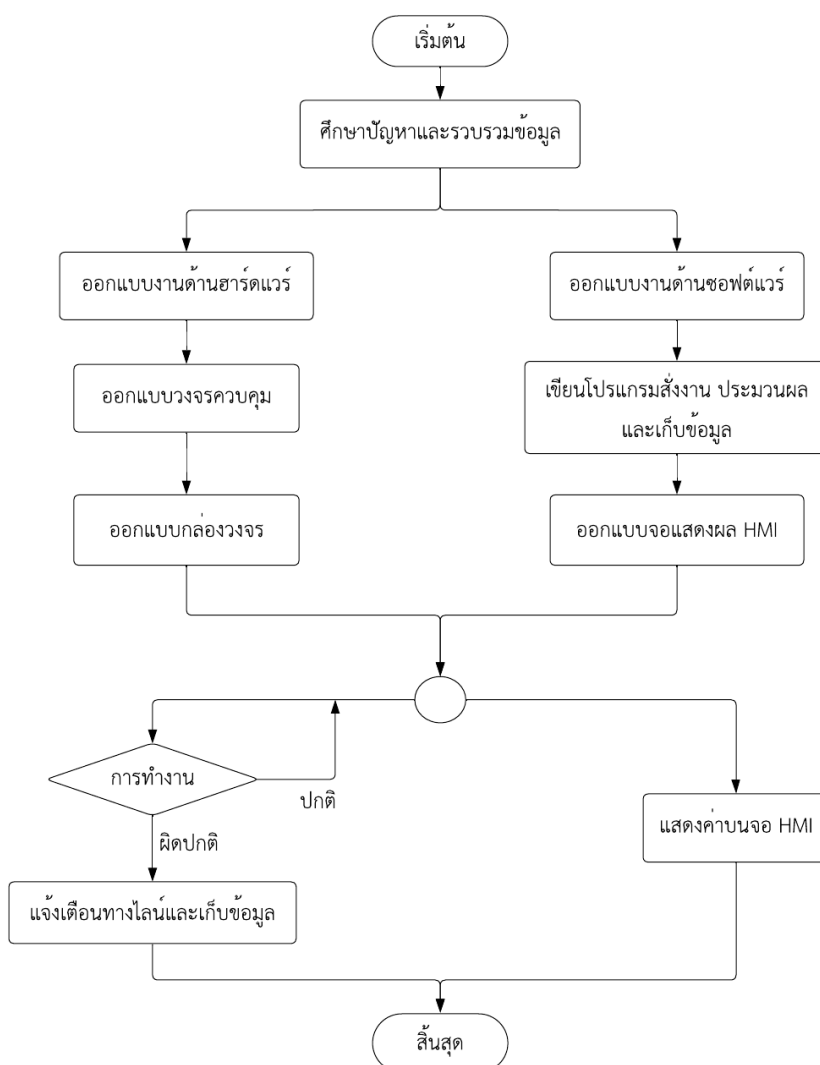
LINE Notify
Connect Everything

บริการที่คุณสามารถรับข้อความแจ้งเตือนจากเว็บเซอร์วิสต่างๆ ที่คุณสนใจได้ทาง LINE โดย
หลังเสร็จสิ้นการเชื่อมต่อกับทางเว็บเซอร์วิสแล้ว คุณจะได้รับการแจ้งเตือนจากบัญชีทางการของ “LINE
Notify” ซึ่งให้บริการโดย LINE นั้นเอง คุณสามารถเชื่อมต่อกับบริการที่หลากหลาย และยังสามารถรับ
การแจ้งเตือนทางกลุ่มได้อีกด้วย

บทที่ 3

วิธีการดำเนินงาน

โครงการ “พัฒนาระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft” เพื่อให้การดำเนินงานเป็นไปด้วยความเรียบร้อยและสำเร็จตามวัตถุประสงค์ จะต้องมีการวางแผนและออกแบบ จัดลำดับขั้นตอนการทำงาน เพื่อให้การทำงานราบรื่นและเกิดผลสำเร็จได้ด้วยดี ในบทนี้จะกล่าวถึงขั้นตอนการดำเนินงานออกแบบและสร้างวงจรแจ้งเตือนปัญหาอุณหภูมิใน Chiller สูง ระดับน้ำ Chiller ต่ำ ระดับน้ำ Soft ต่ำ ป้อนน้ำไม่ทำงาน และแรงดันลมต่ำ ไปจนถึงการติดตั้งในสถานที่ใช้งาน ผู้จัดทำมีขั้นตอนการดำเนินงานสามารถแสดงเป็นแผนผังขั้นตอนการทำงานได้ ดังรูปที่ 3.1 ดังต่อไปนี้



รูปที่ 3.1 แผนผังขั้นตอนการทำงาน

ขั้นตอนในการดำเนินงานมีดังนี้

3.1 ศึกษาปัญหาของสถานประกอบการ

3.2 การสืบค้น เก็บรวบรวมข้อมูลและโครงการที่เกี่ยวข้อง

3.3 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.3.1 การเขียนโปรแกรม Arduino IDE เพื่ออ่านค่าและส่งค่าsensor ลงบนบอร์ด Arduino UNO+WiFi R3 ATmega328P+ESP8266

3.3.2 การเขียนโปรแกรม Arduino IDE เพื่ออ่านค่าและส่งค่าsensor ลงบนบอร์ด ESP8266

3.3.3 การเขียนโปรแกรม App Script เพื่อเก็บข้อมูลและแจ้งเตือนไปยัง Line Notify

3.4 การออกแบบโครงสร้างฮาร์ดแวร์ของการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.4.1 บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์ระบบแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.4.2 การเชื่อมต่ออุปกรณ์ด้านฮาร์ดแวร์

3.4.3 บล็อกไดอะแกรมแสดงการทำงานของระบบการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.5 การติดตั้งอุปกรณ์ลงกล่องจัดเก็บวงจร

3.5.1 การประกอบวงจรแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.5.2 การเชื่อมต่อสายไฟของระบบการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.5.3 การยี่ดวงจรกับกล่อง

3.6 การออกแบบตารางจัดเก็บข้อมูลเมื่อแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.7 การออกแบบจอแสดงผล HMI Samkoon

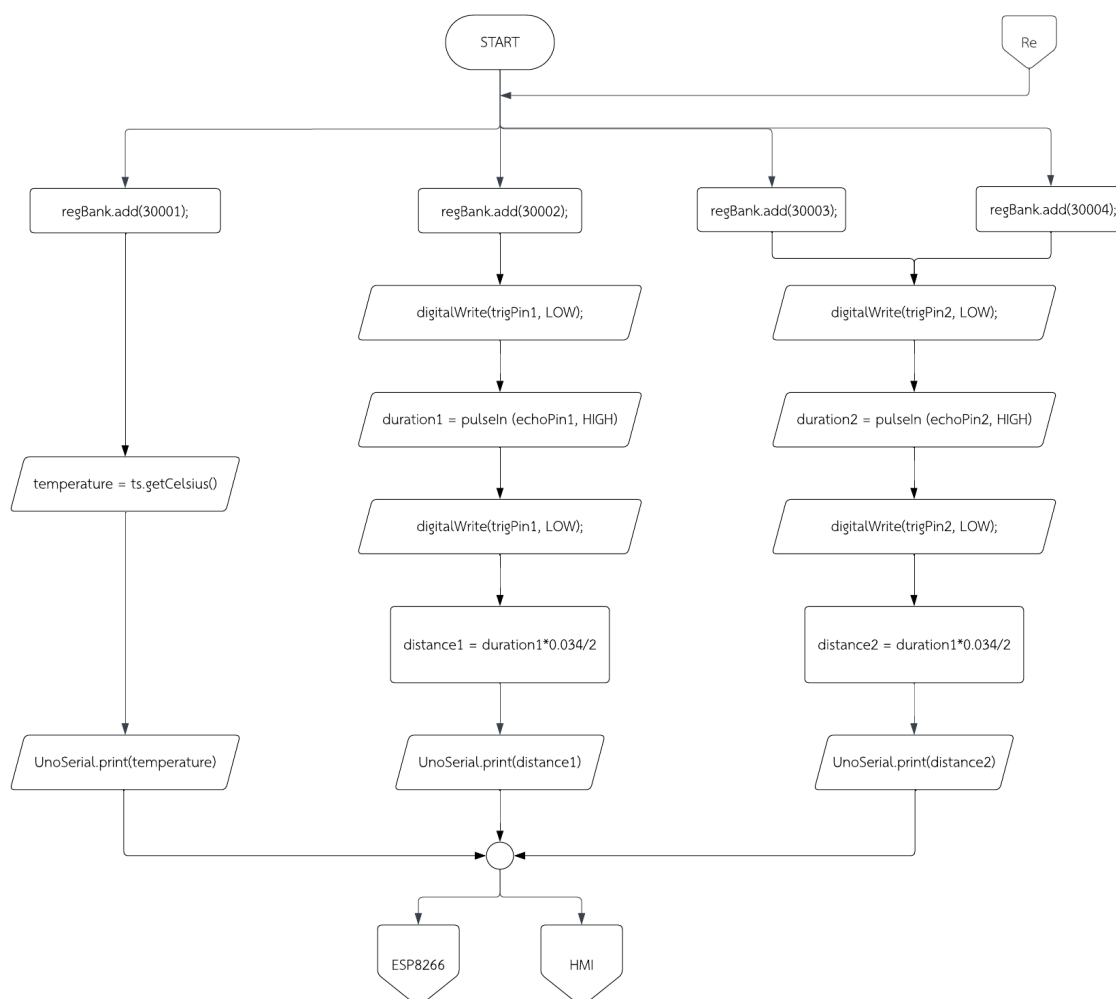
3.1 ศึกษาปัญหาของสถานประกอบการ

ก่อนทำโครงการ “พัฒนาระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft” ทางผู้จัดทำได้พบสาเหตุที่ต้องหยุดการทำงานของเครื่องจักร คือ อุณหภูมิภายใน Chiller ระดับน้ำ Chiller ระดับน้ำ Soft ป้อนน้ำ และแรงดันลม จึงได้คิดพัฒนาระบบ IOT นี้ขึ้น

