เครื่องซักผ้าอัจฉริยะ

SMART WASHING MACHINE

ณัฐนรี ปธานราษฎร์
อนุเทพ แสงงาม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เครื่องซักผ้าอัจฉริยะ

SMART WASHING MACHINE

ณัฐนรี ปธานราษฎร์
อนุเทพ แสงงาม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART WASHING MACHINE

NATNAREE PATANRASD
ANUTHEP SANGNGAM

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องซักผ้าอัจฉริยะ

SMART WASHING MACHINE

ผู้จัดทำ นางสาวณัฐนรี ปธานราษฎร์ 58010394

นายอนุเทพ แสงงาม 58011399

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรมณีล้ำค่า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องซักผ้าอัจฉริยะ

โดย

นางสาวณัฐนรี ปธานราษฎร์ 58010394

นายอนุเทพ แสงงาม 58011399

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรมณีล้ำค่า

ปีการศึกษา 2561

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบเครื่องซักผ้าสาธารณะ ให้มีคุณสมบัติที่ตอบสนองต่อความต้องการของผู้ใช้งาน โดยนำเทคโนโลยี Internet of Things (IoT) มาประยุกต์ใช้เพื่อพัฒนาต่อยอดเครื่องซักผ้าสาธารณะที่มีใช้งานอยู่ในปัจจุบัน โดยเครื่องซักผ้าสาธารณะที่พัฒนาขึ้นนั้น ผู้ใช้งานสามารถรับข้อมูลการรอใช้งาน สั่งงานเครื่องซักผ้าโดยเลือกฟังก์ชันการทำงานได้ รวมถึงการแจ้งเตือนเมื่อเครื่องซักผ้ามีปัญหา หรือทำงานเสร็จผ่านระบบแอปพลิเคชันบนสมาร์ทโฟนหรือแท็บเล็ตได้

ขั้นตอนในการดำเนินงานเริ่มจากศึกษาการทำงานของเครื่องซักผ้าฝาบน ศึกษาและออกแบบวงจรควบคุมการทำงานของเครื่องซักผ้าประกอบด้วยบอร์ดควบคุม ESP32 เซนเซอร์ตรวจวัดกระแส ACS712 และโมดูลรีเลย์สำหรับควบคุมการทำงานเครื่องซักผ้า ทำการเขียนโปรแกรมด้วย Arduino เพื่อควบคุมการทำงานเครื่องซักผ้า จากนั้นเขียนโปรแกรมด้วยแอปพลิเคชัน Blynk เพื่อทำการเชื่อมต่อกับระบบเครือข่ายแบบไร้สายและสามารถใช้งานบนสมาร์ทโฟนได้ สุดท้ายทดสอบระบบการทำงานทั้งหมดและปรับแก้ไขให้ทำงานตามที่ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART WASHING MACHINE

By

Miss Natnaree Patanrasd 58010394

Mr. Anuthep Sangngam 58011399

Advisor

Asst.Prof.Dr. Wandee Petchmaneelumka

Academic Year 2018

ABSTRACT

This thesis presents a design of public washing machine which can be met the client's demands. Internet of Things (IoT) has been merged with existing public washing machine for development. Client can use the public washing machine developed via smartphone or tablet which includes the features such as waiting for its availability, selecting the desired working function and notifications for problem machine and finished work.

Procedure of the operation begins from studying the top lip washing machine operation. The control circuit including ESP32 control board, ACS712 current sensing module and relay module for control washing machine is then studied and designed. After that, Arduino program is used to write code for the operation control of washing machine. Then, Blynk application is employed for interface with Wi-Fi network via smartphone. Finally, whole system is tested and modified to operate as designed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ประสบผลสำเร็จไปได้ด้วยดี ทั้งนี้เนื่องจากคำแนะนำของอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรมณีล้ำค่า และอาจารย์ท่านอื่นๆ ในหลักสูตรวิศวกรรมระบบควบคุม ภาควิชาวิศวกรรมการวัดและควบคุม ผู้จัดทำขอขอบพระคุณในความอนุเคราะห์จากอาจารย์ทุกท่านที่ช่วยเหลือในการทำโครงงานนี้

ขอขอบคุณเพื่อนๆ ทุกคนที่ช่วยเหลือและให้ข้อแนะนำในการทำโครงงานนี้ จึงทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

คณะผู้จัดทำ
ณัฐนรี ปธานราษฎร์
อนุเทพ แสงงาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	**1**
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	2
1.3 ขอบเขตของโครงงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 รายละเอียดของปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	**4**
2.1 กล่าวนำ	4
2.2 อุปกรณ์ที่เกี่ยวข้อง	4
2.2.1 ระบบสื่อสารสัญญาณไร้สายผ่าน WiFi Module ESP32	4
2.2.2 ตัวสั่งการเครื่องใช้ไฟฟ้า 220V (โมดูลรีเลย์ 5V 4 ช่องสัญญาณ)	6
2.2.3 ไอซีวัดกระแสในตระกูล ACS712	8
2.3 โปรแกรมที่เกี่ยวข้อง	9
2.3.1 โปรแกรม Sketch Arduino IDE	9
2.3.2 แอปพลิเคชัน Blynk	10
2.3.2.1 อุปกรณ์ภายในแอปพลิเคชัน Blynk (Widget Box)	12
2.3.2.2 การใช้งานแอปพลิเคชัน Blynk	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีการดำเนินงาน	**18**
3.1 ขั้นตอนการดำเนินงาน	18
3.1.1 ขั้นตอนของการดำเนินงานได้วางแผนไว้ดังนี้	18
3.2 อุปกรณ์ที่เกี่ยวข้อง	18
3.2.1 บอร์ด ESP32	18
3.2.2 โมดูลตรวจจับกระแส ACS712	19
3.2.3 โมดูลรีเลย์ 5V 10A	19
3.2.4 แหล่งจ่ายไฟ 5V 3A	20
3.3 ซอฟต์แวร์ที่เกี่ยวข้อง	20
3.3.1 โปรแกรม Arduino IDE	20
3.3.2 แอปพลิเคชัน Blynk	21
3.4 การออกแบบและการวางแผนการทำงาน	22
3.4.1 การออกแบบและการวางแผนทางด้านฮาร์ดแวร์	22
3.4.2 การออกแบบและการวางแผนทางด้านซอฟต์แวร์	22
3.5 วิธีการดำเนินงาน	23
3.5.1 การติดตั้งเซนเซอร์	23
3.5.2 การติดตั้งโมดูลรีเลย์กับแผงควบคุม	25
3.5.3 การติดตั้งแหล่งจ่ายไฟ	28
3.5.4 การติดตั้งส่วนประกอบต่างๆ ลงตู้	30
3.5.5 การเขียนโปรแกรม Arduino	31
3.5.6 การเขียนโปรแกรมแสดงผลบนแอปพลิเคชัน Blynk	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	**35**
4.1 แผนผังการทำงานของเครื่องซักผ้าอัจฉริยะผ่านเครือข่ายไร้สาย	35
4.2 ผลการออกแบบโปรแกรมแสดงผลผ่านแอปพลิเคชัน Blynk	35
4.2.1 ส่วนของหน้าจอเลือกคำสั่ง	35
4.2.2 ส่วนของหน้าจอแสดงผล	35
4.3 ผลการประดิษฐ์เครื่องซักผ้าอัจฉริยะ	36
4.4 การทดสอบการรับ-ส่งข้อมูลระหว่างเครื่องซักผ้าและแอปพลิเคชัน Blynk	37
4.4.1 ส่วนของผลเมื่อการทำงานของเครื่องซักผ้าเสร็จสิ้น	37
4.4.2 ส่วนของผลเมื่อการทำงานของเครื่องซักผ้ามีปัญหา	37
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	**39**
5.1 สรุปผลการดำเนินงาน	39
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	39
5.3 แนวทางการแก้ไขปัญหา	40
เอกสารอ้างอิง	**39**
ภาคผนวก	**42**
ภาคผนวก ก โมดูลตรวจจับกระแส ACS712	43
ภาคผนวก ข บอร์ด ESP32	45
ภาคผนวก ค โปรแกรมควบคุมการทำงานและการแสดงผล	47
ภาคผนวก ง โปสเตอร์	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 บอร์ด ESP32	5
2.2 ขาของบอร์ด ESP32	6
2.3 โมดูลรีเลย์ 5V 4 ช่องสัญญาณ	7
2.4 ตัวอย่างการต่อระหว่างโมดูลรีเลย์ และบอร์ด Arduino	8
2.5 ไอซีวัดกระแส ACS712	9
2.6 หน้าโปรแกรม Arduino IDE	10
2.7 ตัวอย่างหน้าแอปพลิเคชัน Blynk	11
2.8 สถาปัตยกรรมของแอปพลิเคชัน Blynk	11
2.9 หน้าตา Widget Box ของ Blynk (1)	12
2.10 หน้าตา Widget Box ของ Blynk (1)	13
2.11 แอปพลิเคชัน Blynk ในระบบการทำงาน Android และ IOS	14
2.12 ขั้นตอนการสร้างบัญชีผู้ใช้งานใหม่	15
2.13 ขั้นตอนการสร้างโปรเจคใหม่	15
2.14 ตัวอย่าง Blynk Energy และจำนวน Energy ที่อุปกรณ์ภายในแอปพลิเคชันต้องการใช้	16
2.15 ตัวอย่างโค้ดสำหรับการเชื่อมต่อแอปพลิเคชัน Blynk ด้วยโปรแกรม Arduino IDE	16
2.16 ภาพรวมวิธีการใช้งาน Blynk	17
3.1 บอร์ด ESP32	18
3.2 ขา ADC ที่สามารถใช้งานได้บน บอร์ด ACS712	19
3.3 โมดูลตรวจจับกระแส ACS712	19
3.4 การเดินสายของโมดูลรีเลย์ 5V 10A กับบอร์ด ESP32 และอุปกรณ์อื่นๆ	20
3.5 แหล่งจ่ายไฟ 5V 3A	20
3.6 หน้าต่างของโปรแกรม Arduino	21
3.7 แอปพลิเคชัน Blynk	21
3.8 อุปกรณ์ที่ใช้ในการติดตั้งเซนเซอร์	23
3.9 สายควบคุมวาล์วน้ำเข้าและสายควบคุมวาล์วน้ำออก	23
3.10 สายควบคุมวาล์วน้ำเข้าและสายควบคุมวาล์วน้ำออกที่ตัดแล้ว	24
3.11 การต่อสายจาก ACS712 เข้าหัวแปลงขั้ว	24
3.12 การต่อสายเข้า ACS712	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.13 อุปกรณ์ที่ใช้ในการติดตั้งโมดูลรีเลย์กับแผงควบคุม	25
3.14 การต่อสายไฟเข้ากับสวิตช์ควบคุมเครื่องซักผ้า	26
3.15 การเก็บสายไฟเพื่อรอสำหรับการต่อเข้ากับโมดูลรีเลย์	26
3.16 การต่อสายไฟเข้ากับรีเลย์ (1)	27
3.17 การต่อสายไฟเข้ากับรีเลย์ (2)	27
3.18 การต่อสายไฟจากรีเลย์เข้าบอร์ด ESP32	27
3.19 อุปกรณ์ที่ใช้ในการติดตั้งแหล่งจ่ายไฟ	28
3.20 การติดตั้งแหล่งจ่ายไฟ (1)	29
3.21 การติดตั้งแหล่งจ่ายไฟ (2)	29
3.22 การติดตั้งส่วนประกอบต่างๆ ลงตู้	30
3.23 การประกาศค่าอินพุตที่จะรับค่าจากเซนเซอร์	31
3.24 การกำหนดความเร็วในการสื่อสารข้อมูลและกำหนดโหมดในการใช้ข้อมูล	31
3.25 การกำหนดวงรอบของการทำงานของโปรแกรม	32
3.26 โปรแกรมอ่านค่าเซนเซอร์ ACS712 (1)	32
3.27 โปรแกรมอ่านค่าเซนเซอร์ ACS712 (2)	32
3.28 การเขียนโปรแกรมแสดงผลบนแอปพลิเคชัน Blynk (1)	33
3.29 การเขียนโปรแกรมแสดงผลบนแอปพลิเคชัน Blynk (2)	33
3.30 การเขียนโปรแกรมแสดงผลบนแอปพลิเคชัน Blynk (3)	34
4.1 แผนผังการทำงานของเครื่องซักผ้าอัจฉริยะผ่านเครือข่ายไร้สาย	35
4.2 หน้าแอปพลิเคชัน Blynk	36
4.3 กล่องควบคุมประกอบติดกับเครื่องซักผ้า	36
4.4 การแจ้งเตือนของแอปพลิเคชันผ่านหน้าจอสมาร์ทโฟน (1)	37
4.5 การแจ้งเตือนของแอปพลิเคชันผ่านหน้าจอสมาร์ทโฟน (2)	38
4.6 การแจ้งเตือนของแอปพลิเคชันผ่านระบบอีเมล์ (1)	38
4.7 การแจ้งเตือนของแอปพลิเคชันผ่านระบบอีเมล์ (2)	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในปัจจุบันการซักผ้าเป็นหนึ่งในกิจวัตรประจำวันหรือประจำสัปดาห์สำหรับผู้คนโดยทั่วไป แต่ก็มีผู้คนอีกไม่น้อยที่มีข้อจำกัดบางประการทำให้ไม่สามารถมีเครื่องซักผ้าส่วนตัวได้ เครื่องซักผ้าสาธารณะจึงเข้ามามีบทบาทกับผู้ใช้งานกลุ่มนี้ในการเพิ่มความสะดวกสบายในการซักผ้าแทนการซักด้วยมือ เครื่องซักผ้าสาธารณะปัจจุบันมีหลากหลายประเภท โดยแต่ละประเภทก็ต่างมีค่าธรรมเนียมในการใช้บริการแตกต่างกันไปโดยขึ้นอยู่กับต้นทุนของเครื่องซักผ้า จากผลการสำรวจของ บริษัท โฮม โปรดักส์ เซ็นเตอร์ จำกัด พบว่าเครื่องซักผ้าฝาบนมีราคาที่ต่ำ ระยะเวลาในการซักผ้าใช้น้อย ความจุมาก ค่าบำรุงรักษาต่ำและมีข้อจำกัดเรื่องพื้นที่ในการติดตั้งน้อยกว่าเครื่องซักผ้าประเภทอื่นๆ หากมองเฉพาะในประเทศไทยที่ผู้ใช้งานส่วนใหญ่มีความถนัดในการใช้เครื่องซักผ้าฝาบนมากกว่าเครื่องซักผ้าประเภทอื่นๆ ในทำนองเดียวกันผู้ประกอบการส่วนใหญ่ก็นิยมติดตั้งเครื่องซักผ้าแบบฝาบน พร้อมกับอุปกรณ์รับชำระค่าบริการแบบหยอดเหรียญ เนื่องจากมีต้นทุนที่ต่ำกว่า ส่วนของการทำงานนั้นหลังจากที่ผู้ใช้บริการทำการชำระค่าบริการแล้ว เครื่องซักผ้าจะทำงานโดยอัตโนมัติตามขั้นตอนที่ผู้ประกอบการออกแบบไว้ ด้วยสาเหตุนี้ทำให้ผู้ใช้บริการไม่สามารถสั่งงานเครื่องซักผ้าตามความต้องการของตนเองได้ อีกทั้งเมื่อการซักผ้าดำเนินการอยู่ผู้ใช้งานไม่สามารถตรวจสอบสถานะและปัญหาของเครื่องซักผ้าได้ นอกจากนี้หากการดำเนินการของเครื่องซักผ้าสำเร็จเสร็จสิ้นแล้วผู้ใช้งานก็ไม่สามารถทราบเวลาแน่นอนได้ เนื่องจากการนับเวลาหน้าเครื่องซักผ้าเป็นเพียงการนับเวลาคาดการณ์ โดยเวลาที่เครื่องทำงานจริงก็ขึ้นอยู่กับอีกหลายๆ ปัจจัย เช่น ความแรงของน้ำที่เครื่องซักผ้า เป็นต้น ทำให้ผู้ประกอบการอาจจะเสียโอกาสทางการค้าจากการที่มีผู้ใช้บริการลดลงได้ การพัฒนาและออกแบบเครื่องซักผ้าสาธารณะที่ตอบสนองต่อความต้องการของผู้ใช้งานจึงมีความสำคัญอย่างมาก ในยุคที่ธุรกิจประเภทนี้มีการแข่งขันกันสูงและเทคโนโลยี Internet of Things (IOT) เข้ามามีบทบาทกับคนทุกกลุ่ม ทำให้คณะผู้จัดทำมีความสนใจในการพัฒนาเครื่องซักผ้าสาธารณะแบบฝาบน ให้ผู้ใช้บริการสามารถเลือกใช้ฟังก์ชันการทำงานของเครื่องซักผ้าผ่านระบบแอปพลิเคชันบนสมาร์ทโฟนได้ นอกจากนี้ระบบยังมีการแจ้งเตือนเมื่อเครื่องมีปัญหาระหว่างการซัก รวมถึงแจ้งเตือนเมื่อเครื่องทำการซักเสร็จสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

เพื่อพัฒนาเครื่องซักผ้าสาธารณะแบบฝาบน ให้มีคุณสมบัติที่ตอบสนองต่อความต้องการของผู้ใช้งาน โดยผู้ใช้งานสามารถตรวจสอบสถานะเครื่องซักผ้า เลือกใช้งานฟังก์ชันต่างๆ ของเครื่องซักผ้า และตรวจสอบการทำงานของเครื่องซักผ้าผ่านแอปพลิเคชันบนสมาร์ทโฟนได้

1.3 ขอบเขตของโครงงาน

1. ศึกษารูปแบบของเครื่องซักผ้าแบบฝาบน เซนเซอร์ และอุปกรณ์อิเล็กทรอนิกส์ที่เกี่ยวข้อง
2. ศึกษาโปรโตคอลและโปรแกรมที่นำมาใช้ประมวลผล
3. ออกแบบและพัฒนาเครื่องซักผ้าที่สามารถเลือกใช้งานฟังก์ชันต่างๆ บนสมาร์ทโฟนได้
4. ออกแบบโปรแกรมที่นำมาใช้แสดงผล