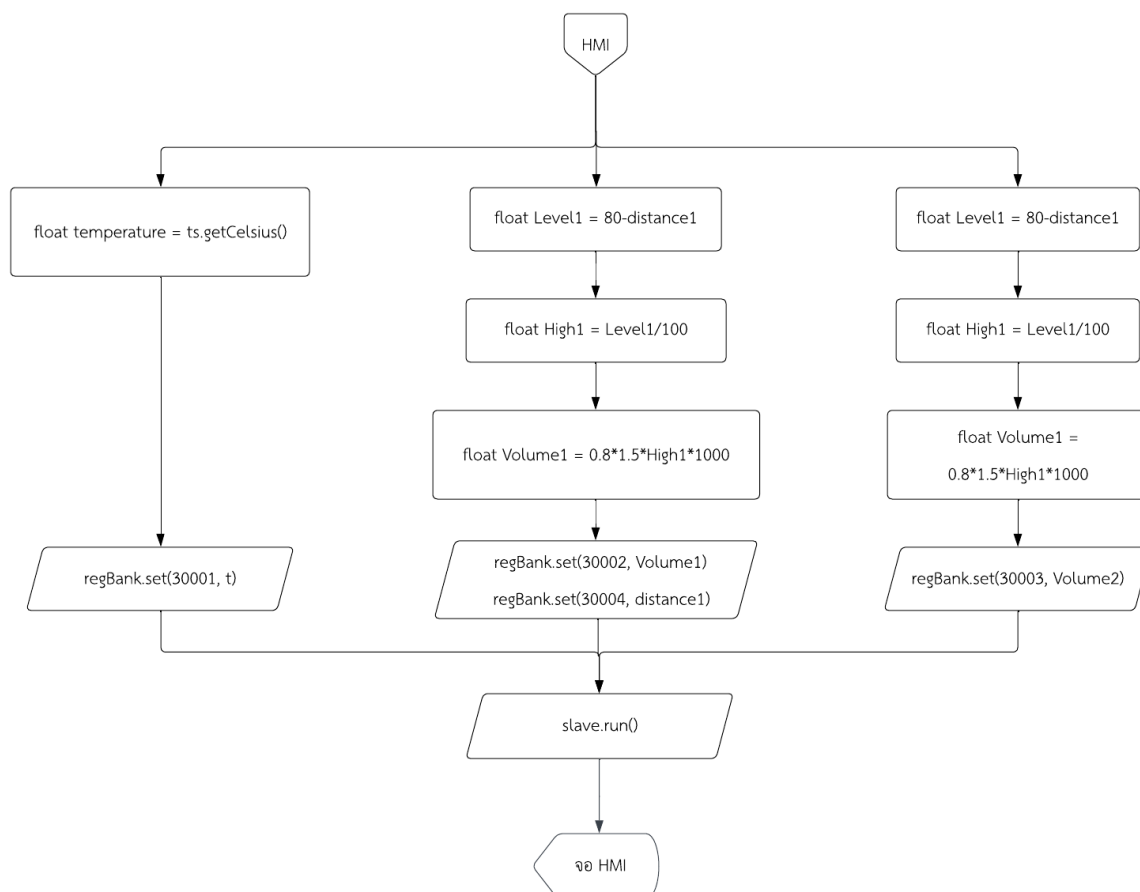
3.2 การสืบค้น เก็บรวบรวมข้อมูลและโครงการที่เกี่ยวข้อง

ก่อนเริ่มทำการออกแบบและสร้างกล่องวงจร ทางผู้จัดทำได้ทำการสืบค้นรวบรวมข้อมูลทาง ทฤษฎีและงานวิจัยที่เกี่ยวข้องเป็นตัวอย่างในการจัดทำโครงการ “พัฒนาระบบ IOT เพื่อแจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ” รวมไปถึงข้อมูลที่เกี่ยวข้องกับการออกแบบระบบ เพื่อสร้างวงจรและวิธีการดำเนินงานต่างๆ ซึ่งเป็นข้อมูลต่อการจัดทำโครงการนี้

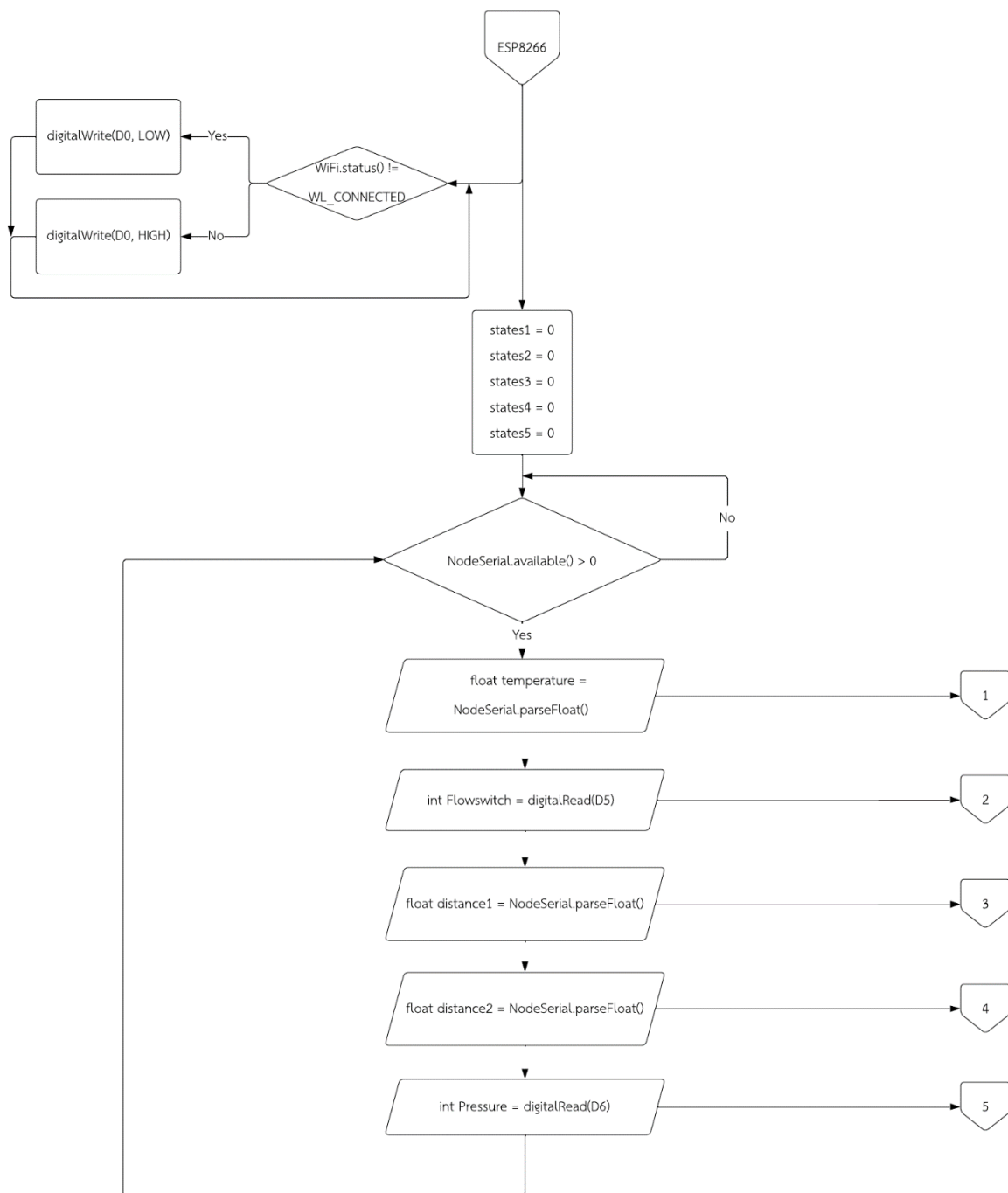
3.3 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft



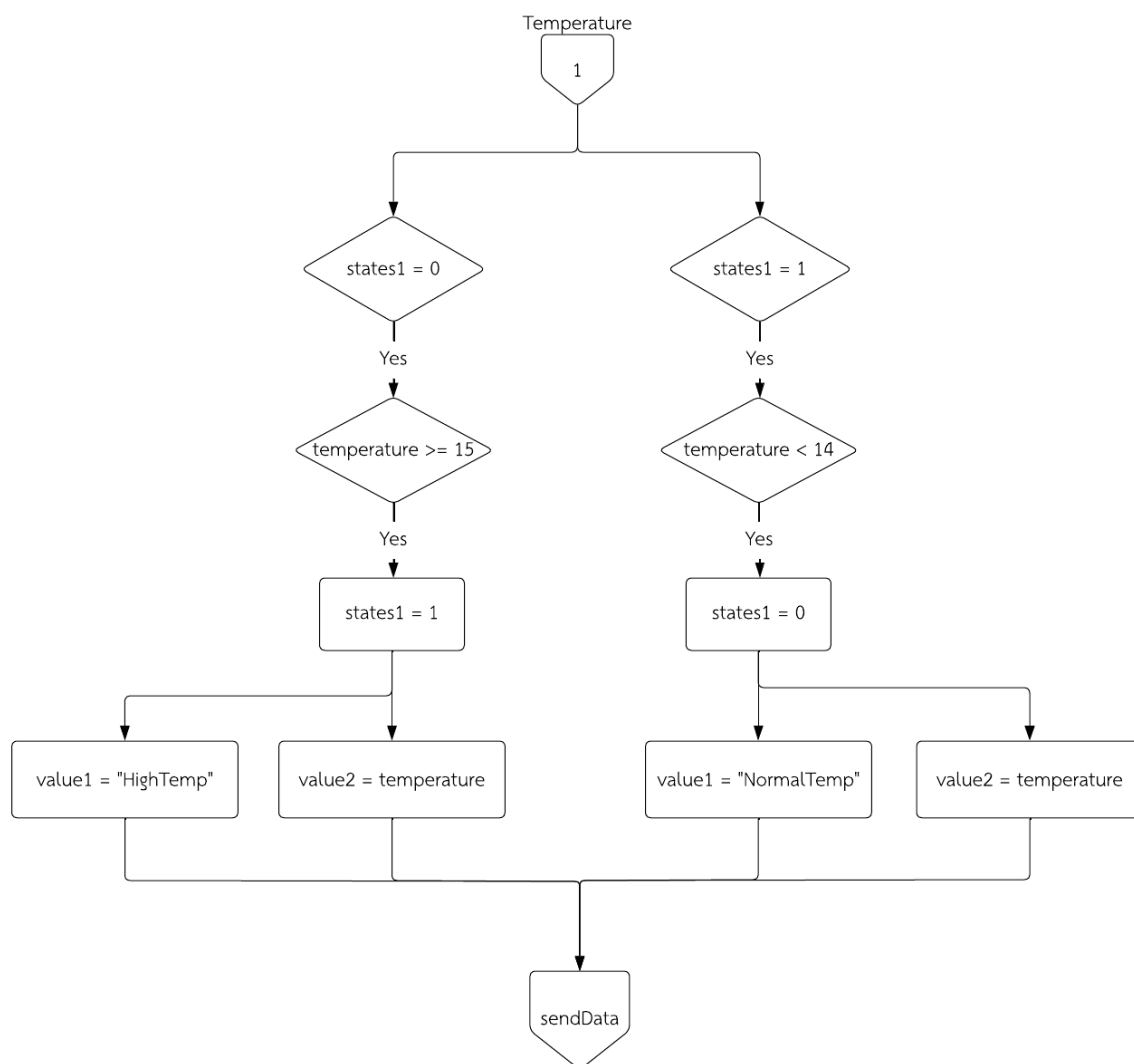
รูปที่ 3.2 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft บนบอร์ด Arduino UNO+WiFi R3 ATmega328P+ESP8266



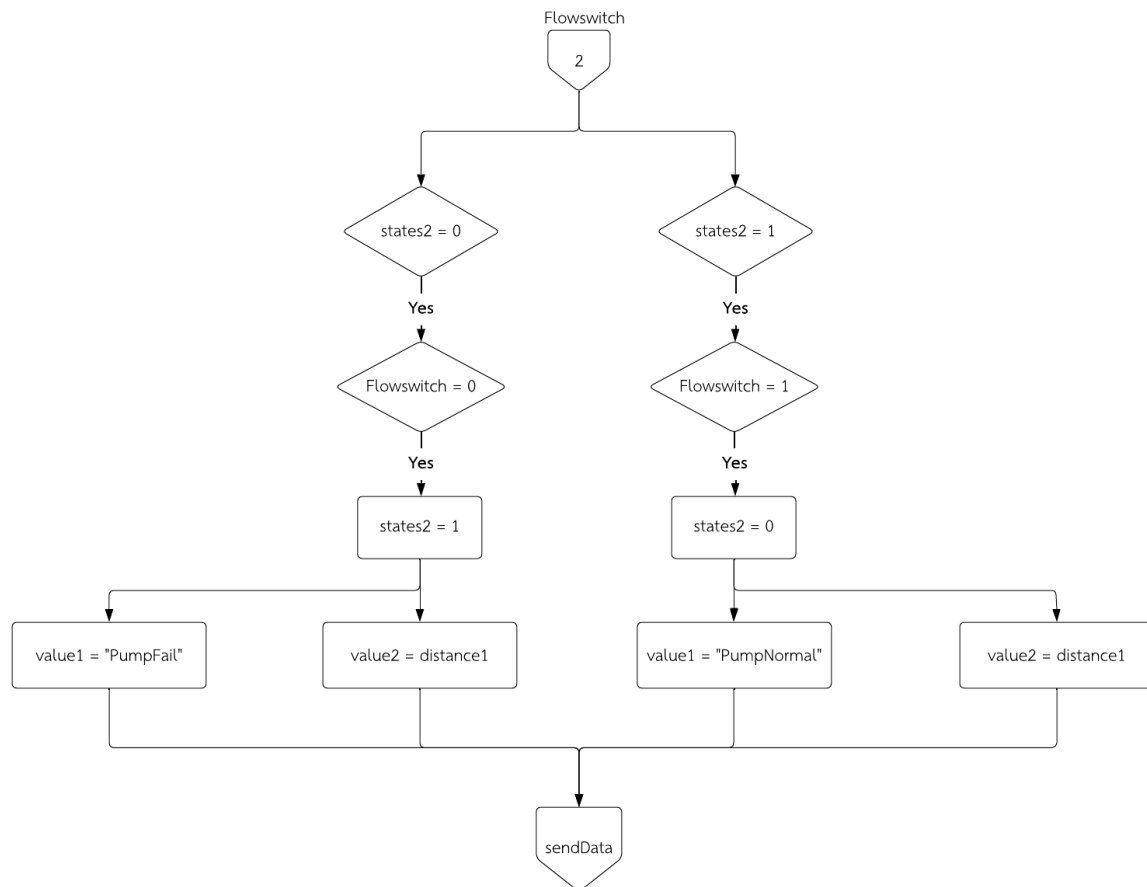
รูปที่ 3.3 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด Arduino UNO+WiFi R3 ATmega328P+ESP8266



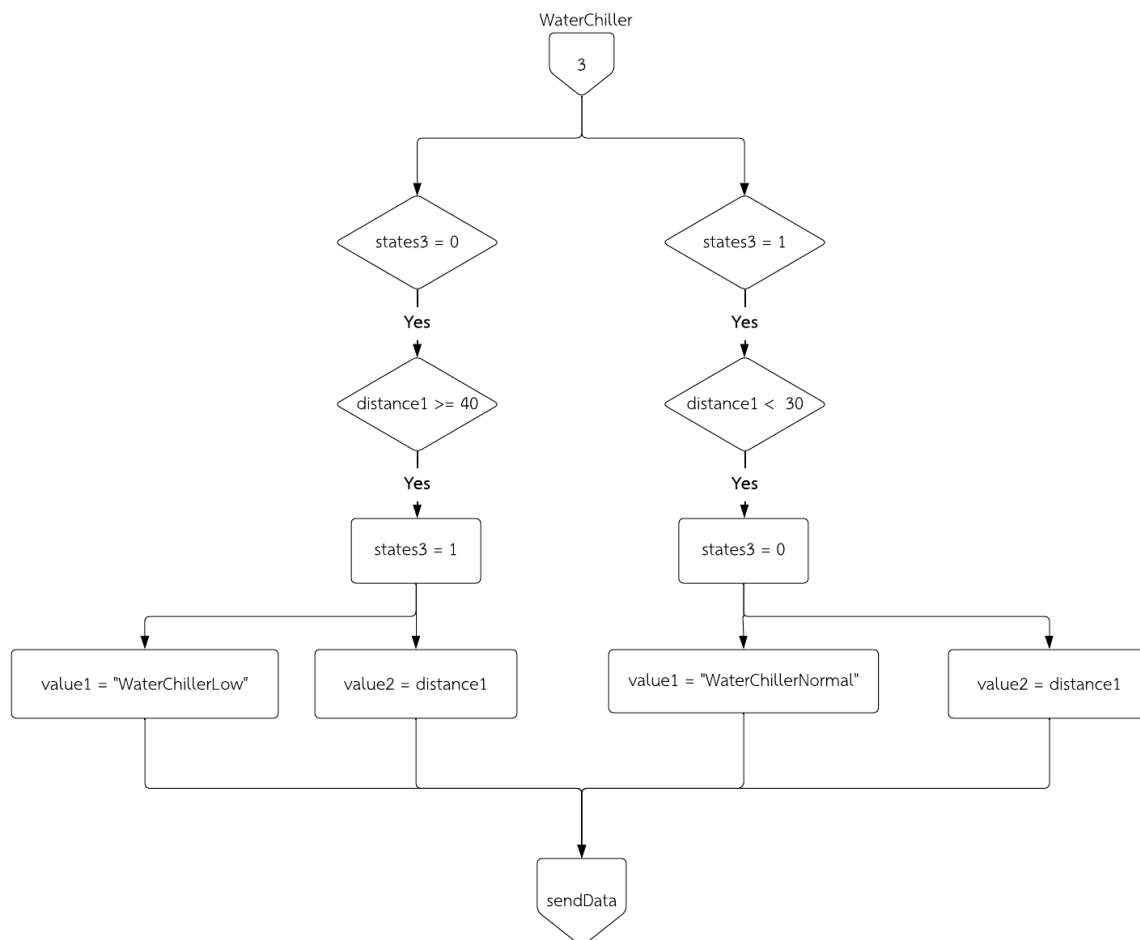
รูปที่ 3.4 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266



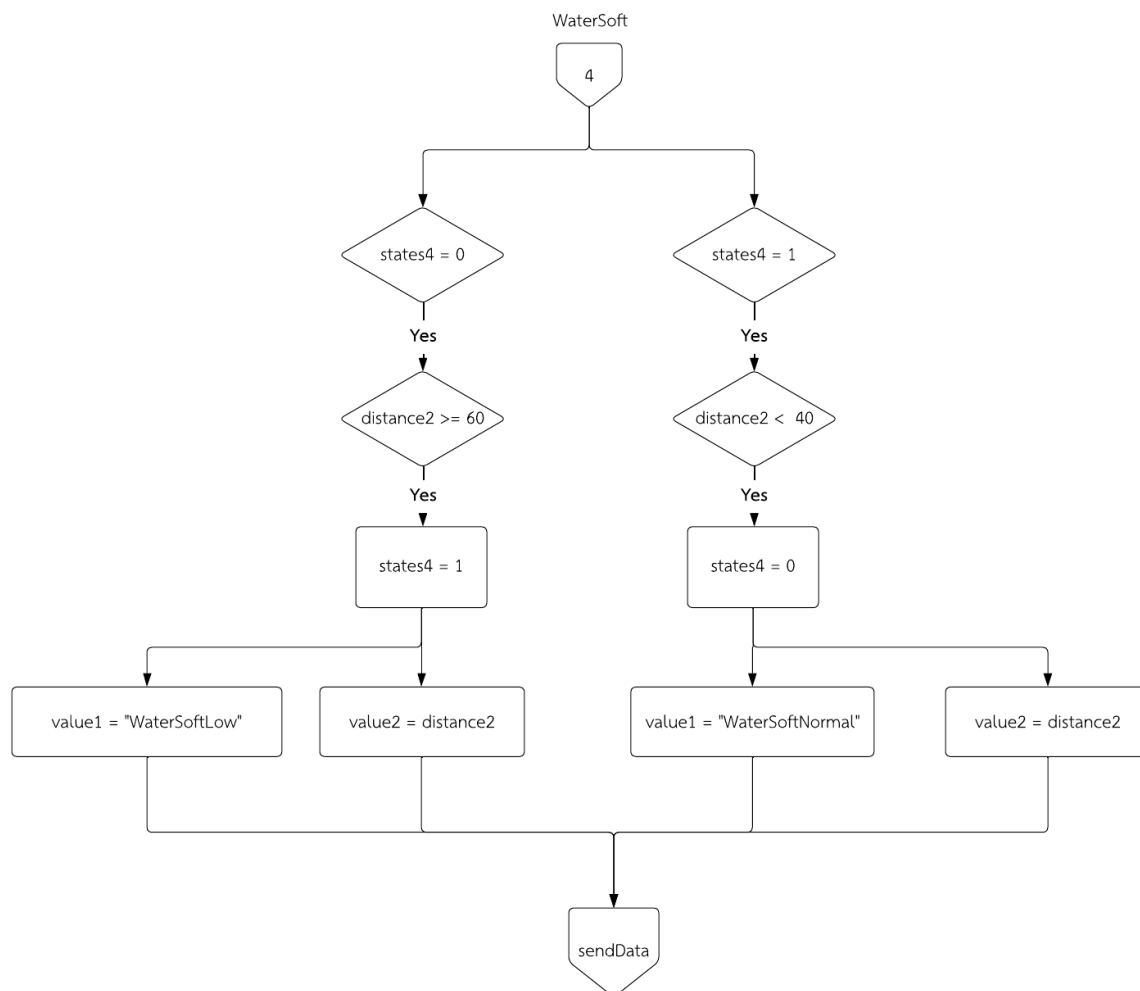
รูปที่ 3.5 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266 (Temperature)