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาค้นคว้าทฤษฎีและเอกสารที่เกี่ยวข้อง
2. ศึกษาหลักการของเซนเซอร์ที่จะนำมาใช้
3. ออกแบบและวางแผนการประยุกต์ใช้เซนเซอร์
4. สั่งซื้ออุปกรณ์
5. ออกแบบและพัฒนาการใช้งานระหว่างเครื่องซักผ้าฝาบนและเซนเซอร์
6. ทดสอบการอ่านค่าของเซนเซอร์
7. ประกอบเครื่องซักผ้ากับเซนเซอร์และอุปกรณ์เสริม
8. เขียนโปรแกรมควบคุมการทำงานและการแสดงผล
9. ทดสอบและแก้ไขโปรแกรมแสดงผล
10. สรุปผลและจัดทำเอกสารรายงานการวิจัย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับทักษะเพิ่มเติมในด้านการคิด การวางแผน การตัดสิน ในระหว่างการทำโครงงาน
2. ได้รับทักษะเพิ่มเติมในการเขียนโปรแกรมและการออกแบบ
3. สามารถประดิษฐ์เครื่องซักผ้าเพื่อตอบสนองความต้องการของผู้ใช้งานตามวัตถุประสงค์ให้สำเร็จได้อย่างสมบูรณ์
4. สามารถประยุกต์เครื่องซักผ้าที่เพื่อตอบสนองความต้องการของผู้ใช้งานในปัจจุบันได้ เช่น การบอกสถานะการทำงานของเครื่องซักผ้าผ่านระบบแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย 5 บท และ 4 ภาคผนวก ซึ่งมีรายละเอียดดังต่อไปนี้

บทที่ 1 บทนำ เป็นการกล่าวถึงที่มาของปริญญานิพนธ์ วัตถุประสงค์ของการทำปริญญานิพนธ์ ขอบเขตของโครงงาน ขั้นตอนการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับและรายละเอียดของปริญญานิพนธ์

บทที่ 2 ทฤษฎี หลักการ อุปกรณ์ และความรู้ที่เกี่ยวข้องในการออกแบบ เป็นการเพิ่มเติมความรู้ทฤษฎีและความเข้าใจในอุปกรณ์ต่างๆ ก่อนจะเริ่มการทำโครงงาน

บทที่ 3 ขั้นตอนการดำเนินงาน เป็นการอธิบายขั้นตอนการทำชิ้นงานโดยละเอียดทั้งในการฮาร์ดแวร์และซอฟต์แวร์

บทที่ 4 ผลการดำเนินงาน เป็นการแสดงผลการทำงานของเครื่องซักผ้าอัจฉริยะ

บทที่ 5 ผลสรุปและข้อเสนอแนะ เป็นบทสรุปภาพรวมของชิ้นงานรวมถึงสิ่งที่จะพัฒนาต่อไปในอนาคต

ภาคผนวก ก โมดูลตรวจจับกระแส ACS712

ภาคผนวก ข บอร์ด ESP32

ภาคผนวก ค โปรแกรมควบคุมการทำงานและการแสดงผล

ภาคผนวก ง โปสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 กล่าวนำ

ในปริญญานิพนธ์ฉบับนี้ได้แบ่งการอธิบายทฤษฎีที่เกี่ยวข้องออกเป็น 2 ส่วน ได้แก่ส่วนของฮาร์ดแวร์ และส่วนของซอฟต์แวร์โดยมีรายละเอียดดังต่อไปนี้

2.2 อุปกรณ์ที่เกี่ยวข้อง

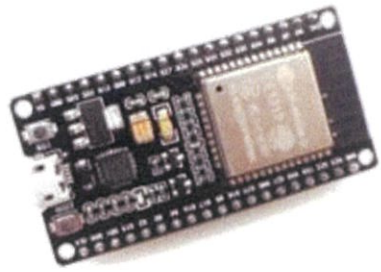
2.2.1 ระบบสื่อสารสัญญาณไร้สายผ่าน Wi-Fi Module ESP32

ESP32 เป็นชื่อของไอซีไมโครคอนโทรลเลอร์ที่มีรองรับการเชื่อมต่อ Wi-Fi และมีความสามารถในการเชื่อมต่อ Bluetooth Low-Energy (BLE BT4.0 และ Bluetooth Smart) ซึ่งไอซี ESP32 เป็น Wi-Fi/BLE SoC (System On Clip) ใช้ USB 2 Serial ตระกูล FTDI Ship เพื่อเป็นช่องทางในการโหลดโปรแกรมแบบอัตโนมัติจากโปรแกรม Arduino IDE

ไอซี ESP32 ได้แก้ไขจุดด้อยต่างๆ ของ ESP8266 ไปจนหมดไม่ว่าจะเป็นเรื่องของ I/O และ Analog Input ที่มีไม่เพียงพอกับการใช้งานและปรับรายละเอียดของฮาร์ดแวร์ให้สูงขึ้น มีความเสถียรภาพสูงเหมาะสำหรับงานพัฒนาต้นแบบอุปกรณ์แบบ Portable และ Wearable โดยตัวไอซี ESP32 มีคุณสมบัติโดยละเอียด ดังนี้

1. ซีพียูใช้ ESP-WROOM-32 ผลิตโดยบริษัท Espressif ซึ่งเป็น Wi-Fi/BLE SoC (System On Clip)
2. Breadboard Friendly มีขนาดกว้าง 0.9" วางบน Breadboard จะเหลือข้างละ 1 ช่อง
3. ใช้ USB 2 Serial ตระกูล FTDI ชิปเพื่อการโหลดโปรแกรมแบบอัตโนมัติ
4. JST 2mm Connector สำหรับเสียบแบตเตอรี่
5. มีวงจรชาร์จ Lithium Ion และ Lithium Polymer (1 Cell) พร้อมทั้งไฟแสดงสถานะ
6. มีวงจร PTC Fuse ตัดกระแสไฟเกินที่ 500mA
7. 3.3V 600mA On-board Voltage Regulator
8. Push Button Switch ที่ขา IO0 และ EN (Reset)
9. Crystal 32.768KHz เพื่อใช้เลี้ยงวงจร RTC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 บอร์ด ESP32

ที่มา http://www.arduino.codemobiles.com/product/255/nodemcu-32s-wifibluetooth-development-board-esp-wroom-32

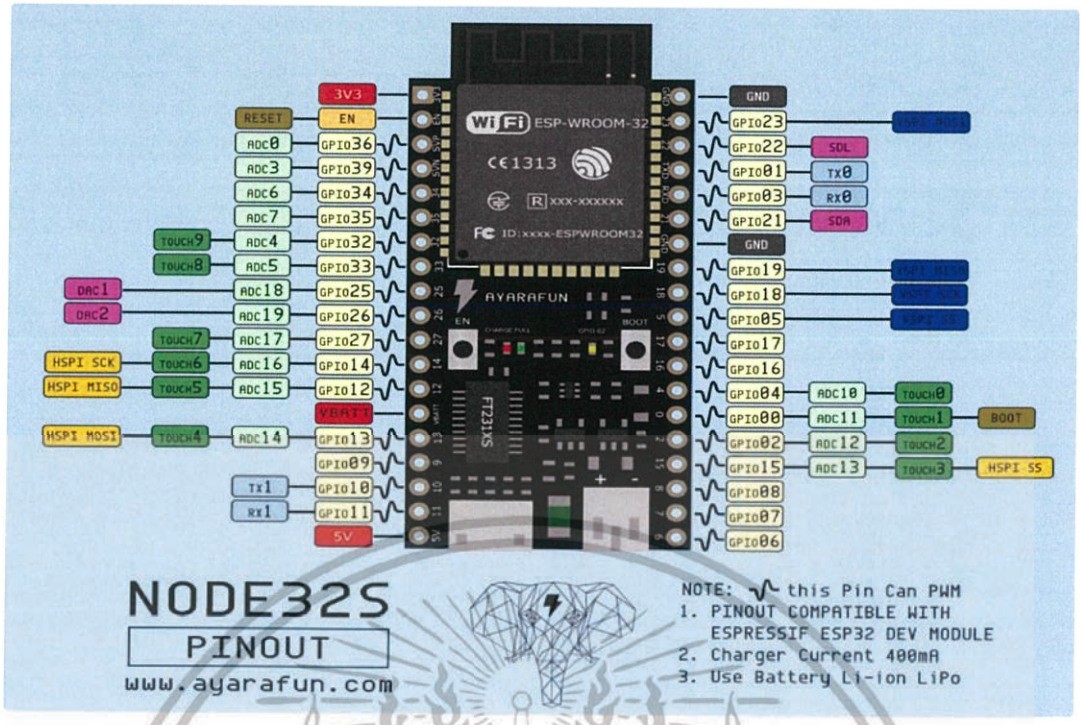
ขาใช้งานต่างๆ ของบอร์ด ESP32 รองรับการเชื่อมต่อบัสต่างๆ ดังนี้

- 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-Car

ในด้านประสิทธิภาพการใช้งาน 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ขาของบอร์ด ESP32

ที่มา http://www.ayarafun.com/2016/11/introduction-node32s/

2.2.2 ตัวสั่งการเครื่องใช้ไฟฟ้า 220V (โมดูลรีเลย์ 5V 4 ช่องสัญญาณ)

โมดูลรีเลย์เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตช์ตัดต่อวงจร โดยใช้แม่เหล็กไฟฟ้าและการที่จะให้รีเลย์ทำงานก็ต้องจ่ายไฟให้กับอุปกรณ์ เมื่อรีเลย์ได้รับการจ่ายไฟจะทำให้หน้าสัมผัสติดกันกลายเป็นวงจรปิด และในทางตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้รีเลย์ก็จะกลายเป็นวงจรเปิดไฟที่ใช้ป้อนให้กับตัวรีเลย์ก็จะเป็นไฟที่มาจากอุปกรณ์ไฟฟ้า ดังนั้นทันทีที่เปิดเครื่องก็จะทำให้โมดูลรีเลย์ทำงาน

หลักการทำงานวงจรของโมดูลรีเลย์คือ การเปิดปิดวงจรด้วยแม่เหล็กไฟฟ้า เมื่อขดลวดเหนี่ยวนำมีกระแสไหลผ่านจะมีคุณสมบัติเป็นแม่เหล็ก สามารถส่งแรงผลักหรือดูดเพื่อเปลี่ยนตำแหน่งสวิตช์ได้ ดังนั้นในวงจรจะประกอบไปด้วยโมดูลรีเลย์และบอร์ด ESP32 เชื่อมโยงสายกันอยู่ ในรูปที่ 2.4 แสดงตัวอย่างการต่อโมดูลรีเลย์ 5V 4 ช่องสัญญาณกับบอร์ดควบคุม โดยมีสวิตช์เปิดปิดวงจรตามแรงดูดของขดลวด หน้าสัมผัสของสวิตช์มี 2 ชนิด คือ

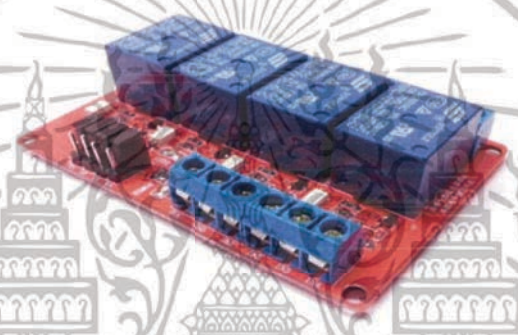
1. หน้าสัมผัสปกติเปิดหรือ NO (Normally Open) หมายถึง หน้าสัมผัสที่เปิดในภาวะขดลวดไม่เหนี่ยวนำ

2. หน้าสัมผัสปกติปิดหรือ NC (Normally Closed) หมายถึง หน้าสัมผัสที่ปิดในภาวะขดลวดไม่เหนี่ยวนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูลรีเลย์ 5V 4 ช่องสัญญาณ มีรายละเอียด ดังนี้

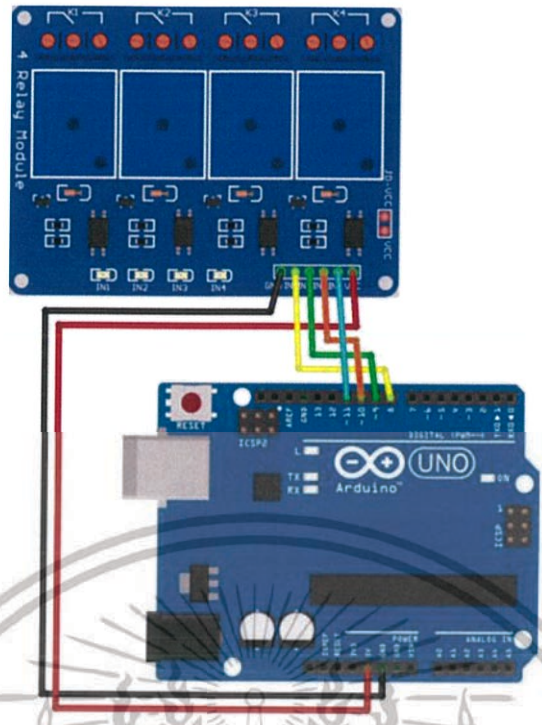
- ไฟเลี้ยงโมดูลรีเลย์ VCC = 5VDC
- ควบคุมโหลดได้ทั้งแรงดันไฟฟ้า AC ได้สูงสุด 250VAC 10A หรือแรงดันไฟฟ้า DC ได้สูงสุด 30VDC 10A (Maximum Load)
- ระดับสัญญาณอินพุตควบคุมแบบ TTL ทำงานด้วยสัญญาณแบบ Active Low
- กระแสขับรีเลย์ (Drive Current) 15-20mA
- มีการออกแบบให้เป็น Isolate ด้วย Optocoupler
- มี LED แสดงสถานะรีเลย์



รูปที่ 2.3 โมดูลรีเลย์ 5V 4 ช่องสัญญาณ

ที่มา http://www.arduino.codemobiles.com/product/29/5v-relay-4-channel-high-low-level-trigger-relay-module-red-pcb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ตัวอย่างการต่อระหว่างโมดูลรีเลย์ และบอร์ด Arduino

ที่มา http://www.arduino.codemobiles.com/product/29/5v-relay-4-channel-high-low-level-trigger-relay-module-red-pcb

2.2.3 ไอซีวัดกระแสในตระกูล ACS712

ไอซีวัดกระแสในตระกูล ACS712 สามารถใช้แรงดันไฟเลี้ยง VCC ในช่วง 4.5V ถึง 5.5V วัดกระแสได้ในสองทิศทาง (DC และ AC) และให้แรงดันเอาต์พุต VOUT แบบเชิงเส้น (Linear) และมีไอซีในตระกูลนี้ให้เลือกใช้แตกต่างกันตามช่วงของการวัดกระแส เช่น ±5A, ±20A และ ±30A เป็นต้น และมีค่าความไวของเอาต์พุตในช่วง 66mV/A ถึง 185mV/A ถ้าไม่มีกระแสไหล จะได้ VOUT เท่ากับ VCC/2 ถ้ามีกระแสไหลทางบวก จะทำให้ VOUT เพิ่มขึ้นสูงกว่า VCC/2 แต่ถ้ามีกระแสไหลในทางลบ จะทำให้ VOUT ลดลงต่ำกว่า VCC/2 ตัวไอซีมีขา กรองสัญญาณสำหรับต่อตัวเก็บประจุเพิ่ม (เช่น ความจุ 1nF) เพื่อใช้ร่วมกับตัวต้านทานที่อยู่ภายในไอซี และทำหน้าที่เป็นวงจรกรองความถี่ต่ำผ่าน (Low-pass RC Filter) สำหรับสัญญาณแรงดันเอาต์พุต

การวัดปริมาณกระแสด้วยโมดูล ACS712 สามารถทำได้โดยนำสัญญาณเอาต์พุต (VOUT) ไปต่อกับขา A0 ของ Arduino และป้อนแรงดันไฟเลี้ยง VCC=5V จากบอร์ด Arduino ให้โมดูล ACS712 เมื่อทำงาน Arduino จะคอยอ่านค่าแรงดันอินพุตแล้วแปลงให้ข้อมูลดิจิทัล 0-1023 ค่าดังกล่าวจะต้องนำไปลบออกจากค่ากลางซึ่งเป็นค่าที่วัดได้เมื่อยังไม่มีกระแสไหล แล้วนำผลต่างที่ได้ไปแปลงให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นแรงดัน (หน่วยเป็น mV) แล้วจึงหารด้วย 185 (mV/A) เพื่อคำนวณปริมาณกระแสที่ได้ในหน่วยเป็นแอมแปร์

รูปที่ 2.5 ไอซีวัดกระแส ACS712

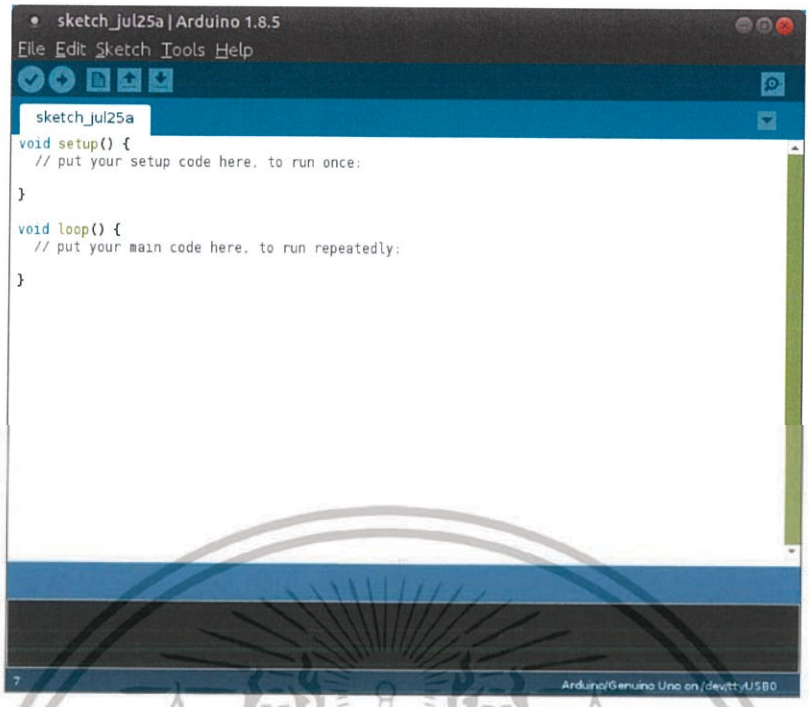
ที่มา https://cdn3.volusion.com/btfzd.umflq/v/vspfiles/photos/AD458-2.jpg?1513941419