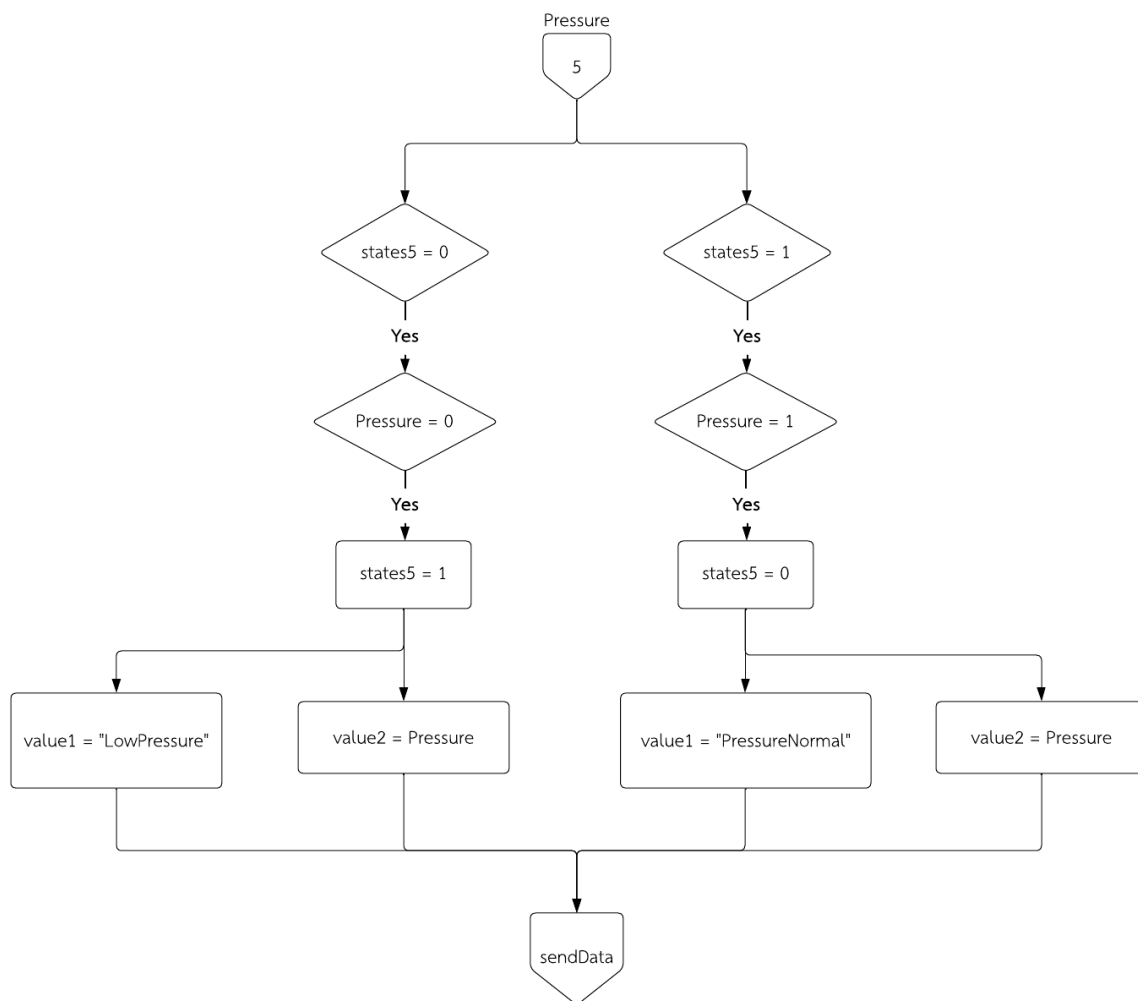
รูปที่ 3.6 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาในระบบน้ำ Chiller ถังลม และระบบน้ำ Soft บนบอร์ด ESP8266 (Flowswitch)



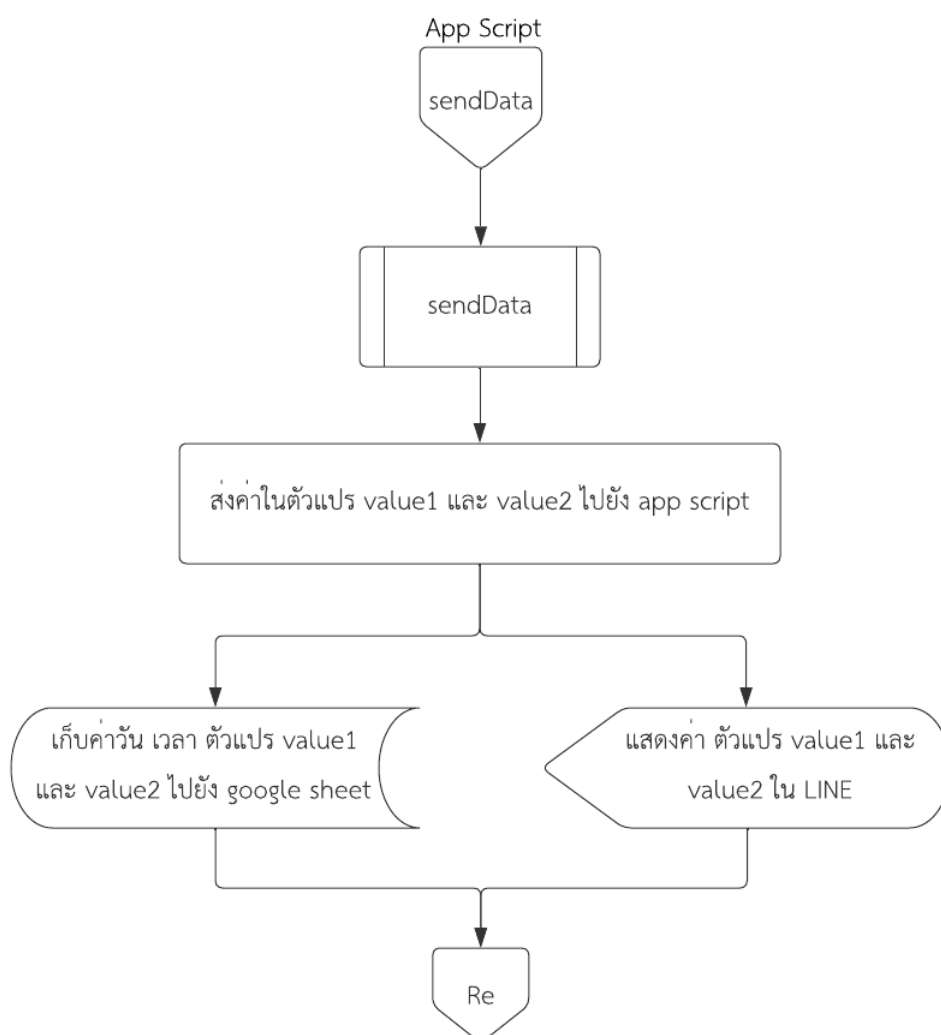
รูปที่ 3.7 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนบอร์ด ESP8266 (WaterChiller)



รูปที่ 3.8 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft บนบอร์ด ESP8266 (WaterSoft)



รูปที่ 3.9 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft บนมอร์ต ESP8266 (Pressure)



รูปที่ 3.10 การออกแบบแผนผังการทำงานของโปรแกรมแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft บน App Script

3.3.1 การเขียนโปรแกรม Arduino IDE เพื่ออ่านค่าและส่งค่าsensor ลงบนบอร์ด

Arduino UNO+WiFi R3 ATmega328P+ESP8266

```
#include <modbus.h>
#include <modbusDevice.h>
#include <modbusRegBank.h>
#include <modbusSlave.h>
#include "Max6675.h"
#include <SoftwareSerial.h>
#include <avr/wdt.h>

#define trigPin1 4 //Ultrasonic1
#define echoPin1 5 //Ultrasonic1
#define trigPin2 6 //Ultrasonic2
#define echoPin2 7 //Ultrasonic2

SoftwareSerial UnoSerial(3, 2); //TX/RX
modbusDevice regBank;
modbusSlave slave;
Max6675 ts(8, 9, 10);

long duration1;
float distance1;
long duration2;
float distance2;
unsigned long period = 5000; //ระยะเวลาที่ต้องการรอ
unsigned long last_time = 0;

void setup()
{
    regBank.setId(1);
    regBank.add(30001);
    regBank.add(30002);
    regBank.add(30003);
    regBank.add(30004);
    pinMode(trigPin1, OUTPUT);
    pinMode(trigPin2, OUTPUT);
    pinMode(echoPin1, INPUT);
    pinMode(echoPin2, INPUT);
    pinMode(3, INPUT);
    pinMode(2, OUTPUT);
    ts.setOffset(0);
    slave._device = &regBank;
    slave.setBaud(9600);
    UnoSerial.begin(57600);
    wdt_enable(WDTO_2S);
}
```

```

void loop()
{
    float temperature = ts.getCelsius();
    float t = temperature*100;

    digitalWrite(trigPin1, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin1, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin1, LOW);
    duration1 = pulseIn (echoPin1, HIGH);
    distance1 = duration1*0.034/2;
    float Level1 = 80-distance1;
    float High1 = Level1/100;
    float Volume1 = 0.8*1.5*High1*1000;
    digitalWrite(trigPin2, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin2, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin2, LOW);
    duration2 = pulseIn (echoPin2, HIGH);
    distance2 = duration2*0.034/2;
    float Level2 = 160-distance2;
    float Volume2 = (3000/160)*Level2;

    if(distance1 > 0 && distance2 > 0 )
    {
        regBank.set(30001, t);
        regBank.set(30002, Volume1);
        regBank.set(30003, Volume2);
        regBank.set(30004, distance1);
        slave.run();

        if( millis() - last_time > period)
        {
            last_time = millis(); //เซฟเวลาปัจจุบันไว้เพื่อรอจนกว่า millis() จะมากกว่าตัวมันเท่า
period
            UnoSerial.print(temperature);
            UnoSerial.print("\n");
            UnoSerial.print(distance1);
            UnoSerial.print("\n");
            UnoSerial.print(distance2);
            UnoSerial.print("\n");
        }
    }
    wdt_reset();
}