2.3 โปรแกรมที่เกี่ยวข้อง

2.3.1 โปรแกรม Sketch Arduino IDE

โปแกรม Arduino IDE เป็นโปรแกรมควบคุมการทำงานของบอร์ด Arduino เป็นโปรแกรมที่ใช้งานง่าย เขียนด้วยภาษา C และเป็นโปรแกรมแบบโอเพนซอร์สทำให้ใช้งานได้โดยไม่มีค่าใช้จ่าย และได้รับความนิยมสูง ในส่วนของบอร์ด Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีพอร์ตอินพุตและเอาต์พุตที่มีมากพอในการนำไปใช้งานจริงสามารถต่อกับเซนเซอร์ได้ทั้งแบบดิจิทัลและแอนะล็อก และยังต่อเพื่อขับเอาต์พุตให้ทำงาน โดยที่จะต้องเขียนโค้ดเพื่อสั่งงานให้กับบอร์ด และบอร์ดจะสามารถควบคุมอุปกรณ์ต่างๆ ได้ดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 หน้าโปรแกรม Arduino IDE

ที่มา https://poundxi.com/วิธีใช้งานโปรแกรม-arduino-ide-เบื้องต้น

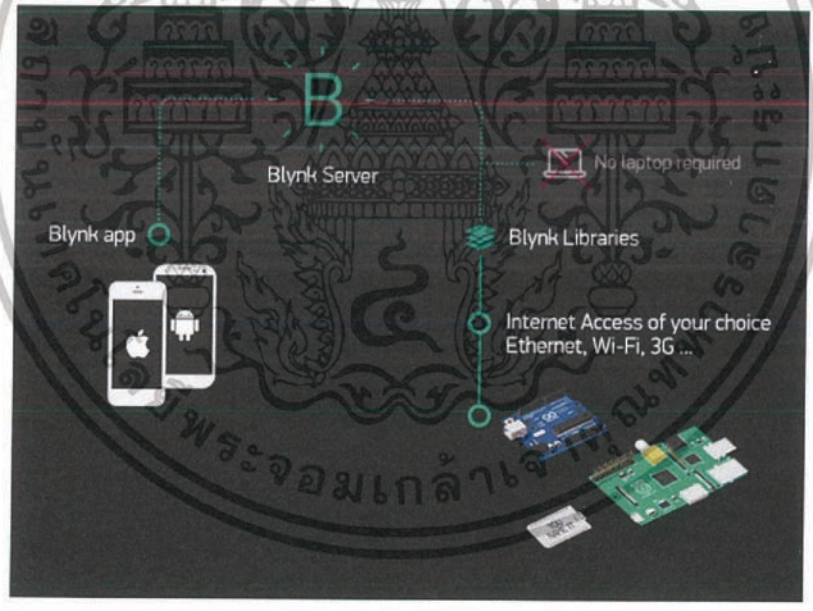
2.3.2 แอปพลิเคชัน Blynk

Blynk คือ แอปพลิเคชัน สำเร็จรูปสำหรับงาน IOT มีจุดเด่นคือ การเขียนโปรแกรมที่ง่ายโดยไม่ต้องเขียนแอปพลิเคชันเอง สามารถเชื่อมต่อกับอุปกรณ์ต่างๆ เข้ากับอินเทอร์เน็ตได้ง่าย ไม่ว่าจะเป็นบอร์ด ESP8266 หรือ ESP32 ก็สามารถนำมาแสดงบนแอปพลิเคชันได้อย่างง่าย Blynk ทำงานผ่านทางอินเทอร์เน็ต ดังนั้นข้อกำหนดเพียงอย่างเดียวคือ อุปกรณ์ต้องสามารถเชื่อมต่อกับอินเทอร์เน็ตได้ โดยสามารถเลือกการเชื่อมต่อแบบใดแบบหนึ่ง เช่น Ethernet Wi-Fi หรืออาจจะเป็น ESP8266 ไลบรารี่ของ Blynk และโค้ดตัวอย่างจะช่วยให้บอร์ดออนไลน์ โดยเชื่อมต่อกับเซิร์ฟเวอร์ของ Blynk และจับคู่กับสมาร์ทโฟนที่สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ โดยเซิร์ฟเวอร์ของ Blynk จะจัดการกับการรับรองความถูกต้อง และการสื่อสารทั้งหมดรวมถึงยังเก็บข้อมูลบนบอร์ดในขณะที่สมาร์ทโฟนออฟไลน์ เซิร์ฟเวอร์ของ Blynk ทำงานบน Java และเป็นแบบโอเพนซอร์ส สามารถเรียกใช้งานได้ภายในเครื่อง การรับส่งข้อความระหว่างแอปพลิเคชันเคลื่อนที่ เซิร์ฟเวอร์ของ Blynk และ Arduino ใช้โปรโตคอลไบนารีที่ง่ายและมีขนาดเล็กและรวดเร็วผ่านซ็อกเก็ต TCP/IP ดังรูปที่ 2.7 และรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 ตัวอย่างหน้าแอปพลิเคชัน Blynk

ที่มา https://www.arduinospro.com/article/23/blynk-app-basic-ep0-blynk-คืออะไร



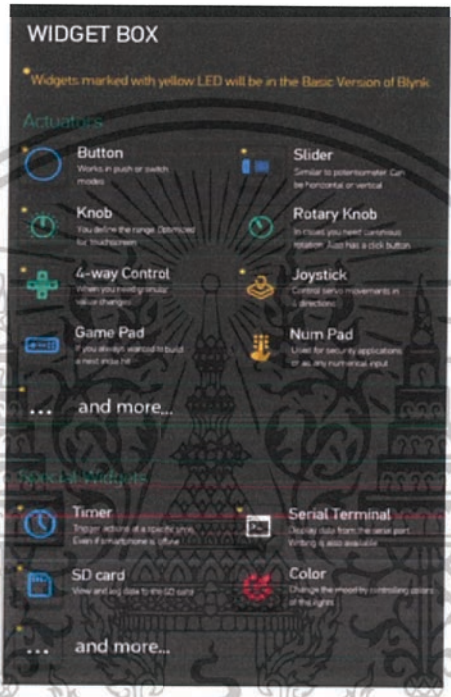
รูปที่ 2.8 สถาปัตยกรรมของแอปพลิเคชัน Blynk

ที่มา https://www.arduinospro.com/article/23/blynk-app-basic-ep0-blynk-คืออะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.1 อุปกรณ์ภายในแอปพลิเคชัน Blynk (Widget Box)

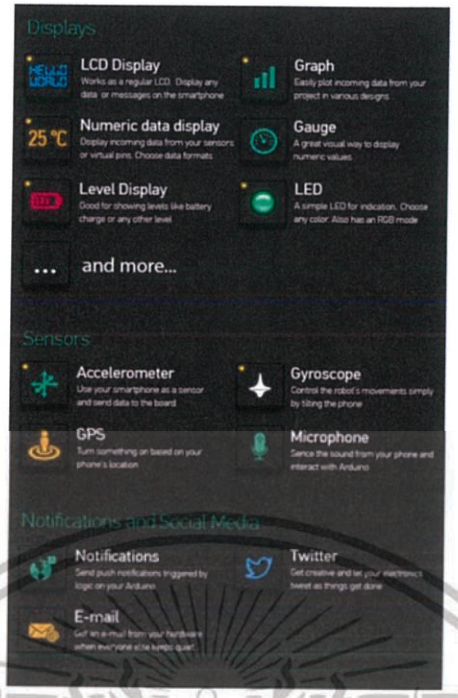
ในการใช้แอปพลิเคชัน Blynk จะมีตัวอุปกรณ์ขึ้นมาเพื่อให้ทำการเลือกตัวอุปกรณ์มาสร้างหน้าแอปพลิเคชันตามที่ต้องการ โดยภายในแอปพลิเคชัน Blynk จะมีตัวอุปกรณ์มากมาย ยกตัวอย่างเช่น ปุ่มกด Button หลอด LED แสดงค่าสถานะไฟ จอแสดงค่า LCD ใช้ในการแสดงค่าผลลัพธ์ต่างๆ รวมไปถึงอุปกรณ์แจ้งเตือนต่างๆ บนสมาร์ทโฟนได้หรืออีเมล์ เป็นต้น ดังรูปที่ 2.9 และรูปที่ 2.10



รูปที่ 2.9 หน้าตา Widget Box ของ Blynk (1)

ที่มา https://www.arduinospro.com/article/23/blynk-app-basic-ep0-blynk-คืออะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 หน้าต่าง Widget Box ของ Blynk (2)

ที่มา https://www.arduinospro.com/article/23/blynk-app-basic-ep0-blynk-คืออะไร

2.3.2.2 การใช้งานแอปพลิเคชัน Blynk

แอปพลิเคชัน Blynk เป็นแอปพลิเคชันที่ไม่มีค่าใช้จ่าย รวมทั้งยังรองรับการทำงานบนระบบ Android และ ระบบ IOS

1. วิธีการติดตั้งเริ่มต้นด้วยการค้นหาชื่อแอปพลิเคชัน โดยให้พิมพ์คำว่า Blynk ในช่องของการค้นหา จะพบแอปพลิเคชัน Blynk และสามารถติดตั้งในสมาร์ทโฟนได้ทันที ดังแสดงรูปที่ 2.11

2. หลังจากการติดตั้งเสร็จสิ้นให้ทำการเปิดแอปพลิเคชัน Blynk และทำการสร้างบัญชีผู้ใช้งานใหม่ ดังแสดงในรูปที่ 2.12

3. ขั้นตอนต่อไปเป็นการสร้างโปรเจคใหม่ เลือกที่คำว่า New Project แล้วใส่ชื่อโปรเจคให้เรียบร้อย หลังจากนั้นเลือกประเภทบอร์ดพัฒนาของผู้ใช้งาน เช่น บอร์ด ESP32 จึงเลือกเป็น ESP32 Dev Board เป็นต้น ดังแสดงในรูปที่ 2.13

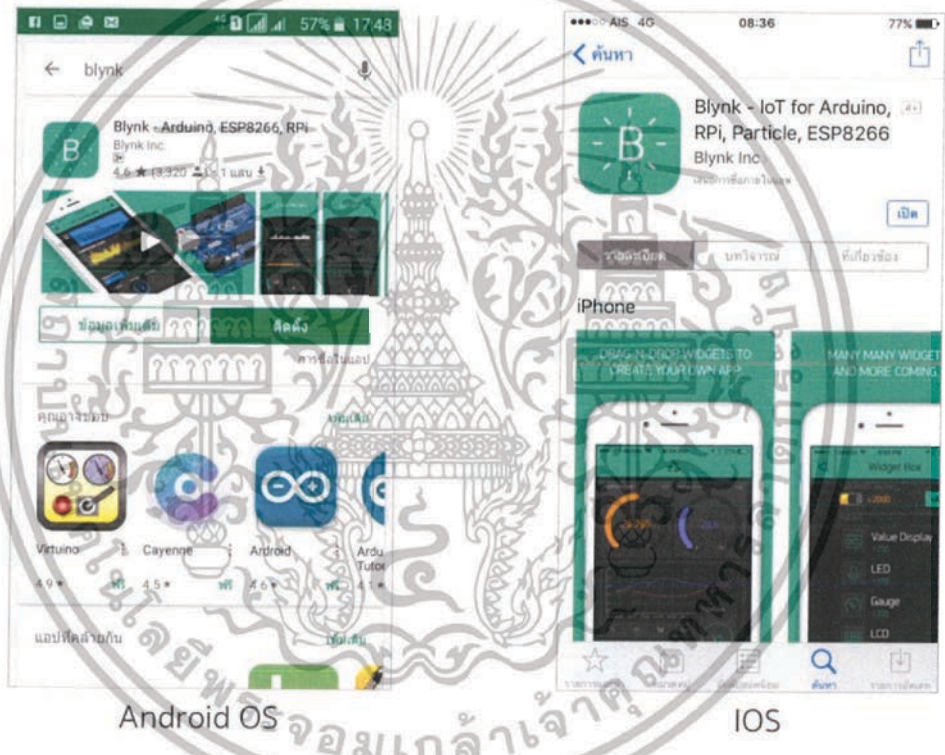
4. ในทางแอปพลิเคชัน Blynk จะมีสิ่งที่เป็นตัวแลกเปลี่ยนในการเลือกใช้อุปกรณ์ภายในแอปพลิเคชันพื้นฐาน ซึ่งจะเรียกว่า Blynk Energy โดยทางแอปพลิเคชันจะให้ผู้ใช้งานใหม่เริ่มต้นที่ 1000 Energy ซึ่งสามารถทำการเพิ่มจำนวน Energy ได้โดยผ่านการซื้อตามราคาที่ทางแอปพลิเคชันกำหนดไว้ ดังแสดงด้านบนของรูปที่ 2.14 หลังจากนั้นให้ทำการเลือกอุปกรณ์ภายในแอปพลิเคชันที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้งาน ตัวอย่าง จำนวน Blynk Energy ที่อุปกรณ์ภายในแอปพลิเคชันต้องการใช้ เช่น Button จะใช้ 200 Energy Labeled Value จะใช้ 400 Energy และ Gauge จะใช้ 300 Energy เป็นต้น

5. หลังจากที่สร้างโปรเจคเสร็จเรียบร้อย จะได้รับรหัสที่ทางเซิร์ฟเวอร์ Blynk ส่งมาให้ทางอีเมล์ของผู้ใช้งาน ซึ่งรหัสนี้คือ ตัวแทนบอร์ดพัฒนาของผู้ใช้งาน ที่ผู้ใช้งานจำเป็นต้องระบุในขั้นตอนการเขียนโปรแกรม Arduino IDE ที่บรรทัด Char Auth[ ] เพื่อทำการเชื่อมต่อระหว่างบอร์ดพัฒนาและแอปพลิเคชัน Blynk

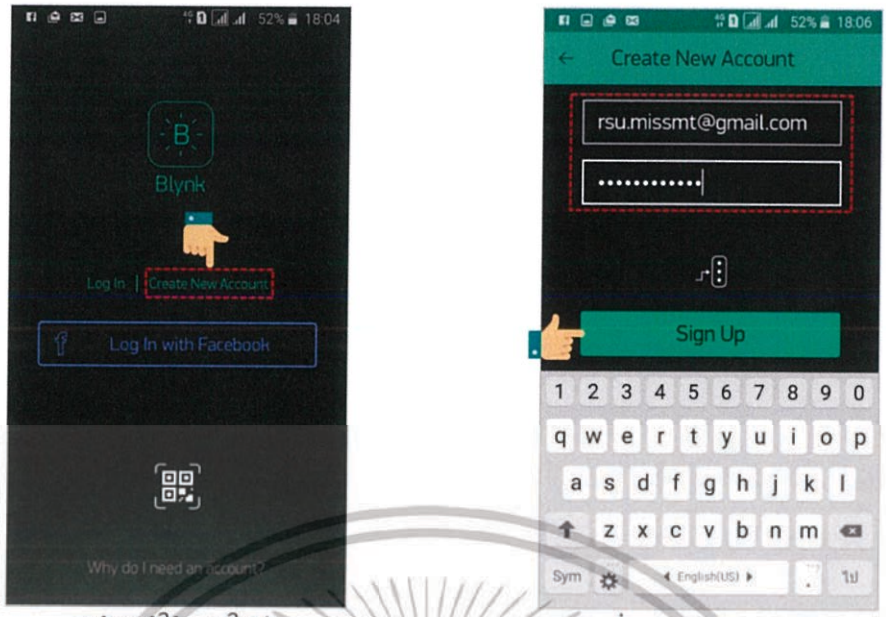
6. ทำการเขียนโปรแกรมเพื่อเชื่อมต่อแอปพลิเคชัน Blynk ด้วยโปรแกรม Arduino IDE ดังแสดงในรูปที่ 2.15 หลังจากนั้นทำการอัปโหลดโปรแกรมใส่บอร์ดพัฒนาของผู้ใช้งาน



รูปที่ 2.11 แอปพลิเคชัน Blynk ในระบบการทำงาน Android และ IOS

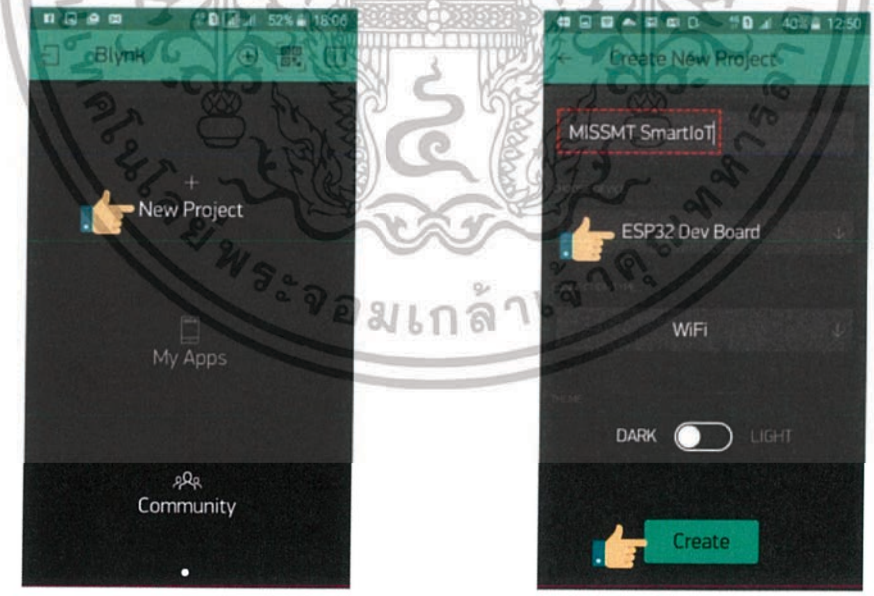
ที่มา https://medium.com/@visitwnk/ใส่ใจ-7-เตรียมความพร้อมก่อนการใช้-blynk-app-1ab60aa1b9e9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 ขั้นตอนการสร้างบัญชีผู้ใช้งานใหม่