```

3.3.2 การเขียนโปรแกรม Arduino IDE เพื่ออ่านค่าและส่งค่าsensor ลงบนบอร์ด

ESP8266

```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <TridentTD_LineNotify.h>
#include <WiFiClientSecure.h>

#define SSID          "Mee"
#define PASSWORD      "000302000"
#define
LINE_TOKEN          "S9ggJPozYih6UUGhWKjTmn2BvgmFTBVTPW7d8H0cVZx"

const char* host = "script.google.com";
const int httpsPort = 443;
WiFiClientSecure client;
String GAS_ID = "AKfycbz_IBI0Hh6W4AKxDx_eaF-
Y5VdSb065v4yZ5FSbgbk7ElK36KxSlRRiF1EOY6qsfvOnvg";

int states1 = 0;
int states2 = 0;
int states3 = 0;
int states4 = 0;
int states5 = 0;

SoftwareSerial NodeSerial(D2, D3); // RX | TX

void setup() {
  pinMode(D0, OUTPUT);
  pinMode(D2, INPUT);
  pinMode(D3, OUTPUT);
  pinMode(D5, INPUT);
  pinMode(D6, INPUT);

  Serial.begin(9600);
  NodeSerial.begin(57600);
  Serial.println();
  Serial.println("NodeMCU/ESP8266 Run");
  WiFi.begin(SSID, PASSWORD);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  LINE.setToken(LINE_TOKEN);
  client.setInsecure();
}
```

```

void loop() {

    while (NodeSerial.available() > 0)
    {
        float temperature = NodeSerial.parseFloat();
        float distance1 = NodeSerial.parseFloat();
        float distance2 = NodeSerial.parseFloat();

        if (NodeSerial.read() == '\n')
        {
            Serial.print("temperature: ");
            Serial.print(temperature);
            Serial.print(" // ");
            Serial.print("Distance1: ");
            Serial.print(distance1);
            Serial.print(" // ");
            Serial.print("Distance2: ");
            Serial.println(distance2);
        }

        if (states1 == 0) {if (temperature >= 15) {states1 = 1;
sendData("HighTemp",temperature );}}
        if (states1 == 1) {if (temperature < 14) {states1 = 0;
sendData("NormalTemp",temperature );}}
//-----
-----

        if (states3 == 0) {if (distance1 >= 40) {states3 = 1;
sendData("WaterChillerLow",distance1 );}}
        if (states3 == 1) {if (distance1 < 30) {states3 = 0;
sendData("WaterChillerNormal",distance1 );}}
//-----
-----

        if (states4 == 0) {if (distance2 >= 60) {states4 = 1;
sendData("WaterSoftLow",distance2 );}}
        if (states4 == 1) {if (distance2 < 40) {states4 = 0;
sendData("WaterSoftNormal",distance2 );}}
//-----
-----

        int Flowswitch = digitalRead(D5);
        if (states2 == 0) {if (Flowswitch == 0) {states2 = 1;
sendData("PumpFail",Flowswitch );}}
        if (states2 == 1) {if (Flowswitch == 1) {states2 = 0;
sendData("PumpNormal",Flowswitch );}}
//-----
-----

```

```

    int Presure = digitalRead(D6);
    if (states5 == 0) {if (Presure == 0) {states5 = 1;
sendData("LowPresure",Presure );}}
    if (states5 == 1) {if (Presure == 1) {states5 = 0;
sendData("PresureNormal",Presure );}}
//-----
}
if (WiFi.status() != WL_CONNECTED)
    {digitalWrite(D0, LOW);}
else {digitalWrite(D0, HIGH);}
}

void sendData(String value1,float value2) {
    Serial.println("=====");
    Serial.print("connecting to ");
    Serial.println(host);
    if (!client.connect(host, httpsPort)) {
        Serial.println("connection failed");
        return;
    }
    String string1 = value1;
    float string2 = value2;
    String url = "/macros/s/" + GAS_ID + "/exec?name=" + string1 +
"&stat=" + string2 ; // 4 variables
    Serial.print("requesting URL: ");
    Serial.println(url);
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
        "Host: " + host + "\r\n" +
        "User-Agent: BuildFailureDetectorESP8266\r\n" +
        "Connection: close\r\n\r\n");
    Serial.println("request sent");
    while (client.connected()) {
        String line = client.readStringUntil('\n');
        if (line == "\r") {
            Serial.println("headers received");
            break;
        }
    }
    String line = client.readStringUntil('\n');
    if (line.startsWith("{\"state\":\"success\"}")) {
        Serial.println("esp8266/Arduino CI successfull!");
    } else {
        Serial.println("esp8266/Arduino CI has failed");
    }
}
}

```

3.3.3 การเขียนโปรแกรม App Script เพื่อเก็บข้อมูลและแจ้งเตือนไปยัง Line Notify

```
function doGet(e) {
    Logger.log( JSON.stringify(e) );
    var token = 'S9ggJPozYih6UUGhWKjTmn2BvgmFTBVPW7d8H0cVZx'//line
    token
    var result = 'Ok';
    if (e.parameter == 'undefined') {
        result = 'No Parameters';
    }
    else {
        var sheet_id = '1hJ_86tx-hngdXikSqwwi0cCFn03YQrkyg-HJe-
xMei8'; // Spreadsheet ID
        var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
        var newRow = sheet.getLastRow() + 1;
        var rowData = [];
        var Curr_Date = new Date();
        rowData[0] = Curr_Date; // Date in column A
        var Curr_Time = Utilities.formatDate(Curr_Date, "Asia/Bangkok",
'HH:mm:ss');
        rowData[1] = Curr_Time; // Time in column B
        for (var param in e.parameter) {
            Logger.log('In for loop, param=' + param);
            var value = stripQuotes(e.parameter[param]);
            Logger.log(param + ':' + e.parameter[param]);
            switch (param) {
                case 'name':
                    rowData[2] = value; // Temperature in column C
                    result = 'temp Written on column C';
                    break;
                case 'stat':
                    rowData[3] = value; // Temperature in column D
                    result = 'humidity Written on column D';
                    break;
            }
        }
        Logger.log(JSON.stringify(rowData));
        var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
        newRange.setValues([rowData]);
    }
    var message = 'Alarm '+rowData[2]+' Status:'+rowData[3]
    sendLineNotify(message, token)
    return ContentService.createTextOutput(result);
}
function stripQuotes( value ) {
    return value.replace(/^[\'"]|['"]$/g, "");
}
function sendLineNotify(message, token) {
```



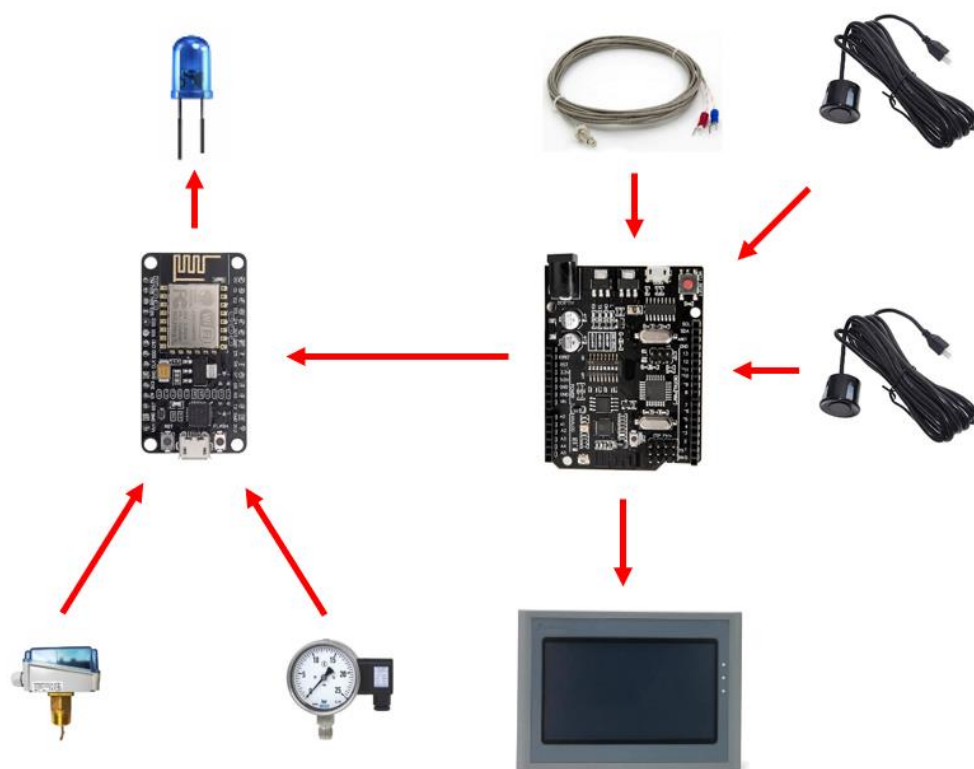
```

var options = {
  "method": "post",
  "payload": {
    "message": message,
  },
  "headers": { "Authorization": "Bearer " + token }
};
UrlFetchApp.fetch("https://notify-api.line.me/api/notify",
options);
}