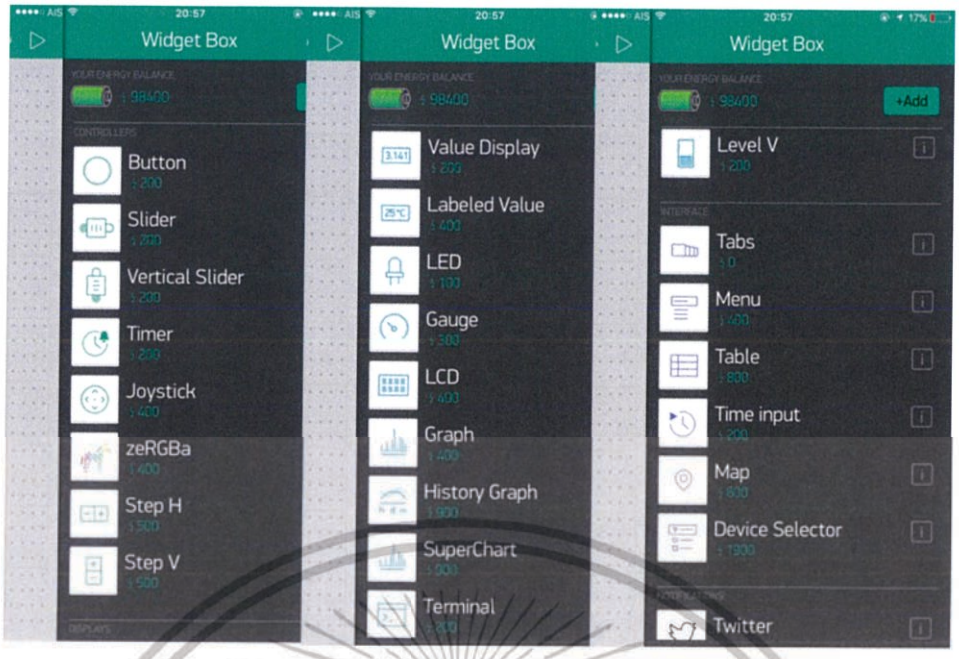
ที่มา https://medium.com/@visitwnk/ใส่ใจ-7-เตรียมความพร้อมก่อนการใช้-blynk-app-1ab60aa1b9e9



รูปที่ 2.13 ขั้นตอนการสร้างโปรเจคใหม่

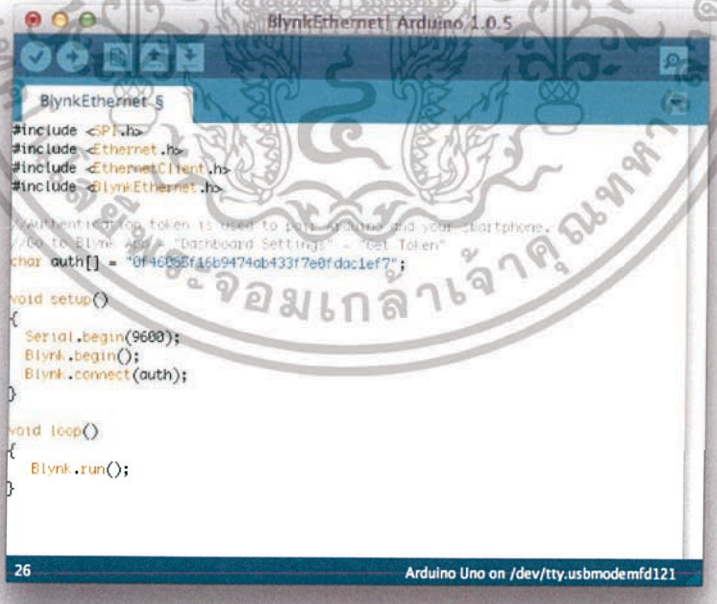
ที่มา https://medium.com/@visitwnk/ใส่ใจ-7-เตรียมความพร้อมก่อนการใช้-blynk-app-1ab60aa1b9e9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ตัวอย่าง Blynk Energy และจำนวน Energy ที่อุปกรณ์ภายในแอปพลิเคชันต้องการใช้

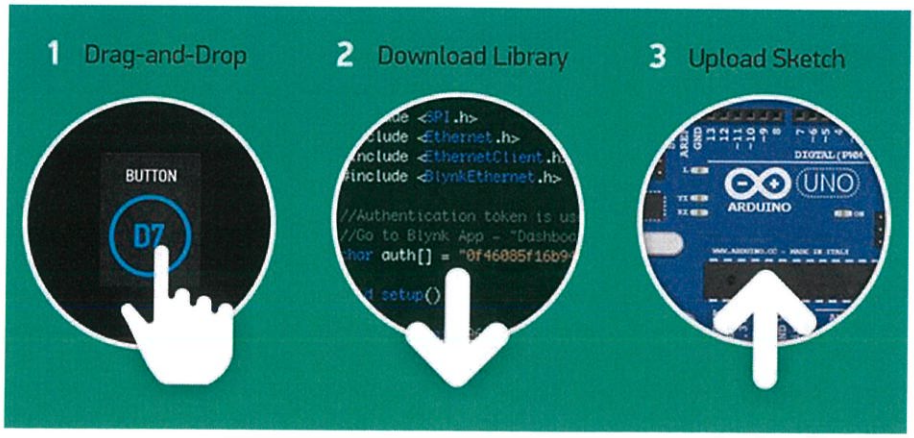
ที่มา https://medium.com/@visitwnk/ใส่ใจ-7-เตรียมความพร้อมก่อนการใช้-blynk-app-1ab60aa1b9e9



รูปที่ 2.15 ตัวอย่างโค้ดสำหรับการเชื่อมต่อแอปพลิเคชัน Blynk ด้วยโปรแกรม Arduino IDE

ที่มา https://www.arduinospro.com/article/23/blynk-app-basic-ep0-blynk-คืออะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 ภาพรวมวิธีการใช้งาน Blynk

ที่มา https://www.arduinospro.com/article/23/blynk-app-basic-ep0-blynk-คืออะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

3.1 ขั้นตอนการดำเนินงาน

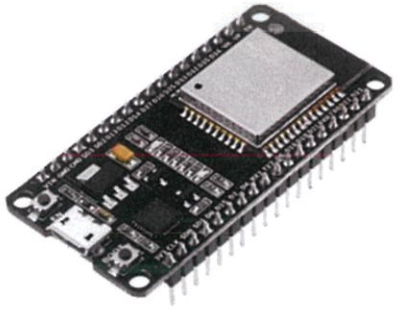
3.1.1 ขั้นตอนของการดำเนินงานได้วางแผนไว้ดังนี้

1. ศึกษาค้นคว้าทฤษฎีและเอกสารที่เกี่ยวข้อง
2. ศึกษาองค์ประกอบและหลักการทำงานของเครื่องซักผ้าแบบฝาบนและเซนเซอร์
3. ออกแบบและวางแผนการพัฒนาเครื่องซักผ้าแบบฝาบนกับเซนเซอร์ และอุปกรณ์อิเล็กทรอนิกส์ที่เกี่ยวข้อง
4. สั่งซื้ออุปกรณ์
5. พัฒนาเครื่องซักผ้าแบบฝาบนกับเซนเซอร์ และอุปกรณ์อิเล็กทรอนิกส์ที่เกี่ยวข้อง
6. ทดสอบการทำงานของเซนเซอร์ และอุปกรณ์อิเล็กทรอนิกส์ที่เกี่ยวข้องกับเครื่องซักผ้า
7. เขียนโปรแกรมและทดสอบการรับค่าจากเซนเซอร์ของไมโครคอนโทรลเลอร์
8. เขียนโปรแกรมและทดสอบการส่งคำสั่งจากไมโครคอลโทรลเลอร์ เพื่อสั่งฟังก์ชันบนเครื่องซักผ้า
9. เขียนโปรแกรมและทดสอบการแสดงผลการทำงานของเครื่องซักผ้าบนแอปพลิเคชัน
10. สรุปผลและจัดทำเอกสารรายงานการวิจัย

3.2 อุปกรณ์ที่เกี่ยวข้อง

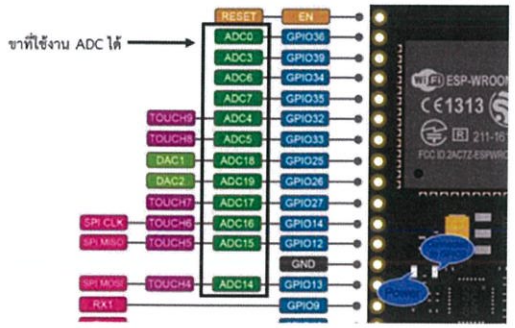
3.2.1 บอร์ด ESP32

บอร์ด ESP32 ทำหน้าที่รับข้อมูลจากเซนเซอร์ที่ส่งเข้ามายังขา ADC ที่สามารถใช้งานได้ เพื่อทำการประมวลผลและส่งไปยังโปรแกรมแสดงผลผ่านเครือข่าย Wi-Fi ดังรูปที่ 3.2



รูปที่ 3.1 บอร์ด ESP32

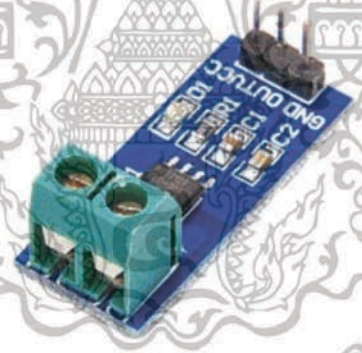
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ขา ADC ที่สามารถใช้งานได้บนบอร์ด ESP32

3.2.2 โมดูลตรวจจับกระแส ACS712

โมดูลตรวจจับกระแส ACS712 ในรูปที่ 3.3 ทำหน้าที่ตรวจจับย่านกระแสไฟฟ้าในภาวะต่างๆ ของเครื่องซักผ้า โดยวัดกระแสได้ในย่าน -30 ถึง +40A (วัดได้ทั้งกระแสแบบ AC และ DC) โดยต่อกับบอร์ด ESP32 ในตำแหน่งขา ADC เพื่ออ่านค่ากระแสที่ไหลผ่าน

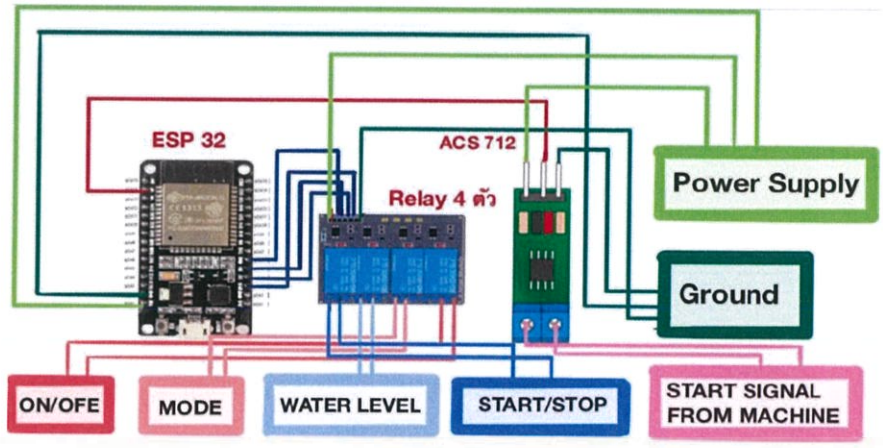


รูปที่ 3.3 โมดูลตรวจจับกระแส ACS712

3.2.3 โมดูลรีเลย์ 5V 10A

โมดูลรีเลย์ 5V 10A เป็นโมดูลที่ใช้ควบคุมแผงวงจรเครื่องซักผ้า โดยสามารถใช้กับแรงดันไฟฟ้าได้ทั้งแรงดันไฟฟ้าแบบ DC และ AC โมดูลรีเลย์ที่ใช้เป็นแบบ 4 ช่องสัญญาณ รูปที่ 3.4 แสดงการต่อโมดูลรีเลย์เข้ากับบอร์ด ESP32

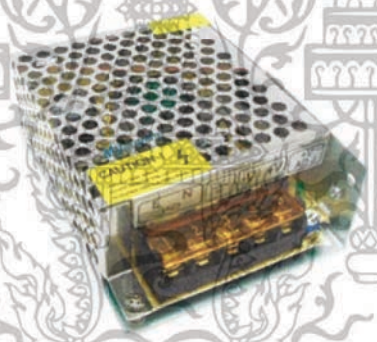
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 การเดินสายของโมดูลรีเลย์ 5V 10A กับบอร์ด ESP32 และอุปกรณ์อื่นๆ

3.2.4 แหล่งจ่ายไฟ 5V 3A

แหล่งจ่ายไฟทำหน้าที่แปลงไฟกระแสสลับ 220V ให้เป็นไฟกระแสตรง ขนาด 5V 3A เพื่อจ่ายไฟเลี้ยงให้กับไมโครคอนโทรลเลอร์ ดังรูปที่ 3.5



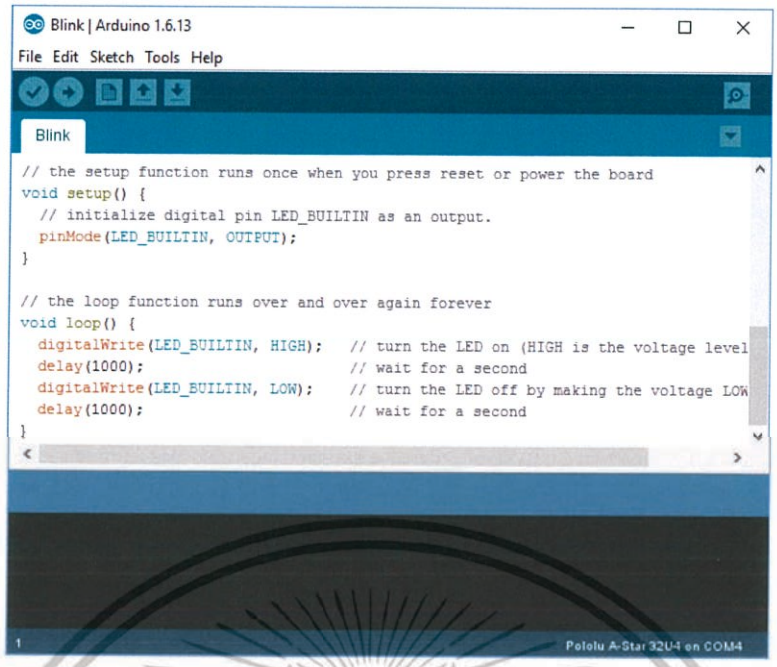
รูปที่ 3.5 แหล่งจ่ายไฟ 5V 3A

3.3 ซอฟต์แวร์ที่เกี่ยวข้อง

3.3.1 โปรแกรม Arduino IDE

โปรแกรม Arduino IDE Program เป็นโปรแกรมสำหรับใช้เขียนโปรแกรมคอมไพล์ และอัปโหลดโปรแกรมลงบอร์ด ESP32 โปรแกรมนี้ใช้สำหรับเขียนคำสั่งเพื่อรับข้อมูลจากเซนเซอร์ และส่งค่าตัวแปรไปยังโปรโตคอลเพื่อแสดงผลบนแอปพลิเคชัน ดังรูปที่ 3.6

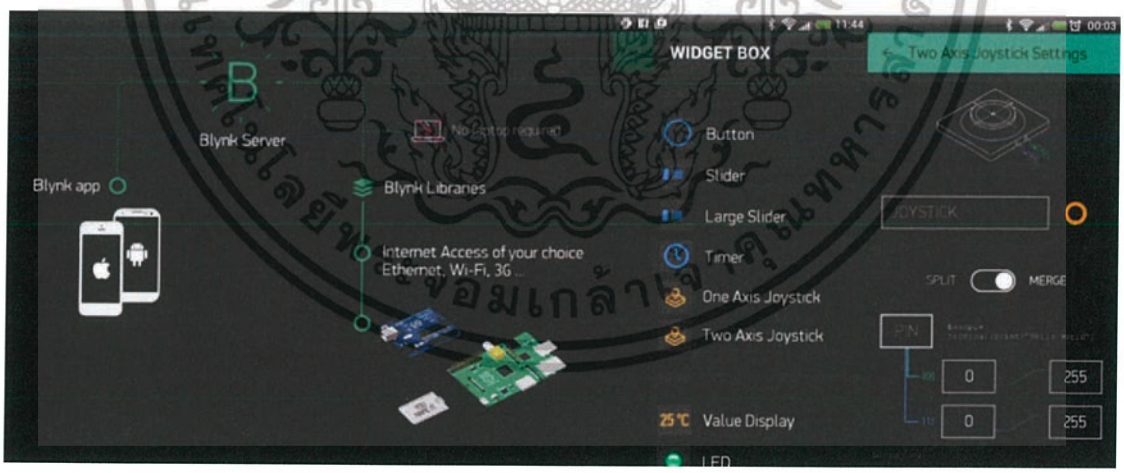
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 หน้าต่างของโปรแกรม Arduino

3.3.2 แอปพลิเคชัน Blynk

ใช้ในการควบคุมอุปกรณ์ IoT ซึ่งมีคุณสมบัติในการควบคุมจากระยะไกลผ่านเครือข่ายอินเทอร์เน็ต และสามารถแสดงผลค่าจากเซนเซอร์ต่างๆ บนแอปพลิเคชัน ดังรูปที่ 3.7



รูปที่ 3.7 แอปพลิเคชัน Blynk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบและการวางแผนการทำงาน

3.4.1 การออกแบบและการวางแผนทางด้านฮาร์ดแวร์

1. ศึกษาหลักการทำงานของเครื่องซักผ้าแบบฝาบน
2. ศึกษาหลักการของไมโครคอนโทรลเลอร์
3. ศึกษาหลักการของเซนเซอร์
4. ออกแบบและทำวงจรระหว่างเครื่องซักผ้าแบบฝาบน ไมโครคอนโทรลเลอร์ เซนเซอร์ และอุปกรณ์อิเล็กทรอนิกส์ที่เกี่ยวข้อง
5. ทำการประกอบชิ้นงาน รวมถึงอุปกรณ์ต่างๆ ตามที่ได้ออกแบบไว้

3.4.2 การออกแบบและการวางแผนทางด้านซอฟต์แวร์

1. ศึกษาโปรแกรมที่จะนำมาใช้งาน
2. วางแผนความต้องการในการแสดงผล
3. เขียนโปรแกรม Arduino เพื่อสั่งไมโครคอนโทรลเลอร์ให้สั่งฟังก์ชันการทำงานไปยังเครื่องซักผ้า
3. เขียนโปรแกรม Arduino เพื่อสั่งไมโครคอนโทรลเลอร์รับค่าจากเซนเซอร์
4. เขียนโปรแกรม Arduino เพื่อรับค่าจากไมโครคอนโทรลเลอร์มาแสดงผล
5. เขียนโปรแกรม Arduino เพื่อแสดงผลจากไมโครคอนโทรลเลอร์บนแอปพลิเคชัน Blynk
6. ตกแต่งหน้าต่างแสดงผลบนแอปพลิเคชัน
7. ทดสอบโปรแกรมกับชิ้นงาน

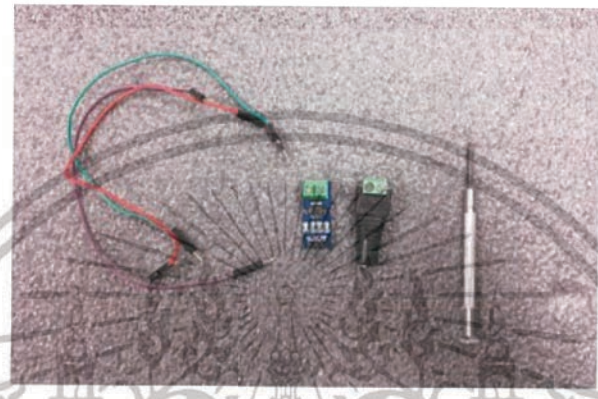
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วิธีการดำเนินงาน

3.5.1 การติดตั้งเซนเซอร์

อุปกรณ์ที่ใช้ในการติดตั้งเซนเซอร์ ดังรูปที่ 3.8

1. โมดูลตรวจจับกระแส ACS712
2. บอร์ด ESP32
3. ไขควง
4. ตัวแปลงไฟ 5V 2A
5. สายไฟ

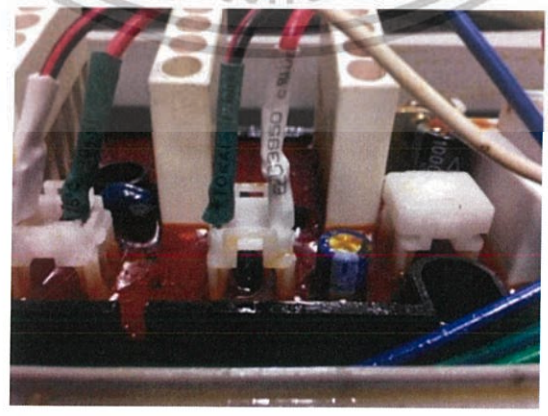


รูปที่ 3.8 อุปกรณ์ที่ใช้ในการติดตั้งเซนเซอร์

ขั้นตอนการติดตั้งเซนเซอร์

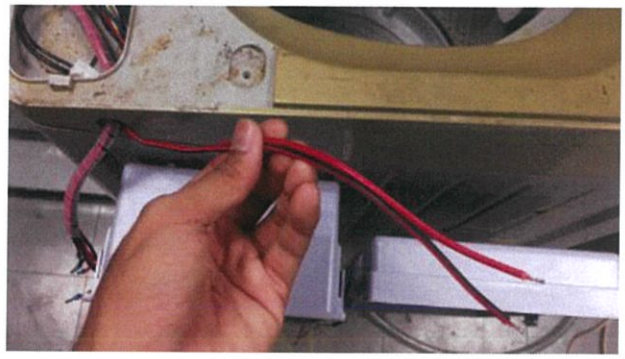
1. ศึกษาสายบนแผงควบคุมเครื่องซักผ้า เพื่อหาสายที่มีการเปลี่ยนแปลงค่ากระแสไฟฟ้าตามสถานะต่างๆ ของเครื่องซักผ้า โดยสายที่สามารถนำมาใช้งานได้คือ สายควบคุมวาล์วน้ำเข้าและสายควบคุมวาล์วน้ำออก

2. ตัดสายควบคุมวาล์วน้ำเข้าและสายควบคุมวาล์วน้ำออกบนเครื่องซักผ้าแบบฝาบน เพื่อเตรียมสำหรับการติดตั้งเซนเซอร์ ดังรูปที่ 3.9 และรูปที่ 3.10



รูปที่ 3.9 สายควบคุมวาล์วน้ำเข้าและสายควบคุมวาล์วน้ำออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



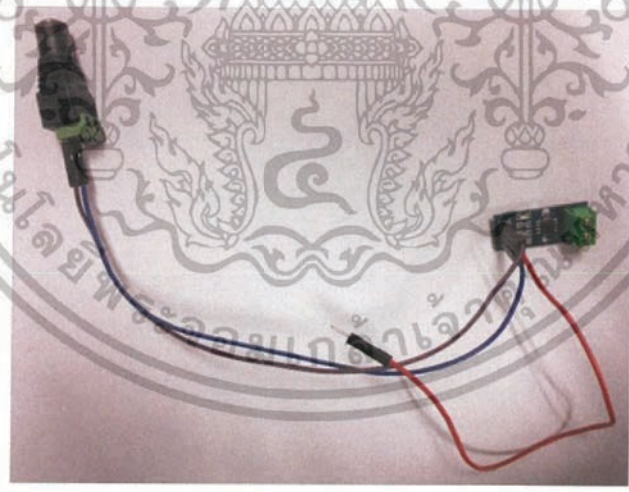
รูปที่ 3.10 สายควบคุมวาล์วน้ำเข้าและสายควบคุมวาล์วน้ำออกที่ตัดแล้ว

3. นำสายควบคุมวาล์วน้ำเข้าและสายควบคุมวาล์วน้ำออกที่ตัดแล้วต่อเข้ากับ ACS712

4. ต่อสายไฟเข้าที่ช่อง VCC บนโมดูลตรวจจับกระแส ACS712 และต่ออีกด้านของสายไฟเข้ากับหัวแปลงขั้วเพื่อใช้ในการต่อเข้ากับตัวแปลงไฟ 5V 2A

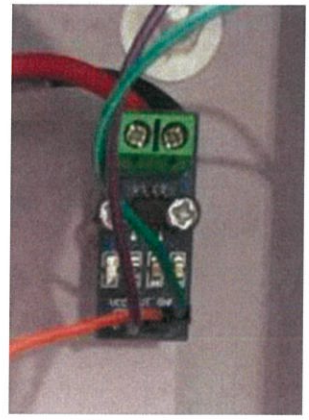
5. ต่อสายไฟเข้าที่ช่อง GND บนโมดูลตรวจจับกระแส ACS712 และต่ออีกด้านของสายไฟสีเข้ากับหัวแปลงขั้วเพื่อใช้ในการต่อเข้ากับตัวแปลงไฟ 5V 2A

6. ต่อสายไฟที่ช่อง OUTPUT บนโมดูลตรวจจับกระแส ACS712 และต่ออีกด้านของสายไฟเข้ากับช่อง GPIO 39 บนบอร์ด ESP32 ดังรูปที่ 3.11



รูปที่ 3.11 การต่อสายจาก ACS712 เข้าหัวแปลงขั้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

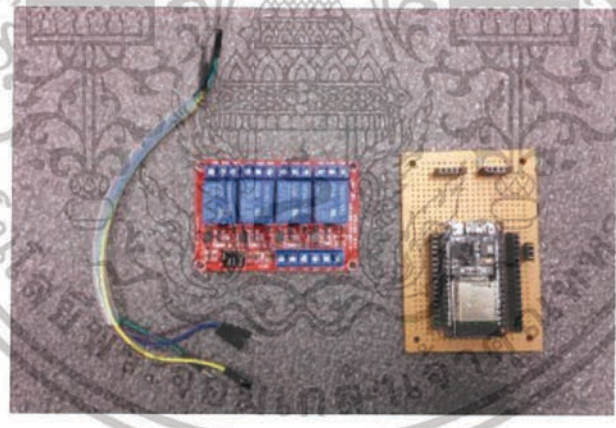


รูปที่ 3.12 การต่อสายเข้า ACS712

3.5.2 การติดตั้งโมดูลรีเลย์กับแผงควบคุม

อุปกรณ์ที่ใช้ในการติดตั้งโมดูลรีเลย์กับแผงควบคุม ดังรูปที่ 3.13

1. โมดูลรีเลย์ 5V 10A
2. สายไฟ
3. บอร์ดสำหรับวางอุปกรณ์อิเล็กทรอนิกส์
4. บอร์ด ESP32
5. ไขควง



รูปที่ 3.13 อุปกรณ์ที่ใช้ในการติดตั้งโมดูลรีเลย์กับแผงควบคุม

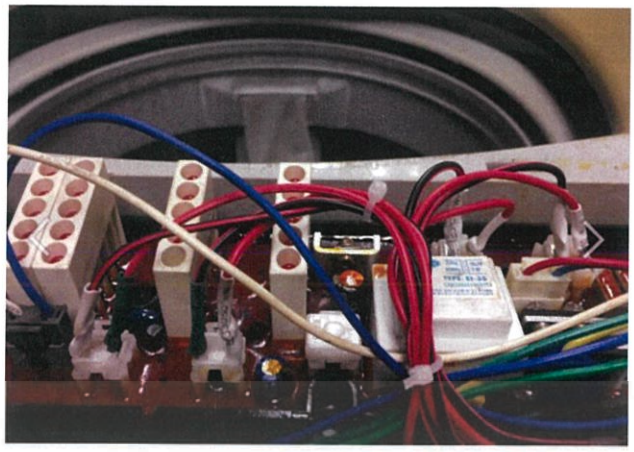
ขั้นตอนการติดตั้งโมดูลรีเลย์กับแผงควบคุม

1. ศึกษาสวิตช์บนแผงควบคุมเครื่องซักผ้า เพื่อระบุว่าแต่ละสวิตช์บนแผงควบคุมเครื่องซักผ้าออกคำสั่งใดให้กับเครื่องซักผ้า

2. ทำการต่อสายไฟเข้ากับสวิตช์ควบคุมระดับน้ำ สวิตช์ควบคุมโหมดของเครื่องซักผ้า สวิตช์ควบคุมคำสั่งทำงานและหยุดการทำงานของเครื่องซักผ้า และคำสั่งเปิดและปิดเครื่องซักผ้า

3. เก็บสายไฟแต่ละเส้นให้เรียบร้อยเพื่อใช้สำหรับการต่อเข้ารีเลย์ ดังรูปที่ 3.14 และรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

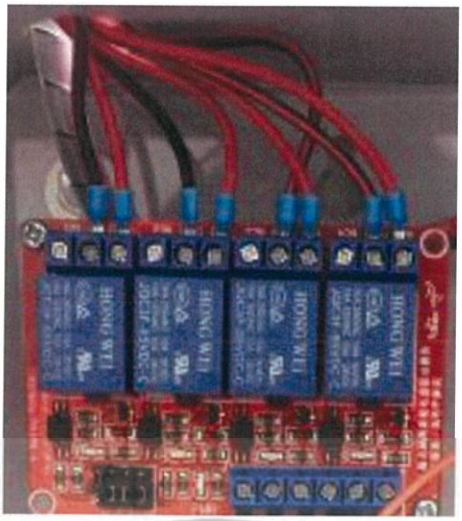


รูปที่ 3.14 การต่อสายไฟเข้ากับสวิตช์ควบคุมเครื่องซักผ้า

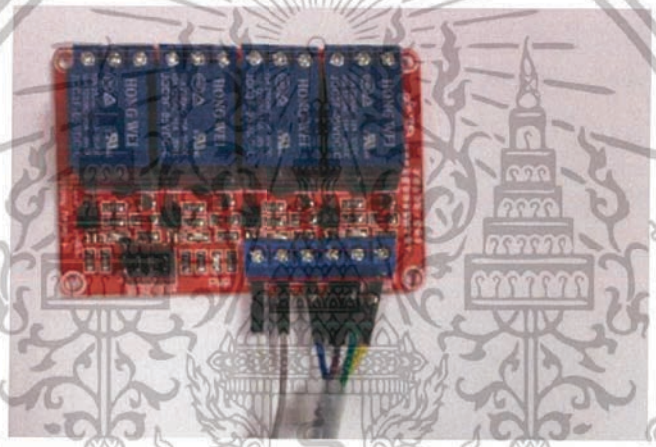
รูปที่ 3.15 การเก็บสายไฟเพื่อรอสำหรับการต่อเข้ากับโมดูลรีเลย์

4. ต่อสายไฟเข้ากับรีเลย์โดยมีรายละเอียดดังนี้
   - สายไฟควบคุมการเปิดปิดเครื่องซักผ้า ต่อเข้ากับรีเลย์ตัวที่ 1
   - สายไฟควบคุมการทำงานของเครื่องซักผ้า ต่อเข้ากับรีเลย์ตัวที่ 2
   - สายไฟควบคุมระดับน้ำ ต่อเข้ากับรีเลย์ตัวที่ 3
   - สายไฟตั้งโหมดการทำงานของเครื่องซักผ้า ต่อเข้ากับรีเลย์ตัวที่ 4
5. วางบอร์ด ESP32 ยึดติดกับบอร์ดสำหรับวางอุปกรณ์อิเล็กทรอนิกส์
6. ต่อสายไฟออกจากรีเลย์เข้ากับ ESP32 โดยมีรายละเอียดดังนี้
   - สายไฟที่ต่อออกจากรีเลย์ตัวที่ 1 ต่อเข้ากับช่อง GPIO 4
   - สายไฟที่ต่อออกจากรีเลย์ตัวที่ 2 ต่อเข้ากับช่อง GPIO 21
   - สายไฟที่ต่อออกจากรีเลย์ตัวที่ 3 ต่อเข้ากับช่อง GPIO 15
   - สายไฟที่ต่อออกจากรีเลย์ตัวที่ 4 ต่อเข้ากับช่อง GPIO 2

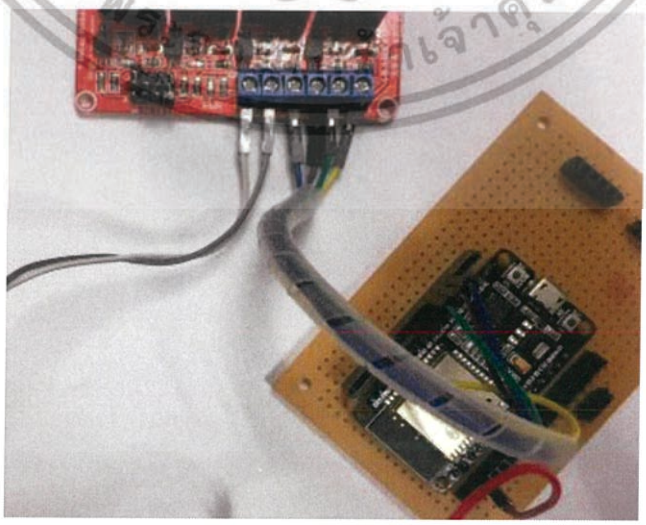
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 การต่อสายไฟเข้ากับรีเลย์ (1)



รูปที่ 3.17 การต่อสายไฟเข้ากับรีเลย์ (2)



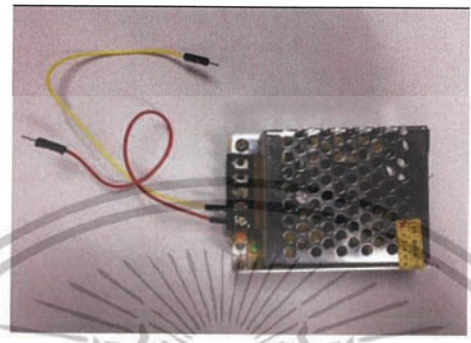
รูปที่ 3.18 การต่อสายไฟจากรีเลย์เข้าบอร์ด ESP32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 การติดตั้งแหล่งจ่ายไฟ

อุปกรณ์ที่ใช้ในการติดตั้งแหล่งจ่ายไฟ ดังรูปที่ 3.19

1. แหล่งจ่ายไฟ 5V 3A
2. โมดูลรีเลย์ 5V 10A
3. สายไฟ



รูปที่ 3.19 อุปกรณ์ที่ใช้ในการติดตั้งแหล่งจ่ายไฟ

ขั้นตอนการติดตั้งแหล่งจ่ายไฟ

1. โยงสายไฟสีเหลืองจากขั้วไฟฟ้า V- ไปยังช่องเสียบบนแท่นวาง ESP32 ด้านขวาบน เพื่อเป็นแหล่งจ่ายไฟลบให้กับอุปกรณ์