```

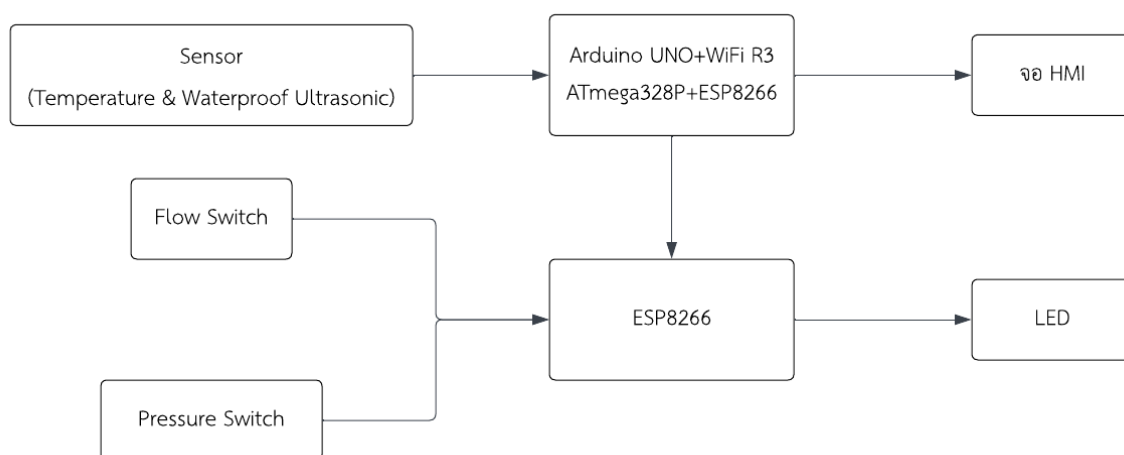
3.4 การออกแบบโครงสร้างฮาร์ดแวร์ของการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

ระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft ประกอบด้วย ชุดSensor วัดอุณหภูมิ ชุดSensor วัดระยะทาง และจอแสดงผล HMI โดยใช้ Arduino UNO+WiFi R3 ATmega328P+ESP8266 ในการประมวลผล



รูปที่ 3.11 การออกแบบโครงสร้างฮาร์ดแวร์ของการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.4.1 บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์ระบบแจ้งเตือนปัญหาระบบน้ำ Chiller ถึง ลม และระบบน้ำ Soft



รูปที่ 3.12 บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์ระบบแจ้งเตือนปัญหาระบบน้ำ Chiller ถึงลม และ
ระบบน้ำ Soft

Arduino UNO+WiFi R3 ATmega328P+ESP8266 จะทำหน้าที่ ประมวลผลการทำงานของ sensor และสั่งการแจ้งเตือนอัตโนมัติ ของระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถึงลม และระบบน้ำ Soft

Thermocouple Type K ใช้งานร่วมกับ MAX6675 จะทำหน้าที่ วัดอุณหภูมิในน้ำ Chiller

Waterproof Ultrasonic Module จะทำหน้าที่ วัดระดับน้ำในถังน้ำ Chiller และ ถังน้ำ Soft

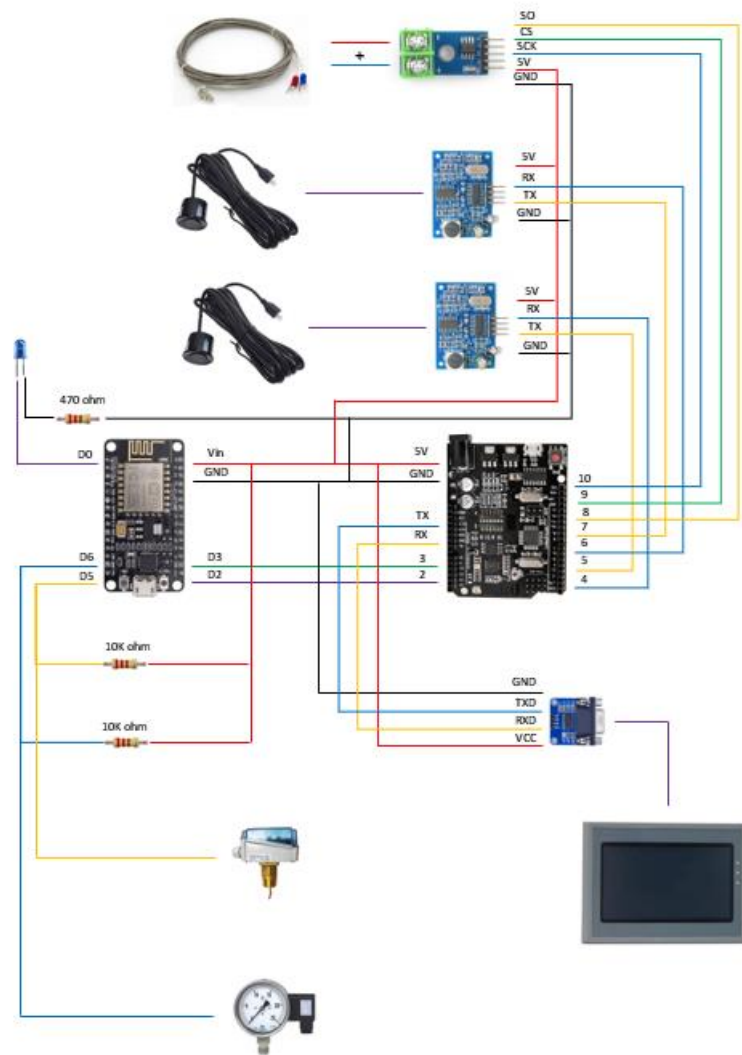
Samkoon SK-050HE HMI Touch Screen ใช้งานร่วมกับ MAX232 RS232 จะทำหน้าที่ แสดงค่า อุณหภูมิ และค่าระดับ

LED จะทำหน้าที่ บอกสถานการณ์เชื่อมต่อ WiFi

Flow Switch จะทำหน้าที่ ตรวจจับการไหลของน้ำในท่อ เพื่อตรวจสอบปั๊มน้ำทำงานหรือไม่

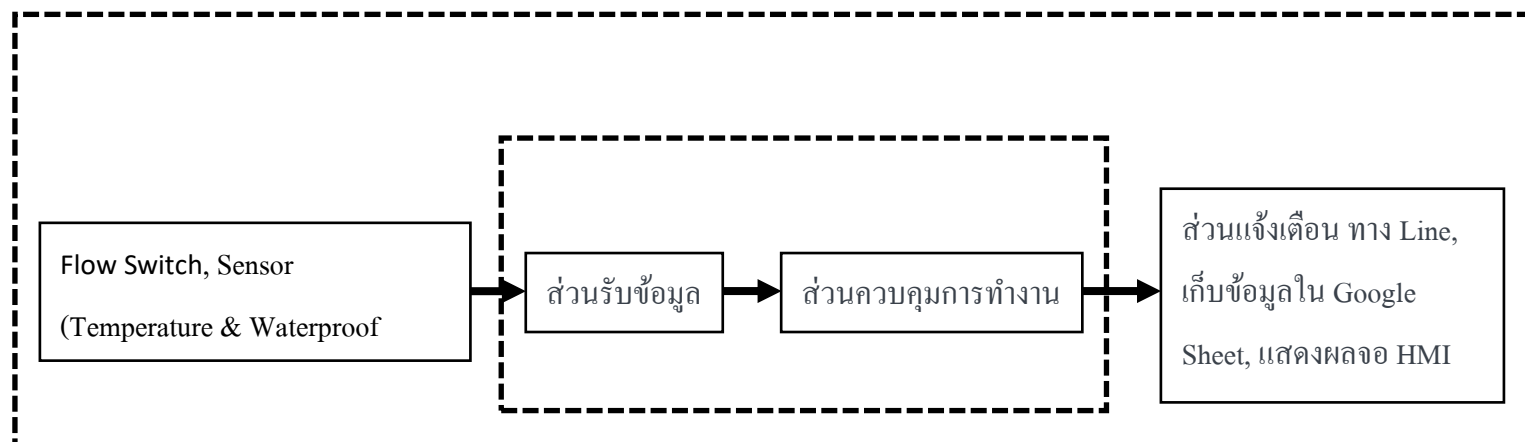
Power Adapter 9V 2A จะทำหน้าที่ เป็นแหล่งจ่ายไฟให้บอร์ด และ sensor

3.4.2 การเชื่อมต่ออุปกรณ์ด้านฮาร์ดแวร์



รูปที่ 3.13 การเชื่อมต่ออุปกรณ์ด้านฮาร์ดแวร์

3.4.3 บล็อกไดอะแกรมแสดงการทำงานของระบบการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft



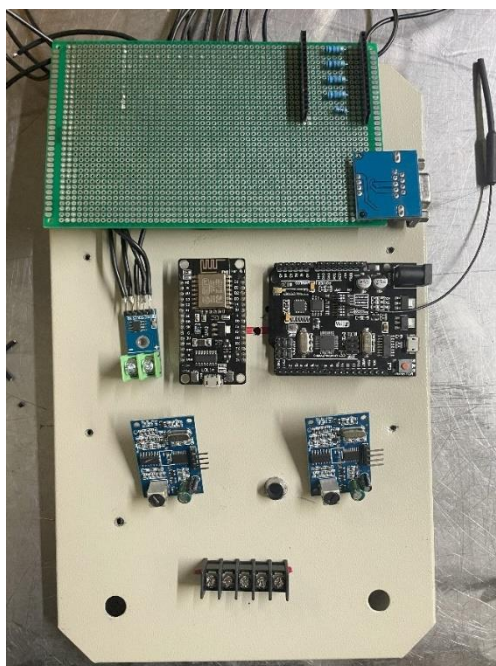
รูปที่ 3.14 บล็อกไดอะแกรมแสดงการทำงานของระบบการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

สำหรับ Flow Switch, Temperature Sensor และ Waterproof Ultrasonic Sensor ประมวลผล โดย Arduino UNO+WiFi R3 ATmega328P+ESP8266 สามารถอธิบายการทำงานของ บล็อกไดอะแกรมแสดงระบบการทำงานของระบบการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบ น้ำซึ่งแบ่งได้เป็น 4 ส่วน ดังนี้