2. โยงสายไฟสีแดงจากขั้วไฟฟ้า V+ ไปยังช่องเสียบบนแท่นวาง ESP32 ด้านขวาล่าง เพื่อเป็นแหล่งจ่ายไฟบวกให้กับอุปกรณ์

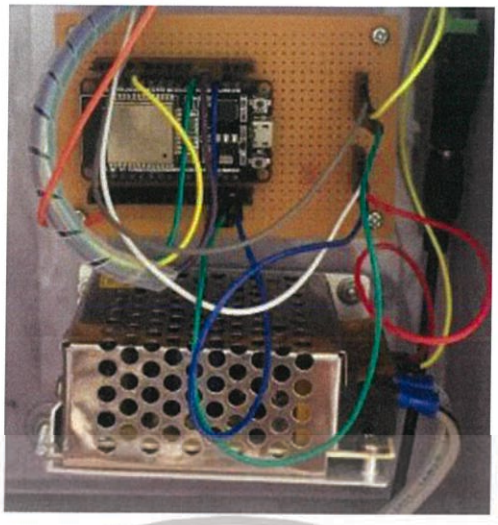
3. โยงสายไฟสีน้ำเงินจากขา VCC บน ESP32 ไปยังช่องเสียบบนแท่นวาง ESP32 ด้านขวาล่างที่เป็นแหล่งจ่ายไฟบวกจากหม้อแปลง

4. โยงสายไฟสีเขียวจากขา GND บน ESP32 ไปยังช่องเสียบบนแท่นวาง ESP32 ด้านขวาบนที่เป็นแหล่งจ่ายไฟลบจากหม้อแปลง

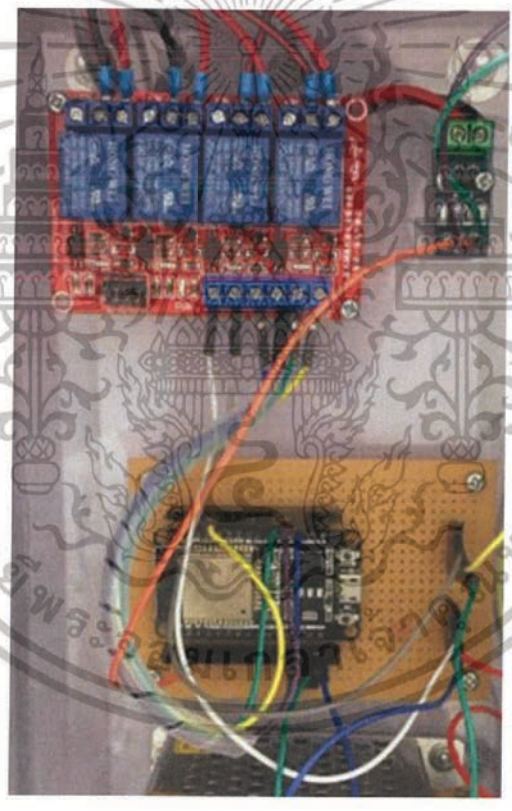
5. โยงสายสีขาวจากช่อง VCC บนรีเลย์ ไปยังช่องเสียบบนแท่นวาง ESP32 ด้านขวาล่างที่เป็นแหล่งจ่ายไฟบวกจากหม้อแปลง

6. โยงสายสีเทาจากช่อง GND บนรีเลย์ ไปยังช่องเสียบบนแท่นวาง ESP32 ด้านขวาบนที่เป็นแหล่งจ่ายไฟลบจากหม้อแปลง ดังรูปที่ 3.20 และรูปที่ 3.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 การติดตั้งแหล่งจ่ายไฟ (1)



รูปที่ 3.21 การติดตั้งแหล่งจ่ายไฟ (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

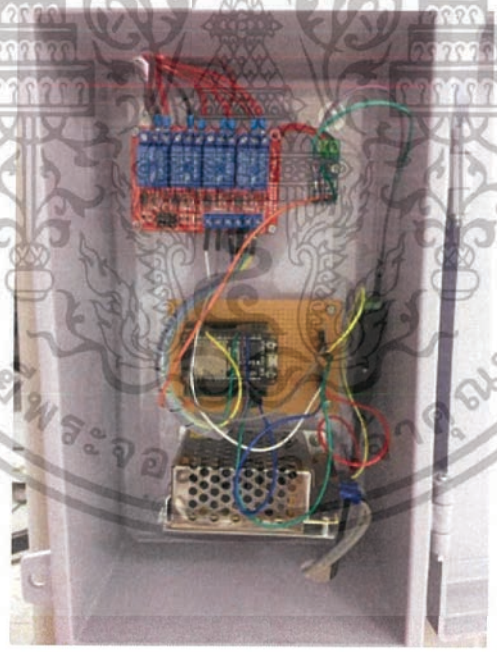
3.5.4 การติดตั้งส่วนประกอบต่างๆ ลงตู้

อุปกรณ์ที่ใช้ในการติดตั้งส่วนประกอบต่างๆ ลงตู้

1. ตู้พลาสติก
2. สว่าน
3. สายไฟ
4. แผ่นอะคริลิค
5. ไขควง
6. น็อต

ขั้นตอนการติดตั้งส่วนประกอบต่างๆ ลงตู้

1. ทำการวัดระยะสายไฟและขนาดเส้นผ่าศูนย์กลางของสายไฟที่จะเดินสายเข้าตู้
2. ทำการวัดขนาดของแผ่นอะคริลิคและวัดระยะการเจาะรูที่มุมทั้งสี่ด้านของแผ่นอะคริลิค
3. ทำการวัดระยะขนาดการเจาะของบนแผ่นอะคริลิคสำหรับอุปกรณ์อิเล็กทรอนิกส์ที่เกี่ยวข้อง
4. เจาะรูตามขนาดที่ต้องการบนผนังตู้พลาสติกและแผ่นอะคริลิค
5. ทำการยึดตู้พลาสติกกับผนังของเครื่องซักผ้า
6. ทำการยึดแผ่นอะคริลิคติดกับตู้พลาสติกด้วยตัวน็อตเสาค้ำ
7. ทำการยึดอุปกรณ์อิเล็กทรอนิกส์ต่างๆ กับแผ่นอะคริลิค
8. ทำการโยงสายไฟเข้าและโยงสายไฟออกจากตู้พลาสติก ดังรูปที่ 3.22

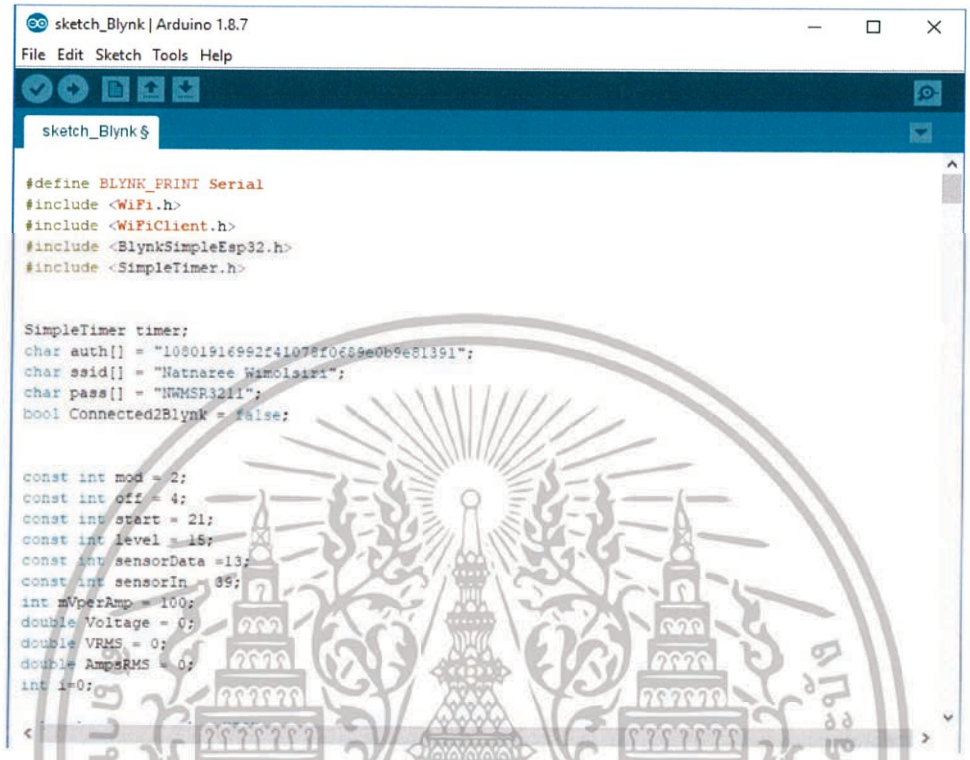


รูปที่ 3.22 การติดตั้งส่วนประกอบต่างๆ ลงตู้

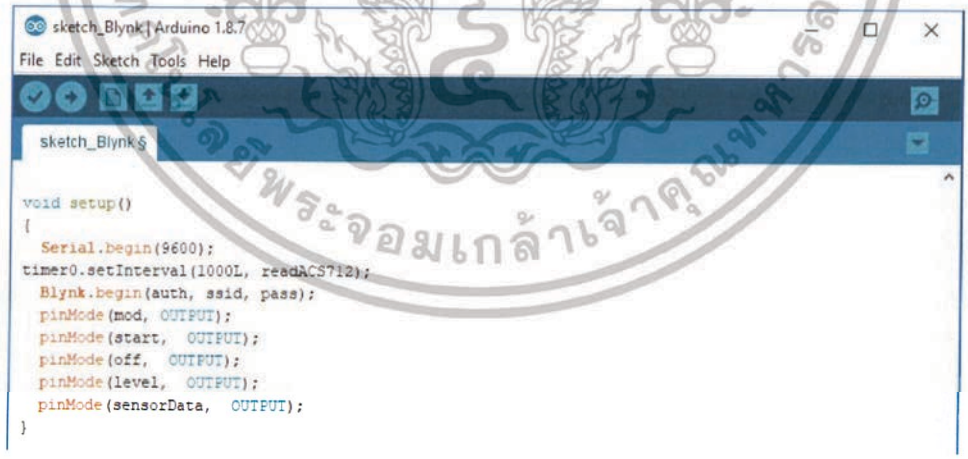
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.5 การเขียนโปรแกรม Arduino

ในขั้นตอนนี้เป็นขั้นตอนการเขียนโปรแกรม Arduino เพื่อรับค่าจากเซนเซอร์เข้ามาในรูปของตัวแปร เพื่อส่งไปให้กับโปรแกรมแสดงผลผ่านเครือข่าย Wi-Fi ดังรูปที่ 3.23 ถึงรูปที่ 3.25

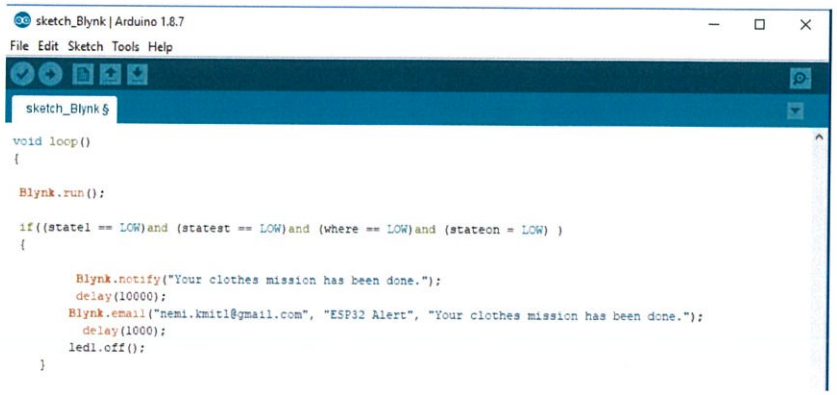


รูปที่ 3.23 การประกาศค่าอินพุตที่จะรับค่าจากเซนเซอร์



รูปที่ 3.24 การกำหนดความเร็วในการสื่อสารข้อมูลและกำหนดโหมดในการใช้ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 การกำหนดวงรอบของการทำงานของโปรแกรม

สำหรับการเขียนโปรแกรมเพื่ออ่านค่าจากเซนเซอร์แสดงในรูปที่ 3.26 ถึง รูปที่ 3.27

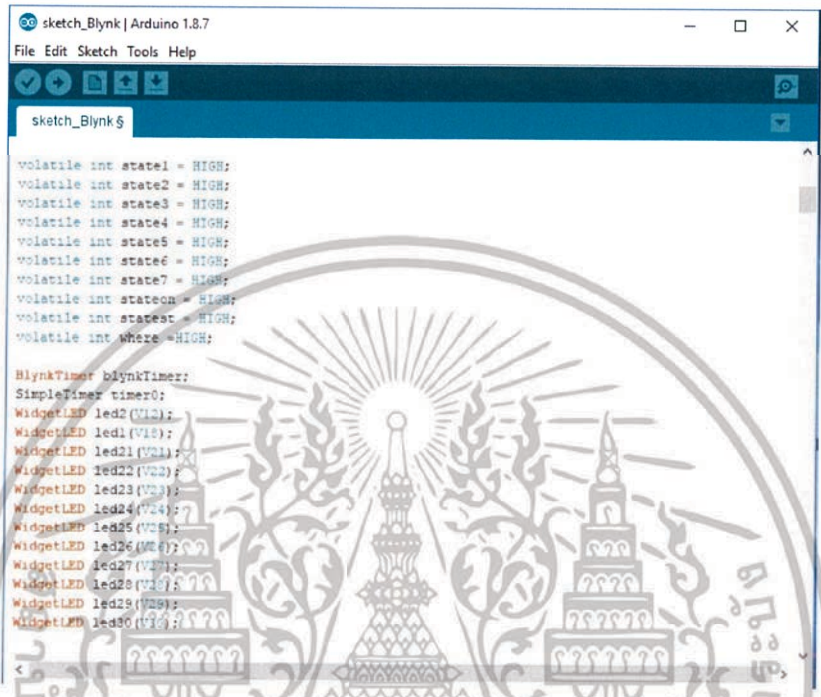
รูปที่ 3.26 โปรแกรมการอ่านค่าเซนเซอร์ ACS712 (1)

รูปที่ 3.27 โปรแกรมการอ่านค่าเซนเซอร์ ACS712 (2)

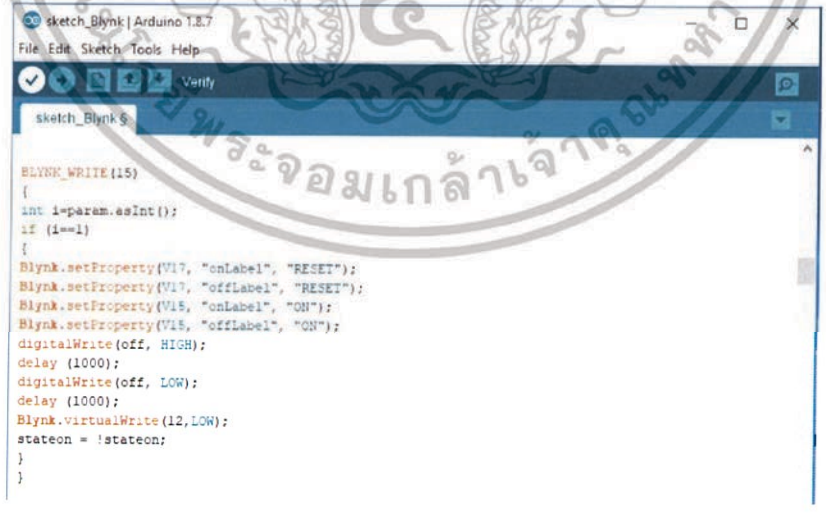
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6 การเขียนโปรแกรมแสดงผลบนแอปพลิเคชัน Blynk

ในขั้นตอนนี้เป็นขั้นตอนการเขียนโปรแกรม Arduino เพื่อส่งค่าจากเซนเซอร์ไปแสดงผลบนแอปพลิเคชัน และรับคำสั่งจากแอปพลิเคชันเพื่อสั่งเครื่องซักผ้าผ่านเครือข่าย Wi-Fi ดังรูปที่ 3.28 ถึงรูปที่ 3.30

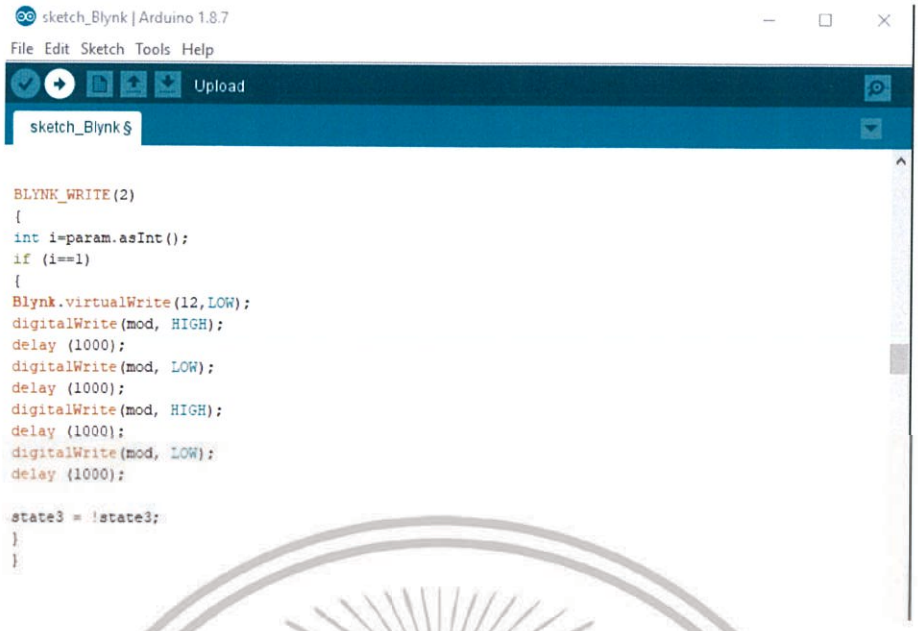


รูปที่ 3.28 การเขียนโปรแกรมแสดงผลบนแอปพลิเคชัน Blynk (1)



รูปที่ 3.29 การเขียนโปรแกรมแสดงผลบนแอปพลิเคชัน Blynk (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



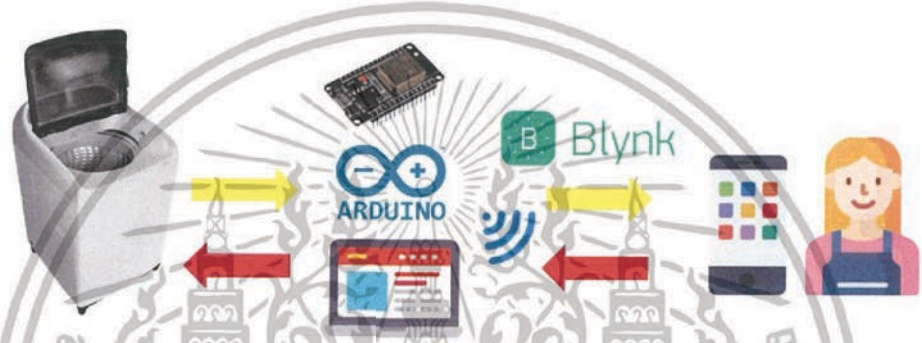
รูปที่ 3.30 การเขียนโปรแกรมแสดงผลบนแอปพลิเคชัน Blynk (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

4.1 แผนผังการทำงานของเครื่องซักผ้าอัจฉริยะผ่านเครือข่ายไร้สาย

เครื่องซักผ้าจะสื่อสารกับบอร์ด ESP32 ซึ่งทางบอร์ด ESP32 จำเป็นต้องถูกอัปโหลดโปรแกรมไว้แล้ว และผู้ใช้ต้องมีแอปพลิเคชัน Blynk เมื่อทางผู้ใช้สั่งการทำงานผ่านแอปพลิเคชัน ข้อมูลชุดคำสั่งจะส่งไปยังบอร์ด ESP32 ที่เชื่อมต่อ Wi-Fi จากนั้นบอร์ด ESP32 จะทำการสั่งการทำงานเครื่องซักผ้าทันที



รูปที่ 4.1 แผนผังการทำงานของเครื่องซักผ้าอัจฉริยะผ่านเครือข่ายไร้สาย

4.2 ผลการออกแบบหน้าแสดงผลผ่านแอปพลิเคชัน Blynk

ภายในหน้าแอปพลิเคชันสามารถแสดงผล 2 ส่วน ได้แก่

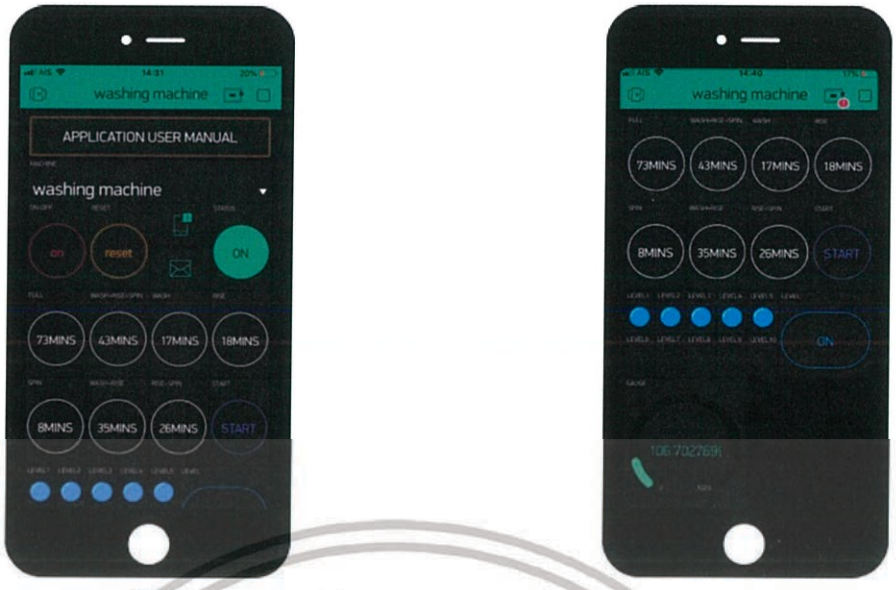
4.2.1 ส่วนของหน้าจอเลือกคำสั่ง

ในส่วนนี้จะแสดงให้เห็นถึงตัวอุปกรณ์ภายในแอปพลิเคชันที่เป็นชุดปุ่มคำสั่งต่างๆ ประกอบไปด้วยคำสั่งเปิดเครื่อง-ปิดเครื่อง คำสั่งโปรแกรมการซักที่จะระบุเวลาของการซักในแต่ละโปรแกรมนั้นๆ และคำสั่งเริ่มทำการซัก หยุดทำการซัก ดังแสดงรูปที่ 4.2 (ก)

4.2.2 ส่วนของหน้าจอแสดงผล

ในส่วนนี้จะแสดงให้เห็นถึงตัวอุปกรณ์ภายในแอปพลิเคชันที่เป็นตัวแสดงค่าผลลัพธ์ ได้แก่ ตัวแสดงค่าสถานะเปิดเครื่อง–ปิดเครื่อง ตัวไฟ LED แสดงสถานะระดับน้ำ และตัวเกจแสดงค่ากระแสไฟของเครื่องซักผ้าที่วัดผ่านโมดูล ACS712 ดังแสดงรูปที่ 4.2 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

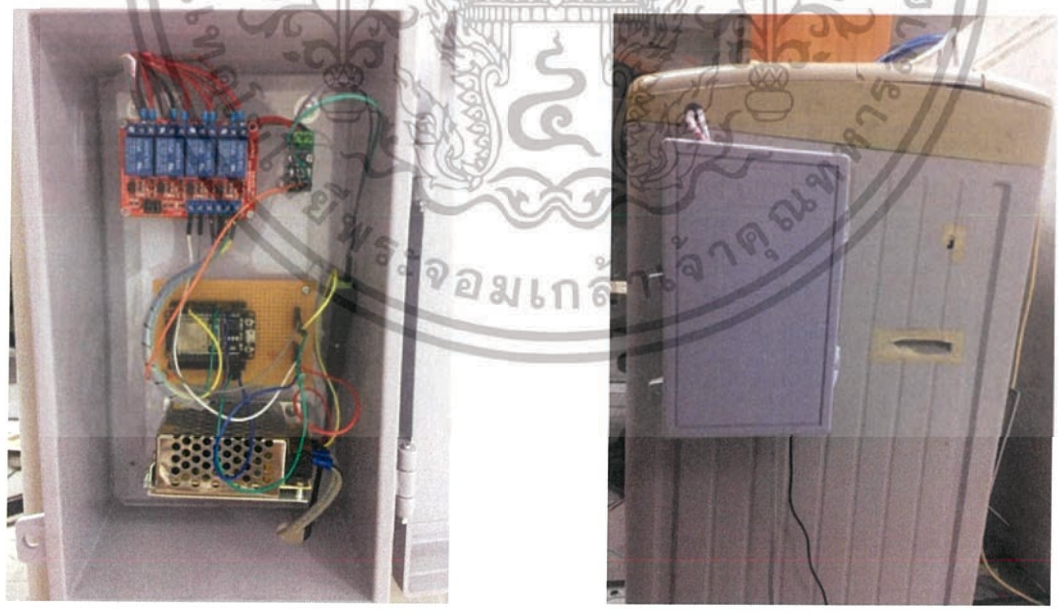


(ก) (ข)

รูปที่ 4.2 หน้าแอปพลิเคชัน Blynk

4.3 ผลการประดิษฐ์เครื่องซักผ้าอัจฉริยะ

ทำการติดตั้งกล่องควบคุมข้างเครื่องซักผ้า โดยภายในกล่องควบคุมประกอบด้วย วงจรอิเล็กทรอนิกส์ต่างๆ ที่ได้ออกแบบขึ้นดังแสดงในรูปที่ 4.3



รูปที่ 4.3 กล่องควบคุมประกอบติดกับเครื่องซักผ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดสอบการรับ–ส่งข้อมูลระหว่างเครื่องซักผ้าและแอปพลิเคชัน Blynk

ภายในหน้าแอปพลิเคชัน Blynk สามารถแสดงผล 2 ส่วน ได้แก่

4.4.1 ส่วนของผลเมื่อการทำงานของเครื่องซักผ้าเสร็จสิ้น

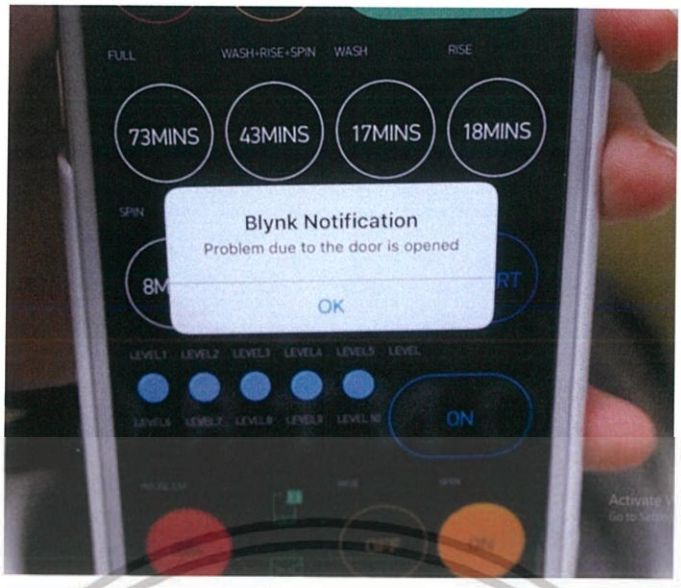
ในส่วนนี้จะแสดงให้เห็นถึงการทำงานของโปรแกรมเครื่องซักผ้าเสร็จสิ้น แอปพลิเคชันจะทำการแจ้งเตือนมายังหน้าสมาร์ทโฟนรวมถึงแจ้งเตือนผ่านระบบอีเมล์เป็นข้อความว่า “Your Clothes Mission has Been Done.” ดังแสดงรูปที่ 4.4

4.4.2 ส่วนของผลเมื่อการทำงานของเครื่องซักผ้ามีปัญหา

ในส่วนนี้จะแสดงให้เห็นถึงการทำงานของโปรแกรมเครื่องซักผ้า ได้เกิดปัญหาขึ้นระหว่างการทำงาน ปัญหาที่เกิดขึ้นดังกล่าว ได้แก่ ฝาบนได้ถูกเปิดขึ้นระหว่างการซัก ไม่มีน้ำไหลเข้าตัวถังของเครื่องซักผ้า รวมไปถึงการที่ไฟตกชั่วขณะทำให้เครื่องซักผ้าได้หยุดชะงักชั่วคราว แอปพลิเคชันจะทำการแจ้งเตือนมายังหน้าสมาร์ทโฟนรวมถึงแจ้งเตือนผ่านระบบอีเมล์เป็นข้อความว่า “Problem” ดังแสดงรูปที่ 4.5

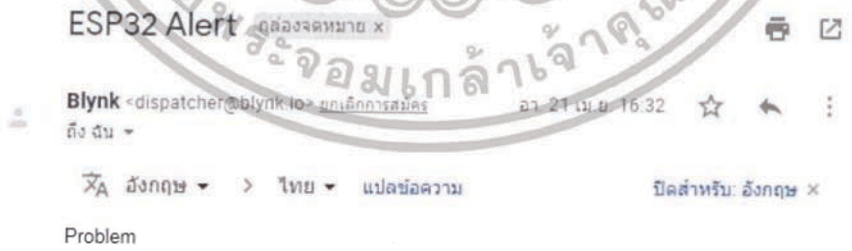
รูปที่ 4.4 การแจ้งเตือนของแอปพลิเคชันผ่านหน้าจอสมาร์ทโฟน (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การแจ้งเตือนของแอปพลิเคชันผ่านหน้าจอสมาร์ทโฟน (2)

รูปที่ 4.6 การแจ้งเตือนของแอปพลิเคชันผ่านระบบอีเมล์ (1)



รูปที่ 4.7 การแจ้งเตือนของแอปพลิเคชันผ่านระบบอีเมล์ (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานตามขั้นตอนที่ได้กล่าวมา เริ่มจากขั้นตอนแรกคือ ขั้นตอนของการศึกษาทฤษฎีและหลักการพร้อมทำความเข้าใจกับขอบเขตของโครงงาน ดังนั้นจึงได้แบ่งส่วนของการดำเนินงานเป็นทั้งหมด 3 ส่วน คือ

1. ส่วนของการสร้างด้านฮาร์ดแวร์ ในส่วนนี้ได้ศึกษาถึงหลักการการทำงานของเครื่องซักผ้า อุปกรณ์ด้านฮาร์ดแวร์ต่างๆ รวมถึงการออกแบบชิ้นงานให้สอดคล้องกับความต้องการ และวัตถุประสงค์ให้มากที่สุด

2. ส่วนของการเขียนโปรแกรมด้านซอร์ฟแวร์ เป็นส่วนของการเขียนโปรแกรมเพื่อการสื่อสารระหว่างแอปพลิเคชันและเครื่องซักผ้า รวมถึงการรับค่าของไอซีวัดกระแส ACS712 ด้วยโปรแกรม Arduino IDE อีกทั้งทำการออกแบบหน้าแอปพลิเคชัน Blynk ในการติดต่อระหว่างผู้ใช้กับเครื่องซักผ้าเพื่อให้ผู้ใช้สามารถใช้งานได้โดยง่าย

3. หลังจากที่ได้ออกแบบและประดิษฐ์เครื่องซักผ้าอัจฉริยะเรียบร้อยแล้ว จึงนำชิ้นงานมาทำการทดสอบด้วยการสั่งการทำงานผ่านแอปพลิเคชัน Blynk ด้วยสมาร์ทโฟน พบว่าเครื่องซักผ้าสามารถใช้งานได้จริง บอร์ด ESP32 สามารถเชื่อมต่ออินเทอร์เน็ตได้เพื่อส่งข้อมูลแบบ Real-Time เป็นตัวกลางในการสื่อสารระหว่างเครื่องซักผ้าและแอปพลิเคชันได้

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

1. เนื่องจากโครงงานนี้มีเรื่องใหม่ๆ ที่ยังไม่เชี่ยวชาญ จึงจำเป็นที่ต้องใช้เวลาในการศึกษาหาความรู้เพิ่มเติมเพื่อที่จะนำมาใช้ในโครงงานนี้

2. ในการประกอบระหว่างอุปกรณ์ภายในกล่องควบคุมและเครื่องซักผ้า จำเป็นต้องใช้ทักษะในการบัดกรีและเชื่อมโยงสาย จึงต้องใช้เวลาในการวางแผนและปฏิบัติ

3. การเขียนโปรแกรม Arduino IDE ยังขาดความชำนาญในการเขียน ซึ่งจำเป็นต้องใช้เวลาในการศึกษารูปแบบการเขียนโปรแกรมเพิ่มเติมในช่วงแรก และยังมีข้อบกพร่องในความละเอียด คืออาจมีการตกหล่นของตัวอักษรในระหว่างการเขียนโปรแกรม ทำให้การรันโค้ดในโปรแกรมจึงไม่ผ่านบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการแก้ไขปัญหา

1. ศึกษาและสอบถามแนวทางเพิ่มเติมจากสื่อการสอนต่างๆ รวมถึงอาจารย์ที่ปรึกษา
2. วางแผนโครงสร้างของด้านฮาร์ดแวร์ทั้งหมดก่อนทำการประกอบ
3. ศึกษาการเขียนโปรแกรมและสอบถามจากเพื่อนที่มีความชำนาญในการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] “**หลักการของ บอร์ด ESP 32**” เข้าถึงได้จาก: https://www.espressif.com/en /products/hardware/esp32/overview

[2] “**หลักการของโมดูลตรวจจับกระแส ACS712**” เข้าถึงได้จาก: http://henrysbench.capnfatz.com/henrys-bench/arduino-current-measurements/the-acs712-current-sensor-with-an-arduino/

[3] “**การเขียนโปรแกรม Arduino**” เข้าถึงได้จาก: https://www.arduino.cc/en/Guide/HomePage

[4] “**การเขียนแอปพลิเคชัน Blynk** ” เข้าถึงได้จาก: https://examples.blynk.cc/?board=ESP8266&shield=ESP8266%20WiFi&example=GettingStarted%2FBlynkBlink

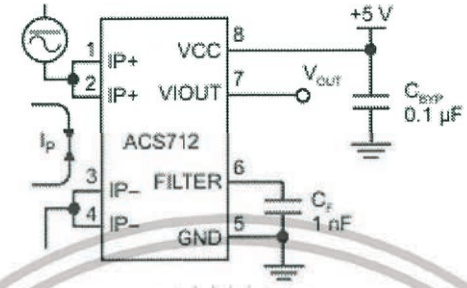
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

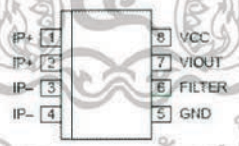
โมดูลตรวจจับกระแส ACS712



Application 1. The ACS712 outputs an analog signal, V_{OUT}, that varies linearly with the uni- or bi-directional AC or DC primary sensed current, I_P, within the range specified. C_F is recommended for noise management, with values that depend on the application.

รูปที่ ก.1 คุณสมบัติทั่วไปของ โมดูลตรวจจับกระแส ACS712 (1)

Pin-out Diagram

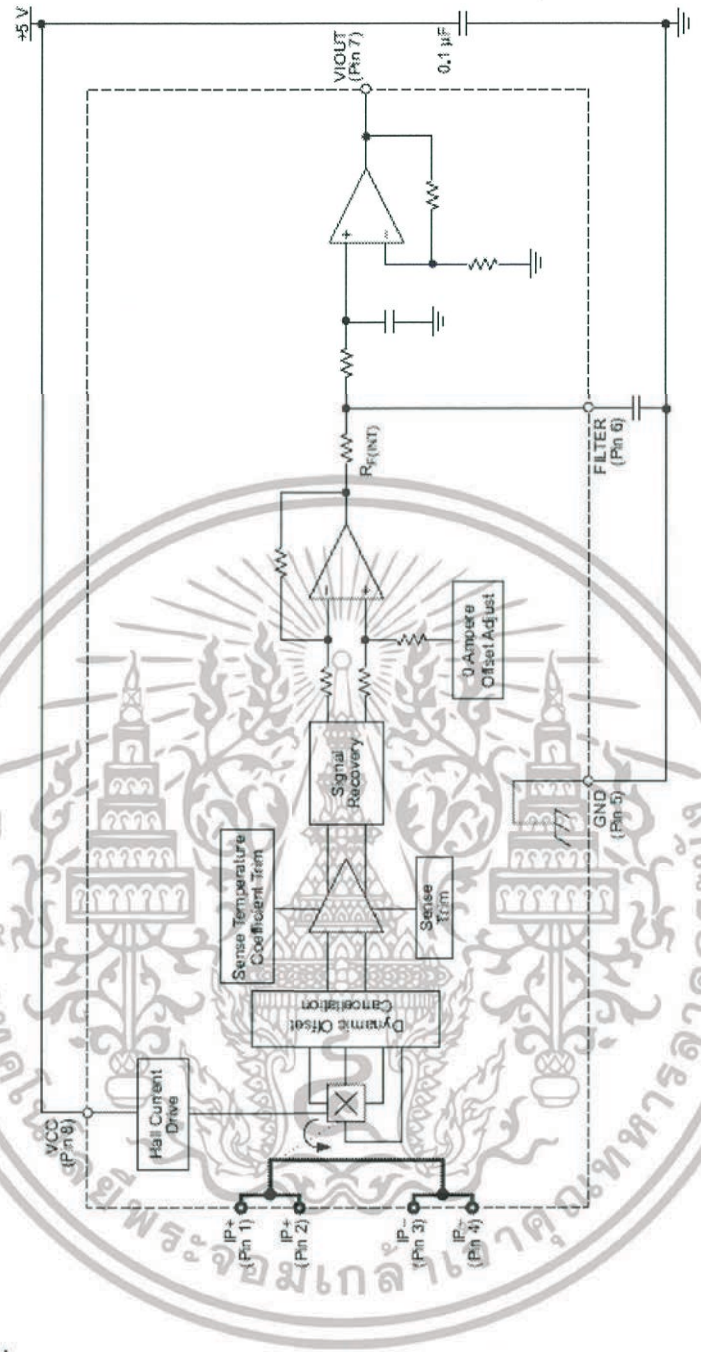


Terminal List Table

Number	Name	Description
1 and 2	IP+	Terminals for current being sampled; fused internally
3 and 4	IP–	Terminals for current being sampled; fused internally
5	GND	Signal ground terminal
6	FILTER	Terminal for external capacitor that sets bandwidth
7	VIOUT	Analog output signal
8	VCC	Device power supply terminal

รูปที่ ก.2 คุณสมบัติทั่วไปของ โมดูลตรวจจับกระแส ACS712 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

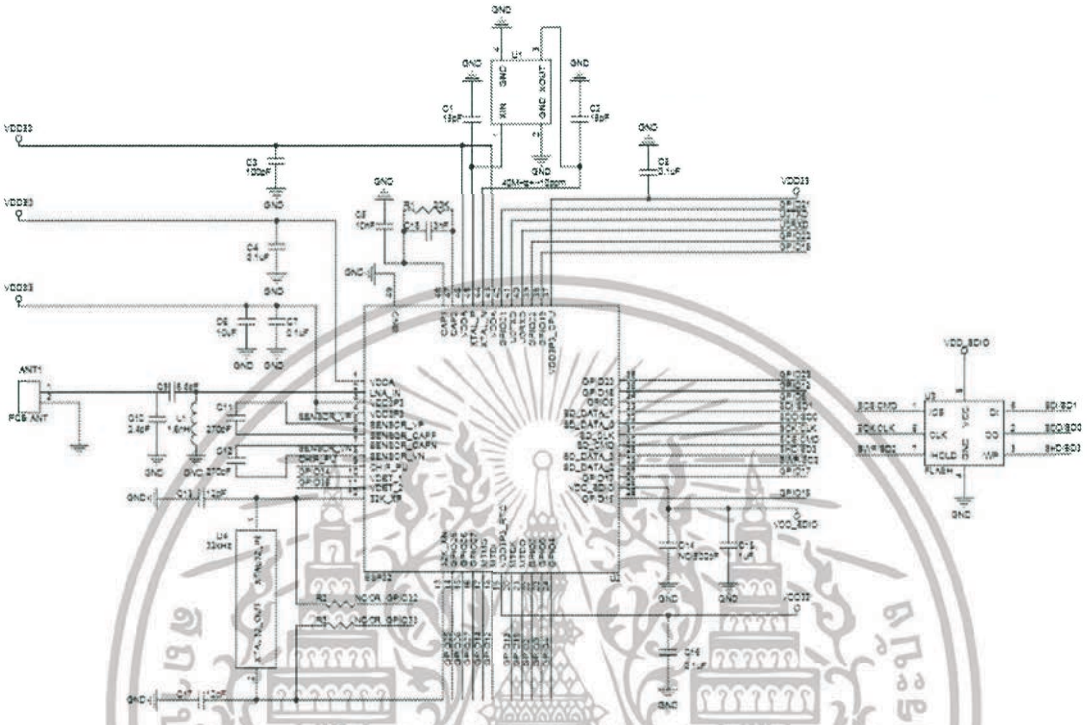


รูปที่ ก.3 Block Diagram ของ โมดูลตรวจจับกระแส ACS712

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

บอร์ด ESP32



รูปที่ ข.1 คุณสมบัติของบอร์ด ESP32 (1)

Table 2: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8

รูปที่ ข.2 คุณสมบัติของบอร์ด ESP32 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name	No.	Type	Function
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

รูปที่ ข.3 คุณสมบัติทั่วไปของ บอร์ด ESP32 (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โปรแกรมควบคุมการทำงานและการแสดงผล

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <SimpleTimer.h>

SimpleTimer timer;
char auth[] = "10801916992f41078f0689e0b9e81391";
char ssid[] = "Natnaree Wimolsiri";
char pass[] = "NWMSR3211";
bool Connected2Blynk = false;

const int mod = 2;
const int off = 4;
const int start = 21;
const int level = 15;
const int sensorData =13;
const int sensorIn = 39;
int mVperAmp = 100;
double Voltage = 0;
double VRMS = 0;
double AmpsRMS = 0;
int i=0;

volatile int state1 = HIGH;
volatile int state2 = HIGH;
volatile int state3 = HIGH;
volatile int state4 = HIGH;
volatile int state5 = HIGH;
volatile int state6 = HIGH;
volatile int state7 = HIGH;
volatile int stateon = HIGH;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
volatile int statest = HIGH;
volatile int where =HIGH;

BlynkTimer blynkTimer;
SimpleTimer timer0;
WidgetLED led2(V12);
WidgetLED led1(V18);
WidgetLED led21(V21);
WidgetLED led22(V22);
WidgetLED led23(V23);
WidgetLED led24(V24);
WidgetLED led25(V25);
WidgetLED led26(V26);
WidgetLED led27(V27);
WidgetLED led28(V28);
WidgetLED led29(V29);
WidgetLED led30(V30);

BLYNK_WRITE(20)
{
int i=param.asInt();
if (i==1)
{
digitalWrite(start, HIGH);
delay (1000);
digitalWrite(start, LOW);
delay (1000);
Blynk.setProperty(V17, "onLabel", "DISABLE");
Blynk.setProperty(V17, "offLabel", "DISABLE");
Blynk.setProperty(V15, "onLabel", "DISABLE");
Blynk.setProperty(V15, "offLabel", "DISABLE");
statest = !statest;
}
}

BLYNK_WRITE(17)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int i=param.asInt();
if (i==1)
{
digitalWrite(off, HIGH);
delay (1000);
digitalWrite(off, LOW);
delay (1000);
digitalWrite(off, HIGH);
delay (1000);
digitalWrite(off, LOW);
delay (1000);
Blynk.virtualWrite(12,LOW);
led21.on();
led22.on();
led23.on();
led24.on();
led25.on();
led26.off();
}
}

BLYNK_WRITE(15)
{
int i=param.asInt();
if (i==1)
{
Blynk.setProperty(V17, "onLabel", "RESET");
Blynk.setProperty(V17, "offLabel", "RESET");
Blynk.setProperty(V15, "onLabel", "ON");
Blynk.setProperty(V15, "offLabel", "ON");
digitalWrite(off, HIGH);
delay (1000);
digitalWrite(off, LOW);
delay (1000);
Blynk.virtualWrite(12,LOW);
led1.on();
led21.on();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
led22.on();
led23.on();
led24.on();
led25.on();
led26.off();
stateon = !stateon;
}
}

BLYNK_WRITE(0)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(12,LOW);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
state1 = !state1;
}
}

BLYNK_WRITE(1)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(12,LOW);
state2 = !state2;
}
}

BLYNK_WRITE(2)
{
int i=param.asInt();
if (i==1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
Blynk.virtualWrite(12,LOW);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
state3 = !state3;
}
}

BLYNK_WRITE(3)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(12,LOW);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
state4 = !state4;
}
}

BLYNK_WRITE(4)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(12,LOW);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
state5 = !state5;
}
}

BLYNK_WRITE(5)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(12,LOW);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
state6 = !state6;
}
}

BLYNK_WRITE(6)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(12,LOW);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
digitalWrite(mod, HIGH);
delay (1000);
digitalWrite(mod, LOW);
delay (1000);
state7 = !state7;
}
}

BLYNK_WRITE(7)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(12,LOW);
digitalWrite(level, LOW);
delay (1000);
digitalWrite(level, HIGH);
delay (1000);
led26.on();
}
}

BLYNK_WRITE(33)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(V12, HIGH);
delay (1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Blynk.virtualWrite(V9, HIGH);
delay (1000);
}
}

BLYNK_WRITE(34)
{
int i=param.asInt();
if (i==1)
{
Blynk.notify("Your clothes mission has been done");
}
}

BLYNK_WRITE(35)
{
int i=param.asInt();
if (i==1)
{
Blynk.virtualWrite(V36, HIGH);
delay (1000);
Blynk.notify("Problem due to the door is opened");
}
}
boolean connectInProgress;
int connectTimerId;
float getVPP()
{
  float result;
  int readValue;
  int maxValue = 0;
  int minValue = 1024;
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000)
   {
       readValue = analogRead(sensorIn);
       if (readValue > maxValue)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            maxValue = readValue;
        }
        if (readValue < minValue)
        {
            minValue = readValue;
        }
    }
    result = ((maxValue - minValue) * 5.0)/1024.0;
        return result;
        }
void readACS712()
{
 Voltage = getVPP();
 VRMS = (Voltage/2.0) *0.707;
 AmpsRMS = (VRMS * 1000)/mVperAmp;
 Serial.print(AmpsRMS);
 Serial.println(" Amps RMS");
 Blynk.virtualWrite(V13, AmpsRMS);
 pinMode(sensorIn, INPUT);
}

void setup()
{
  Serial.begin(9600);
timer0.setInterval(1000L, readACS712);
  Blynk.begin(auth, ssid, pass);
  pinMode(mod, OUTPUT);
  pinMode(start,  OUTPUT);
  pinMode(off,  OUTPUT);
  pinMode(level,  OUTPUT);
  pinMode(sensorData,  OUTPUT);
}

void loop()
{
 Blynk.run();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if((state1 == LOW)and (statest == LOW)and (where == LOW)and (stateon = LOW) )
{
        Blynk.notify("Your clothes mission has been done.");
        delay(10000);
       Blynk.email("nemi.kmitl@gmail.com", "ESP32 Alert", "Your clothes mission has been done.");
         delay(1000);
       led1.off();
   }
   if((state2 == LOW)and (statest == LOW)and (where == LOW)and (stateon = LOW) )
{
       Blynk.notify("Your clothes mission has been done.");
       delay(10000);
        Blynk.email("nemi.kmitl@gmail.com", "ESP32 Alert", "Your clothes mission has been done.");
        delay(1000);
       led1.off();
   }
   if((state3 == LOW)and (statest == LOW)and (where == LOW)and (stateon= LOW) )
{
       Blynk.notify("Your clothes mission has been done.");
       delay(10000);
        Blynk.email("nemi.kmitl@gmail.com", "ESP32 Alert", "Your clothes mission has been done.");
        delay(1000);
       led1.off();
   }
   if((state4 == LOW)and (statest == LOW)and (where == LOW)and (stateon= LOW) )
{
       delay(1000);
        Blynk.email("nemi.kmitl@gmail.com", "ESP32 Alert", "Your clothes mission has been done.");
        delay(1000);
       led1.off();
   }
   if((state5 == LOW)and (statest == LOW)and (where == LOW)and (stateon = LOW))
{
       Blynk.notify("Your clothes mission has been done.");
       delay(1000);
        Blynk.email("nemi.kmitl@gmail.com", "ESP32 Alert", "Your clothes mission has been done.");
        delay(1000);
```

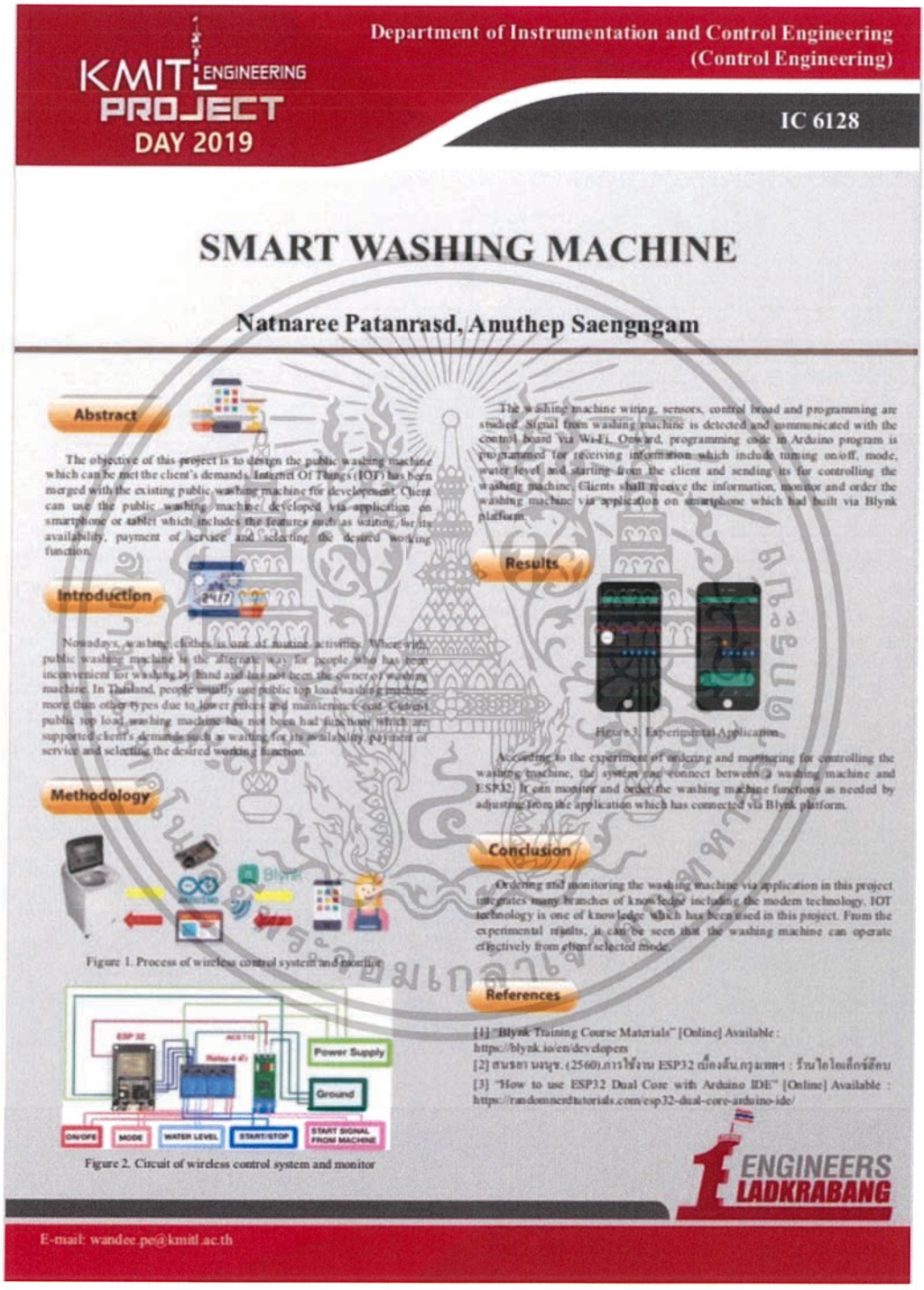
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        led1.off();
    }
    if((state6 == LOW)and (statest == LOW)and (where == LOW) and (stateon = LOW))
{
        Blynk.notify("Your clothes mission has been done.");
         Blynk.email("nemi.kmitl@gmail.com", "ESP32 Alert", "Your clothes mission has been done.");
         delay(1000);
        led1.off();
    }
    if((state7 == LOW)and (statest == LOW)and (where == LOW)and (stateon= LOW))
{
        Blynk.notify("Your clothes mission has been done.");
        delay(10000);
         Blynk.email("nemi.kmitl@gmail.com", "ESP32 Alert", "Your clothes mission has been done.");
        delay(1000);
        led1.off();
    }
blynkTimer.run();
 timer0.run();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

โปสเตอร์



รูปที่ ง.1 โปสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้