- 1.1 ส่วนป้อนข้อมูล
- 1.2 ส่วนรับข้อมูล
- 1.3 ส่วนประมวลผล
- 1.4 ส่วนแสดงผล

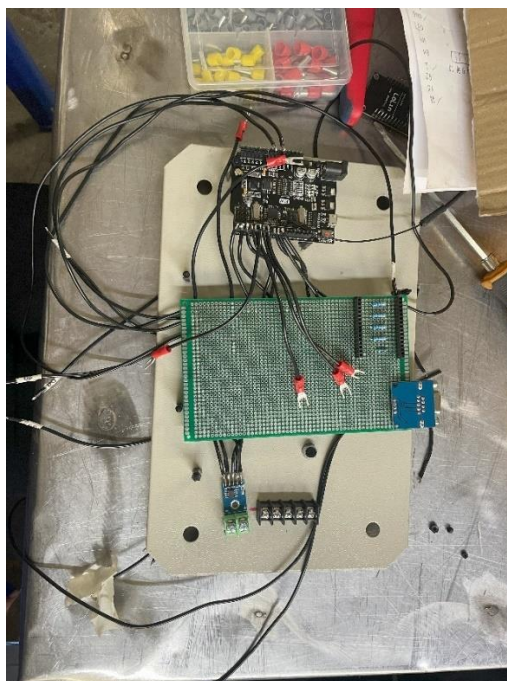
3.5 การติดตั้งอุปกรณ์ลงกล่องจัดเก็บวงจร

3.5.1 การประกอบวงจรแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

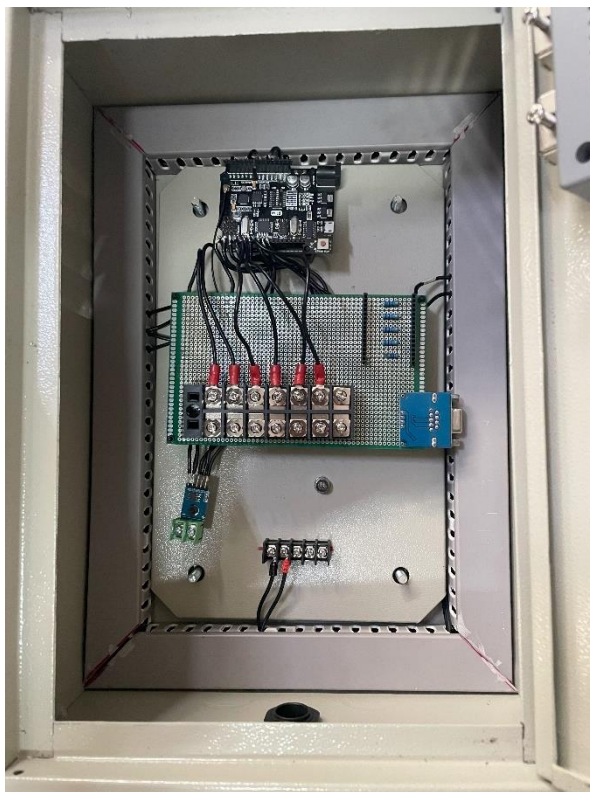


รูปที่ 3.15 การประกอบวงจรแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.5.2 การเชื่อมต่อสายไฟของระบบการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft



รูปที่ 3.16 การเชื่อมต่อสายไฟของระบบการแจ้งเตือนปัญหาระบบน้ำ Chiller ถังลม และระบบน้ำ Soft



รูปที่ 3.17 การยึดอุปกรณ์กับวงจรแรงดันไฟฟ้าระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

3.5.3 การยึดวงจรกับกล่อง



รูปที่ 3.18 การประกอบจอ Hmi Samkoon SK-050HE Touch Screen Touch screen เข้ากับตู้



รูปที่ 3.19 การประกอบขาเซิร์ฟเวอร์ติดตั้งตู้



รูปที่ 3.20 การประกอบเซ็นเซอร์วัดระยะทาง Waterproof Ultrasonic Module (JSN-SR04T) เข้ากับ
กล่องและท่อ



รูปที่ 3.21 เซ็นเซอร์วัดระยะทาง (JSN-SR04T) วัดระดับน้ำ Chiller



รูปที่ 3.22 ภายในของกล่องใส่เซ็นเซอร์วัดระยะทาง (JSN-SR04T) วัดระดับน้ำ Chiller



รูปที่ 3.23 ภายในของกล่องใส่เซ็นเซอร์วัดระยะทาง (JSN-SR04T) วัดระดับน้ำ Soft



รูปที่ 3.24 ต่อเซ็นเซอร์วัดอุณหภูมิกับเซ็นเซอร์วัดระยะทาง (JSN-SR04T) เข้ากับถังน้ำ Chiller

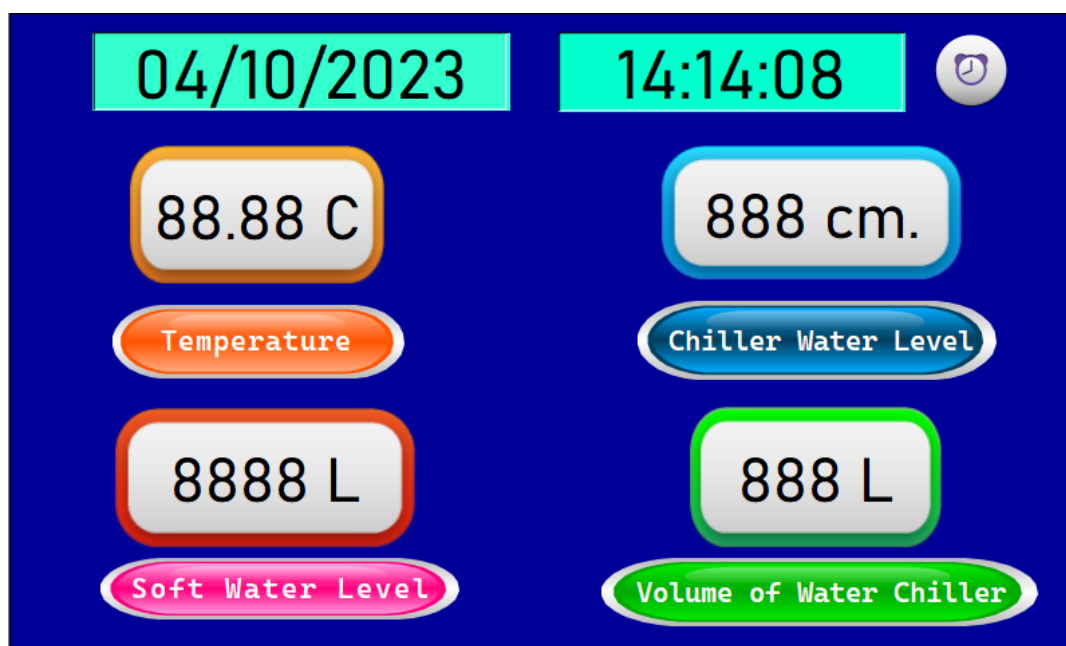


รูปที่ 3.25 ต่อเซ็นเซอร์วัดระยะทาง (JSN-SR04T) เข้ากับถังน้ำ Soft



รูปที่ 3.26 ต่อเซ็นเซอร์วัดระยะทาง (JSN-SR04T) เข้ากับถังน้ำ Soft

3.7 การออกแบบจอแสดงผล Hmi Samkoon



รูปที่ 3.28 การออกแบบจอแสดงผล Hmi Samkoon

บทที่ 4

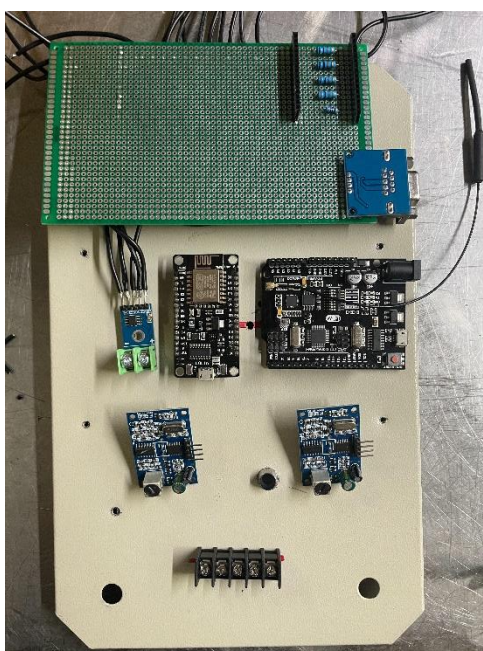
ผลการดำเนินการ

ผลการดำเนินงานระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft คณะผู้จัดทำได้นำข้อมูลจากการทำงานจริงของตัวระบบ มานำเสนอในรูปแบบของรูปภาพและประกอบคำบรรยาย โดยมีหัวข้อดังต่อไปนี้

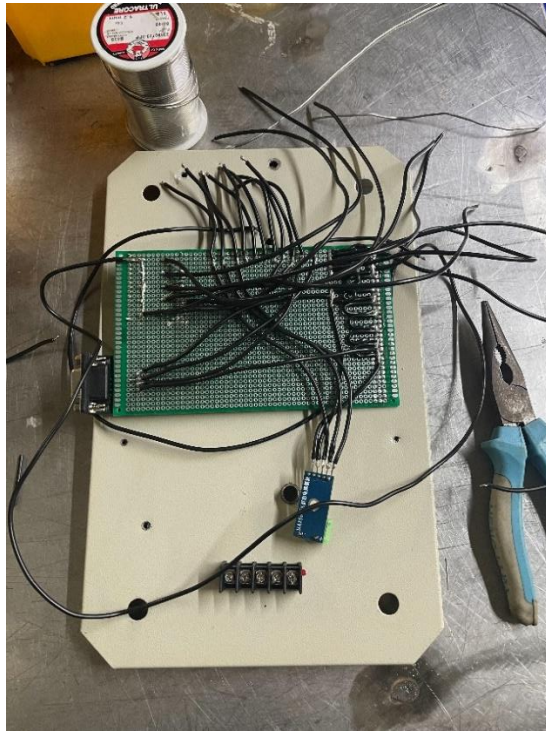
4.1 ผลการต่อวงจรระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft

- 4.2 ผลการติดตั้งวงจรเข้าตู้คอนโทรล
- 4.3 ผลการแสดงค่าบนหน้าจอ Touch screen
- 4.4 ผลการเก็บข้อมูล
- 4.5 ต้นทุนในการประเมินการสร้างหรือผลิตนวัตกรรม

4.1 ผลการต่อวงจรระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft



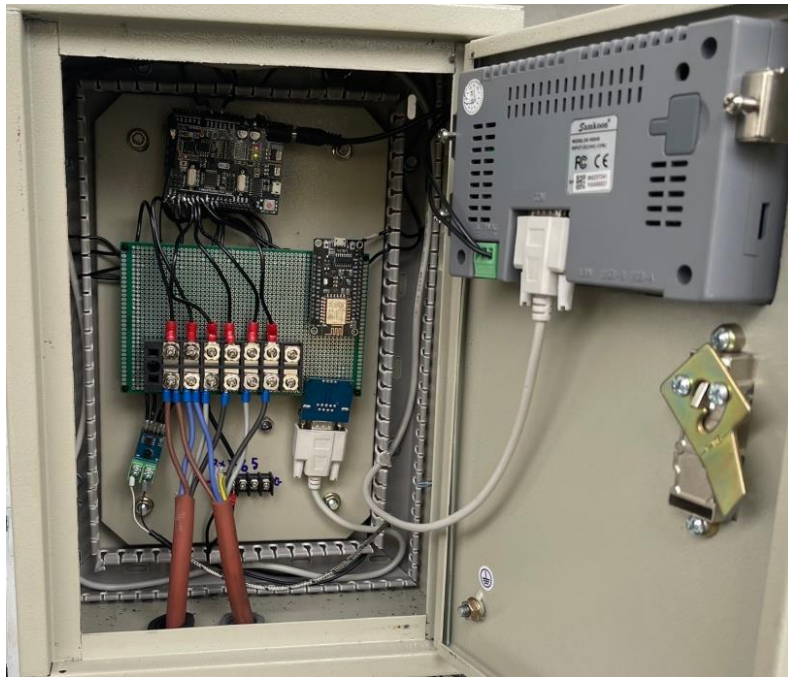
รูปที่ 4.1 การเตรียมอุปกรณ์



รูปที่ 4.2 การต่อวงจรและติดตั้งบนแผ่น PCB



รูปที่ 4.3 การติดตั้งจอ Touch screen



รูปที่ 4.4 การประกอบอุปกรณ์พร้อมต่อสายวงจร

4.2 ผลการติดตั้งวงจรเข้าตู้คอนโทรล



รูปที่ 4.5 ด้านหน้าของตู้คอนโทรล



รูปที่ 4.6 ด้านล่างของตู้คอนโทรล

4.3 ผลการแสดงผลหน้าจอ Hmi Samkoon Touch Screen



รูปที่ 4.7 ผลการแสดงผลหน้าจอ Hmi Samkoon Touch Screen

4.4 ผลการเก็บข้อมูล

Date	Time	Name	Status
5/10/2023	1:09:38	HighTemp	16
5/10/2023	1:10:01	NormalTemp	14
6/10/2023	1:10:24	HighTemp	17
6/10/2023	1:10:47	NormalTemp	14
6/10/2023	1:10:06	WaterChillerLow	42
6/10/2023	1:10:11	WaterChillerNormal	31
6/10/2023	1:10:17	WaterSoftLow	64
6/10/2023	1:10:51	WaterSoftNormal	40
6/10/2023	1:10:58	PumpFail	0
6/10/2023	1:11:04	PumpNormal	1
6/10/2023	1:11:11	LowPressure	0
6/10/2023	1:11:16	PressureNormal	1

รูปที่ 4.8 ผลการเก็บข้อมูล

4.5 ต้นทุนในการประเมินการสร้างหรือผลิตนวัตกรรม

ตารางที่ 4.1 การประเมินราคา

รายการ	จำนวน	ราคา(บาท)
Power Adaptor 9V กระแส 2A หัวแจ๊ค 5.5x2.5mm	1	150
MAX6675 Digital for K-Type Thermocouple	1	150
Thermocouple Type-K ขนาด M6 สายยาว 5 เมตร	1	120
Arduino UNO+WiFi R3 ATmega328P+ESP8266 (32Mb memory)	1	300
LED 5mm หลอดใส สีน้ำเงิน 5 ชิ้น	1	10
ตัวต้านทาน 1/2W 10k Ohm Metal film 1% 10 ชิ้น	1	12
ตัวต้านทาน 470 Ohm 1/4W Metal film 1% 10 ชิ้น	1	5
RS232 To TTL Converter Module MAX3232 MAX232CSE	1	50
บอร์ดทดลอง Breadboard 830 holes MB-102	1	48
Waterproof Ultrasonic Module แบบมีสาย (JSN-SR04T)	2	285
สายไฟ jumper ผู้-เมีย ยาว 15 cm. 40 เส้น	1	30
ก้างปลา Pin Header Connector Male 1x40 Pin 2.54mm	2	5
ปลอก LED 5mm ฝารอบ ซุปโครเมียม	1	5
สาย USB 2.0 (Type A To Type B) ยาว 1 เมตร	1	45

สาย RS232 9Pin ผู้ - เมีย ยาว 1.5 เมตร	1	35
SK-050HE Samkoon HMI Touch Screen จอ 5 นิ้ว	1	3000
สวิตชิงเพาเวอร์ซัพพลาย Switching Power Supply 24V 1A 25W	1	120
ก้างปลาตัวเมีย , ก้างปลาตัวผู้ 40 PIN	1	6
7.62 mm. KF7.62 MG 762 spliced ขั้วต่อขั้วต่อสกรูสีดำสีเขียว 5pin	1	15
สายจัมป์ 20 cm. ผู้-เมีย (แผงละ 40 เส้น)	1	25
กล่อง ขนาด 4x4 กันน้ำ	2	50
PCB แผ่นปรินต์ แผ่นวงจร สีเขียว 9x15 cm.	1	35
น็อตตัวผู้หัวกลม น็อตในลอนพลาสติก M2x6 10ตัว	1	23
น็อตตัวเมีย น็อตในลอนพลาสติก M2 10ตัว	1	25
Spacer MF เสารองบอร์ด ในลอน ผู้เมีย พลาสติก M2x10+6 10ตัว	2	26
น็อตตัวผู้หัวเตปเปอร์ น็อตในลอนพลาสติก M3x6 10ตัว	2	25
น็อตตัวเมีย น็อตในลอนพลาสติก M3 10ตัว	2	23
Spacer MF เสารองบอร์ด ในลอน ผู้เมีย พลาสติก M3x15+6 10ตัว	2	28
เคเบิลแกนลวด ขั้วต่อกันน้ำ PG7	4	3
เคเบิลแกนลวด ขั้วต่อกันน้ำ PG13.5	2	6
waterproof junction box 4*4	2	50
รวม		5,217

บทที่ 5

สรุปผลและข้อเสนอแนะ

โครงการระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft ได้จัดทำเพื่อสามารถรู้ปัญหาและแก้ไขได้ทันที่ที่ ไม่กระทบต่อการทำงานของเครื่องจักร ลดปัญหา Down Times เครื่องจักรหยุด เพิ่มคุณภาพผลิตภัณฑ์สม่ำเสมอที่ ทางผู้จัดทำจึงทำระบบนี้ขึ้นมาให้ สะดวกต่อการเข้าทำงานของช่างและลดการทำงานของพนักงาน

5.1 สรุปผล

จากการทดลองสรุปได้ว่า ระบบ IOT แจ้งเตือนการทำงานผิดปกติของระบบน้ำ Chiller ถังลม และระบบน้ำ Soft สามารถวัดค่าอุณหภูมิและระดับน้ำแสดงบนจอ Touch screen แจ้งเตือนทาง Line และเก็บข้อมูลได้จริง ซึ่งเป็นไปตามวัตถุประสงค์ของโครงการ

ผลจากการวัดระดับน้ำด้วยเซ็นเซอร์วัดระยะทางอัลตราโซนิก ผลออกมาสามารถอ่านค่าความสูงของระดับน้ำได้ใกล้เคียงกับความสูงของระดับน้ำจริงโดยมีค่าความผิดพลาดคลาดเคลื่อนอยู่ในช่วงที่ยอมรับได้ ซึ่งค่าความผิดพลาดนี้ไม่ได้ส่งผลกระทบต่อการทำงานของระบบ

5.2 ปัญหาและข้อเสนอแนะ

1. การใช้ sensor waterproof ultrasonic ในการวัดระดับน้ำแสดงค่าไม่เสถียร
2. การต่อวงจรแบบใช้การบัดกรีสายไฟ จะทำให้วงจรนั้นแกหรือเปลี่ยนอุปกรณ์ไม่สะดวก เมื่อเทียบกับการต่อ เทอร์มินอลบล็อก

5.3 แนวทางพัฒนาต่อยอด

พัฒนาการต่อสัญญาณผ่านจอ Touchscreen Samkoon เพื่อสั่งการทำงานของระบบน้ำ กำหนดอุณหภูมิ กำหนดประมาตรของน้ำ

เอกสารอ้างอิง

- [1] “การใช้งาน Arduino UNO+WiFi R3 ATmega328P+ESP8266” [Online]. Available : <https://www.cybertice.com/product/1723/> [Accessed: 11 กรกฎาคม 2566].
- [2] “การใช้งาน Waterproof Ultrasonic Module” [Online]. Available : <https://www.cybertice.com/product/638/waterproof-ultrasonic-module> [Accessed: 12 กรกฎาคม 2566].
- [3] “การใช้งาน MAX6675 ร่วมกับ Thermocouple Type K” [Online]. Available : <https://www.cybertice.com/product/893/max6675-digital-converter-module-for-k-type-thermocouple-max6675> [Accessed: 13 กรกฎาคม 2566].
- [4] “การเขียนโค้ดโปรแกรมเก็บข้อมูลใน Google Sheet” [Online]. Available : https://www.youtube.com/watch?v=QDR-t8xoUTM&list=PL7IkDWFWSJPxYJCagqXFmpcgg3f-pFHoo&index=22&t=2s&ab_channel=SPEDY [Accessed: 22 สิงหาคม 2566].
- [5] “การส่ง-รับข้อมูลระหว่าง Board Arduino R3 กับ Board NodeMCU ESP8266” [Online]. Available : <https://www.cybertice.com/article/> [Accessed: 2 ตุลาคม 2566].
- [6] “การเขียนโค้ดแสดงค่าบนจอ Hmi Samkoon Touch Screen” [Online]. Available : <https://www.youtube.com/watch?v=kTnSnwv0Pjo&list> [Accessed: 4 ตุลาคม 2566].

ประวัติผู้จัดทำรายงาน

ชื่อ – ชื่อสกุล

นายพีรวัส เจียจำรูญ

รหัสนักศึกษา

630910483

สถานที่อยู่ปัจจุบัน

100/5 หมู่ 10 ตำบลปากแพรก

อำเภอเมืองกาญจนบุรี จังหวัดกาญจนบุรี

71000



ประวัติการศึกษา

ระดับมัธยมศึกษาปลาย

โรงเรียนวิสุทธิรังสี จังหวัดกาญจนบุรี

ระดับปริญญาตรี

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยศิลปากร

ประวัติผู้จัดทำรายงาน

ชื่อ ชื่อสกุล

สริต วนาวิเศษศักดิ์

รหัสนักศึกษา

630910590

สถานที่อยู่ปัจจุบัน

40/3 หมู่2 ตำบลสร้อยฟ้า

อำเภอโพธาราม จังหวัดราชบุรี

70120



ประวัติการศึกษา

ระดับมัธยมศึกษาตอนปลาย

โรงเรียนสารสิทธิ์พิทยาลัย

ระดับปริญญาตรี

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยศิลปากร