ระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร
CACTUS GREENHOUSE CONTROL MONITORING SYSTEM

โดย

นางสาวกาญจนา ทองวิจิตร

นางสาวเกศแก้ว เจริญสุข

นางสาวประภัสสร ปั่นเมืองปัก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร
CACTUS GREENHOUSE CONTROL MONITORING SYSTEM

โดย

นางสาวกาญจนา	ทองวิจิตร	61010068
นางสาวเกศแก้ว	เจริญสุข	61010100
นางสาวประภัสสร	ปั่นเมืองปัก	61010632

อาจารย์ที่ปรึกษา

ศ. ดร.ชูวงค์ พงศ์เจริญพาณิชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร

CACTUS GREENHOUSE CONTROL MONITORING SYSTEM

ผู้จัดทำ

1. นางสาวกาญจนา ทองวิจิตร 61010068
2. นางสาวเกศแก้ว เจริญสุข 61010100
3. นางสาวประภัสสร ปั่นเมืองปัก 61010632

..... **อาจารย์ที่ปรึกษา**

(ศ. ดร.ชูวงค์ พงศ์เจริญพาณิชย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินปริญญานิพนธ์เรื่อง “ระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความความอนุเคราะห์และสนับสนุนอย่างดียิ่งจาก อาจารย์ที่ปรึกษาปริญญานิพนธ์คือ ศ.ดร.ชูวงค์ พงศ์เจริญพาณิชย์ รวมทั้งรุ่นพี่ทุกท่านในห้องปฏิบัติการทดลองที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษา ค้นคว้าวิจัยให้ปริญญานิพนธ์นี้สำเร็จสมบูรณ์ยิ่งขึ้น รวมถึงสนับสนุนสถานที่ เครื่องมือและอุปกรณ์ต่าง ๆ ที่จำเป็นต้องใช้ในการจัดทำปริญญานิพนธ์ ขอขอบพระคุณในความห่วงใย และความปรารถนาดีให้แก่คณะผู้จัดทำเป็นอย่างยิ่ง

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอนประสิทธิ์ประสาทวิชา ความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาในยามที่เกิดปัญหา จนกระทั่งปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดีและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษาอันมีค่ายิ่งแก่ผู้จัดทำ

นางสาวกาญจนา ทองวิจิตร
นางสาวเกศแก้ว เจริญสุข
นางสาวประภัสสร ปั่นเมืองปัก
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร
CACTUS GREENHOUSE CONTROL MONITORING SYSTEM

โดย นางสาวกาญจนา ทองวิจิตร 61010068
นางสาวเกศแก้ว เจริญสุข 61010100
นางสาวประภัสสร ปั่นเมืองปัก 61010632

อาจารย์ที่ปรึกษา ศ. ดร.ชูวงค์ พงศ์เจริญพาณิชย์

บทคัดย่อ

ในปัจจุบันได้มีการนำเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งมาประยุกต์ใช้ในชีวิตประจำวันมากขึ้นโดยเฉพาะด้านการเกษตร ทางผู้จัดทำเล็งเห็นถึงความสำคัญของพืชพันธุ์กระบองเพชร เนื่องจากเป็นพืชด้านการเกษตรที่กำลังได้รับความนิยมเป็นอย่างยิ่ง จึงได้นำเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งมาประยุกต์ใช้กับพืชพันธุ์กระบองเพชรเพื่อคิดค้นอุปกรณ์และพัฒนาเป็นระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร โดยระบบประกอบด้วย 2 ส่วนหลัก ๆ คือส่วนแสดงผลค่าเซนเซอร์ต่าง ๆ ภายในโรงเรือน ซึ่งจะทำงานร่วมกับไมโครคอนโทรลเลอร์และจัดเก็บไปยังฐานข้อมูลแบบตามเวลาจริงและส่วนการควบคุมอุปกรณ์ต่าง ๆ ภายในโรงเรือนกระบองเพชร โดยการทำงานในส่วนนี้จะมีการทำงานสองรูปแบบคือรูปแบบอัตโนมัติ และการทำงานรูปแบบควบคุมด้วยตัวผู้ใช้เอง ซึ่งการทำงานของระบบทั้งหมดจะสั่งงานผ่านแอปพลิเคชันบนโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

Nowadays, Internet of Things technology is increasingly applied in daily life, especially in agriculture. We see the importance of cactus plants because it is an agricultural plant that is gaining popularity. Therefore, the Internet of Things technology has been applied to cactus plants in order to invent a device and develop as a cactus greenhouse control monitoring system. The system consists of two main sections. It begins with the display section of the sensor values in the greenhouse by using a real-time database with a microcontroller to store sensor values and the devices control section within the cactus greenhouse. By working in this section, there are two modes of operation, i.e., automatic mode and manual mode, which all system functions are ordered through an application on a mobile phone.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		IV
สารบัญรูป		VI
สารบัญตาราง		IX
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของปริญญานิพนธ์	2
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 การศึกษาลักษณะทั่วไปของกระบองเพชร	3
	2.2 การศึกษาสภาพแวดล้อมที่เหมาะสมสำหรับการเพาะเลี้ยงกระบองเพชร	4
	2.3 การศึกษารูปแบบโรงเรือน	8
	2.4 การศึกษาส่วนประกอบโรงเรือน	9
	2.5 NODEMCU ESP32	11
	2.6 โมดูลเซนเซอร์วัดความเข้มแสง GY-302 BH1750	13
	2.7 โมดูลเซนเซอร์วัดอุณภูมิและความชื้นในอากาศ DHT22	14
	2.8 โมดูลเซนเซอร์วัดความชื้นในดิน	15
	2.9 โมดูลรีเลย์ 3 โวลต์ (4 Channel)	16
	2.10 หลอดไฟปลูกพืช (LED Grow Light)	16
	2.11 พัดลมระบายอากาศ	17
	2.12 ปั๊มแรงดันขนาด 4.8 บาร์	18
	2.13 อุปกรณ์จ่ายไฟฟ้ากระแสตรง	18
	2.14 ท่อ PE ขนาด 0.9 เซนติเมตร	19
	2.15 ชุดสปริงเกอร์ขนาดเล็กและข้อต่อ	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	2.16 ระบบปฏิบัติการ Android	19
	2.17 Android Studio	20
	2.18 ศึกษาการเก็บข้อมูลแบบตามเวลาจริง	23
	2.19 Arduino IDE	24
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**25**
	3.1 การออกแบบ	25
	3.2 เครื่องมือที่ใช้ในการทดสอบ	54
	3.3 การจัดเก็บผลการทดสอบ	57
บทที่ 4	**ผลการทดลอง**	**58**
	4.1 ผลการทดสอบการทำงานของชุดคำสั่งควบคุมอุปกรณ์	58
	4.2 ผลการทดสอบการรับส่งข้อมูลระหว่าง NODEMCU ESP32 กับฐานข้อมูล FIREBASE REALTIME	58
	4.3 ผลการทดสอบการใช้งานเซนเซอร์และอุปกรณ์ภายในโรงเรือน	60
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**67**
	5.1 สรุปผล	67
	5.2 ข้อเสนอแนะ	68
บรรณานุกรม		**69**
ภาคผนวก ก	คำสั่งควบคุมการทำงานของอุปกรณ์	73
ภาคผนวก ข	คำสั่งสำหรับการเข้าสู่ระบบบนแอปพลิเคชัน	83
ภาคผนวก ค	คำสั่งสำหรับลงทะเบียนผู้ใช้บนแอปพลิเคชัน	93
ภาคผนวก ง	คำสั่งกรณีผู้ใช้ลืมรหัสผ่านบนแอปพลิเคชัน	98
ภาคผนวก จ	คำสั่งแสดงสถานะการทำงานของอุปกรณ์บนแอปพลิเคชัน	102
ภาคผนวก ฉ	คำสั่งควบคุมระบบการทำงานของอุปกรณ์บนแอปพลิเคชัน	111
ภาคผนวก ช	คำสั่งสำหรับแสดงคู่มือการใช้งานหน้าควบคุม	131
ภาคผนวก ซ	คำสั่งสำหรับออกจากระบบบนแอปพลิเคชัน	134

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	โรงเรือนอัจฉริยะ SMART GREEN HOUSE ระบบปิด	8
2.2	เหล็กกัลวาไนซ์ขนาด 1.9 เซนติเมตร ความหนา 0.15 เซนติเมตร	9
2.3	เหล็กแบน	10
2.4	เหล็กเส้นกลม	10
2.5	พลาสติก PE 200 ไมครอน	11
2.6	NODEMCU ESP32 DEV MODULE	12
2.7	โมดูลวัดความเข้มแสง GY-302 BH1750	14
2.8	โมดูลเซนเซอร์วัดอุณหภูมิและความชื้น DHT22	15
2.9	โมดูลเซนเซอร์วัดความชื้นในดิน	15
2.10	โมดูลรีเลย์ 3 โวลต์ (Active low)	16
2.11	LED Grow Light 50 วัตต์	17
2.12	พัดลมระบายอากาศขนาด 12 เซนติเมตร	17
2.13	ปั๊มน้ำ รุ่น Green-01 DC 12 โวลต์	18
2.14	อุปกรณ์จ่ายไฟฟ้ากระแสตรง	18
2.15	ท่อ PE ขนาด 0.9 เซนติเมตร	19
2.16	ชุดสปริงเกอร์ขนาดเล็กและข้อต่อ	19
2.17	หน้าต่างโปรแกรม Android Studio	20
2.18	ANDROID SOFTWARE DEVELOPMENT KIT	21
2.19	หน้าต่างตัวอย่างการออกแบบหน้าจอเข้าสู่ระบบโดยใช้ภาษา XML ผ่าน ANDROID STUDIO	22
2.20	ตัวอย่างการเก็บข้อมูลแบบตามเวลาจริงที่ได้จากการลงทะเบียนผ่านทางแอปพลิเคชัน	23
2.21	ตัวอย่างบริการตรวจสอบผู้ใช้ที่ได้จากการลงทะเบียนผ่านทางแอปพลิเคชัน	24
2.22	หน้าต่างโปรแกรม Arduino IDE	24
3.1	แผนผังการทำงานของระบบ	26
3.2	แผนผังควบคุมการเปิดและปิดของสวิตซ์หลอดไฟแบบอัตโนมัติ	27
3.3	แผนผังควบคุมการเปิดและปิดของสวิตซ์พัดลมแบบอัตโนมัติ	28
3.4	แผนผังควบคุมการเปิดและปิดของสวิตซ์ปั๊มน้ำแบบอัตโนมัติ	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.5	ภาพวงจรโดยรวมของระบบ	29
3.6	แสดงชื่อที่อยู่ไฟล์ที่เชื่อมต่อระหว่างอุปกรณ์และ Firebase	31
3.7	หน้าสำหรับเข้าสู่ระบบ	32
3.8	หน้าจอแสดงลิงค์สำหรับยืนยันตัวตนก่อนทำการเข้าสู่ระบบครั้งแรก	33
3.9	การยืนยันตัวตนเสร็จสิ้น	33
3.10	หน้าสำหรับลงทะเบียนบัญชีผู้ใช้	34
3.11	หน้าสำหรับกรอกอีเมลของผู้ใช้ในกรณีลืมรหัสผ่าน	35
3.12	การแจ้งเตือนสำหรับการตั้งรหัสผ่านใหม่	36
3.13	หน้าแสดงค่าที่ได้รับจากเซนเซอร์ที่เก็บบนฐานข้อมูล Firebase	37
3.14	สถานะบน Firebase กรณีที่อุปกรณ์มีการเปิดใช้งาน	37
3.15	สถานการณ์ทำงานของอุปกรณ์บนแอปพลิเคชันเมื่อมีการเปิดใช้งาน	38
3.16	สถานะบน Firebase กรณีที่อุปกรณ์ไม่มีการเปิดใช้งาน	38
3.17	สถานะการทำงานของอุปกรณ์บนแอปพลิเคชันเมื่อไม่มีการเปิดใช้งาน	38
3.18	หน้าต่างสำหรับควบคุมการทำงานของอุปกรณ์	39
3.19	หน้าสำหรับแสดงคู่มือผู้ใช้	40
3.20	หน้าสำหรับออกจากระบบ	41
3.21	แผนผังการทำงานของแอปพลิเคชัน	42
3.22	โครงสร้างด้านหน้าของหลังคาโรงเรือน	43
3.23	โครงสร้างด้านบนของหลังคาโรงเรือน	43
3.24	โครงสร้างด้านข้างของหลังคาโรงเรือน	44
3.25	โครงสร้างด้านหน้าของชั้นวางกระถางต้นกระบองเพชร	44
3.26	โครงสร้างด้านบนของชั้นวางกระถางต้นกระบองเพชร	45
3.27	โครงสร้างด้านข้างของชั้นวางกระถางต้นกระบองเพชร	45
3.28	เสาสำหรับโครงสร้างโรงเรือน	46
3.29	เสาที่ออกแบบมาเพื่อติดตั้งพัดลม	46
3.30	ข้อต่อที่ออกแบบมาเพื่อติดตั้งพัดลม	47
3.31	ข้อต่อระหว่างฐานโครงสร้าง	47
3.32	แบบตัดเย็บพลาสติกคลุมโรงเรือน	48
3.33	ออกแบบเพื่อติดตั้งบริเวณคานของโรงเรือน	49
3.34	แสดงระยะห่างระหว่างสปริงเกอร์หัวพ่นหมอก	49
3.35	โครงสร้างด้านหน้าของหลังคาโรงเรือน	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.36 โครงสร้างด้านหน้าของชั้นวางกระถางต้นกระบองเพชร 50
3.37 เสาสำหรับโครงสร้างโรงเรือน 51
3.38 เสาที่ออกแบบมาเพื่อติดตั้งพัดลม 51
3.39 ข้อต่อที่เป็นมุมทั้ง 4 ของฐานโรงเรือน 52
3.40 ข้อต่อที่ออกแบบมาเพื่อติดตั้งพัดลม 52
3.41 แบบตัดเย็บพลาสติกคลุมโรงเรือน 53
3.42 แบบตัดเย็บตะข่ายกันแมลง 53
3.43 ติดตั้งชุดมินิสปริงเกอร์ตามที่ออกแบบไว้ 54
3.44 อุปกรณ์แปลงแรงดันไฟฟ้า 56
3.45 Xiaomi Redmi note9 56
4.1 ผลการทดสอบการทำงานตามเงื่อนไขที่กำหนดในโหมดอัตโนมัติ 58
4.2 แสดงผลการเชื่อมต่อ Wi-Fi ในโหมด STP 59
4.3 แสดงชื่อที่อยู่ของไฟล์ที่เชื่อมต่อระหว่างอุปกรณ์และ Firebase 59
4.4 ตำแหน่งการทดสอบติดตั้งเซนเซอร์เพื่อวัดความเข้มแสงภายในโรงเรือน 60
4.5 ตำแหน่งการติดตั้งเซนเซอร์ภายในโรงเรือน 61
4.6 กราฟการเปลี่ยนแปลงค่าอุณหภูมิเมื่อเปิดใช้งานพัดลมเป็นเวลา 15 นาที 62
4.7 กราฟความสัมพันธ์ระหว่างความเข้มแสงกับช่วงเวลา 65
4.8 ตำแหน่งการวัดค่าความชื้นในดินในกระถางกระบองเพชร 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	การใช้งานขาสัญญาณของโมดูลเซนเซอร์วัดความเข้มแสง GY-302 BH1750	14
2.2	การใช้งานขาสัญญาณของโมดูลเซนเซอร์วัดอุณหภูมิและความชื้น DHT22	15
2.3	การใช้งานขาสัญญาณของโมดูลเซนเซอร์วัดความชื้นในดิน	16
3.1	วัสดุที่ใช้ในการสร้างโรงเรือนกระบองเพชร	49
4.1	ความเข้มแสงในระยะความสูงและตำแหน่งที่ต่างกันภายในโรงเรือน	60
4.2	ค่าอุณหภูมิและความชื้นในอากาศเมื่อเปิดใช้งานพัดลมระบายอากาศภายในโรงเรือน	62
4.3	ความสัมพันธ์ระหว่างปริมาตรห้องกับปริมาณลมระบายอากาศที่เหมาะสมของพัดลม	63
4.4	ความสัมพันธ์ระหว่างขนาดของพัดลมและปริมาณลมระบายอากาศ	63
4.5	ความสัมพันธ์ระหว่างความเข้มแสงกับช่วงเวลา	64
4.6	ค่าความชื้นในดินทั้ง 5 ตำแหน่งเมื่อเปิดใช้งานระบบให้น้ำภายในโรงเรือน	65
4.7	ค่าความชื้นในดินทั้ง 5 ตำแหน่งหลังจากการให้น้ำผ่านไปเป็นระยะเวลา 5 วัน	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันสังคมส่วนมากได้มีการนำเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งมาใช้ในหลาย ๆ ด้านด้วยกัน อาทิ ด้านอุตสาหกรรม ด้านคมนาคม และด้านการเกษตร และในขณะนี้การเจริญเติบโตทางด้านการเกษตรในประเทศไทยซึ่งกำลังได้รับความนิยมเป็นอย่างมาก จึงมีการนำเทคโนโลยีมาปรับปรุงและพัฒนาเพื่อเพิ่มความสะดวกสบายให้กับชาวเกษตรกร โดยในปัจจุบันนี้มีพืชพันธุ์ต่าง ๆ ที่ได้รับความนิยมจากกระแสสังคมเป็นจำนวนไม่น้อย หนึ่งในนั้นคือพืชพันธุ์กระบองเพชรซึ่งมีรูปทรงสวยงาม สีสันสดใส แปลกตา และมีการพัฒนาสายพันธุ์หลากหลายสายพันธุ์ด้วยกัน จนทำให้ธุรกิจกระบองเพชรที่ผ่านมาได้รับความนิยมอย่างต่อเนื่องทั้งด้านธุรกิจการค้าขายในตลาดหรือด้วยความชื่นชอบส่วนตัวก็ตาม จึงทำให้ปัจจุบันพืชพันธุ์กระบองเพชรเป็นสายพันธุ์ยอดนิยมอันดับต้น ๆ ในธุรกิจการเกษตร

ระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชรได้รับการออกแบบ เพื่อศึกษาและพัฒนาการนำเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งมาประยุกต์ใช้ในงานด้านการเกษตร โดยระบบจะแบ่งออกเป็น 2 ส่วนหลัก ๆ คือ ส่วนแสดงผลค่าเซนเซอร์ต่าง ๆ ภายในโรงเรือน ซึ่งประกอบไปด้วยเซนเซอร์วัดความเข้มแสง อุณหภูมิ ความชื้นในอากาศ และเซนเซอร์วัดความชื้นในดิน ซึ่งจะทำงานร่วมกับไมโครคอนโทรลเลอร์และจัดเก็บไปยังฐานข้อมูลแบบตามเวลาจริง อีกส่วนหนึ่งคือการควบคุมการทำงานของอุปกรณ์ภายในโรงเรือนกระบองเพชรซึ่งได้แก่ พัดลมระบายอากาศ หลอดไฟปลูกพืช และปั๊มน้ำ โดยการทำงานในส่วนนี้จะมีการทำงานสองรูปแบบคือรูปแบบอัตโนมัติ และการทำงานรูปแบบควบคุมด้วยตัวผู้ใช้เอง ซึ่งการทำงานดังกล่าวจะสั่งงานผ่านแอปพลิเคชันบนโทรศัพท์มือถือ โดยจะมีการรับและส่งคำสั่งควบคุมต่าง ๆ ผ่านฐานข้อมูลเพื่อให้ผู้ใช้งานได้ทราบการเปลี่ยนแปลงค่าเซนเซอร์ต่าง ๆ ที่สัมพันธ์กับสภาพแวดล้อมภายในโรงเรือนแบบทันทีทันใด อีกทั้งช่วยเพิ่มความสะดวกสบายให้กับผู้ใช้งานยิ่งขึ้น

1.2 วัตถุประสงค์

1) เพื่อศึกษาสภาพแวดล้อมที่เหมาะสมสำหรับการเพาะเลี้ยงกระบองเพชรในระบบโรงเรือนแบบปิด

2) เพื่อเลือกใช้เซนเซอร์ในการติดตามสภาพแวดล้อมในโรงเรือนและควบคุมเพื่อให้กระบองเพชรสามารถเจริญเติบโตได้อย่างมีประสิทธิภาพ

3) เพื่อศึกษาการจัดเก็บเป็นฐานข้อมูลโดยใช้เครือข่ายอินเทอร์เน็ตพร้อมสามารถแสดงผลแบบตามเวลาจริง

4) เพื่อออกแบบและพัฒนาแอปพลิเคชันให้สามารถควบคุมระบบการทำงานผ่านโทรศัพท์มือถือได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

ออกแบบและคิดค้นอุปกรณ์เพื่อใช้สำหรับสังเกตการณ์ควบคุมโรงเรือนกระบองเพชรที่สามารถจับค่าความเข้มแสง อุณหภูมิ ความชื้นในอากาศ และความชื้นในดิน โดยการประมวลผลด้วย NodeMCU ESP32 ที่ทำงานร่วมกับเซนเซอร์วัดความเข้มแสง เซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ และเซนเซอร์วัดความชื้นในดิน ซึ่งอุปกรณ์ดังกล่าวจะทำงานร่วมกับพัดลมระบายอากาศ หลอดไฟปลูกพืช และปั๊มน้ำ โดยระบบจะส่งข้อมูลไปยังฐานข้อมูลแบบตามเวลาจริงเพื่อแสดงค่าจากเซนเซอร์ต่าง ๆ และสามารถควบคุมอุปกรณ์ในโรงเรือนโดยการสั่งงานผ่านแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์ “ระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร” ได้จัดทำขึ้นเพื่อสร้างระบบและคิดค้นอุปกรณ์ที่สามารถช่วยอำนวยความสะดวกแก่ผู้ใช้งานโรงเรือนกระบองเพชร โดยระบบสังเกตการณ์จะทำการเก็บค่าเซนเซอร์ต่าง ๆ ภายในโรงเรือนเพื่อส่งไปยังฐานข้อมูลและแสดงผลผ่านแอปพลิเคชันและระบบการควบคุมอุปกรณ์ภายในโรงเรือนจะรับและส่งคำสั่งควบคุมผ่านฐานข้อมูลแบบตามเวลาจริง โดยทำงานร่วมกับไมโครคอนโทรลเลอร์ในการควบคุมอุปกรณ์ต่าง ๆ ภายในโรงเรือน ซึ่งมีทฤษฎีและหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 การศึกษาลักษณะทั่วไปของกระบองเพชร

กระบองเพชรโดยปกติแล้วคนส่วนใหญ่มักคิดว่าต้นไม้ที่มีหนามมักเป็นกระบองเพชร ซึ่งในความเป็นจริงแล้วบางสกุล อาทิ สกุลแอสโตรไฟตัม บางชนิดก็ไม่มีหนามแต่ถูกจัดว่าเป็นกระบองเพชร ในขณะที่ไม้อวบน้ำ (Succulent) บางสกุล เช่น สกุลยูฟอร์เบีย แม้มีหนามแต่ไม่ถูกจัดว่าเป็นกระบองเพชร หลักพฤกษศาสตร์กล่าวว่า พืชที่จัดว่าเป็นกระบองเพชรหรือจัดอยู่ในวงศ์กระบองเพชร (Cactaceae) นั้นเป็นไม้ยืนต้นและจะต้องมีบริเวณพื้นที่ที่เรียกว่า ตุ่มหนาม บริเวณนี้จะเป็นที่ที่พบกลุ่มของหนามหรือขนแข็งขึ้นอยู่และเรียงไปตามแนวซี่หรือสันสูงของต้นอย่างเป็นระเบียบ อีกทั้งยังเป็นบริเวณที่เกิดตาดอกและแตกกิ่งใหม่ของต้นอีกด้วย ส่วนในไม้อวบน้ำประเภทที่มีหนามนั้นหนามจะขึ้นกระจัดกระจายไม่เป็นระเบียบรอบบริเวณลำต้น และไม่พบบริเวณตุ่มหนามเหมือนกระบองเพชร อีกทั้งพืชทั้งสองกลุ่มที่มีหนามนั้นอยู่กันคนละวงศ์ สิ่งสำคัญคือในกลุ่มของกระบองเพชรนั้นดอกจะมีกลีบเลี้ยงและกลีบดอกแยกกัน รังไข่จะอยู่ต่ำกว่าส่วนอื่น ๆ ส่วนกลุ่มยูฟอร์เบีย ดอกจะไม่มีทั้งกลีบเลี้ยง และกลีบดอก และรังไข่จะอยู่เหนือส่วนอื่น ซึ่งจะเห็นได้ว่ามีความแตกต่างกันอย่างสิ้นเชิง ซึ่งในการจัดทำปริญญานิพนธ์ผู้จัดทำได้เลือกพืชในกลุ่มของกระบองเพชรเพื่อนำมาศึกษาและทำการทดลองโดยมุ่งเน้นศึกษาปัจจัยหลัก ดังนี้

2.1.1 รูปร่างหน้าตาของกระบองเพชร

ลำต้นของกระบองเพชรมีลักษณะอวบน้ำ รูปทรงต่างกันไปหลายแบบตั้งแต่ทรงกลม ทรงกระบอก ไปจนถึงรูปร่างคล้ายกระบอง มีทั้งที่ขึ้นเป็นต้นเดี่ยว แตกกอเป็นกลุ่ม และที่ขึ้นรวมกันเป็นกลุ่ม มีผิวต้นเรียบเป็นมันคล้ายเคลือบด้วยขี้ผึ้ง เพื่อช่วยลดการสูญเสียน้ำของต้น ส่วนใหญ่จะมีสีเขียวเพื่อใช้สังเคราะห์แสงแทนใบสร้างอาหารเพื่อเลี้ยงต้น ต้นจะประกอบไปด้วยส่วนที่เรียกว่าตุ่มหนาม (Areole) ตุ่มหนามอาจเรียงต่อกันอยู่บนแนวซี่หรือสันสูงของต้นที่เรียกว่า สันต้น (Rib) หรือเรียงต่อกันอยู่บนเนินนูนที่เรียกว่า เนินหนาม (Tubercles) ของต้นก็ได้ หนามถือว่าเป็นจุดเด่นของกระบองเพชร เนื่องจากมีความหลากหลายทั้งในเรื่องของรูปร่างลักษณะ เป็นขนอ่อนนุ่มคล้ายขนสัตว์หรือแหลมแข็ง อาจจะปลายตรงหรือปลายงอเป็นตะขอ สีสันหลากหลายมากมายตั้งแต่สีขาว สีเหลือง สีส้ม สีแดง สีน้ำตาล ไปจนถึงสีดำ ในบางครั้งหนามยังเปลี่ยนสีไปตามอายุหรือตามสภาพและการปลูกเลี้ยงได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ดอกของกระบองเพชร

กระบองเพชรจัดเป็นต้นไม้พันธุ์ที่มีดอกสวยงามไม่แพ้ต้นไม้ชนิดอื่น อาทิ สกุลเอพิฟิลลัม หรือที่รู้จักในชื่อของออร์คิดแคคตัส นอกจากจะมีดอกที่สวยงามแล้วกระบองเพชรบางชนิดก็มีกลิ่นหอมอ่อนอีกด้วย ดอกของกระบองเพชรเป็นชนิดไม่มีก้านดอกรูปร่างลักษณะของดอกมีหลายแบบ อาทิ รูปกรวย (Funnel-shaped) รูประฆัง (Bell-shaped) รูปจาน (Dish-like) หรืออาจมีลักษณะเป็นหลอด (Tubular) มีสีสันสดใสต่างกันไปตั้งแต่สีขาว สีครีม สีเหลือง สีชมพู สีส้ม สีแดง บางชนิดอาจจะมีสองสีในดอกเดียวกัน ส่วนขนาดของดอกจะแตกต่างกันไปขึ้นอยู่กับชนิดและสายพันธุ์ของกระบองเพชร

2.1.3 สายพันธุ์ของกระบองเพชร

ปัจจุบันมีว่าพืชในกลุ่มกระบองเพชรประมาณ 50 ถึง 150 สกุลมากกว่า 2,000 ชนิด โดยผู้จัดทำเลือกศึกษา 6 สายพันธุ์ต้นกระบองเพชรที่นิยมในไทยเพื่อทำการศึกษาและมาทดลองเลี้ยง ได้แก่

1) สกุลยิมโนคาไลเซียม (Gymnocalycium)
2) สกุลแอสโตรไฟตัม (Astrophytum)
3) สกุลแมมมิลลาเรีย (Mammillaria)
4) สกุลโครีแฟนทา (Coryphantha)
5) สกุลอิชิโนฟอสซูโลแคคตัส (Echinofossulocactus)
6) สกุลโลโฟโฟร่า (Lophophora) [1]

2.2 การศึกษาสภาพแวดล้อมที่เหมาะสมสำหรับการเพาะเลี้ยงกระบองเพชร

2.2.1 ดินปลูกกระบองเพชร

ดินที่ใช้ในการปลูกกระบองเพชรต้องเป็นดินหรือวัสดุปลูกที่มีลักษณะ โปร่ง ร่วนซุย เพื่อให้รากเติบโตแผ่ขยายออกไปได้ ระบายน้ำได้ดี ไม่เปียกและอุ้มน้ำอยู่ตลอดเวลา ในขณะเดียวกันต้องสามารถเก็บความชื้นไว้ได้พอสมควร การผสมวัสดุสำหรับปลูกกระบองเพชรเป็นเรื่องที่น่าสนใจอย่างมากเพราะกระบองเพชรแต่ละชนิดต้องการวัสดุปลูกที่แตกต่างกันไป ต้องหมั่นศึกษาทดลองผสมวัสดุปลูกอยู่เสมอ เพื่อให้ได้สูตรที่เหมาะสมในการปลูก และต้นกระบองเพชรสามารถผลิหนามออกดอกสวยงาม ซึ่งในปัจจุบันวัสดุปลุกที่ใช้ได้ดีมีหลายสูตรขึ้นกับความเหมาะสมและแหล่งของวัสดุที่ใช้เป็นส่วนผสม

2.2.2 การเลือกภาชนะสำหรับการปลูกกระบองเพชร

กระถางสำหรับปลูกกระบองเพชรนั้นสามารถเลือกใช้ได้ตามความพอใจอาจจะเป็นกระถางดินเผา กระถางพลาสติก กระถางเซรามิก หรือภาชนะอื่น แต่มีหลักสำคัญ คือ กระถางต้องสามารถระบายน้ำและอากาศได้ดีพอสมควรรวมทั้งรักษาความโปร่งและร่วนซุยของวัสดุปลูกนั้นได้ ข้อสำคัญของการเลือกกระถางสำหรับปลูกกระบองเพชรคือ ไม่ควรเลือกกระถางที่มีขนาดใหญ่เกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะนอกจากจะต้องใช้วัสดุปลูกจำนวนมากและยังทำให้วัสดุนั้นอุ้มน้ำมากจนเกินไป ส่งผลให้ต้นกระบองเพชรมีการเจริญเติบโตช้าบางครั้งอาจมีผลทำให้รากเน่าได้

2.2.3 การให้น้ำ

กระบองเพชรเป็นต้นไม้ที่ไม่ต้องการน้ำในปริมาณมาก การรดน้ำที่มากเกินความต้องการมีผลทำให้รากเน่า เป็นเหตุให้ต้นตายในที่สุด ความจริงแล้วกระบองเพชรสามารถทนทานต่อสภาพอากาศที่แห้งแล้งและทนอยู่ในสภาพแวดล้อมที่ไม่เหมาะสมได้ดี ต้นจะเพียงแค่หยุดการเจริญเติบโตเท่านั้น เมื่อความอุดมสมบูรณ์กลับคืนมาต้นก็จะเริ่มเจริญเติบโตอีกครั้ง ตามธรรมชาติแล้วกระบองเพชรไม่ได้ต้องการน้ำตลอดเวลาต้นจะต้องการน้ำเฉพาะช่วงที่มีการเจริญเติบโตเท่านั้น ซึ่งส่วนใหญ่ก็จะตรงกับช่วงฤดูฝนที่อากาศมีความชื้นสูง ส่วนช่วงฤดูหนาวกระบองเพชรจะพักตัวและเป็นช่วงที่ต้องการน้ำปริมาณน้อยมาก ดังนั้นในฤดูฝนจึงควรรดน้ำสัปดาห์ละ 1 ถึง 2 ครั้ง เมื่อเข้าสู่ฤดูหนาวจะมีปริมาณความชื้นในอากาศน้อยลงกระบองเพชรจะเข้าสู่ภาวะพักตัวช่วงนี้ควรให้น้ำในปริมาณน้อยและเพิ่มปริมาณขึ้นในช่วงเข้าสู่ฤดูร้อน การให้น้ำมากไปในช่วงที่ต้นพักตัวอาจเป็นสาเหตุทำให้กระบองเพชรไม่ผลิดอก เพราะฉะนั้นเมื่อซื้อกระบองเพชรมาควรสอบถามผู้ขายถึงช่วงเวลาในการพักตัวและออกดอก เพื่อให้ทราบถึงช่วงเวลาการงดรดน้ำที่เหมาะสมและส่งผลให้ต้นกระบองเพชรผลิดอกตามฤดูกาล

วิธีการรดน้ำที่ถูกต้องคือ ต้องรดน้ำให้เปียกชุ่มจนน้ำไหลออกมาทางรูระบายที่ก้นกระถาง วิธีรดน้ำดังกล่าวมีประโยชน์ช่วยชะล้างเอาสารพิษต่าง ๆ ที่อาจเกิดสะสมให้เจือจางลงหรือหมดไป และจะรดน้ำอีกครั้งเมื่อวัสดุปลูกแห้งเท่านั้น แต่ทั้งนี้ความถี่ในการรดน้ำขึ้นอยู่กับปริมาณแสงที่ได้รับ รวมทั้งฤดูกาล ความโปร่งร่วนของวัสดุปลูก ขนาดและชนิดของภาชนะปลูก ถ้าเป็นกระถางดินเผาก็ต้องรดน้ำให้บ่อยกว่ากระถางพลาสติก ไม่ว่าจะเป็นชนิดหรือขนาดของกระถาง ชนิดของวัสดุปลูก รวมทั้งสถานที่ตั้งกระถางต่างก็เป็นองค์ประกอบสำคัญที่มีผลต่อการรดน้ำทั้งสิ้น ตัวอย่างเช่น กระถางต่างชนิดก็มีความสามารถในการระเหยน้ำต่างกัน ถ้ารดน้ำให้ต้นที่ปลูกในกระถาง 3 ชนิดต่างกันในปริมาณที่เท่ากัน เมื่อเวลาผ่านไปหนึ่งวันน้ำในกระถางดินเผาจะแห้งไปหมด ขณะที่ถ้าเป็นกระถางพลาสติกจะแห้งใน 2 วัน กระถางเซรามิกจะแห้งใน 3 วัน สรุปคือ

1) กระถางต่างชนิดกัน ขนาดเท่ากัน ระเหยน้ำด้วยความเร็วไม่เท่ากัน

2) กระถางชนิดเดียวกัน ขนาดแตกต่างกัน ระเหยน้ำด้วยความเร็วไม่เท่ากัน

3) กระถางชนิดเดียวกัน ขนาดเท่ากัน แต่ใช้วัสดุปลูกต่างกัน ระเหยน้ำด้วยความเร็วไม่เท่ากัน

4) กระถางชนิดเดียวกัน ขนาดเท่ากัน ใช้วัสดุปลูกเหมือนกัน แต่ตั้งไว้คนละที่กัน ระเหยน้ำด้วยความเร็วไม่เท่ากัน

5) กระถางชนิดเดียวกัน ขนาดเท่ากัน ใช้วัสดุปลูกเหมือนกัน ตั้งไว้ที่เดียวกันแต่สภาพอากาศแตกต่างกันในแต่ละวัน ระเหยน้ำด้วยความเร็วไม่เท่ากัน

จากสรุปข้างต้นสิ่งที่ควรทำในกรณีที่ซื้อกระบองเพชรมาจากหลายแหล่งคือ ใช้กระถางชนิดเดียวกัน ใช้วัสดุปลูกที่เหมือนกันหรือใกล้เคียงกัน ทั้งนี้ขึ้นอยู่กับชนิดและสายพันธุ์ของต้นกระบองเพชร และควรตั้งกระถางในที่มีสภาพแวดล้อมที่คล้ายกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 แสงและอุณหภูมิ

กระบองเพชรต้องการแสงที่มีความเข้มมากกว่า 1000 ลักซ์ แต่ไม่ควรได้รับในปริมาณมากเกินไปเพราะจะส่งผลให้ผิวของกระบองเพชรไหม้ได้ การแก้ปัญหากรณีต้นกระบองเพชรได้รับแสงไม่เพียงพอ อาทิ ย้าย เปลี่ยนหลังคาโรงเรือน หรือการติดไฟกรณีที่ในห้องได้รับแสงไม่เพียงพอ กระบองเพชรที่ได้รับแสงเพียงพอนั้นต้นจะเจริญเติบโตได้ดี สีสันของต้นสดใสสวยงามและออกดอกในช่วงระยะเวลาที่ตรงตามลักษณะพันธุ์ ต้นที่ได้รับแสงมากเกินไปจะมีลักษณะแห้งกร้านสีสันไม่สดใส อาจไหม้ตายได้ แต่ถ้าแสงน้อยเกินไปหนามก็จะหดสั้น ต้นยืดสูงขึ้นรากไม่งอกงามและต้นไม่เจริญเติบโต สำหรับอุณหภูมิที่เหมาะสมสำหรับการเจริญเติบโตของกระบองเพชรไม่ควรต่ำกว่า 25 องศาเซลเซียส และไม่ควรเกิน 37 องศาเซลเซียส ช่วงอุณหภูมิที่ดีที่สุดควรอยู่ระหว่าง 27 องศาเซลเซียส ถึง 35 องศาเซลเซียส

2.2.5 ความชื้น

กระบองเพชรเลี้ยงในระดับความชื้นสัมพัทธ์ที่ 25 เปอร์เซ็นต์ ถึง 45 เปอร์เซ็นต์ จะมีดอกสวย ผิวสวย ต้องการความชื้นในดินประมาณ 3 เปอร์เซ็นต์ ถึง 6 เปอร์เซ็นต์ ไม่ต้องการน้ำมาก ถ้าเปรียบเทียบระหว่างภาวะขาดน้ำกับภาวะที่แสงและอุณหภูมิไม่เหมาะสมแล้ว กระบองเพชรมีโอกาสตายได้สูงจากกรณีภาวะที่แสงและอุณหภูมิมากกว่า เพราะถ้ากระบองเพชรขาดน้ำต้นจะเพียงแค่เข้าสู่ภาวะพักตัวและจะเริ่มเจริญเติบโตอีกครั้งเมื่อได้รับน้ำและปุ๋ย

2.2.6 การให้ปุ๋ย

ปริมาณของธาตุอาหารของวัสดุปลูกที่ต้นกระบองเพชรได้รับจากวัสดุใส่ผสม อาทิ ปุ๋ยคอก หรือใบไม้ผุนั้นอาจไม่เพียงพอต่อความต้องการ ดังนั้นการใช้ปุ๋ยจึงเป็นตัวเลือกที่เหมาะสมในการแก้ปัญหาดังกล่าวโดยเลือกปุ๋ยสูตรที่มีปริมาณธาตุฟอสฟอรัส (P) และโพแทสเซียม (K) สูงกว่าไนโตรเจน (N) เนื่องจาก

1) ธาตุฟอสฟอรัสจะช่วยบำรุงให้ต้นออกดอกติดเมล็ดได้ดี

2) ธาตุโพแทสเซียมช่วยเพิ่มความต้านทานให้แก่ต้น และช่วยให้ต้นดูดซึมน้ำและอาหารได้ดีขึ้น

3) ธาตุไนโตรเจนนั้นช่วยเพิ่มอัตราการเจริญเติบโต

สิ่งที่ต้องระวัง คือ การที่กระบองเพชรได้รับธาตุไนโตรเจนมากเกินไปเพราะจะทำให้ต้นโตเร็ว เซลล์ขยายตัวมากแต่ผนังเซลล์จะบางอาจทำให้ต้นปริแตกหรือที่เรียกว่าต้นระเบิด อีกทั้งต้นจะอ่อนแอเป็นโรคได้ง่าย ปุ๋ยที่เหมาะสมสำหรับกระบองเพชร ได้แก่ ปุ๋ยออสโมโคทสูตรเร่งดอก โรยรอบกระถาง 3 เม็ด ถึง 5 เม็ด ทุก 3 เดือนหรือปุ๋ยเกล็ดละลายน้ำของกล้วยไม้สูตรเร่งดอก ผสมในน้ำปริมาณเจือจางกว่าที่ฉลากกำหนดครึ่งหนึ่ง รดแทนน้ำทุก 2 สัปดาห์

2.2.7 การดูแลกระบองเพชรให้เติบโต

2.2.7.1 วิธีการเปลี่ยนกระถาง

หลังจากปลูกกระบองเพชรไประยะหนึ่ง ต้นอาจจะไม่เติบโตงอกงามเหมือนเช่นที่ผ่านมา หรือเมื่อยกกระถางดูแล้วเห็นรากเริ่มโผล่พ้นกระถางออกมานั่นเป็นสัญญาณเตือนให้รู้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงเวลาที่ต้องทำการเปลี่ยนกระถาง การเปลี่ยนกระถางจะก่อให้เกิดผลดีต่อกระบองเพชร 3 ประการ คือ

1) ทำให้รากมีพื้นที่เพียงพอในการเจริญเติบโต เนื่องจากรากที่อัดแน่นเกินไปจะทำให้ความสามารถในการดูดสารอาหารน้อยลง เป็นสาเหตุให้ต้นหยุดการเจริญเติบโต

2) ทำให้ดินไม่แน่นจนเกินไป อากาศและน้ำไม่สามารถไหลเวียนได้อย่างทั่วถึง

3) ช่วยปรับสภาพดินที่ไม่สมดุลอันเกิดจากการให้ปุ๋ยและน้ำเป็นระยะเวลานาน

ควรเปลี่ยนกระถางในขณะที่วัสดุปลูกมีความชื้นพอสมควร คือไม่แห้งหรือไม่เปียกจนเกินไป ถ้าเป็นกระถางดินเผาให้ใช้นิ้วค่อย ๆ ดันรูระบายน้ำที่ก้นกระถางจนต้นขยับออกจากกระถาง แต่ถ้าเป็นกระถางพลาสติกให้บีบปากกระถางเบา ๆ จนดินภายในกระถางร่วงออกจากขอบกระถาง แล้วจึงค่อย ๆ เอียงกระถางเทต้นออกมา สังเกตรากว่ายังแข็งแรงดีอยู่หรือไม่ ถ้าจำเป็นก็ให้ตัดแต่งรากที่เน่าเสียออก หลังจากนั้นค่อย ๆ เขย่าต้นไม้อย่างเบามือ เพื่อให้ดินเก่าหลุดออกบ้างบางส่วน และรากที่เกาะกลุ่มกันอยู่คลายตัวแล้วจึงย้ายปลูกลงในกระถางใหม่ที่มีขนาดใหญ่กว่าเดิม

เมื่อปลูกกระบองเพชรไประยะหนึ่งอาจมีฝุ่นเกาะอยู่ตามผิวต้น ก็ควรดูแลทำความสะอาดเพื่อให้ต้นได้รับแสงเพื่อปรุงอาหารได้อย่างเต็มประสิทธิภาพ ใช้แปรงขนนุ่มปัดไล่ฝุ่นออกหรือใช้ที่เป่าผม (Dryer) โดยเลือกใช้ระดับลมที่เบาที่สุด เป่าให้ห่างจากต้นไม้ประมาณ 15 เซนติเมตร ค่อย ๆ เป่าไล่ฝุ่นออก การปล่อยให้ฝุ่นเกาะตามผิวต้นจำนวนมาก อาจทำให้ต้นหยุดการเจริญเติบโตได้ หากเห็นส่วนใดเน่าแห้งให้ตัดทิ้ง เพราะอาจเป็นแหล่งสะสมเชื้อโรคและแมลง สำหรับต้นที่ถูกละเลยไม่สนใจมาเป็นเวลานานจนทรุดโทรม ให้จัดการตัดแต่งส่วนที่เสียหายออกให้หมด โดยตัดให้เหนือจุดที่เสียหายประมาณ 2.5 เซนติเมตร ถึง 5 เซนติเมตร หลังจากนั้นจึงรดน้ำและให้ปุ๋ย

สำหรับกระบองเพชรที่แตกหน่อเติบโตรวมกันเป็นกลุ่ม การตัดแต่งรูปทรงเป็นสิ่งจำเป็นเพราะนอกจากจะทำให้ต้นสวยงามแล้ว ยังทำให้ต้นแตกหน่อใหม่ที่แข็งแรงกว่าเดิม หลังจากที่กระบองเพชรผลิดอกจนกระทั่งดอกเริ่มร่วง ควรเด็ดดอกที่เหี่ยวแห้งออกด้วยนิ้วหัวแม่มือและนิ้วชี้ ทั้งนี้เพื่อความสวยงามและเพื่อสุขภาพที่ดีของต้น และควรทำความสะอาดบริเวณโคนต้น ตรวจดูว่ามีแมลงรบกวนหรือไม่ถ้าพบก็กำจัดออก ถ้าต้องการปลูกกระบองเพชรไว้ในบ้านก็ควรตั้งไว้ในที่ที่มีแสงแดดส่องถึง สังเกตทิศทางของแสงถ้าแสงเปลี่ยนทิศทางควรย้ายตำแหน่งของต้นตามทิศทางของแสง ควรหมั่นสังเกตต้นว่ามีสีซีดลงหรือไม่ ลำต้นยืดยาวเกินไปหรือไม่ หนามหดสั้นหรือดูแคระแกร็นหรือไม่ ถ้าเกิดอาการเหล่านี้สามารถสรุปได้ว่าต้นได้รับแสงไม่เพียงพอกับความต้องการ ให้รีบตัดแต่งลำต้นส่วนที่ได้รับความเสียหายทิ้งและย้ายต้นไม้ไปยังบริเวณที่มีแสงเพียงพอ ต้นไม้ก็ค่อย ๆ แข็งแรงดังเดิม ส่วนกรณีที่ต้องการยกกระถางมาตั้งภายในห้องหรือในที่ร่มนั้น ควรปล่อยให้ดินปลูกแห้งเสียก่อนที่จะนำเข้ามาไว้ในห้อง และไม่ควรรดน้ำในขณะที่ต้นยังตั้งอยู่ในห้องเพราะในห้องนั้นไม่มีแสงแดดที่จะให้ต้นดูดน้ำเพื่อใช้ในการบวนการสังเคราะห์แสง หากวัสดุปลูกเปียกต้นอาจเน่าตายได้ จะรดน้ำก็ต่อเมื่อนำต้นออกไปรับแสงเท่านั้น ปัญหานี้ถือว่าเป็นปัญหาหลักของผู้ปลูกกระบองเพชรส่วนมาก คือปัญหาที่กระบองเพชรบางพันธุ์ค่อนข้างจะอ่อนไหวต่อแสง ต้นจะไม่ยอมผลิดอกจนกว่าจะได้รับแสงไม่น้อยกว่า 12 ชั่วโมงต่อวัน ดังนั้นเมื่อเลือกซื้อกระบองเพชรควรสอบถามข้อมูลจากผู้ขายถึงปริมาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

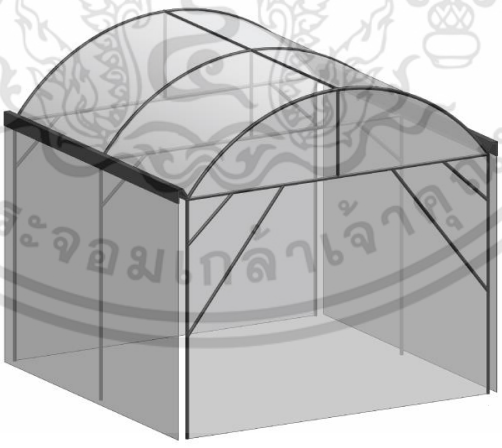
แสงที่เพียงพอต่อความต้องการ ช่วงเวลาออกดอกและช่วงเวลารดน้ำ เพื่อที่ผู้ปลูกนั้นสามารถกำหนดบริเวณที่จะตั้งกระถางและช่วงเวลาการรดน้ำได้ถูกต้องตรงตามความต้องการ

2.2.7.2 ปัญหาจากศัตรูพืช

แมลงที่เป็นศัตรูพืชตัวสำคัญของกระบองเพชร คือ เพลี้ย ซึ่งมีลำตัวอ่อนนุ่ม ปกคลุมด้วยผงสีขาวและไขมัน มักซ่อนอยู่ในบริเวณที่มองเห็นได้ยาก อาทิ รอบฐานตุ่มหนาม ซอกหนาม โคนต้น และราก เพลี้ยจะดูดกินน้ำเลี้ยงจากส่วนต่าง ๆ ของต้น ทำให้ต้นหงิกงอเหี่ยวแห้ง หยุดการเจริญเติบโตและอาจตายได้ในที่สุด สามารถป้องกันได้โดย ใช้แอลกอฮอล์เช็ดที่ผิวต้น หากเพลี้ยมีปริมาณมากให้ฉีดพ่นด้วยสารประเภทดูดซึมอย่างมาลาไธออนหรือไพรีทรอยด์ทุก 7 วัน ถึง 10 วัน [2]

2.3 การศึกษารูปแบบโรงเรือน

โรงเรือนอัจฉริยะ Smart Green House เป็นระบบปิด ดังแสดงในรูปที่ 2.1 สามารถควบคุมอุณหภูมิให้คงที่ตามที่ตั้งไว้ได้โดยสามารถเปิดปิดพัดลมระบายอากาศภายในโรงเรือน ป้องกันแมลงและสามารถควบคุมความชื้นให้เหมาะกับความต้องการของต้นกระบองเพชรได้ ข้อดีของระบบปิด คือการควบคุมอุณหภูมิให้อยู่ในระดับที่เหมาะสมและป้องกันแมลง การติดตั้งระบบ Smart Sensor เหมาะสำหรับทำการเพาะเลี้ยงในพื้นที่ที่มีอย่างจำกัด เช่น คอนโด หอพัก ครั้งนี้ได้เลือกทำการทดสอบในโรงเรือนอัจฉริยะ Smart Green House รูปทรงโรงเรือนทรงหลังคาโค้ง (High Tunnel Greenhouse) โครงสร้างโรงเรือนใช้เป็นเหล็กกัลป์วาไนซ์ขนาด 19.05 มิลลิเมตร ความหนา 1.50 มิลลิเมตร และมีอายุการใช้งาน 9 ปี ถึง 10 ปีขึ้นไป



รูปที่ 2.1 โรงเรือนอัจฉริยะ Smart Green House เป็นระบบปิด [3]

การใช้งานเหมาะสำหรับการปลูกพืชกินต้นและใบ ตระกูลผักนำหน้า อาทิ ผักสลัด ผักคะน้า ผักกาดขาว ผักกาดเขียว ผักบุ้ง ผักกวางตุ้ง เป็นต้น และยังเหมาะสำหรับพืชที่ต้องขึ้นค้างหรือพืชเถาเลื้อย อาทิ แตงโม เมล่อน แคนตาลูป สตรอเบอร์รี่ มะเขือเทศ เหมาะสำหรับการเพาะกล้าพืชทุกชนิด อาทิ ผัก ดอกไม้ ไม้ผล ปาล์ม และยางพารา เป็นต้น เหมาะสำหรับพืชที่ต้องการคุณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของผลผลิตสูง ผลผลิตสะอาด ปลอดภัย สีสันสดใหม่ ขนาดของผลสม่ำเสมอ และป้องกันแมลงศัตรูพืชที่ทำให้เกิดความเสียหายกับพืชที่เราต้องการ [3]

2.4 การศึกษาส่วนประกอบโรงเรือน

2.4.1 โครงสร้างของโรงเรือน

2.4.1.1 เหล็กกล่องกัลวาไนซ์

เหล็กกล่องนี้จะนิยมนำมาใช้เป็นองค์ประกอบของโครงสร้างทุกส่วนประกอบโดยเฉพาะโครงสร้างหลังคาที่เป็นสิ่งปลูกสร้าง อย่างเช่น โรงจอดรถ หรือโรงเรือน โดยเลือกเหล็กกล่องกัลวาไนซ์ คือ เหล็กชุบสังกะสี ซึ่งในส่วนนี้ไม่ต้องทาสีกันสนิม แต่เมื่อทำการเชื่อมเหล็กกล่องแล้วต้องทาสีกันสนิมที่รอยเชื่อมด้วยเพื่อป้องกันไม่ให้เกิดสนิมภายหลังโดยใช้เป็นเหล็กกัลวาไนซ์ขนาด 19.05 มิลลิเมตร ความหนา 1.50 มิลลิเมตร และมีอายุการใช้งาน 9 ปี ถึง 10 ปีขึ้นไป ดังแสดงในรูปที่ 2.2 [4]

รูปที่ 2.2 เหล็กกัลวาไนซ์ขนาด 19.05 มิลลิเมตร ความหนา 1.50 มิลลิเมตร [4]

2.4.1.2 เหล็กแบน

เหล็กแบน หรือ Flat Bar มีชื่อที่นิยมเรียกใช้กันหลายชื่อ อาทิ เหล็กพืด เหล็กเส้นแบน เหล็กแบนรีด ฯลฯ เหล็กแบนมีลักษณะเป็นเหล็กสี่เหลี่ยมผืนผ้า แบนยาว สามารถนำไปใช้งานได้หลากหลาย ทนแรงยืดพับได้ดี นิยมใช้กันหลากหลายในการก่อสร้าง เกิดจากการนำเหล็กแผ่นมาตัดเป็นเส้น ๆ และมีความหนาขนาดเท่ากันตลอดทั้งเส้น ส่วนความยาวมาตรฐานจะอยู่ที่ 6 เมตร เหล็กแบนมี 2 แบบคือ เหล็กแบนรีด และเหล็กแบนตัด มีเลือกใช้แบบรีด เนื่องจากมีลักษณะเป็นสี่เหลี่ยมผืนผ้ายาว นิยมใช้สำหรับงานเชื่อม ทำเหล็กดัด งานฝาท่อ ทำแหนบรถยนต์ ฯลฯ มีหลายขนาดให้เลือก สามารถทนแรงยืดพับได้ดี ดังแสดงในรูปที่ 2.3 [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 เหล็กแบน [5]

2.4.1.3 เหล็กเส้น

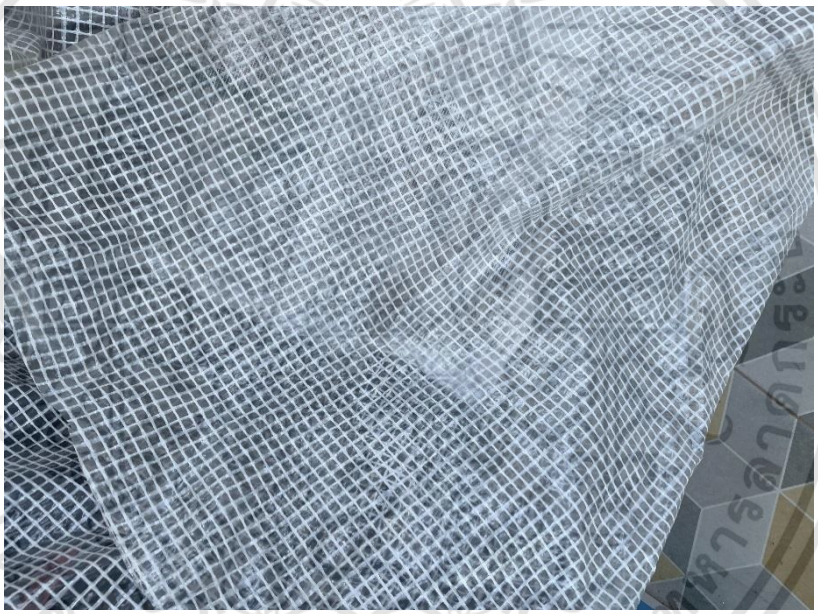
เหล็กเส้นกลม (Round Bars) หรือ RB คือเหล็กเส้นที่นำมาใช้ในงานก่อสร้างอาคาร เป็นเหล็กพื้นฐานช่วยเพิ่มความแข็งแรงและรับแรงโครงสร้างอาคาร โดยมักจะนำมาใช้ในงานคอนกรีตเสริมเหล็ก งานก่ออิฐทั่วไป และงานปลอกเสา คานต่าง ๆ เป็นต้น เหล็กเส้นกลมคุณภาพดีต้องผ่านมาตรฐานอุตสาหกรรม มอก. 20-2559 เกรด SR 24 ซึ่งมีเส้นผ่านศูนย์กลางของเหล็กเส้นเริ่มต้นจาก 6 มิลลิเมตร ถึง 34 มิลลิเมตร มีความยาวมาตรฐาน 10 เมตร และ 12 เมตรตามลำดับ เหล็กขนาด 6 มิลลิเมตร ถึง 9 มิลลิเมตร มักถูกใช้เป็นเหล็กปลอกเสา และคาน ส่วนเหล็กขนาด 12 มิลลิเมตร ขึ้นไปจะใช้ในงานเสริมแกนโครงสร้างอาคาร ดังแสดงในรูปที่ 2.4 [6]

รูปที่ 2.4 เหล็กเส้นกลม [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 วัสดุคลุมโรงเรือน

พื้นฐานพลาสติกโรงเรือน คือ พลาสติกที่เติมสารต้านการเสื่อมสลายจากแสงอัลตราไวโอเลต (UV Resistant) โดยมีการผสมสารดังกล่าวลงในเนื้อพลาสติก ซึ่งมีสัดส่วนสูงถึง 7 เปอร์เซ็นต์ ทำให้พลาสติกมีความทดทานต่อแสง สามารถใช้งานกลางแจ้งได้อย่างยาวนาน โดยวัสดุคลุมโรงเรือนด้านบนคลุมด้วยพลาสติก PE 200 ไมครอน ด้านข้างคลุมด้วยตาข่ายกันแมลงสีขาว ความถี่ 20 ตา อายุการใช้งาน 3 ถึง 5 ปีโดยประมาณ พลาสติกโรงเรือนใสอัลตราไวโอเลต 7 เปอร์เซ็นต์ ดังแสดงในรูปที่ 2.5 ถูกใช้ทำโรงเรือนเพื่อป้องกันแมลงศัตรูพืช โรงเรือนกระบองเพชร โรงเรือนปลูกผักปลอดสาร โรงเรือนปลูกพืชแบบไฮโดรโปนิกส์ เป็นต้น ทำให้ภูมิอากาศเหมาะกับการปลูกพืชนั้น ๆ ตลอดปี ทำให้พืชได้รับแสงอย่างต่อเนื่องระยะเวลาปลูกพืชสั้นลงปรับแสงที่เป็นอันตรายต่อพืช อาทิ อัลตราไวโอเลต และอินฟราเรด [7]



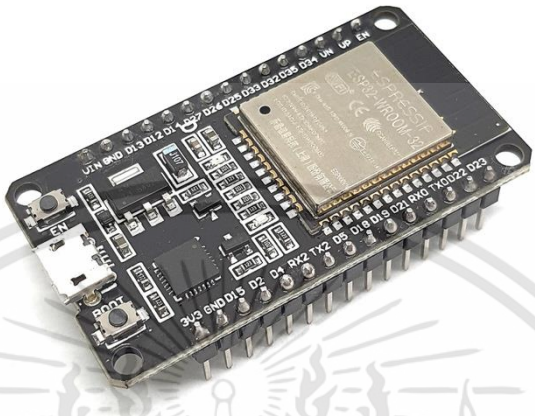
รูปที่ 2.5 พลาสติก PE 200 ไมครอน [7]

2.5 NodeMCU ESP32

โครงงานนี้เลือกใช้ NodeMCU ESP32 ดังแสดงในรูปที่ 2.6 ซึ่งเป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีลักษณะการทำงานตามคำสั่งภาษา C คล้ายกับ Arduino แต่จะมีฟังก์ชันการทำงานที่พิเศษกว่าโดยสามารถรองรับการเชื่อมต่อ Wi-Fi และ Blutooth โดย ESP32 นั้นเป็นชื่อของไอซีไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัท Epressif จากประเทศจีน และก่อนที่ ESP32 จะเริ่มผลิตขึ้น ได้มีไอซีไมโครคอนโทรลเลอร์ที่มี Wi-Fi ผลิตจากบริษัทเดียวกันอย่าง ESP8266 ทำยอดขายได้จำนวนมากเพราะเป็นที่นิยมใช้อย่างแพร่หลาย เมื่อความนิยมใช้มีจำนวนเพิ่มขึ้นจึงทำให้นักพัฒนาบอร์ด NodeMCU ได้พัฒนาต่อยอดความสามารถของบอร์ด ESP8266 พัฒนามาเป็น ESP32 โดยบริษัทได้ผลิตแบบโมดูลออกมาเพียงอย่างเดียวในชื่อ ESP-WROOM-32 มีการพัฒนาชุดแปลงคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Arduino ในชื่อ Arduino core ESP32 Wi-Fi chip ทำให้การพัฒนาเป็นไปด้วยความรวดเร็วมากขึ้น โดย NodeMCU ESP32 มีคุณสมบัติดังนี้ [8]



รูปที่ 2.6 NodeMCU ESP32 Dev Module [8]

1) หน่วยประมวลผลใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240 เมกะเฮิรตซ์

2) มีแรมในตัว 512 กิโลไบต์

3) รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16 เมกะไบต์

4) รองรับการเชื่อมต่อ Wi-Fi มาตรฐาน 802.11/b/g/n และรองรับการใช้งานในโหมด Station softAP และ Wi-Fi direct

5) รองรับการใช้งาน Bluetooth ในโหมด 2.0 และโหมด 4.0 BLE

6) ใช้แรงดันไฟฟ้าในการทำงาน 2.6 โวลต์ ถึง 3.3 โวลต์

7) สามารถทำงานได้ที่อุณหภูมิ -40 องศาเซลเซียส ถึง 125 องศาเซลเซียส

ในด้านประสิทธิภาพการทำงานของบอร์ดสามารถทำงานได้ดี โดยมีความเร็วในการรับและส่งข้อมูลที่ความเร็วสูงสุด 150 เมกะบิตต่อวินาที และเมื่อเชื่อมต่อผ่านโปรโตคอล UDP จะสามารถรับและส่งข้อมูลได้ที่ความเร็ว 135 เมกะบิตต่อวินาที ส่วนในโหมด Sleep จะใช้กระแสไฟฟ้าเพียง 2.5 ไมโครแอมแปร์ นอกจากนี้ ESP32 ยังมีเซนเซอร์ต่าง ๆ ในตัว ประกอบไปด้วย วงจรกรองสัญญาณรบกวนในวงจรขยายสัญญาณ เซนเซอร์แม่เหล็ก และเซนเซอร์ตรวจจับสัมผัส (Capacity touch) รองรับ 10 ช่อง โดยรองรับการเชื่อมต่อคลิสตอล 32.768 กิโลเฮิรตซ์ สำหรับใช้กับส่วนวรจรนับเวลาโดยเฉพาะ และในส่วนของการเชื่อมต่อขาใช้งานของ ESP32 รองรับการเชื่อมต่อบัสต่าง ๆ ซึ่งมีรายละเอียดดังนี้

1) มี GPIO 30 ขา

2) รองรับ UART จำนวน 3 ช่อง

3) รองรับ SPI จำนวน 3 ช่อง

4) รองรับ I^2C จำนวน 2 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) รองรับ ADC จำนวน 10 ช่อง
6) รองรับ DAC จำนวน 2 ช่อง
7) รองรับ I^2S จำนวน 2 ช่อง
8) รองรับ PWM/Timer ทุกช่อง
9) รองรับการเชื่อมต่อกับ SD-Card

การใช้งาน Wi-Fi ของ NodeMCU ESP 32 มีการทำงานทั้งหมด 3 โหมด ประกอบไปด้วย

1) โหมด Access Point AP

โหมดที่ใช้ ESP32 เป็นตัวปล่อยสัญญาณ Wi-Fi ให้กับอุปกรณ์ต่าง ๆ ทั้งนี้ ESP32 สามารถรองรับการเชื่อมต่ออุปกรณ์ได้เพียง 1 ตัวเท่านั้น เหมาะสมสำหรับงานที่ไม่ต้องการใช้งานบนอินเทอร์เน็ตหรือติดต่อกับอุปกรณ์อื่นในการเชื่อมต่อเครือข่ายระยะใกล้ และการนำไปควบคุมอุปกรณ์ต่าง ๆ ผ่าน Wi-Fi

2) โหมด Station STA

โหมดที่ใช้ ESP32 เชื่อมต่อกับอุปกรณ์ปล่อยสัญญาณอื่น ๆ อาทิ อุปกรณ์กระจายสัญญาณหรือโทรศัพท์มือถือที่สามารถเปิดบริการอินเทอร์เน็ตไร้สายสาธารณะได้ โหมดนี้นิยมใช้กับงานที่ต้องการเชื่อมต่ออินเทอร์เน็ตหรือมีการสื่อสารกับหลายอุปกรณ์ในเครือข่ายระยะใกล้ โดยงานด้าน IoT ด้าน Smart Home และนอกจากนี้ด้าน Smart Farm ก็มีการใช้งานในโหมดนี้เพื่อส่งข้อมูลจากเซนเซอร์ขึ้นไปบนระบบคลาวด์หรือระบบฐานข้อมูลแบบตามเวลาจริงและเชื่อมต่อกับระบบแอปพลิเคชันเพื่อรับคำสั่งควบคุมอุปกรณ์ผ่านอินเทอร์เน็ต

3) โหมด Access Point AP ร่วมกับ โหมด Station STA

โหมดที่ ESP32 ทำงานทั้ง 2 โหมดร่วมกัน ข้อดีของการใช้โหมดนี้คืออุปกรณ์ภายนอกสามารถเชื่อมต่อกับ ESP32 ได้ และ ESP32 สามารถเชื่อมต่อกับอุปกรณ์ภายนอกได้ นิยมใช้กับงานที่ต้องการใช้งานอินเทอร์เน็ตและมีการเคลื่อนย้าย หรือสามารถเปลี่ยนสถานที่ใช้งานได้ โดยในโหมด AP จะทำให้อุปกรณ์ต่าง ๆ สามารถเข้ามาตั้งค่าการเชื่อมต่อ Wi-Fi กับอุปกรณ์กระจายสัญญาณหรืออุปกรณ์อื่น ๆ โดยโหมด STA ก็จะนำค่าที่ตั้งค่าไว้มาใช้เชื่อมต่อกับอุปกรณ์กระจายสัญญาณหรืออุปกรณ์อื่น ๆ

2.6 โมดูลเซนเซอร์วัดความเข้มแสง GY-302 BH1750

GY-302 BH1750 ดังแสดงในรูปที่ 2.7 เป็นโมดูลเซ็นเซอร์ความเข้มแสงที่ผลิตโดย ROHM Semiconductor ใช้การเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยการสื่อสารแบบอนุกรมความเร็วต่ำ (I^2C) ทำให้สามารถนำไปใช้งานต่อได้ง่าย ให้ผลการวัดในหน่วย ลักซ์ สามารถวัดค่าความเข้มแสงได้ตั้งแต่ 1 ลักซ์ ถึง 65535 ลักซ์ ตัวเซ็นเซอร์ภายในประกอบด้วยอุปกรณ์สารกึ่งตัวนำที่สามารถเปลี่ยนสัญญาณแสงให้เป็นสัญญาณไฟฟ้า และใช้วงจร ADC ค่าความละเอียด 16 บิต สามารถใช้งานได้ที่แรงดันไฟฟ้า 2.4 โวลต์ ถึง 3.6 โวลต์ ใช้กระแสไฟฟ้า 200 ไมโครแอมแปร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่สำหรับโมดูล GY-302 ที่ใช้ในการทดสอบครั้งนี้จะมีวงจรควบคุมแรงดันไฟฟ้าแบบ LDO ที่สามารถควบคุมแรงดันในช่วง 3.3 โวลต์ ถึง 5 โวลต์ ให้อยู่ที่ 3.3 โวลต์ ได้ทำให้สามารถใช้งานกับอุปกรณ์ที่ใช้แรงดันไฟฟ้า 5 โวลต์ ได้ นอกจากนี้ตัวโมดูลยังมีตัวต้านทานภายในที่ต่อกับสัญญาณขาเข้าเพื่อรักษาระดับแรงดันไฟฟ้าให้อยู่ในสถานะ HIGH แล้วส่งไปยังขา SCL และ SDA ทำให้ไม่ต้องต่อสายเพิ่มเติม นอกจากนี้ยังต่อขาสำหรับเลือก Address ออกทำให้สามารถใช้งานโมดูลสองตัวได้พร้อมกัน โดยการเชื่อมต่อการใช้งานเซนเซอร์แสดงดังตารางที่ 1 [8]

รูปที่ 2.7 โมดูลวัดความเข้มแสง GY-302 BH1750 [8]

ตารางที่ 2.1 การใช้งานขาสัญญาณของโมดูลเซนเซอร์วัดความเข้มแสง GY-302 BH1750

ลำดับที่	ชื่อขาสัญญาณ	คำอธิบาย
1	VCC	แรงดันไฟฟ้าขนาด 3.3 โวลต์ ถึง 5 โวลต์
2	GND	กราวนด์
3	SCL	สัญญาณ SCL ของการเชื่อมต่อแบบอนุกรมความเร็วต่ำ
4	SDA	สัญญาณ SDA ของการเชื่อมต่อแบบอนุกรมความเร็วต่ำ
5	ADDR	เลือก Address ของบอร์ด

2.7 โมดูลเซนเซอร์วัดอุณภูมิและความชื้นในอากาศ DHT22

DHT22 ดังแสดงในรูปที่ 2.8 เป็นโมดูลเซนเซอร์สำหรับวัดอุณหภูมิและความชื้นที่มีความแม่นยำสูงในการวัด สามารถวัดได้ในช่วงอุณหภูมิ ตั้งแต่ −40 องศาเซลเซียส ถึง +80 องศาเซลเซียส ความคลาดเคลื่อนน้อยกว่า ± 0.5 องศาเซลเซียส และวัดความชื้นสัมพัทธ์ได้ในย่าน 0 เปอร์เซ็นต์ ถึง 100 เปอร์เซ็นต์ ความแม่นยำ ± 2 เปอร์เซ็นต์ ถึง 5 เปอร์เซ็นต์ มีความละเอียดในระดับทศนิยม 1 ตำแหน่ง ใช้งานได้นานและทนทาน เหมาะสำหรับงานที่ต้องการความแม่นยำสูง โมดูลมาพร้อมแผ่นวงจรพิมพ์ สายไฟ และตัวต้านทานภายในที่ต่อกับขาสัญญาณขาเข้าเพื่อรักษาระดับแรงดันไฟฟ้าให้อยู่ในสถานะ HIGH จึงสามารถต่อสายใช้งานได้ทันที และใช้ไฟเลี้ยงช่วงแรงดันไฟฟ้าขนาด 3.3 โวลต์ ถึง 6 โวลต์ โดยการเชื่อมต่อการใช้งานเซนเซอร์แสดงดังตารางที่ 4 [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

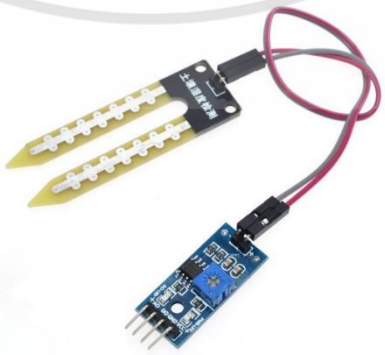
รูปที่ 2.8 โมดูลเซนเซอร์วัดอุณหภูมิและความชื้น DHT22 [9]

ตารางที่ 2.2 การใช้งานขาสัญญาณของโมดูลเซนเซอร์วัดอุณหภูมิและความชื้น DHT22

ลำดับที่	ชื่อขาสัญญาณ	คำอธิบาย
1	VCC (+)	แรงดันไฟฟ้าขนาด 3.3 โวลต์ ถึง 6 โวลต์
2	OUT	อ่านข้อมูลเซนเซอร์ ใช้กำหนด pinMode
3	GND (–)	กราวนด์

2.8 โมดูลเซนเซอร์วัดความชื้นในดิน

Soil Moisture Sensor ดังแสดงในรูปที่ 2.9 เป็นเซนเซอร์ที่นิยมใช้ในงานภาคการเกษตรอย่างมากในปัจจุบัน ด้วยคุณสมบัติที่สามารถวัดความชื้นในดินได้อย่างแม่นยำเหมาะสมกับการใช้งานเพื่อการทำระบบรดน้ำอัตโนมัติ ซึ่งหลักการทำงานของเซนเซอร์ชนิดนี้จะเริ่มจากการวัดค่าความต้านทานระหว่างอิเล็กโทรดที่ชุบโลหะอย่างดีมีลักษณะคล้ายส้อมเพื่อป้องกันการเกิดออกซิเดชั่น เพิ่มอายุการใช้งานและลดการสึกหรอเนื่องจากความชื้น โดยอิเล็กโทรดที่ปักลงในดินจะดึงค่าที่ได้ส่งต่อไปประมวลผลในโมดูลของเซนเซอร์ LM393 โดยการเชื่อมต่อการการใช้งานเซนเซอร์แสดงดังตารางที่ 3 [10]



รูปที่ 2.9 โมดูลเซนเซอร์วัดความชื้นในดิน [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 การใช้งานขาสัญญาณของโมดูลเซนเซอร์วัดความชื้นในดิน

ลำดับที่	ชื่อขาสัญญาณ	คำอธิบาย
1	VCC (+)	แรงดันไฟฟ้าขนาด 3.3 โวลต์ ถึง 5 โวลต์
2	INPUT	สัญญาณขาเข้า
3	GND (–)	กราวนด์

2.9 โมดูลรีเลย์ 3 โวลต์ (4 Channel)

รีเลย์ คืออุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เปรียบเสมือนสวิตช์ไฟที่ใช้แรงดันไฟฟ้าในการเปิดและปิดอุปกรณ์ไฟฟ้าเพื่อควบคุมวงจรต่าง ๆ โดยหลักการทำงานของรีเลย์จะทำงานโดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อเปลี่ยนแรงดันไฟฟ้าให้เป็นพลังงานแม่เหล็กสำหรับใช้ดึงดูดหน้าสัมผัสให้เปลี่ยนทิศทางการไหลของกระแสไฟฟ้า เพื่อควบคุมการจ่ายกระแสไฟฟ้าให้กับอุปกรณ์ต่าง ๆ คล้ายกับสวิตช์ โดยการทดสอบครั้งนี้ผู้จัดทำได้เลือกใช้ โมดูลรีเลย์ 3 โวลต์ (Active low) เพราะสามารถต่อกับบอร์ด ESP32 ได้โดยตรงใช้แรงดันไฟฟ้าขนาด 3 โวลต์ สามารถใช้งานได้ทั้งไฟฟ้ากระแสตรงและกระแสสลับ และมี LED แสดงสถานการณ์ทำงานของรีเลย์แสดงดังรูปที่ 2.10 [11]

รูปที่ 2.10 โมดูลรีเลย์ 3 โวลต์ (Active low) [11]

2.10 หลอดไฟปลูกพืช (LED Grow Light)

หลอดไฟปลูกพืช คือหลอดไฟสำหรับปลูกพืชซึ่งเป็นที่นิยมใช้กันในส่วนมากของโรงเรือนหรือแปลงปลูกพืช หลอดไฟชนิดนี้มีคุณสมบัติการให้แสงที่ช่วยในการเจริญเติบโตของพืชโดยภายในหลอดจะประกอบไปด้วยหลอด LED ขนาดเล็กจำนวนมาก โดยแบ่งออกเป็นหลอด LED สีแดงและ LED สีน้ำเงินซึ่งให้คุณสมบัติที่เฉพาะแตกต่างกันดังนี้ LED แสงสีแดงมีความยาวคลื่นแสง 630 นาโนเมตร ถึง 660 นาโนเมตร ซึ่งจะช่วยเพิ่มประสิทธิภาพในการผลิตดอกและทำให้ต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบองเพชรมีสีสันสวยงามและชัดเจน และ LED แสงสีน้ำเงินมีความยาวคลื่นแสง 430 นาโนเมตร ถึง 460 นาโนเมตร จะช่วยเพิ่มประสิทธิภาพในการสังเคราะห์แสงของพืช ช่วยให้ต้นไม้เจริญเติบโตไม่ยืด บำรุงลำต้นและใบ ซึ่งหลอดไฟปลูกพืชขนาด 50 วัตต์ ประกอบไปด้วยหลอด LED ขนาดเล็กจำนวน 80 หลอด ดังแสดงในรูปที่ 2.11 โดยหลอดไฟชนิดนี้สามารถใช้ให้ความสว่างกับพืชครอบคลุมพื้นที่ประมาณ 1 ตารางเมตร และในส่วนของตำแหน่งการติดตั้งควรติดตั้งให้ห่างจากพืชเป็นระยะประมาณ 15 เซนติเมตร ถึง 20 เซนติเมตร [12]

รูปที่ 2.11 LED Grow Light 50 วัตต์ [12]

2.11 พัดลมระบายอากาศ

พัดลมระบายอากาศเป็นพัดลมที่ใช้สำหรับติดในห้องหรือตู้ควบคุม เพื่อช่วยให้อากาศถ่ายเทได้สะดวก ลดการสะสมของความร้อน โดยพัดลมระบายอากาศมีให้เลือกใช้หลายขนาดขึ้นอยู่กับวัตถุประสงค์การใช้งาน ซึ่งโดยปกติจะมีใบพัดตั้งแต่ 3 ใบพัดขึ้นไป โครงสร้างประกอบไปด้วยพลาสติกและเหล็ก ส่วนของการทดสอบในโรงเรือนแบบปิดได้เลือกใช้พัดลมระบายอากาศขนาด 12 เซนติเมตร แรงดันไฟฟ้ากระแสสลับขนาด 220 โวลต์ ถึง 240 โวลต์ และใช้ความถี่ตั้งแต่ 50 เฮิรตซ์ ถึง 60 เฮิรตซ์ มีใบพัดพลาสติกทั้งหมด 5 ใบ โครงสร้างของตัวพัดลมเป็นโครงเหล็ก แสดงดังรูปที่ 2.12 [13]

รูปที่ 2.12 พัดลมระบายอากาศขนาด 12 เซนติเมตร [13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 ปั๊มแรงดันขนาด 4.8 บาร์

ปั๊มแรงดันขนาด 4.8 บาร์ เป็นปั๊มขนาดเล็กสามารถต่อใช้งานได้กับแหล่งจ่ายไฟฟ้ากระแสตรงขนาด 12 โวลต์ และใช้กระแสไฟฟ้า 2 แอมแปร์ อัตราการไหลของน้ำสูงสุดอยู่ที่ 210 ลิตรต่อชั่วโมง ขนาดท่อดูดน้ำเข้าและท่อส่งน้ำออกสามารถต่อใช้งานกับสายยางขนาด 0.9 เซนติเมตรได้ เหมาะสำหรับการนำไปใช้งานในโรงเรือนการเกษตรขนาดเล็ก ซึ่งขนาดแรงดันปั๊มน้ำชนิดนี้สามารถใช้ได้กับหัวพ่นหมอกหรือสปริงเกอร์ขนาดเล็กเพื่อนำไปใช้งานในการให้น้ำแบบละอองฝอยขนาดเล็ก แสดงดังรูปที่ 2.13 [14]

รูปที่ 2.13 ปั๊มน้ำ รุ่น Green-01 DC 12 โวลต์ [14]

2.13 อุปกรณ์จ่ายไฟฟ้ากระแสตรง

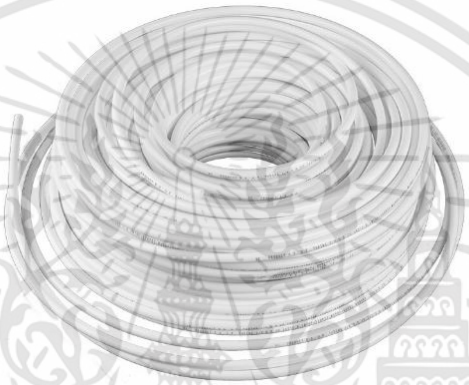
Switching Power Supply ดังแสดงในรูปที่ 2.14 เป็นอุปกรณ์แหล่งจ่ายไฟฟ้ากระแสตรงคงค่าแรงดันแบบหนึ่งและสามารถเปลี่ยนแรงดันไฟสลับค่าสูงเป็นแรงดันไฟตรงค่าต่ำได้ โดยในปริญญานิพนธ์นี้ ผู้จัดทำได้เลือกใช้อุปกรณ์จ่ายไฟฟ้ากระแสตรงที่ให้แรงดันขาออกขนาด 12 โวลต์ กระแสไฟฟ้า 5 แอมแปร์ โดยใช้เป็นแหล่งจ่ายไฟกระแสตรงให้กับปั๊มน้ำขนาด 12 โวลต์ ที่ใช้ภายในโรงเรือน [15]

รูปที่ 2.14 อุปกรณ์จ่ายไฟฟ้ากระแสตรง [15]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14 ท่อ PE ขนาด 0.9 เซนติเมตร

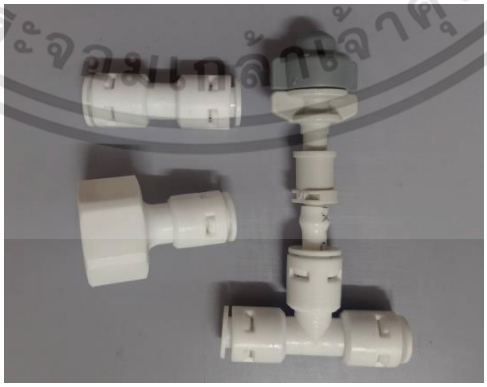
ท่อ PE ชนิดนี้เหมะสำหรับนำมาใช้ในระบบให้น้ำที่จะต้องมีการติดตั้งหัวพ่นหมอกหรือชุดสปริงเกอร์ขนาดเล็ก เนื่องจากท่อมีขนาดให้เลือกใช้หลากหลาย มีคุณสมบัติที่ทนต่อสภาพแวดล้อมได้ดีและตัวท่อมีความยืดหยุ่นเหมาะกับการเดินท่อในพื้นที่โรงเรือนและสามารถใช้กับชุดสปริงเกอร์ได้หลากหลายขนาด ตัวอย่างท่อ PE แสดงดังรูปที่ 2.15 [16]



รูปที่ 2.15 ท่อ PE ขนาด 0.9 เซนติเมตร [16]

2.15 ชุดสปริงเกอร์ขนาดเล็กและข้อต่อ

ผู้จัดทำได้เลือกใช้ชุดสปริงเกอร์ขนาดเล็กหรือหัวพ่นหมอกที่ให้ละอองน้ำแบบฝอยขนาดเล็ก โดยชุดสปริงเกอร์ขนาดเล็กจะใช้ร่วมกับข้อต่อและข้องอขนาด 0.9 เซนติเมตร ซึ่งสามารถต่อใช้งานได้โดยตรงกับท่อ PE และเหมาะกับการนำไปใช้งานร่วมกับปั๊มน้ำที่มีขนาดแรงดันตั้งแต่ 2 บาร์ขึ้นไป ตัวอย่างของชุดสปริงเกอร์และข้อต่อแสดงดังรูปที่ 4รูปที่ 2.16 ท่อ PE ขนาด 0.9 เซนติเมตร



รูปที่ 2.16 ชุดสปริงเกอร์ขนาดเล็กและข้อต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

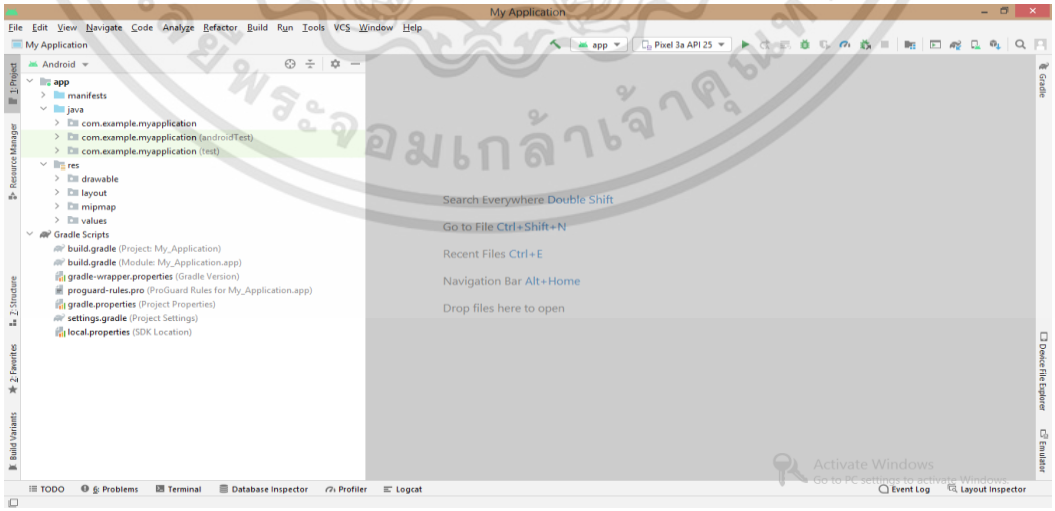
2.16 ระบบปฏิบัติการ Android

ระบบปฏิบัติการแบบเปิดเผยซอฟต์แวร์ต้นฉบับ (Open Source) โดยบริษัท Google (Google Inc.) ซึ่งเผยแพร่ภายใต้ลิขสิทธิ์ Apache ที่ได้รับความนิยมเป็นอย่างสูง เนื่องจากอุปกรณ์ที่ใช้ระบบปฏิบัติการ Android มีอย่างแพร่หลาย ซึ่งสามารถทำงานได้บนอุปกรณ์ที่มีขนาดหน้าจอและความละเอียดแตกต่างกันได้ โดยบริษัท Google อนุญาตให้ผู้ผลิตและนักพัฒนาสามารถปรับแต่งและวางจำหน่ายได้ภายใต้เงื่อนไขที่กำหนด

สำหรับนักพัฒนาโปรแกรม (Programmer) เพื่อใช้งานบนระบบปฏิบัติการ Android ในปัจจุบันไม่ใช่เรื่องยาก เนื่องจากมีข้อมูลในการพัฒนารวมทั้ง Android SDK (Software Development Kit) เตรียมไว้สำหรับนักพัฒนาได้เรียนรู้ และเมื่อนักพัฒนาต้องการจะเผยแพร่หรือจำหน่ายโปรแกรมก็สามารถเผยแพร่โปรแกรม ผ่าน Android Market ได้ ในส่วนของโครงสร้างภาษาที่ใช้ในการพัฒนาสำหรับ Android SDK จะใช้โครงสร้างของภาษา Java ในการเขียนโปรแกรม [17]

2.17 Android Studio

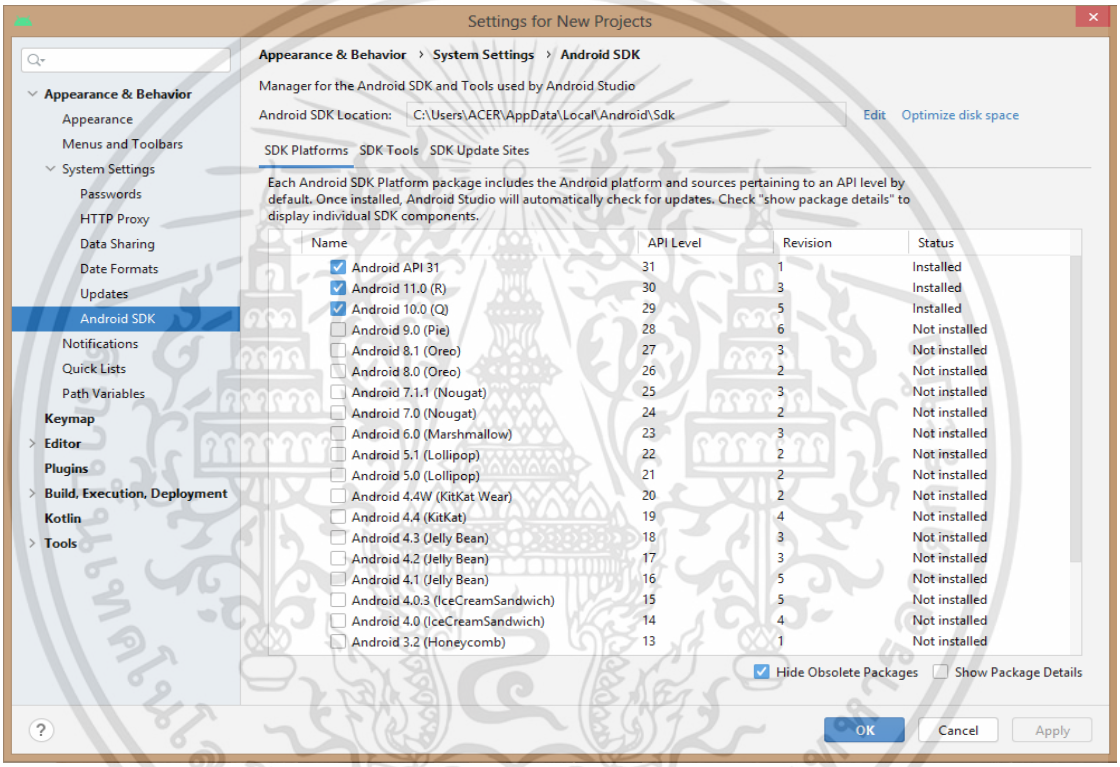
Android Studio เป็นเครื่องมือ IDE สำหรับพัฒนาระบบปฏิบัติการ Android จากบริษัท Google โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนาแอปพลิเคชันบน Android ให้มีประสิทธิภาพมากยิ่งขึ้น อีกทั้งยังสามารถแสดงตัวอย่างของแอปพลิเคชันในมุมมองที่แตกต่างบนโทรศัพท์หรืออุปกรณ์ในแต่ละรุ่นได้ และช่วยแก้ปัญหาเรื่องความเร็วในการแสดงผลของโปรแกรมจำลองการทำงานบนอุปกรณ์ต่าง ๆ (Emulator) โดยสามารถแสดงผลการทำงานได้ทันทีบนอุปกรณ์จริงโดยไม่ต้องทำการรันแอปพลิเคชันผ่านโปรแกรมจำลองการทำงานบนอุปกรณ์ โดยปัจจุบันการพัฒนา Native Mobile Application เป็นสิ่งที่น่าสนใจและง่ายสำหรับผู้เริ่มต้นและไม่ยุ่งยากเท่าระบบปฏิบัติการ iOS โดยหน้าต่างของโปรแกรมแสดงดังรูปที่ 2.17[18]



รูปที่ 2.17 หน้าต่างโปรแกรม Android Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Android Software Development Kit (Android SDK) เปรียบเสมือนไลบรารี่ (Library) ของ Android การพัฒนาแอปพลิเคชัน เนื่องจากตัว Android มีหลายเวอร์ชั่นและแต่ละเวอร์ชั่นมี ลักษณะเฉพาะและ Graphical User Interface (GUI) ที่ไม่เหมือนกันทำให้เกิด Android SDK ออกมาให้เลือกใช้งานหลากหลายเวอร์ชั่น ดังแสดงในรูปที่ 2.18 โดยค่า API Level หมายถึงเวอร์ชั่นของ API ที่ให้นักพัฒนานำมาใช้งานใน SDK เวอร์ชั่นนั้น ๆ โดยจะมีลักษณะเฉพาะตัวใหม่ที่เพิ่มขึ้นมาทำให้อุปกรณ์หรือโทรศัพท์มือถือที่ใช้งานในเวอร์ชั่นเก่ากว่าจะไม่สามารถใช้งาน API Level นี้ได้



รูปที่ 2.18 Android Software Development Kit

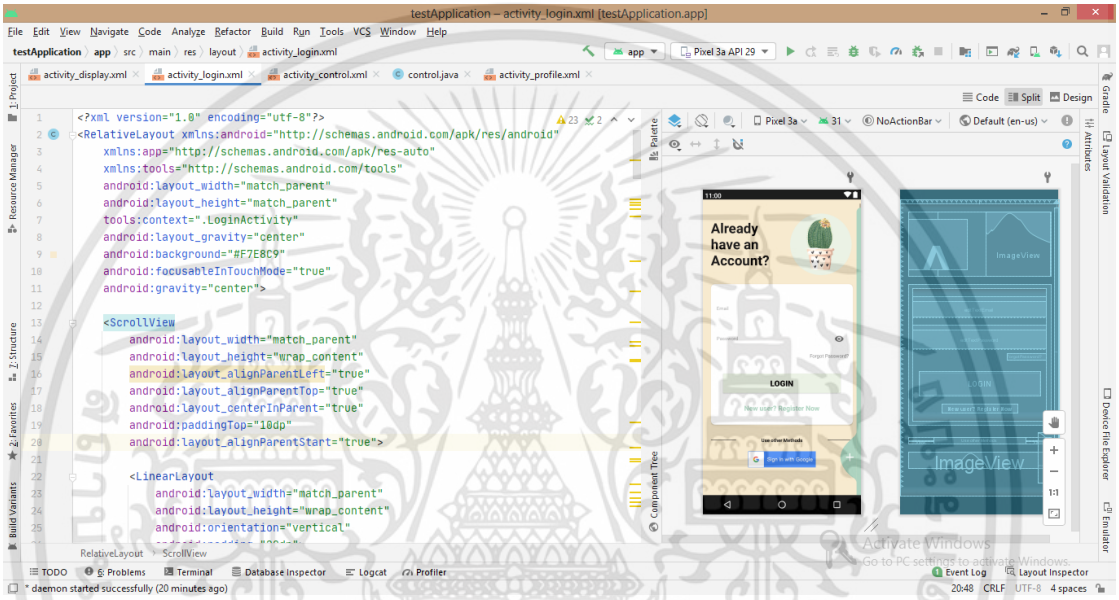
2.17.1 ส่วนต่อประสานผู้ใช้ (User Interface)

ส่วนต่อประสานผู้ใช้ (User Interface) คือส่วนที่ผู้ใช้สามารถมองเห็นและโต้ตอบได้ โดย Android Studio มีส่วนประกอบของส่วนต่อประสานผู้ใช้สำเร็จรูปมากมาย อาทิ รูปแบบที่มีโครงสร้างและการควบคุมส่วนต่อประสานผู้ใช้ที่ช่วยให้สามารถสร้างและออกแบบกราฟิกสำหรับส่วนต่อประสานผู้ใช้ภายในแอปพลิเคชันได้ อีกทั้ง Android Studio ยังมีโมดูลส่วนต่อประสานผู้ใช้อื่น ๆ สำหรับออกแบบส่วนต่อประสานพิเศษ เช่น กล่องโต้ตอบ การแจ้งเตือน และเมนู โดยจะใช้ภาษา XML ในการสร้างโครงสร้าง [19]

XML หรือ Extensible Markup Language เป็นภาษาที่ใช้สำหรับแสดงผลข้อมูล โดยแตกต่างจากภาษา HTML โดยภาษา HTML ถูกออกแบบมาสำหรับการแสดงผลเท่านั้น ส่วนภาษา XML ถูกออกแบบมาเพื่อเก็บโครงสร้างและข้อมูล ซึ่งภาษาที่ใช้จะกำหนดรูปแบบของคำสั่งภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HTML หรือเรียกว่า Meta Data ใช้สำหรับกำหนดรูปแบบของคำสั่ง Markup ต่าง ๆ โดยภาษา XML สามารถจัดการด้านระบบการติดต่อกับผู้ใช้จากโครงสร้างของข้อมูลเองได้และอาจจะต้องใช้ร่วมกับภาษาอื่น อาทิ HTML JSP PHP ASP หรือภาษาอื่น ๆ ที่สนับสนุน XML เพื่อให้สามารถแสดงผลได้อย่างถูกต้อง โดยจะมีนามสกุลเป็น .XML ซึ่งหน้าจอแอปพลิเคชันสามารถออกแบบร่วมกันได้ทั้งจาก XML และ Virtual แบบลากวางได้ทำให้สะดวกในการออกแบบและใช้งาน [20] หน้าต่างตัวอย่างการออกแบบแสดงดังรูปที่ 2.19



รูปที่ 2.19 หน้าต่างตัวอย่างการออกแบบหน้าจอเข้าสู่ระบบโดยใช้ภาษา XML ผ่าน Android Studio

2.17.2 ภาษา Java

Java หรือ Java programming language เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) โดยมีจุดประสงค์เพื่อใช้แทนภาษา C++ ที่มีความซับซ้อนมากกว่าและมีจำนวนคำสั่งมากกว่าทำให้เกิดข้อผิดพลาดได้ง่าย ซึ่งรูปแบบที่เพิ่มขึ้นมีความคล้ายคลึงกับภาษา Objective-C โดยจุดเด่นของภาษา Java นักพัฒนาสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนเองด้วยภาษา Java ได้ เหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน สามารถทำงานในระบบปฏิบัติการที่แตกต่างกันได้ อีกทั้งภาษา Java มีการตรวจสอบข้อผิดพลาดในขณะที่ compile time และ runtime ช่วยลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรมและทำให้ debug โปรแกรมได้ง่าย [21]

2.17.3 คำสั่งควบคุม Android ขั้นสูง

Android Debug Bridge หรือ ADB เป็นเครื่องมือที่ติดมาพร้อมกับตัว Android SDK (Software Development Kit) โดยใช้คำสั่งแบบ command line เพื่อควบคุมหรือออกคำสั่งให้กับอุปกรณ์ Android ซึ่งในกรณีที่ใช้อุปกรณ์จริงจะต้องทำการเชื่อมต่อผ่าน USB Driver เพื่อให้สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผลการทำงานของแอปพลิเคชันบนอุปกรณ์จริงได้ เช่นเดียวกับส่วนแสดงผลในโปรแกรมจำลองการทำงานบนอุปกรณ์

2.18 การเก็บข้อมูลแบบเรียลไทม์

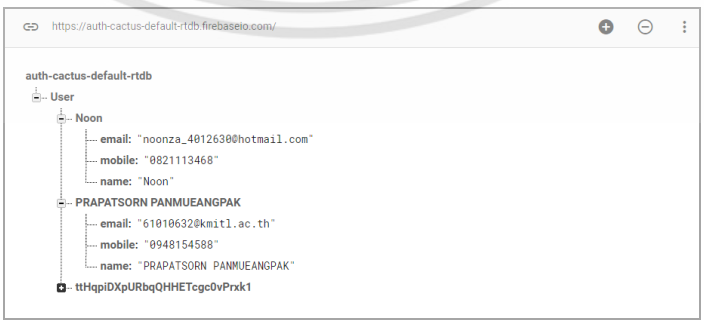
2.18.1 Firebase

Firebase คือแพลตฟอร์มที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วนของ Backend หรือ Server side ซึ่งทำให้สร้างแอปพลิเคชันสำหรับอุปกรณ์สื่อสารได้อย่างมีประสิทธิภาพ ช่วยลดระยะเวลาและค่าใช้จ่ายในการทำ Server side หรือการวิเคราะห์ข้อมูล โดยในแพลตฟอร์มมีทั้งเครื่องมือที่สามารถใช้งานได้ฟรีและเครื่องมือที่มีค่าใช้จ่าย โดย Firebase เป็นบริการหนึ่งของ Google เพื่อใช้งานรับและส่งข้อมูลในรูปแบบฐานข้อมูลตามเวลาจริงสำหรับ Application และ Web Application ซึ่งสามารถประยุกต์ใช้ร่วมกับงานด้าน IoT (Internet of Things) ได้ โดยโครงงานครั้งนี้มีการนำ Firebase มาเป็นตัวกลางในการเชื่อมต่ออุปกรณ์เข้าด้วยกัน เช่น NodeMCU ESP32 ทำการส่งข้อมูลเซนเซอร์ไปแสดงยังแอปพลิเคชันบนระบบปฏิบัติการ Android [22]

Firebase มีบริการหลายอย่างโดยหลัก ๆ จะแบ่งออกเป็น 3 หมวดหมู่ คือ 1) Build better apps 2) Improve app quality และ 3) Grow your business ซึ่งผลิตภัณฑ์หลัก ๆ ที่ใช้มีดังนี้ [23]

1. ฐานข้อมูลตามเวลาจริง (RealTime Database) คือบริการฐานข้อมูลแบบ NoSQL โดยไม่ใช้ภาษา SQL ในการจัดการข้อมูล แต่ใช้วิธีการเก็บข้อมูลเป็น JSON Tree ขนาดใหญ่ โดยที่มีตารางเหมือนกับ SQL แต่ไม่มีหลัก ในแถวสามารถเก็บข้อมูลได้ทั้งข้อความ (String) ตัวเลข (Number) และอื่น ๆ ได้ Realtime Database สามารถเชื่อมต่อสถานะข้ามระบบได้ตามเวลาจริง หากมีการเชื่อมต่อฐานข้อมูลเดียวกัน 2 ที่ ถ้าที่ใดที่หนึ่งมีการอัพเดตข้อมูลอีกที่หนึ่งก็จะทำการอัพเดตข้อมูลให้เหมือนกันโดยอัตโนมัติ อีกทั้งยังสามารถทำงานได้บนแอปพลิเคชันทั้งในระบบปฏิบัติการ Android และระบบปฏิบัติการ iOS

ตัวอย่างการเก็บข้อมูลแบบตามเวลาจริงที่ได้รับจากการลงทะเบียนบนแอปพลิเคชันที่ทำการเชื่อมต่อกับ Firebase โดยมีการกำหนดให้เก็บข้อมูล 3 ค่าได้แก่ อีเมล หมายเลขโทรศัพท์ และชื่อ ดังแสดงในรูปที่ 2.20

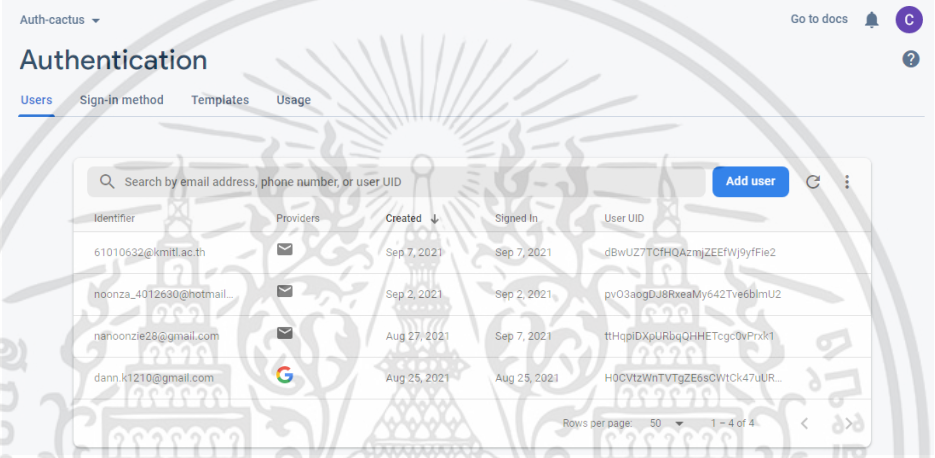


รูปที่ 2.20 ตัวอย่างการเก็บข้อมูลแบบตามเวลาจริงที่ได้จากการลงทะเบียนผ่านทางแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Authentication คือบริการตรวจสอบผู้ใช้ โดยสามารถตรวจสอบได้หลายวิธี เช่น Email-Password เบอร์โทรศัพท์ บัญชี Google Facebook Twitter Github เป็นต้น ซึ่งใช้สำหรับการเข้าสู่ระบบ โดยมีฐานข้อมูลภายในทำให้ไม่จำเป็นต้องออกแบบหรือสร้างฐานข้อมูลใหม่ ซึ่งสามารถแสดงวิธีการและช่วงเวลาการสมัคร รวมไปถึงช่วงเวลาล่าสุดในการเข้าสู่ระบบ

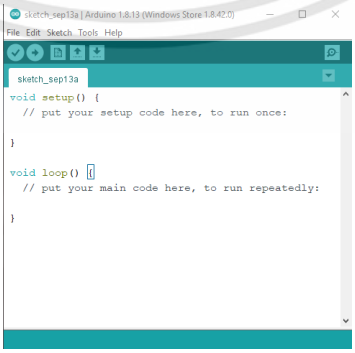
จากการออกแบบแอปพลิเคชันมีการกำหนดให้สามารถเข้าระบบและลงทะเบียนได้ 2 ช่องทางคือการสมัครโดยใช้ Email-Password และอีกวิธีคือการเข้าสู่ระบบโดยการใช้บัญชี Google ตัวอย่างการตรวจสอบผู้ใช้ที่ปรากฏในแพลตฟอร์ม Firebase จะแสดงดังรูปที่ 2.21



รูปที่ 2.21 ตัวอย่างบริการตรวจสอบผู้ใช้ที่ได้จากการลงทะเบียนผ่านทางแอปพลิเคชัน

2.19 Arduino IDE

โปรแกรม Arduino IDE คือโปรแกรมสำหรับใช้เขียนโปรแกรม compile และอัปโหลดโปรแกรมลงบอร์ด Arduino หรือบอร์ดตัวอื่น ๆ ที่คล้ายกัน อาทิ Generic ESP8266 modules NodeMCU หรือ WeMos D1 เป็นต้น โดยใช้ภาษา C หรือ C++ มาประยุกต์ในการเขียนคำสั่ง ซึ่งโปรแกรมมีลักษณะ ดังรูปที่ 2.22 [24]



รูปที่ 2.22 หน้าต่างโปรแกรม Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

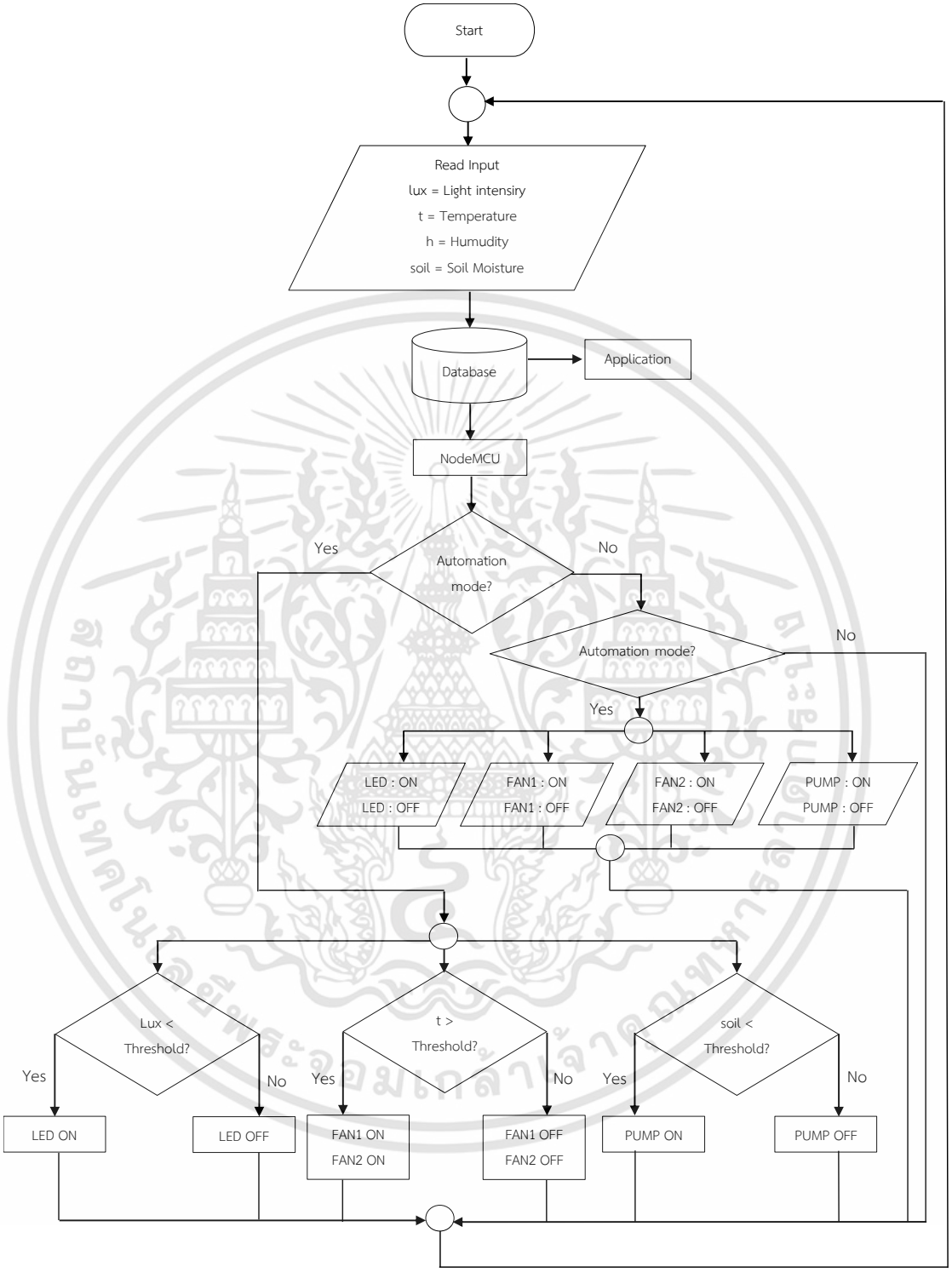
บทที่ 3
การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของระบบ

ระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชรมีลำดับขั้นตอนการทำงานของระบบแสดงดังแผนผังในรูปที่ 3.1 ซึ่งการทำงานควบคุมอุปกรณ์ผ่านแอปพลิเคชันจะทำงานร่วมกับเซนเซอร์วัดความเข้มแสง (GY-302 BH1750) เซนเซอร์วัดอุณหภูมิ (DHT22) และเซนเซอร์วัดความชื้นในดิน ซึ่งจะทำงานร่วมกับ NodeMCU ESP32 โดยขั้นตอนการใช้งานควบคุมอุปกรณ์ผ่านแอปพลิเคชันประกอบไปด้วยโหมดการทำงาน 2 โหมด ได้แก่ โหมดการทำงานแบบอัตโนมัติ ในโหมดการทำงานนี้ผู้ใช้จะสามารถกำหนดค่าคงที่ซึ่งจะประกอบไปด้วยค่าความเข้มแสง ค่าอุณหภูมิ และค่าความชื้นในดิน ในโครงงานนี้ผู้จัดทำได้เลือกค่าความเข้มแสงขั้นต่ำที่ใช้เป็นเกณฑ์ในการเปิดปิดการทำงานของหลอดไฟปลูกพืชแบบอัตโนมัติโดยอ้างอิงจากความต้องการแสงของพืชพันธุ์กระบองเพชร เมื่อค่าความเข้มแสงมีค่าน้อยกว่า 2,000 ลักซ์ หลอดไฟจะเปิดใช้งานโดยอัตโนมัติ เช่นเดียวกับการทำงานของพัดลมจะสัมพันธ์กับค่าของอุณหภูมิจากเซนเซอร์ที่อ่านได้ในขณะนั้น โดยค่าอุณหภูมิที่เหมาะสมต่อการเจริญเติบโตของกระบองเพชรไม่ควรจะเกิน 38 องศาเซลเซียส ดังนั้นผู้จัดทำจึงเลือกกำหนดค่าในโหมดการใช้งานแบบอัตโนมัติของพัดลมให้ทำงานเมื่ออุณหภูมิในโรงเรือนมีค่ามากกว่า 38 องศาเซลเซียส และในส่วนของการให้น้ำระบบจะทำงานโดยอัตโนมัติเมื่อค่าความชื้นในดินมีค่าน้อยกว่า 70 เปอร์เซ็นต์ เนื่องจากกระบองเพชรเป็นพืชที่ต้องให้น้ำโดยประมาณสัปดาห์ละ 2 ครั้ง และไม่ต้องการน้ำในปริมาณมากเกินไปเพราะจะทำให้รากเน่าและหยุดการเจริญเติบโตได้ ดั้งนั้นในการให้น้ำ 2 ครั้งต่อสัปดาห์ควรเว้นระยะห่างประมาณ 3 ถึง 4 วัน หลังจากที่ให้น้ำในปริมาณเพียงพอต่อการเจริญเติบโตแล้ว โหมดการทำงานที่ 2 ได้แก่ โหมดการควบคุมอุปกรณ์โดยผู้ใช้ ซึ่งการใช้งานในโหมดนี้ผู้ใช้สามารถเป็นผู้กำหนดการใช้งานอุปกรณ์ได้ด้วยตนเองได้แก่ หลอดไฟ พัดลม 2 ตัว และระบบให้น้ำละอองฝอยขนาดเล็กจากชุดสปริงเกอร์ขนาดเล็ก ทั้งนี้โหมดดังกล่าวถูกออกแบบมาเพื่อรองรับความต้องการของผู้ใช้แต่ละคนซึ่งอาจจะพิจารณาจากปัจจัยสภาพอากาศ หรือ ความต้องการของต้นกระบองเพชร เป็นต้น โดยเมื่อเปิดใช้งานอุปกรณ์ทั้งหมด NodeMCU ESP32 จะเชื่อมต่อกับสัญญาณ Wi-Fi โดยอัตโนมัติจากนั้นจะเริ่มอ่านค่าเซนเซอร์ได้แก่ เซนเซอร์วัดความเข้มแสงและเซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ และค่าความชื้นในดิน ค่าของเซนเซอร์ที่อ่านได้จะจัดเก็บอยู่ในฐานข้อมูล Firebase ซึ่งจะแสดงผ่านหน้าจอแสดงผลของแอปพลิเคชัน และในการใช้งานเริ่มต้นผู้ใช้สามารถเลือกที่จะใช้งานในโหมดใดก็ได้ ซึ่งถ้าเลือกใช้งานในโหมดอัตโนมัติก็จะสามารถปรับปรุงเงื่อนไขของความเข้มแสง อุณหภูมิ และค่าความชื้นในดิน เพื่อใช้ในการควบคุมหลอดไฟปลูกพืช พัดลมระบายอากาศทั้ง 2 ตัวได้ รวมถึงปั๊มน้ำ หรือถ้าต้องการใช้งานในโหมดที่ผู้ใช้สามารถควบคุมอุปกรณ์ด้วยตนเองโดยที่ไม่ต้องพิจารณาเงื่อนไขที่ตั้งไว้ ก็สามารถเปิดการใช้งานโหมดนี้ผ่านแอปพลิเคชันได้เช่นกัน

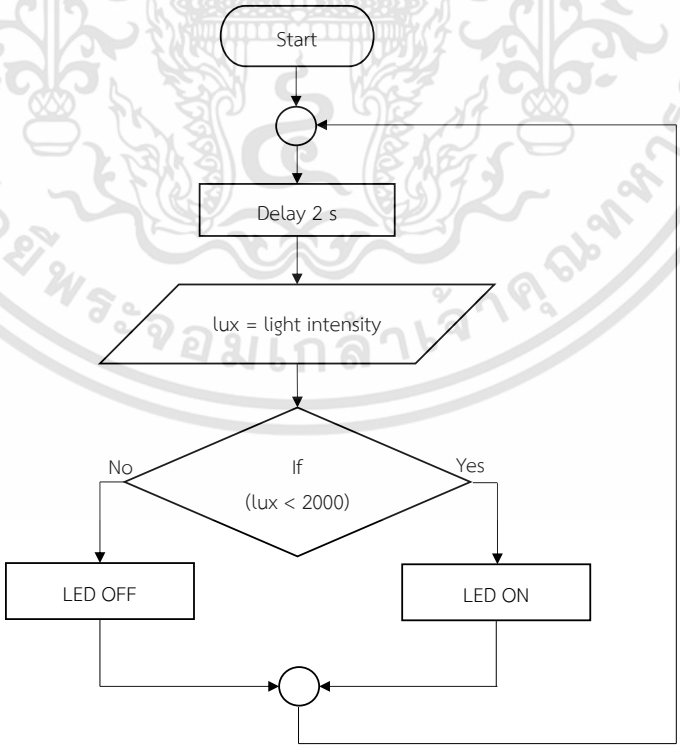
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แผนผังการทำงานของระบบ

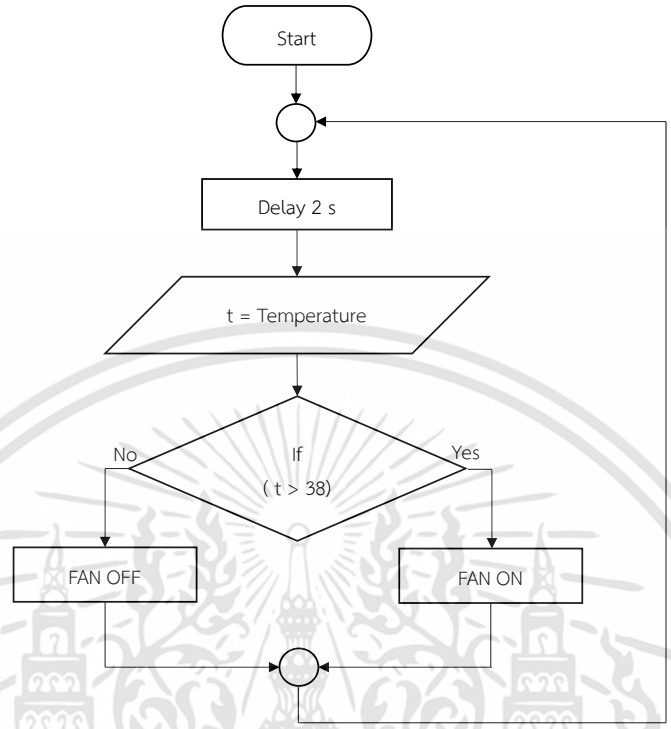
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบชุดคำสั่งในโหมดการทำงานควบคุมอุปกรณ์แบบอัติโนมัติจะมีการทำงานร่วมกันระหว่างเซนเซอร์วัดความเข้มแสง (GY-302 BH1750) เซนเซอร์วัดค่าอุณหภูมิ (DHT22) และเซนเซอร์วัดค่าความชื้นในดินโดยทำงานร่วมกับ NodeMCU ESP32 ซึ่งอุปกรณ์ที่ใช้ได้แก่ หลอดไฟปลูกพืช 1 หลอด พัดลมระบายอากาศ 2 ตัว และปั๊มน้ำ เงื่อนไขการควบคุมหลอดไฟปลูกพืชแบบอัตโนมัติ แสดงดังรูปที่ 3.2 โดยการทำงานจะเริ่มจากการอ่านค่าเซนเซอร์วัดความเข้มแสงทุก 2 วินาที จากนั้นโปรแกรมจะตรวจสอบตามเงื่อนไขที่กำหนดไว้ หากค่าความเข้มแสงที่วัดได้มีค่าน้อยกว่า 2,000 ลักซ์ จะควบคุมให้หลอดไฟเปิดโดยอัตโนมัติ ซึ่งจากการศึกษาค่าความเข้มแสงที่เหมาะสมต่อการเจริญเติบโตของต้นกระบองเพชรที่ดีควรจะมีค่าความเข้มแสงตั้งแต่ 2,000 ลักซ์ เป็นต้นไป และในส่วนของเงื่อนไขการควบคุมพัดลมระบายอากาศแบบอัตโนมัติ แสดงดังรูปที่ 3.3 โดยการทำงานจะเริ่มจากการอ่านเซนเซอร์ทำการวัดค่าอุณหภูมิทุก 2 วินาที จากนั้นโปรแกรมจะตรวจสอบตามเงื่อนไขที่กำหนดไว้ หากค่าอุณหภูมิมีค่ามากกว่า 38 องศาเซลเซียส จะควบคุมให้พัดลมทั้ง 2 ตัวทำงานพร้อมกัน ซึ่งจากผลการศึกษาค่าอุณหภูมิที่เหมาะสมต่อการเจริญเติบโตของต้นกระบองเพชร ค่าอุณหภูมิไม่ควรเกิน 38 องศาเซลเซียส เพราะอาจจะส่งผลต่อการการเจริญเติบโตของต้นกระบองเพชร ส่วนของเงื่อนไขการควบคุมปั๊มน้ำแบบอัตโนมัติ แสดงดังรูปที่ 3.4 โดยการทำงานเริ่มจากการอ่านค่าเซนเซอร์วัดความชื้นในดินทุก 2 วินาที จากนั้นโปรแกรมจะตรวจสอบตามเงื่อนไขที่กำหนดไว้ หากค่าความชื้นในดินมีค่าน้อยกว่า 70 เปอร์เซ็นต์ จะควบคุมให้ปั๊มน้ำและระบบให้น้ำแบบละอองฝอยภายในโรงเรือนทำงาน

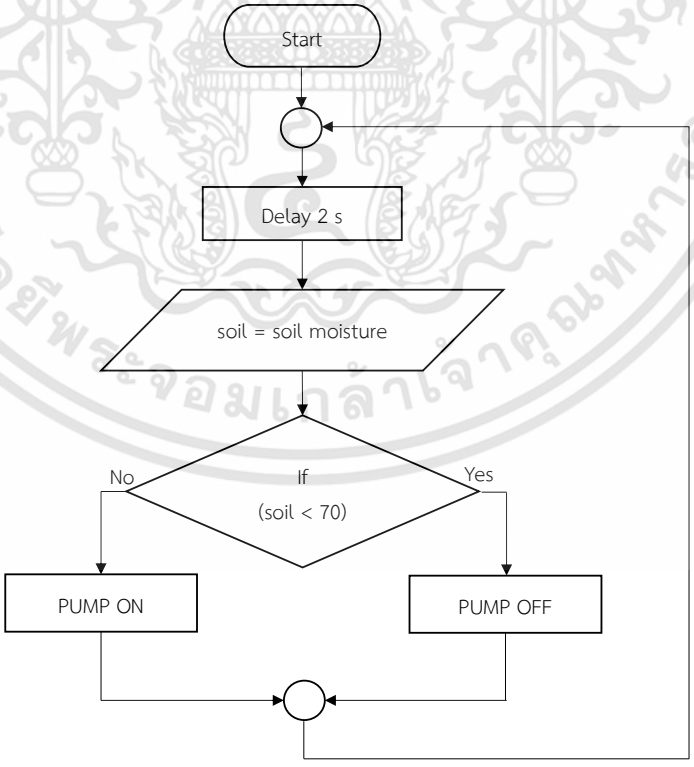


รูปที่ 3.2 แผนผังควบคุมการเปิดและปิดของสวิตช์หลอดไฟแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผนผังควบคุมการเปิดและปิดของสวิตช์พัดลมแบบอัตโนมัติ

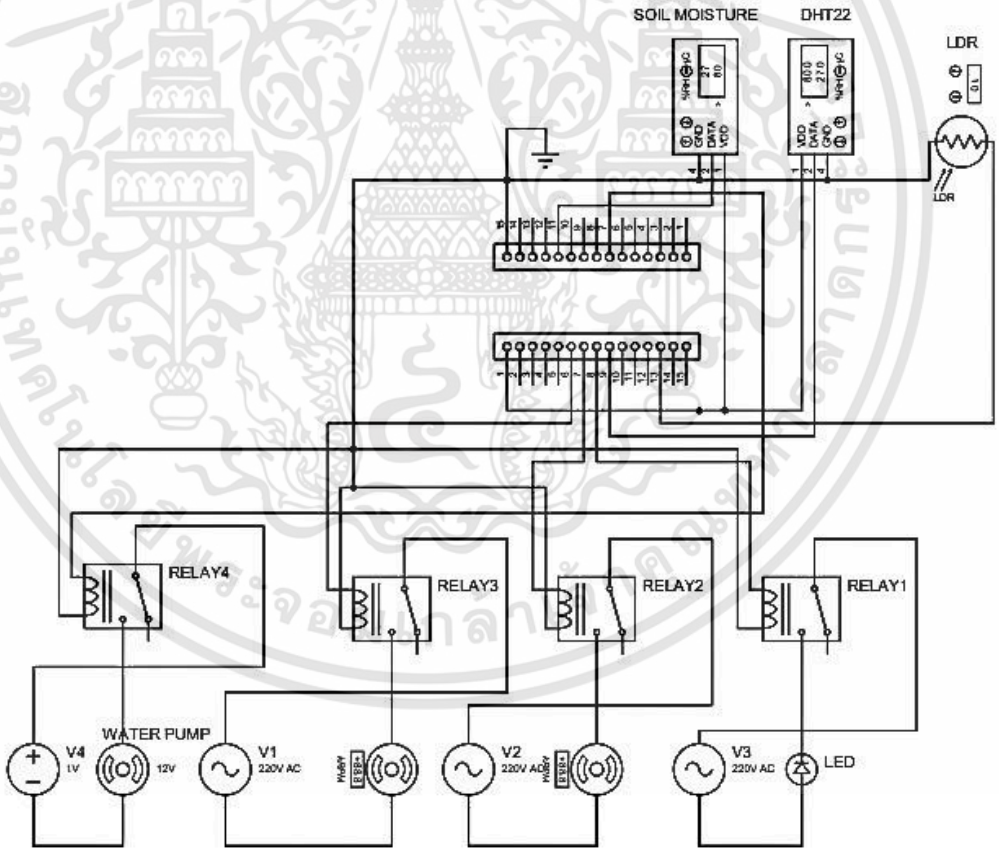


รูปที่ 3.4 แผนผังควบคุมการเปิดและปิดของสวิตช์ปั๊มน้ำแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การออกแบบวงจร

การออกแบบวงจรและเขียนชุดคำสั่งเพื่อควบคุมการทำงานของอุปกรณ์ตามแผนผังในรูปที่ 3.2 รูปที่ 3.3 และรูปที่ 3.4 แล้วนั้น โดยในปริญญานิพนธ์นี้ได้ทำการออกแบบชุดคำสั่งร่วมกับรีเลย์ทั้งหมด 4 channel กำหนดใช้ขา GPIO ที่ 5 18 19 และ 13 เป็นสัญญาณขาออกต่อเข้ากับรีเลย์ทั้ง 4 channel ตามลำดับ โดยกำหนดสถานะเริ่มต้นการทำงานของรีเลย์เป็นสถานะ HIGH เนื่องจากรีเลย์ชนิดนี้จะทำงานเมื่อสถานะเป็น LOW (Active low) และเขียนคำสั่งให้โปรแกรมทำงานแบบวนซ้ำ (Loop) โดยเมื่อค่าความเข้มแสง ค่าอุณหภูมิ และค่าความชื้นในดินเป็นไปตามเงื่อนไขที่กำหนด ซึ่งสามารถปรับค่าเกณฑ์ได้จากแอปพลิเคชันในโหมดการทำงานแบบอัตโนมัติ ส่วนในโหมดการทำงานที่ผู้ใช้กำหนดด้วยตนเองสามารถทำได้ผ่านแอปพลิเคชัน เนื่องจากระบบจะมีการรับและส่งคำสั่งผ่านฐานข้อมูลมายัง NodeMCU ESP32 เพื่อทำการควบคุมการเปิดและปิดอุปกรณ์ภายในโรงเรือน ซึ่งอุปกรณ์ภายในโรงเรือนจะใช้แรงดันไฟฟ้ากระแสสลับ 220 โวลต์ และแรงดันไฟฟ้ากระแสตรง 12 โวลต์ การออกแบบวงจรแสดงดังรูปที่ 3.5



รูปที่ 3.5 ภาพวงจรโดยรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การออกแบบชุดคำสั่งการรับส่งข้อมูลไปยังฐานข้อมูล Firebase Realtime

ออกแบบชุดคำสั่งการรับส่งข้อมูลและการเชื่อมต่อกับฐานข้อมูล Firebase Realtime โดยสร้างโครงงานทดลองที่มีชื่อว่า Auth-cactus เพื่อทำการเชื่อมต่อกับแอปพลิเคชันบน Android โดยได้สร้างที่อยู่ของไฟล์ไว้สำหรับเชื่อมต่อข้อมูลระหว่างอุปกรณ์และแอปพลิเคชันและที่อยู่ของไฟล์สำหรับการส่งค่าจากเซนเซอร์ต่าง ๆ ไปแสดงบนแอปพลิเคชัน และในหน้าแสดงผลการทำงานของระบบ และ หน้าการควบคุมอุปกรณ์ ซึ่งจะมีชื่อที่อยู่ของไฟล์แทนคำสั่งทั้งหมดดังต่อไปนี้ แสดงดังรูปที่ 3.6

1. Auto light หมายถึง การกำหนดเงื่อนไขค่าความเข้มแสงต่ำสุดที่จะสั่งให้เปิดหลอดไฟแบบอัตโนมัติโดยได้ออกแบบให้สามารถแก้ไขค่าได้ผ่านทาง Firebase และแอปพลิเคชัน

2. Auto soil หมายถึง การกำหนดเงื่อนไขค่าความชื้นต่ำสุดที่จะสั่งให้เปิดปั๊มน้ำแบบอัตโนมัติ โดยได้ออกแบบให้สามารถแก้ไขค่าได้ผ่านทาง Firebase และแอปพลิเคชัน

3. Auto temp หมายถึง การกำหนดเงื่อนไขค่าอุณหภูมิสูงสุดที่จะสั่งให้เปิดพัดลมระบายอากาศทั้ง 2 ตัวพร้อมกันแบบอัตโนมัติ โดยได้ออกแบบให้สามารถแก้ไขค่าได้ผ่านทาง Firebase และแอปพลิเคชัน

4. S1_Auto หมายถึง การกำหนดค่าสถานการณ์เปิดและปิดของสวิตช์ควบคุมโหมดการทำงานแบบอัตโนมัติ โดยถ้าสถานะเป็น 0 หมายถึงปิดการทำงานในโหมดนี้ และถ้าสถานะเป็น 1 หมายถึงเปิดการใช้งานในโหมดนี้

5. S1_Manual หมายถึง การกำหนดค่าสถานะการเปิดและปิดของสวิตช์ควบคุมโหมดการทำงานที่ผู้ใช้ควบคุมด้วยตนเองผ่านแอปพลิเคชัน โดยถ้าสถานะเป็น 0 หมายถึงเปิดการทำงานในโหมดนี้ และถ้าสถานะเป็น 1 หมายถึงปิดการใช้งานในโหมดนี้

6. S2_Lamp หมายถึง การกำหนดสถานะเปิดและปิดสวิตช์ควบคุมหลอดไฟในโหมด โดยถ้าสถานะเป็น 0 หมายถึง ปิดการใช้งานหลอดไฟ และถ้าสถานะเป็น 1 หมายถึง เปิดการใช้งานหลอดไฟ

7. S2_Fan1 หมายถึง การกำหนดสถานะเปิดและปิดสวิตช์ควบคุมพัดลมตัวที่ 1 ในโหมดผู้ใช้กำหนดด้วยตนเอง โดยถ้าสถานะเป็น 0 หมายถึง ปิดการใช้งานพัดลม และถ้าสถานะเป็น 1 หมายถึง เปิดการใช้งานพัดลม

8. S2_Fan2 หมายถึง การกำหนดสถานะเปิดและปิดสวิตช์ควบคุมพัดลมตัวที่ 2 ในโหมดผู้ใช้กำหนดด้วยตนเอง โดยถ้าสถานะเป็น 0 หมายถึง ปิดการใช้งานพัดลม และถ้าสถานะเป็น 1 หมายถึง เปิดการใช้งานพัดลม

9. S5_Pump หมายถึง การกำหนดสถานะเปิดและปิดสวิตช์ควบคุมปั๊มน้ำในโหมดผู้ใช้กำหนดด้วยตนเอง โดยถ้าสถานะเป็น 0 หมายถึง ปิดการใช้งานปั๊มน้ำ และถ้าสถานะเป็น 1 หมายถึง เปิดการใช้งานปั๊มน้ำ

10. Status Fan1 หมายถึง การแสดงสถานะเปิดและปิดของพัดลมตัวที่ 1 แบบตามเวลาจริงบนหน้าแสดงผลการทำงานของระบบในแอปพลิเคชัน

11. Status Fan2 หมายถึง การแสดงสถานะเปิดและปิดของพัดลมตัวที่ 2 แบบตามเวลาจริงบนหน้าแสดงผลการทำงานของระบบในแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. Status Lamp หมายถึง การแสดงสถานะเปิดและปิดของหลอดไฟ แบบตามเวลาจริงบนหน้าแสดงผลการทำงานของระบบในแอปพลิเคชัน

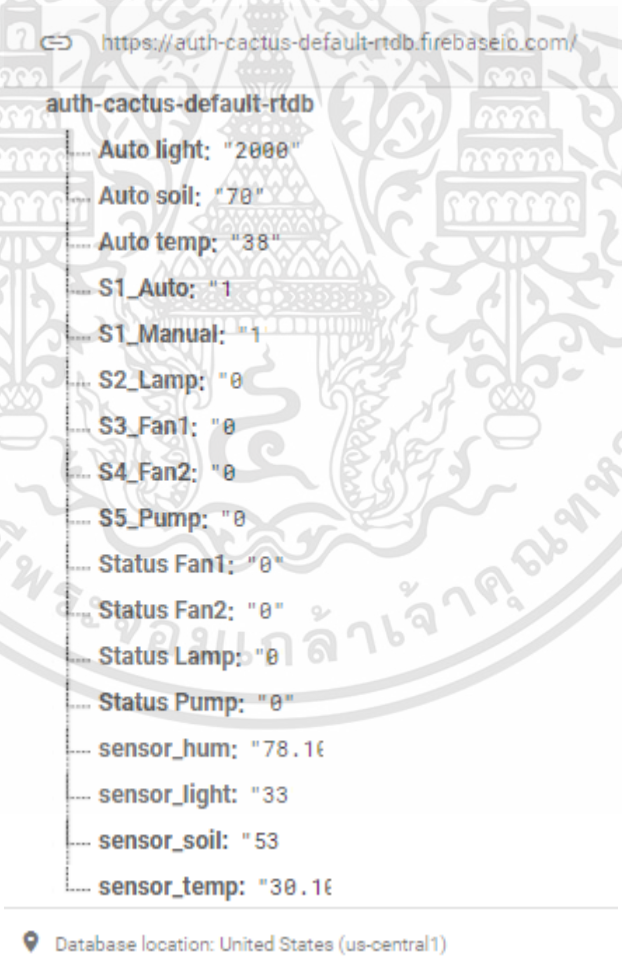
13. Status Pump หมายถึง การแสดงสถานะเปิดและปิดของปั๊มน้ำ แบบตามเวลาจริงบนหน้าแสดงผลการทำงานของระบบในแอปพลิเคชัน

14. sensor_hum หมายถึง การแสดงค่าความชื้นในอากาศจากเซนเซอร์วัดอุณหภูมิให้แสดงบนหน้าแสดงผลการทำงานของระบบในแอปพลิเคชันแบบตามเวลาจริง

15. sensor_light หมายถึง การแสดงค่าความเข้มแสงจากเซนเซอร์วัดวัดความเข้มแสงให้แสดงบนหน้าแสดงผลการทำงานของระบบในแอปพลิเคชันแบบตามเวลาจริง

16. sensor_soil หมายถึง การแสดงค่าความชื้นในดินให้แสดงบนหน้าแสดงผลการทำงานของระบบในแอปพลิเคชันแบบตามเวลาจริง

17. sensor_temp หมายถึง การแสดงค่าอุณหภูมิจากเซนเซอร์วัดอุณหภูมิให้แสดงบนหน้าแสดงผลการทำงานของระบบในแอปพลิเคชันแบบตามเวลาจริง



รูปที่ 3.6 ชื่อที่อยู่ของไฟล์ที่เชื่อมต่อระหว่างอุปกรณ์และ Firebase

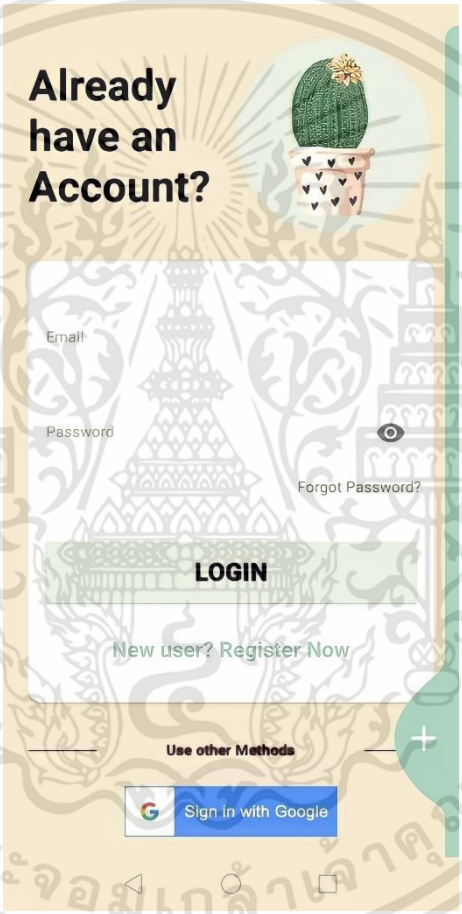
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การออกแบบแอปพลิเคชัน

จากการศึกษาได้มีการทดลองออกแบบแอปพลิเคชันสำหรับระบบควบคุมโรงเรือน โดยได้ทำการออกแบบหน้าใช้งานสำหรับผู้ใช้ ดังนี้

1. หน้าสำหรับเข้าสู่ระบบ

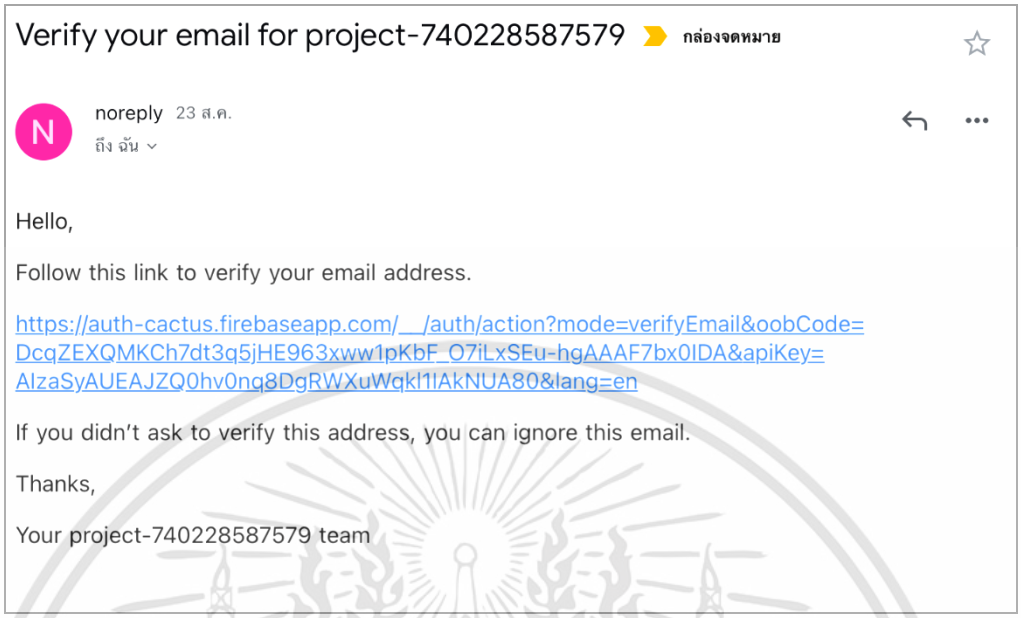
เมื่อเข้าสู่แอปพลิเคชันจะพบกับหน้าสำหรับเข้าสู่ระบบ โดยผู้ใช้จะต้องทำการกรอกข้อมูลอีเมล (Email) และรหัสผ่าน (Password) ที่ได้ทำการลงทะเบียนไว้ ดังแสดงในรูปที่ 3.7



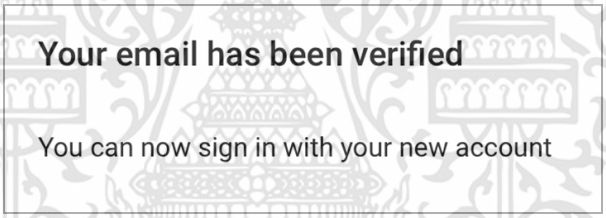
รูปที่ 3.7 หน้าสำหรับเข้าสู่ระบบ

จากรูปที่ 3.7 เมื่อทำการกรอกข้อมูลอีเมลและรหัสผ่านที่ทำการลงทะเบียนแล้วในครั้งแรกที่เข้าสู่ระบบผู้ใช้จะได้รับอีเมลเพื่อยืนยันตัวตน ดังแสดงในรูปที่ 3.8 และเมื่อผู้ใช้ซึ่งเป็นเจ้าของอีเมลทำการกดเพื่อยืนยันตัวตนระบบจะเข้าสู่หน้ายืนยันตัวตนเสร็จเรียบร้อย จากนั้นจึงจะสามารถทำการเข้าสู่ระบบได้ซึ่งแสดงดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 หน้าจอแสดงลิงค์สำหรับยืนยันตัวตนก่อนทำการเข้าสู่ระบบครั้งแรก



รูปที่ 3.9 การยืนยันตัวตนเสร็จสิ้น

2. หน้าสำหรับลงทะเบียน

ผู้ใช้สามารถทำการลงทะเบียนเพื่อสร้างบัญชีผู้ใช้สำหรับเข้าสู่ระบบได้ โดยกำหนดให้ทำการกรอกข้อมูลชื่อ (Name) อีเมล (Email) หมายเลขโทรศัพท์ (Mobile Number) และรหัสผ่าน (Password) ดังแสดงในรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 หน้าสำหรับลงทะเบียนบัญชีผู้ใช้

3. หน้าสำหรับกรณีผู้ใช้ลืมรหัสผ่าน

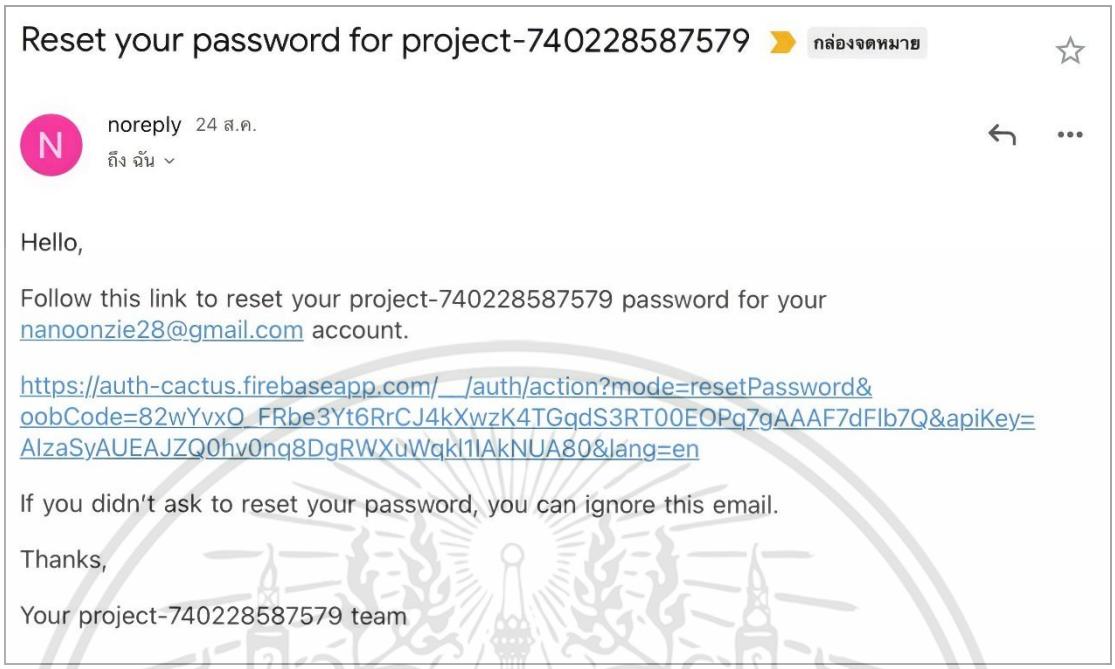
กรณีผู้ใช้ลืมรหัสผ่านให้ทำการกดไปที่ลืมรหัสผ่าน (Forgot Password) ระบบจะแสดงหน้าจอสำหรับกรอกอีเมลของผู้ใช้เพื่อทำการตั้งรหัสผ่านใหม่ ดังแสดงในรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 หน้าสำหรับกรอกอีเมลของผู้ใช้ในกรณีลืมรหัสผ่าน

จากนั้นผู้ใช้จะได้รับการแจ้งเตือนส่งไปทางอีเมลเพื่อยืนยันสำหรับการตั้งรหัสผ่านใหม่ซึ่งแสดงดังรูปที่ 3.12 และเมื่อทำการกดเข้าไปก็จะแสดงหน้าสำหรับสร้างรหัสผ่านใหม่เพื่อใช้ในการเข้าสู่ระบบอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


Reset your password for project-740228587579 กล่องจดหมาย

noreply 24 ส.ค.
ถึง ฉัน

Hello,

Follow this link to reset your project-740228587579 password for your nanoonzie28@gmail.com account.

https://auth-cactus.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=82wYvxO_FRbe3Yt6RrCJ4kXwzK4TGqdS3RT00EOPq7gAAAF7dFlb7Q&apiKey=AIzaSyAUEAJZQ0hv0nq8DgRWXuWqkl1lAkNUA80&lang=en

If you didn't ask to reset your password, you can ignore this email.

Thanks,

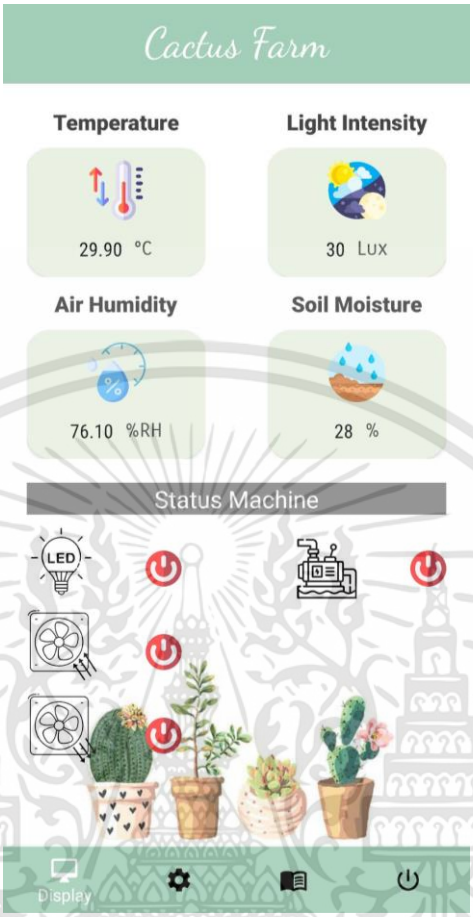
Your project-740228587579 team

รูปที่ 3.12 การแจ้งเตือนสำหรับการตั้งรหัสผ่านใหม่

4. หน้าแสดงผลการทำงานของระบบ

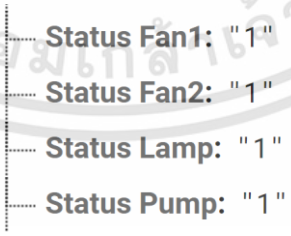
ออกแบบแอปพลิเคชันสำหรับแสดงค่าที่ได้รับจากเซนเซอร์แบบตามเวลาจริงบนฐานข้อมูล Firebase และแสดงสถานะการทำงานของอุปกรณ์แบบทันทีทันใด โดยกำหนดให้ส่วนแรกของแอปพลิเคชันทำการแสดงค่าที่ได้จากเซนเซอร์ทั้งหมด 4 ค่า ได้แก่ อุณหภูมิ ความเข้มแสง ความชื้นในอากาศและความชื้นในดิน อีกทั้งยังออกแบบให้มีการแสดงสถานะการทำงานของอุปกรณ์แต่ละตัว ซึ่งได้แก่ หลอดไฟ LED พัดลม และปั๊มน้ำ เพื่อให้ผู้ใช้รับรู้และสะดวกต่อการควบคุม ดังแสดงในรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



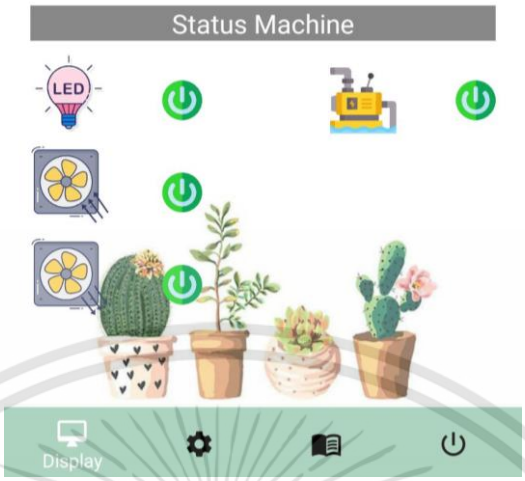
รูปที่ 3.13 หน้าแสดงค่าที่ได้รับจากเซนเซอร์ที่เก็บบนฐานข้อมูล Firebase แบบตามเวลาจริง

ด้านล่างของหน้าแสดงผลคือส่วนแสดงสถานะการทำงานของอุปกรณ์ โดยกำหนดว่าเมื่ออุปกรณ์มีการทำงานหรือสถานะใน Firebase เป็น 1 ดังแสดงในรูปที่ 3.14 โดยแอปพลิเคชันจะแสดงสถานะสีเขียวและภาพอุปกรณ์จะปรากฏสี ดังแสดงในรูปที่ 3.15



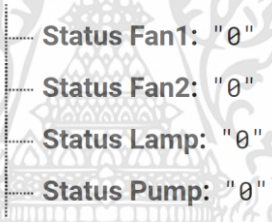
รูปที่ 3.14 สถานะบน Firebase กรณีที่อุปกรณ์มีการเปิดใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

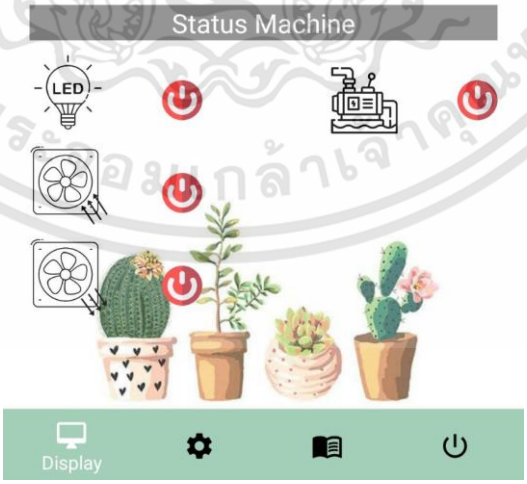


รูปที่ 3.15 สถานะการทำงานของอุปกรณ์บนแอพพลิเคชันเมื่อมีการเปิดใช้งาน

หากอุปกรณ์ไม่มีการทำงานหรือสถานะใน Firebase เป็น 0 ดังแสดงในรูปที่ 3.16 ซึ่งแอปพลิเคชันจะแสดงสถานะสีแดงและภาพอุปกรณ์จะไม่ปรากฏสี ดังแสดงดังรูปที่ 3.17



รูปที่ 3.16 สถานะบน Firebase กรณีที่อุปกรณ์ไม่มีการเปิดใช้งาน

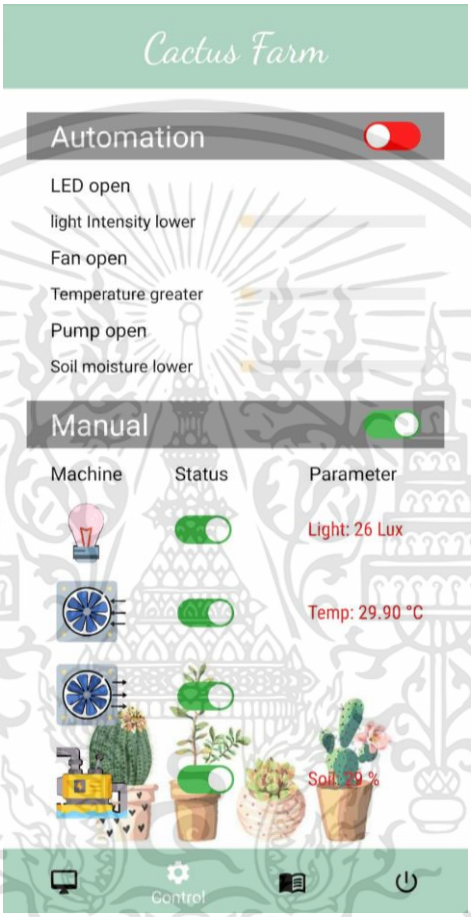


รูปที่ 3.17 สถานะการทำงานของอุปกรณ์บนแอปพลิเคชันเมื่อไม่มีการเปิดใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หน้าต่างสำหรับควบคุมการทำงานของอุปกรณ์

ออกแบบแอปพลิเคชันสำหรับควบคุมการทำงานของอุปกรณ์และเซนเซอร์ โดยกำหนดให้สามารถควบคุมการทำงานได้ทั้งในโหมดอัตโนมัติและโหมดควบคุมโดยผู้ใช้เอง ซึ่งแสดงดังรูปที่ 3.18



รูปที่ 3.18 หน้าต่างสำหรับควบคุมการทำงานของอุปกรณ์

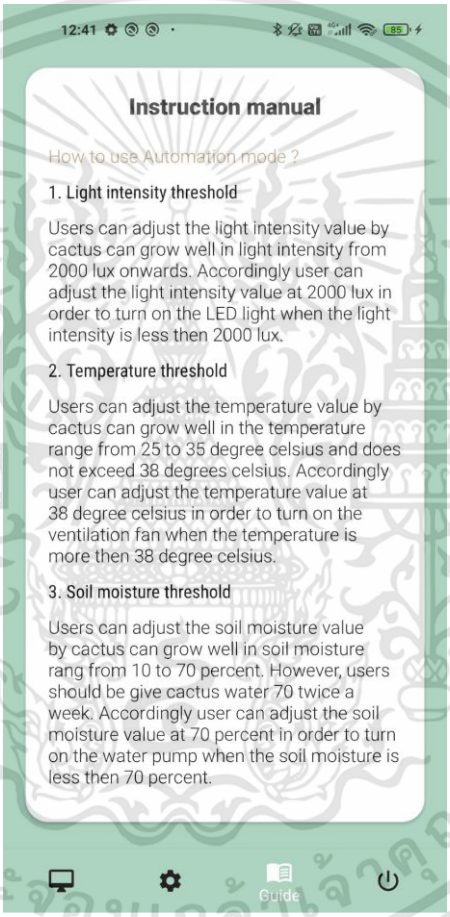
จากรูปที่ 3.18 หากผู้ใช้ทำการเปิดสวิตช์ในโหมดอัตโนมัติ อุปกรณ์จะทำงานแบบอัตโนมัติตามค่าที่ตั้งไว้ซึ่งก็คือความเข้มแสง อุณหภูมิและความชื้นในดิน โดยจากการออกแบบหากค่าความเข้มแสงที่เซนเซอร์วัดได้มีค่าน้อยกว่าค่าคงที่ที่กำหนดหลอดไฟจะเปิด แต่ถ้าความเข้มแสงที่เซนเซอร์วัดได้มีค่ามากกว่าค่าคงที่ที่กำหนดหลอดไฟจะปิด พัดลมจะทำงานเมื่อค่าอุณหภูมิที่วัดได้จากเซนเซอร์มีค่ามากกว่าค่าคงที่กำหนดไว้ หากค่าอุณหภูมิที่วัดได้จากเซนเซอร์มีค่าน้อยกว่าค่าคงที่กำหนดไว้พัดลมก็จะหยุดทำงานทั้งสองตัว หากค่าความชื้นในดินที่เซนเซอร์วัดได้มีค่าน้อยกว่าค่าคงที่ที่กำหนดปั๊มน้ำจะเปิด แต่ถ้าความชื้นในดินที่เซนเซอร์วัดได้มีค่ามากกว่าค่าคงที่ที่กำหนดปั๊มน้ำจะปิด และถ้าผู้ใช้ต้องการทำงานในโหมดควบคุมโดยผู้ใช้เองจะต้องทำการปิดสวิตช์ในโหมดอัตโนมัติและทำการเปิดสวิตช์ในโหมดควบคุมด้วยมือ ซึ่งผู้ใช้สามารถเปิดและปิดสวิตช์เพื่อควบคุมอุปกรณ์ได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตนเอง กรณีที่ผู้ใช้ทำการปิดสวิตช์ทั้ง 2 โหมดระบบจะสั่งงานให้อุปกรณ์ทำงานแบบอัตโนมัติตามค่าคงที่ล่าสุดที่ทำการตั้งค่าไว้

6. หน้าต่างแสดงคู่มือสำหรับผู้ใช้

ออกแบบหน้าต่างแสดงคู่มือกรณีที่เป็นผู้ใช้รายใหม่และยังไม่ทราบข้อมูลพื้นฐานที่เหมาะสมในการเพาะปลูกต้นกระบองเพชร ดังแสดงในรูปที่ 3.19



รูปที่ 3.19 หน้าสำหรับแสดงคู่มือสำหรับผู้ใช้

7. หน้าต่างสำหรับออกจากระบบ

ออกแบบหน้าต่างกรณีที่ผู้ใช้ต้องการออกจากระบบหลังจากจบการทำงาน ดังแสดงในรูปที่ 3.20

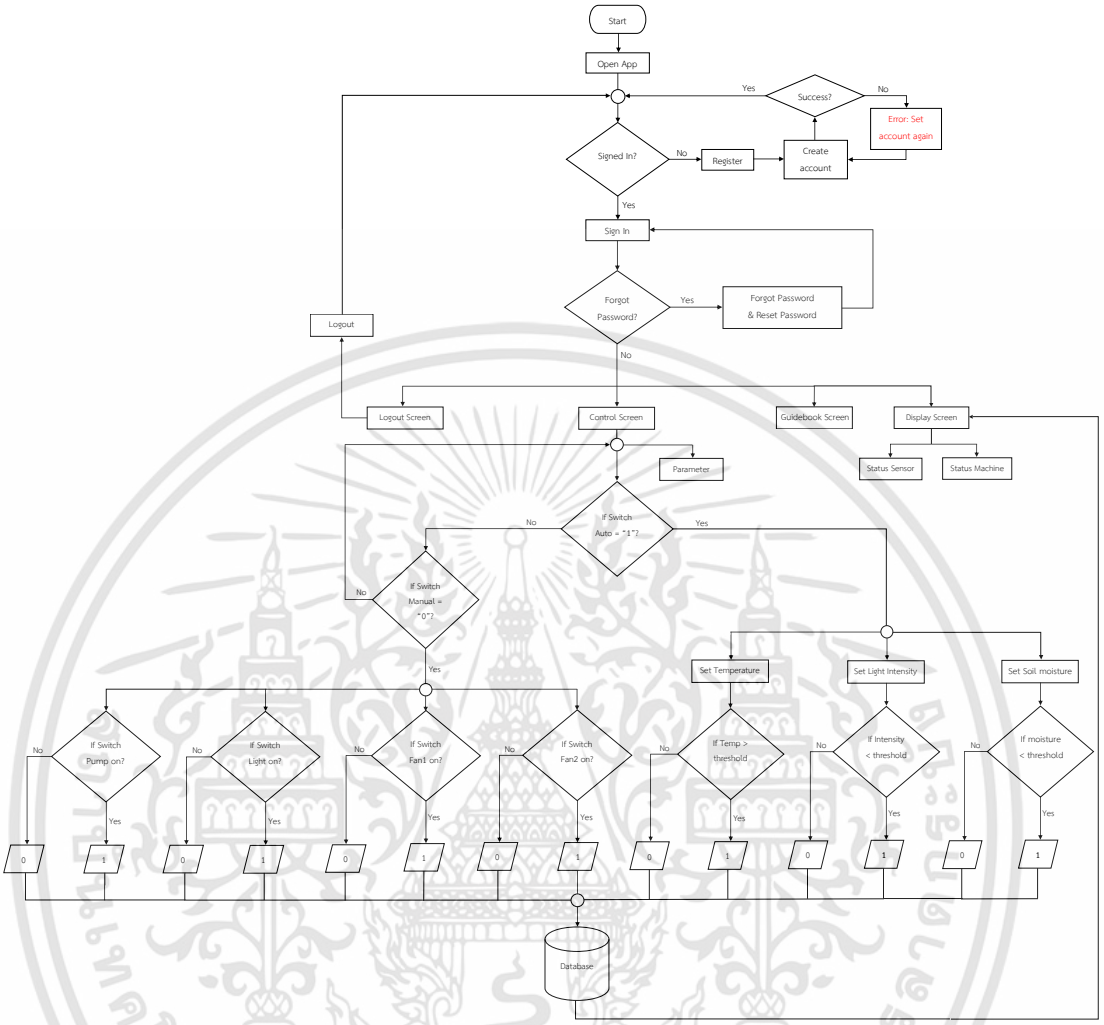
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.20 หน้าสำหรับออกจากระบบ

3.1.5 ออกแบบระบบการทำงานของแอปพลิเคชัน

ออกแบบการทำงานของแอปพลิเคชันโดยจะเริ่มจากผู้ใช้งานเปิดแอปพลิเคชันจะแสดงหน้าสำหรับเข้าสู่ระบบ หากผู้ใช้ไม่มีบัญชีอยู่ในระบบจะต้องไปที่หน้าลงทะเบียน (Register) เพื่อสร้างบัญชีผู้ใช้ จากนั้นจึงกลับมาที่หน้าเริ่มต้นเพื่อเข้าสู่ระบบ และในกรณีที่ผู้ใช้ลืมรหัสผ่านผู้ใช้ต้องกำหนดรหัสผ่านใหม่ในหน้าลืมรหัสผ่าน (Forgot Password) เมื่อเข้าสู่ระบบได้เรียบร้อยแล้วแอปพลิเคชันจะทำการแสดงหน้าแสดงสถานะเซนเซอร์และอุปกรณ์ (Display Screen) เป็นหน้าเริ่มต้น ซึ่งในหน้านี้ผู้ใช้สามารถเลือกไปที่หน้าสำหรับแสดงคู่มือการตั้งค่า (Guidebook Screen) และหน้าสำหรับควบคุมอุปกรณ์ (Control Screen) ได้ โดยหน้าสำหรับควบคุมอุปกรณ์จะมีทั้งหมด 2 โหมดคือโหมดอัตโนมัติและโหมดควบคุมโดยผู้ใช้ เมื่อทำการเปิดและปิดสวิตช์ในแอปพลิเคชันระบบจะทำการส่งค่า 0 ซึ่งก็คือการปิดใช้งานโหมดอัตโนมัติและอุปกรณ์ ส่วนโหมดควบคุมด้วยมือจะเปิดการใช้งานถ้าส่งค่าเป็น 0 หากระบบส่งค่าเป็น 1 ซึ่งก็คือการเปิดใช้งานโหมดอัตโนมัติและอุปกรณ์ ส่วนโหมดควบคุมด้วยมือจะปิดการใช้งานถ้าส่งค่าเป็น 1 เข้าไปที่ฐานข้อมูล จากนั้นฐานข้อมูลจะเป็นตัวกลางเพื่อส่งต่อข้อมูลไปกำหนดการทำงานของอุปกรณ์ตามที่ได้ออกแบบไว้ ดังแสดงในรูปที่ 3.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 แผนผังการทำงานของแอปพลิเคชัน

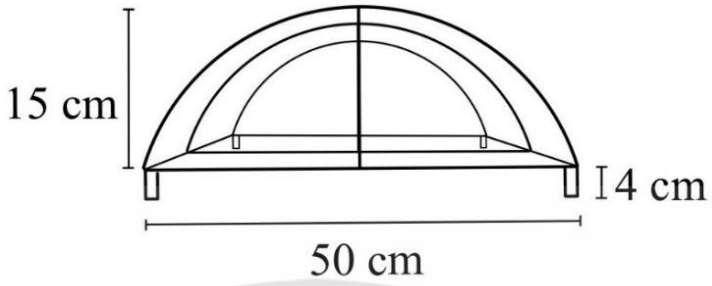
3.1.6 การออกแบบโรงเรือน

การออกแบบโรงเรือนกระบอกเพชรเพื่อใช้งานในพื้นที่ที่มีอย่างจำกัด เช่น ระเบียงหอพัก หรือในห้องที่มีพื้นที่อย่างจำกัด จึงออกแบบให้สามารถถอดเก็บและประกอบได้ง่ายต่อการขนย้ายและมีความแข็งแรงทนทาน โดยมีส่วนประกอบดังนี้

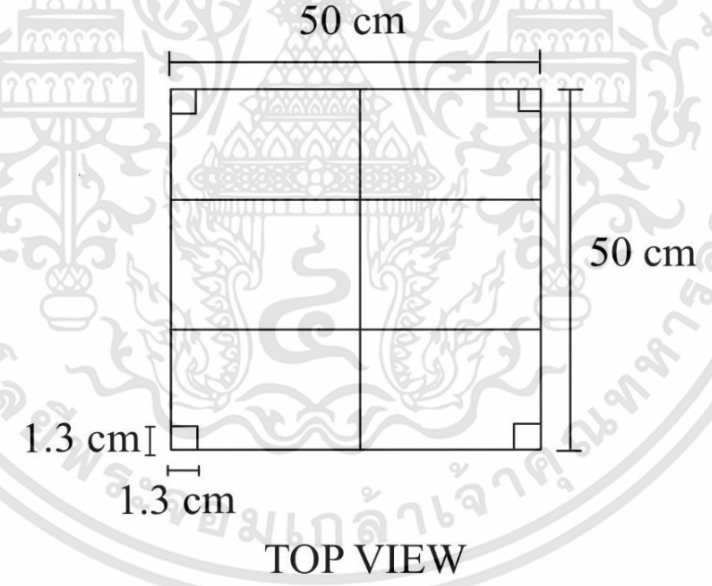
3.1.6.1. ออกแบบหลังคาโรงเรือน

ผู้จัดทำได้มีการออกแบบโครงสร้างบริเวณหลังคาโรงเรือน โดยเลือกรูปทรงหลังคาโค้ง (High Tunnel Greenhouse) เนื่องจากการใช้งานเหมาะสำหรับการปลูกพืชกินต้นและใบตระกูลผัก รวมถึงไม้อวบน้ำ ซึ่งประกอบไปด้วยภาพจำลองขนาดโครงสร้างด้านหน้า ด้านบน และด้านข้าง ดังแสดงในรูปที่ 3.22 3.23 และ 3.24 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

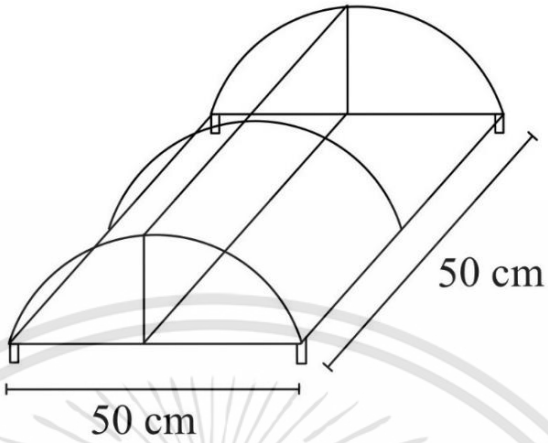


รูปที่ 3.22 โครงสร้างด้านหน้าของหลังคาโรงเรือน



รูปที่ 3.23 โครงสร้างด้านบนของหลังคาโรงเรือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

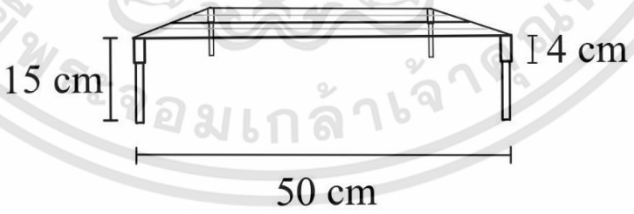


SIDE VIEW

รูปที่ 3.24 โครงสร้างด้านข้างของหลังคาโรงเรือน

3.1.6.2. ออกแบบชั้นวางกระถางต้นกระบองเพชร

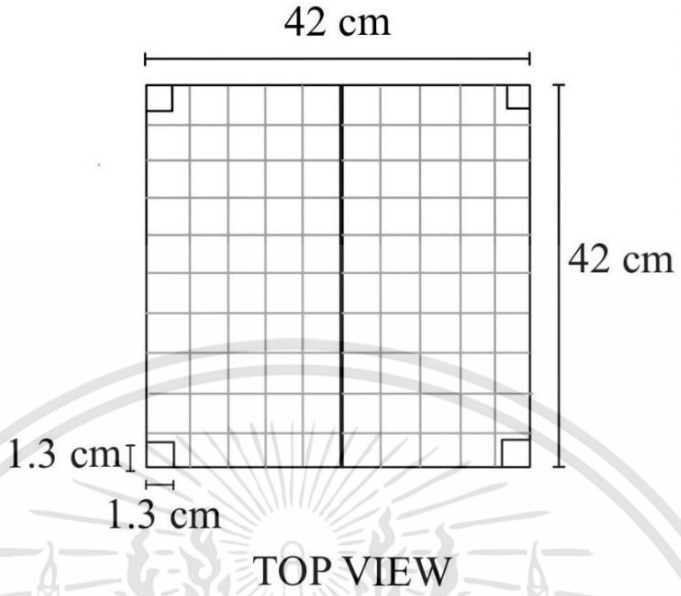
การออกแบบโครงสร้างบริเวณชั้นวางกระถางต้นกระบองเพชรได้คำนึงถึงความแข็งแรง และไม่เก็บน้ำบริเวณใต้กระถาง เนื่องจากการมีน้ำเป็นปัจจัยหลักที่นำไปสู่สาเหตุที่ทำให้ไม้ในโรงเรือนตายได้เช่น ก่อให้เกิดลำต้นเน่าและเกิดเชื้อรา จึงออกแบบให้ชั้นวางกระบอกเพชรเป็นพื้นโปร่ง ซึ่งประกอบไปด้วยภาพจำลองขนาดโครงสร้างด้านหน้า ด้านบน และด้านข้าง ดังแสดงในรูปที่ 3.25 3.26 และ 3.27 ตามลำดับ



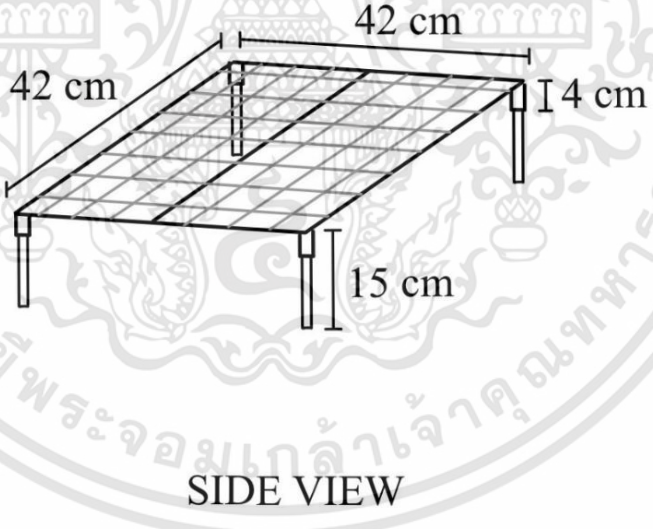
FRONT VIEW

รูปที่ 3.25 โครงสร้างด้านหน้าของชั้นวางกระถางต้นกระบองเพชร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 โครงสร้างด้านบนของชั้นวางกระถางต้นกระบองเพชร



รูปที่ 3.27 โครงสร้างด้านข้างของชั้นวางกระถางต้นกระบองเพชร

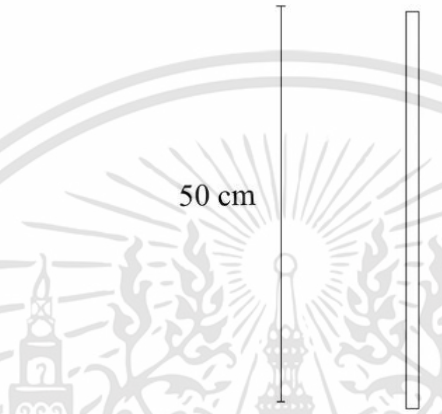
3.1.6.3. ออกแบบโครงสร้างเสาและฐานโรงเรือน

เนื่องจากการออกแบบโรงเรือนกระบอกเพชรเพื่อใช้งานในพื้นที่ที่มีอย่างจำกัด เช่น ระเบียงหอพัก หรือในห้องที่มีพื้นที่อย่างจำกัด จึงออกแบบให้สามารถถอดออกมาจัดเก็บและประกอบกลับคืนได้ เสาของโรงเรือนจึงมีการออกแบบให้ประกอบได้ง่ายโดยใช้ข้อต่อเชื่อมระหว่างเสาโดยใช้เหล็กเส้นยาว 50 เซนติเมตร เป็นเสาโครงสร้างโรงเรือนดังรูปที่ 3.28 และมีการออกแบบเสาเพิ่มเติมด้านข้างของโรงเรือนเพื่อติดตั้งพัดลมดังรูปที่ 3.29 โดยการประกอบเสาทั้งสองแบบต้องอาศัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

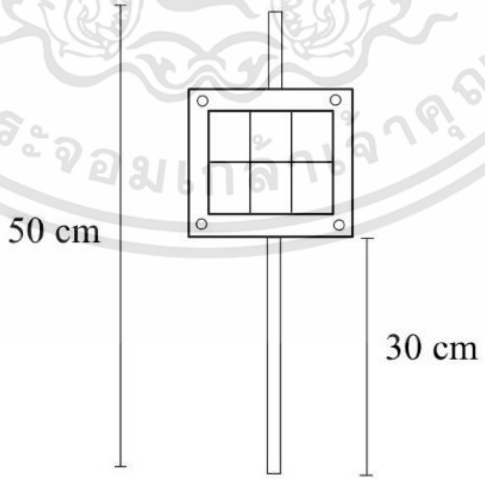
ข้อต่อซึ่งมีลักษณะแตกต่างกันตามตำแหน่งและการใช้งานซึ่งแสดงดังรูปที่ 3.30 ประกอบกับข้อต่อที่แสดงดังรูปที่ 3.31 จะได้โครงสร้างโรงเรือนเป็นสี่เหลี่ยมจัตุรัส โดยแสดงการออกแบบดังนี้

1.แบบโครงสร้างเสา โดยใช้เหล็กเส้นยาว 50 เซนติเมตร



รูปที่ 3.28 เสาสำหรับโครงสร้างโรงเรือน

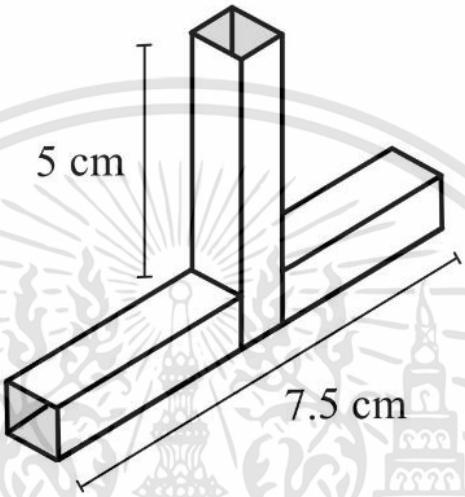
2. แบบโครงสร้างเสาที่ออกแบบมาเพื่อติดตั้งพัดลม



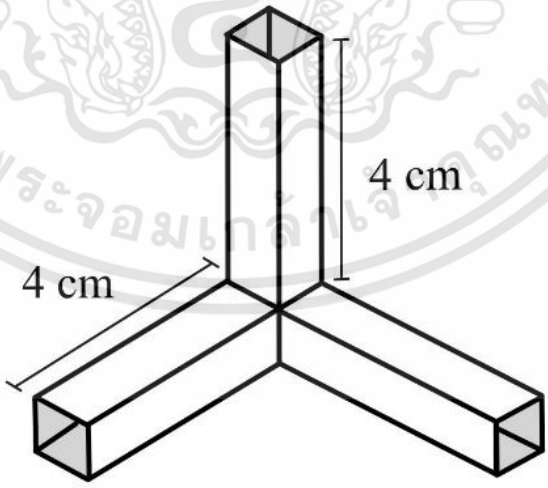
รูปที่ 3.29 เสาที่ออกแบบมาเพื่อติดตั้งพัดลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แบบข้อต่อระหว่างฐานโครงสร้าง ซึ่งมีข้อต่อสองแบบเนื่องจากการใช้งานที่แตกต่างกัน โดยข้อต่อที่เป็นมุมทั้งสี่ของฐานโรงเรือนจะแสดงดังรูปที่ 3.28 และข้อต่อที่มีการเชื่อมระหว่างโครงสร้างเสาพัดลมกับโครงสร้างฐานโรงเรือนแสดงดังรูปที่ 3.29



รูปที่ 3.30 ข้อต่อที่ออกแบบมาเพื่อติดตั้งพัดลม

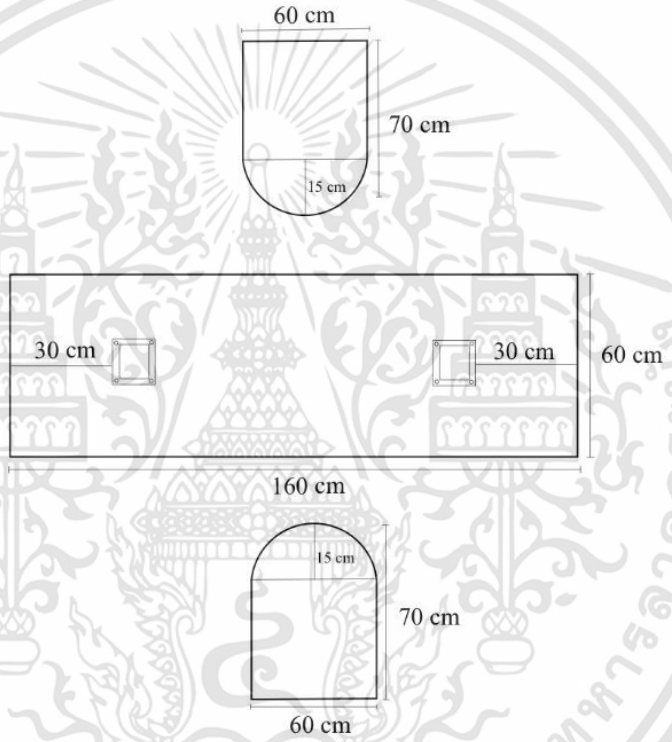


รูปที่ 3.31 ข้อต่อระหว่างฐานโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แบบตัดเย็บพลาสติกคลุมโรงเรือน

โรงเรือนคลุมด้วยพลาสติก PE 200 ไมครอน ด้านข้างคลุมด้วยตาข่ายกันแมลงสีขาวความถี่ 20 ตา อายุการใช้งาน 3 ถึง 5 ปีโดยประมาณ พลาสติกโรงเรือนใส UV 7 เปอร์เซ็นต์ ดังแสดงในรูปที่ 2.5 ถูกใช้ทำโรงเรือนเพื่อป้องกันแมลงศัตรูพืช โรงเรือนกระบองเพชร โรงเรือนปลูกผักปลอดสาร โรงเรือนปลูกพืชแบบไฮโดรโปนิกส์ เป็นต้น ทำให้ภูมิอากาศเหมาะกับการปลูกพืชนั้น ๆ ตลอดปี ทำให้พืชได้รับแสงอย่างต่อเนื่องระยะเวลาปลูกพืชสั้นลงปรับแสงที่เป็นอันตรายต่อพืช เช่น UV และอินฟราเรด โดยมีแบบตัดเย็บดังรูปที่ 3.32

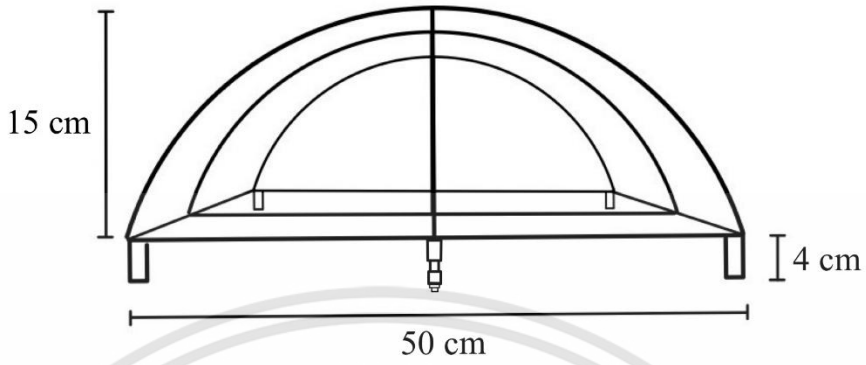


รูปที่ 3.32 แบบตัดเย็บพลาสติกคลุมโรงเรือน

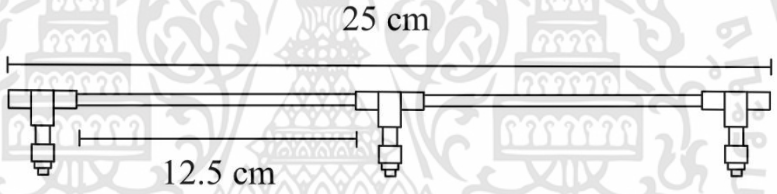
3.1.6.4. ออกแบบการติดตั้งชุดสปริงเกอร์ขนาดเล็ก

ผู้จัดทำได้เลือกใช้ชุดสปริงเกอร์หัวพ่นหมอก โดยออกแบบเพื่อติดตั้งบริเวณคานของโรงเรือน แสดงดังรูปที่ 3.33 ให้มีประสิทธิภาพในการรดน้ำได้อย่างทั่วถึง โดยออกแบบให้สปริงเกอร์มีระยะห่างจากกัน 12.5 เซนติเมตร แสดงดังรูปที่ 3.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.33 ออกแบบเพื่อติดตั้งบริเวณคานของโรงเรือน



รูปที่ 3.34 แสดงระยะห่างระหว่างสปริงเกอร์หัวพ่นหมอก

3.1.7 จัดทำโรงเรือน

วัสดุที่ใช้ในการสร้างโรงเรือนกระบองเพชรแสดงดังตารางที่ 3.1

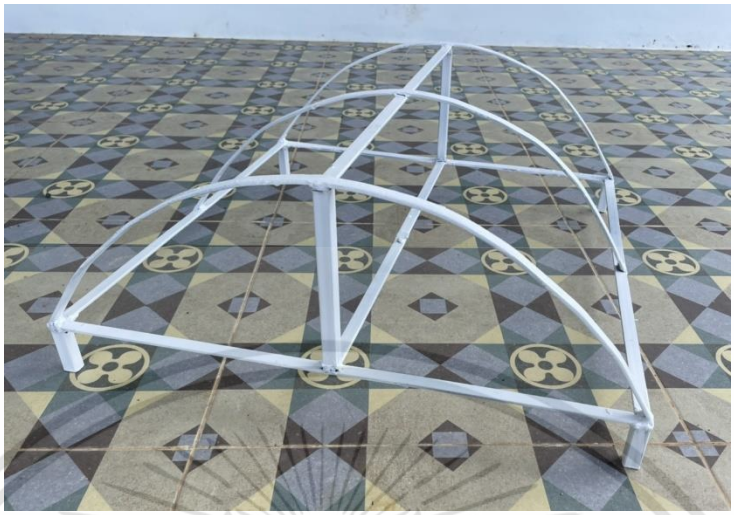
ตารางที่ 3.1 วัสดุที่ใช้ในการสร้างโรงเรือนกระบองเพชร

รายการวัสดุ	ขนาด (เซนติเมตร)	ความยาว (เซนติเมตร)
เหล็กกล่อง	0.95 × 0.95	200
เหล็กกล่อง	1.27 × 1.27	600
เหล็กแบน	1.27	600
เหล็กกลมกัลป์วาไนซ์	0.64	600

3.1.7.1. จัดทำหลังคาโรงเรือนดังแสดงในรูปที่ 3.35 โดยวัสดุที่ใช้ทำหลังคาโรงเรือนมีดังนี้

1. เหล็กกล่อง ขนาด 1.27 × 1.27 เซนติเมตร
2. เหล็กแบน ขนาด 1.27 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.35 โครงสร้างด้านหน้าของหลังคาโรงเรือน

3.1.7.2. จัดทำชั้นวางกระถางต้นกระบองเพชรดังแสดงในรูปที่ 3.36 โดยวัสดุที่ใช้ทำหลังคาโรงเรือนมีดังนี้

1. เหล็กกล่อง ขนาด 0.95 × 0.95 เซนติเมตร
2. เหล็กกล่อง ขนาด 1.27 × 1.27 เซนติเมตร
3. เหล็กแบน ขนาด 1.27 เซนติเมตร

รูปที่ 3.36 โครงสร้างด้านหน้าของชั้นวางกระถางต้นกระบองเพชร

3.1.7.3. จัดทำโครงสร้างเสาและฐานโรงเรือนดังแสดงในรูปที่ 3.37 และ 3.38

1. จัดทำโครงสร้างเสาใช้เหล็กเส้นกลมยาว 50 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.37 เสาสำหรับโครงสร้างโรงเรือน

3.จัดทำโครงสร้างเสาที่ออกแบบมาเพื่อติดตั้งพัดลม

รูปที่ 3.38 เสาที่ออกแบบมาเพื่อติดตั้งพัดลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.7.4 จัดทำข้อต่อระหว่างฐานโครงสร้าง

โดยมีข้อต่อสองแบบเนื่องจากการใช้งานที่แตกต่างกัน โดยข้อต่อที่เป็นมุมทั้ง 4 ของฐานโรงเรือนจะแสดงดังรูปที่ 3.39 และข้อต่อที่มีการเชื่อมระหว่างโครงสร้างเสาพัดลมกับโครงสร้างฐานโรงเรือนแสดงดังรูปที่ 3.40

รูปที่ 3.39 ข้อต่อที่เป็นมุมทั้ง 4 ของฐานโรงเรือน

รูปที่ 3.40 ข้อต่อที่ออกแบบมาเพื่อติดตั้งพัดลม

3.1.7.5. ตัดเย็บพลาสติกคลุมโรงเรือน ดังแสดงในรูปที่ 3.41 โดยวัสดุที่ใช้ทำหลังคาโรงเรือนมีดังนี้

1. พลาสติก PE 200 ไมครอน
2. ซิป ขนาด 63.50 เซนติเมตร
3. ผ้าตีนตุ๊กแกตัวผู้และตัวเมีย สำหรับใช้เปิดปิดหน้าต่างดังรูปที่ 3.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.41 แบบตัดเย็บพลาสติกคลุมโรงเรือน

รูปที่ 3.42 แบบตัดเย็บตะข่ายกันแมลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.7.6 ติดตั้งชุดปริงเกอร์ขนาดเล็ก

ผู้จัดทำได้เลือกใช้ชุดสปริงเกอร์ขนาดเล็กหรือหัวพ่นหมอกที่ให้ละอองน้ำแบบฝอยขนาดเล็ก โดยชุดมินิสปริงเกอร์จะใช้ร่วมกับข้อต่อและข้องอขนาด 0.9 เซนติเมตร ติดตั้งบริเวณคานของโรงเรือนโดยมีระยะห่างระหว่างมินิสปริงเกอร์หัวพ่นหมอก 12.5 เซนติเมตร แสดงดังรูปที่ 3.43

รูปที่ 3.43 ติดตั้งชุดมินิสปริงเกอร์ตามที่ออกแบบไว้

3.2 เครื่องมือที่ใช้ในการทดสอบ

ในปริญญานิพนธ์นี้มีอุปกรณ์และเครื่องมือในการทดสอบดังนี้

3.2.1 NodeMCU ESP32

ในปริญญานิพนธ์นี้ผู้จัดทำใช้ NodeMCU ESP32 เป็นไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ Wi-Fi มาตรฐาน 802.11/b/g/n ซึ่งตัวชิปประมวลผลได้ถูกพัฒนามาจากไมโครคอนโทรลเลอร์รุ่น ESP8266 เพื่อเพิ่มประสิทธิภาพในประมวลผลข้อมูลและรองรับการเชื่อมต่อ GPIO ที่มากขึ้นกว่าเดิม

3.2.2 โมดูลเซนเซอร์วัดความเข้มแสง GY-302 BH1750

ในปริญญานิพนธ์นี้เลือกใช้โมดูลเซนเซอร์วัดความเข้มแสง GY-302 BH1750 ซึ่งสามารถวัดความเข้มแสงได้ตั้งแต่ 0 ลักซ์ ถึง 65,535 ลักซ์ ใช้การเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยใช้การสื่อสารแบบอนุกรมความเร็วต่ำ ทำให้สามารถนำไปใช้งานต่อได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 โมดูลเซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ DHT22

ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้โมดูลเซนเซอร์ DHT22 เป็นเซนเซอร์วัดค่าอุณหภูมิและความชื้นในอากาศ ซึ่งโมดูลชนิดนี้มีความแม่นยำสูงในการวัด สามารถวัดได้ในช่วง −40 องศาเซลเซียส ถึง +80 องศาเซลเซียส และวัดความชื้นสัมพัทธ์ได้ในช่วง 0 เปอร์เซ็นต์ ถึง 100 เปอร์เซ็นต์ มีความละเอียดในทศนิยม 1 ตำแหน่ง ใช้งานได้นานและทนทาน

3.2.4 โมดูลเซนเซอร์วัดค่าความชื้นในดิน

ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้โมดูลเซนเซอร์วัดค่าความชื้นในดิน เป็นเซนเซอร์วัดค่าความชื้นในดินโดยให้ค่าสัญญาณขาออกเป็นค่าดิจิทัลและสามารถแปลงค่าให้เป็นรูปแบบเปอร์เซ็นต์ความชื้นได้ ซึ่งโมดูลชนิดนี้มีความทนทานต่อการสึกกร่อนจากสภาพดินชื้น และทนทานต่อกรดเบสได้เป็นอย่างดี ใช้งานได้นานและทนทาน

3.2.5 โมดูลรีเลย์ 3 โวลต์ (4 channel)

สำหรับโมดูลรีเลย์ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้โมดูลรีเลย์ 3 โวลต์ 4 Channel เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เปรียบเสมือนสวิตช์ไฟที่ใช้แรงดันไฟฟ้าในการเปิดและปิดอุปกรณ์ไฟฟ้าเพื่อควบคุมวงจรต่าง ๆ โดยโมดูลรีเลย์นี้สามารถต่อกับ NodeMCU ESP32 ได้โดยตรง ใช้แรงดันไฟฟ้าขนาด 3 โวลต์ สามารถใช้งานได้ทั้งไฟฟ้ากระแสตรงและกระแสสลับ

3.2.6 หลอดไฟปลูกพืช (LED Grow Light)

ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้หลอดไฟปลูกพืชขนาด 50 วัตต์ เป็นหลอดไฟที่ให้แสงที่มีคุณสมบัติช่วยในการเจริญเติบโตของพืชพันธุ์กระบองเพชร โดยภายในหลอดไฟจะประกอบไปด้วยหลอด LED ขนาดเล็กจำนวนมาก ประกอบไปด้วย หลอด LED สีแดง ซึ่งจะช่วยเพิ่มประสิทธิภาพในการผลิตดอกและทำให้ต้นกระบองเพชรมีสีสันสวยงาน หลอด LED สีน้ำเงิน ช่วยเพิ่มประสิทธิภาพในการสังเคราะห์แสงของพืช ช่วยให้ต้นไม้เจริญเติบโต

3.2.7 พัดลมระบายอากาศ

ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้พัดลมระบายอากาศขนาด 12 เซนติเมตร ทำหน้าที่ช่วยให้อากาศถ่ายเทได้สะดวกลดการสะสมความร้อน โครงสร้างของพัดลมประกอบไปด้วยพลาสติกและเหล็ก มีใบพัดพลาสติกทั้งหมด 5 ใบ ใช้แรงดันไฟฟ้ากระแสสลับขนาด 220 โวลต์ และใช้ความถี่ตั้งแต่ 50 เฮิรตซ์ ถึง 60 เฮิรตซ์

3.2.8 ปั๊มแรงดันขนาด 4.8 บาร์

ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้ปั๊มน้ำแรงดันขนาด 4.8 บาร์ ใช้แรงดันไฟฟ้ากระแสตรง 12 โวลต์ และกระแสไฟฟ้า 2 แอมแปร์ ทำหน้าที่ปั๊มน้ำจากถังบรรจุน้ำเพื่อใช้กับระบบหัวฉีดมินิสปริงเกอร์ซึ่งจะให้น้ำเป็นละอองฝอยขนาดเล็ก โดยใช้อุปกรณ์จ่ายไฟฟ้ากระแสตรง 12 โวลต์ 5 แอมแปร์ เป็นแหล่งจ่ายไฟให้กับปั๊มน้ำ

3.2.9 อุปกรณ์จ่ายไฟฟ้ากระแสตรง

ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้อุปกรณ์จ่ายไฟฟ้ากระแสตรง ทำเป็นอุปกรณ์แหล่งจ่ายไฟฟ้ากระแสตรงคงค่าแรงดันแบบหนึ่งและสามารถเปลี่ยนแรงดันไฟสลับค่าสูงเป็นแรงดันไฟตรงค่าต่ำได้ โดยในปริญญานิพนธ์นี้ ผู้จัดทำได้เลือกใช้อุปกรณ์จ่ายไฟฟ้ากระแสตรงที่ให้แรงดันขาออกขนาด 12 โวลต์ กระแสไฟฟ้า 5 แอมแปร์ เป็นแหล่งจ่ายไฟกระแสตรงให้กับปั๊มน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.10 อุปกรณ์แปลงแรงดันไฟฟ้า

ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้อุปกรณ์แปลงแรงดันไฟฟ้า ที่ให้แรงดันไฟฟ้าขาออก ขนาด 6 โวลต์ และกระแสไฟฟ้า 2 แอมแปร์ เป็นแหล่งจ่ายไฟฟ้าให้กับ NodeMCU ESP32 แสดงดังรูปที่ 3.44

รูปที่ 3.44 อุปกรณ์แปลงแรงดันไฟฟ้า [21]

3.2.11 โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

ในปริญญานิพนธ์นี้ผู้จัดทำเลือกใช้โทรศัพท์มือถือ Xiaomi Redmi note9 แสดงดังรูปที่ 3.45 ในการควบคุมการทำงานของแอปพลิเคชัน

รูปที่ 3.45 Xiaomi Redmi note9 [22]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การจัดเก็บผลการทดสอบ

3.3.1 การทดสอบการทำงานของชุดคำสั่งควบคุมอุปกรณ์

ทำการทดสอบการทำงานร่วมกับระหว่างเซนเซอร์วัดความเข้มแสง เซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ โดยการเขียนโปรแกรมภาษา C ให้กับไมโครคอนโทรลเลอร์ NodeMCU ESP32 สั่งการควบคุมอุปกรณ์ผ่านสวิตช์รีเลย์ จากนั้นสังเกตผลจาก Serial Monitor ในโปรแกรม Arduino IDE

3.3.2 การทดสอบการรับส่งข้อมูลระหว่าง NodeMCU ESP32 กับฐานข้อมูล Firebase Realtime

ทำการทดสอบการเชื่อมต่อ Wi-Fi ของNodeMCU ESP32 กับฐานข้อมูล Firebase Realtime โดยสร้างโครงงานทดลองที่มีชื่อว่า Auth-cactus เพื่อทำการทดสอบการรับและส่งข้อมูลระหว่าง NodeMCU ESP32 กับฐานข้อมูล จากนั้นบันทึกผลการเปลี่ยนแปลงของข้อมูลที่ส่งไปยังฐานข้อมูล

3.3.3 การทดสอบการใช้งานเซนเซอร์และอุปกรณ์ภายในโรงเรือน

ทำการทดสอบการติดตั้งเซนเซอร์ในตำแหน่งต่าง ๆ ภายในโรงเรือน จากนั้นทดสอบการทำงานของเซนเซอร์และอุกรณ์ในโรงเรือนในสถานที่กลางแจ้ง เพื่อบันทึกผลค่าความเข้มแสง ค่าอุณหภูมิและความชื้นในอากาศ เพื่อเป็นข้อมูลในการวิเคราะห์ผลต่อไปในอนาคต

3.3.4 การทดสอบการเชื่อมต่อระหว่างฐานข้อมูล Firebase Realtime กับแอปพลิเคชัน

ทำการทดสอบด้วยการเขียนโปรแกรมภาษา Java สั่งการแอปพลิเคชันให้ทำการเชื่อมต่อกับฐานข้อมูล Firebase เพื่อแสดงผลข้อมูลที่ได้รับจากเซนเซอร์และทำการส่งคำสั่งควบคุมอุปกรณ์ไปเก็บไว้บนฐานข้อมูล

3.3.5 การทดสอบการทำงานของแอปพลิเคชัน

ทดสอบการแสดงผลการทำงานของเซนเซอร์และทดสอบการทำงานของคำสั่งควบคุมอุปกรณ์ภายในโรงเรือน เพื่อทดสอบการเชื่อมต่อระหว่างแอปพลิเคชันและฐานข้อมูล Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

สำหรับการทำงานของระบบสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร ผู้จัดทำได้ทำการเก็บผลการทำงานของระบบโดยแบ่งการทดสอบและจัดเก็บผลการทดสอบออกเป็นส่วนต่าง ๆ ดังต่อไปนี้

4.1 ผลการทดสอบการทำงานของชุดคำสั่งควบคุมอุปกรณ์

ในการออกแบบและเขียนชุดคำสั่งเพื่อควบคุมการทำงานของอุปกรณ์ตามภาพการเชื่อมต่อของระบบโดยรวมในรูปที่ 3.5 ผู้จัดทำได้ทำการทดสอบการทำงานชุดคำสั่งร่วมกับรีเลย์ทั้งหมด 4 channel และเขียนคำสั่งให้โปรแกรมทำงานแบบวนซ้ำ (Loop) โดยแสดงผลการทดสอบผ่านหน้าจอ Serial Monitor ดังรูปที่ 4.1 ซึ่งสังเกตได้ว่าเมื่อค่าความเข้มแสงเป็นไปตามเงื่อนไขซึ่งมีค่าน้อยกว่า 2,000 ลักซ์ หลอดไฟจะทำงานแบบอัตโนมัติและแสดงสถานะ ON ในส่วนของอุณหภูมิซึ่งไม่ตรงตามเงื่อนไขเนื่องจากอุณหภูมิมีค่าน้อยกว่า 38 องศาเซลเซียส ส่งผลให้รีเลย์ที่ควบคุมการทำงานของพัดลมไม่ทำงานและพัดลมทั้งสองตัวอยู่ในสถานะ OFF และในส่วนของระบบให้น้ำจะสังเกตได้ว่าเมื่อความชื้นในดินมีค่าน้อยกว่า 70 เปอร์เซ็นต์ รีเลย์ที่ควบคุมการทำงานของปั๊มน้ำจะเปิดการทำงานซึ่งจะแสดงสถานะ ON

```
Temperature: 22.00 C  Light: 60 lux  Soil moisture: 46 %
LED : ON
FAN1 : OFF
FAN2 : OFF
PUMP : ON
```

รูปที่ 4.1 ผลการทดสอบการทำงานตามเงื่อนไขที่กำหนดในโหมดอัตโนมัติ

4.2 ผลการทดสอบการรับส่งข้อมูลระหว่าง NodeMCU ESP32 กับฐานข้อมูล Firebase Realtime

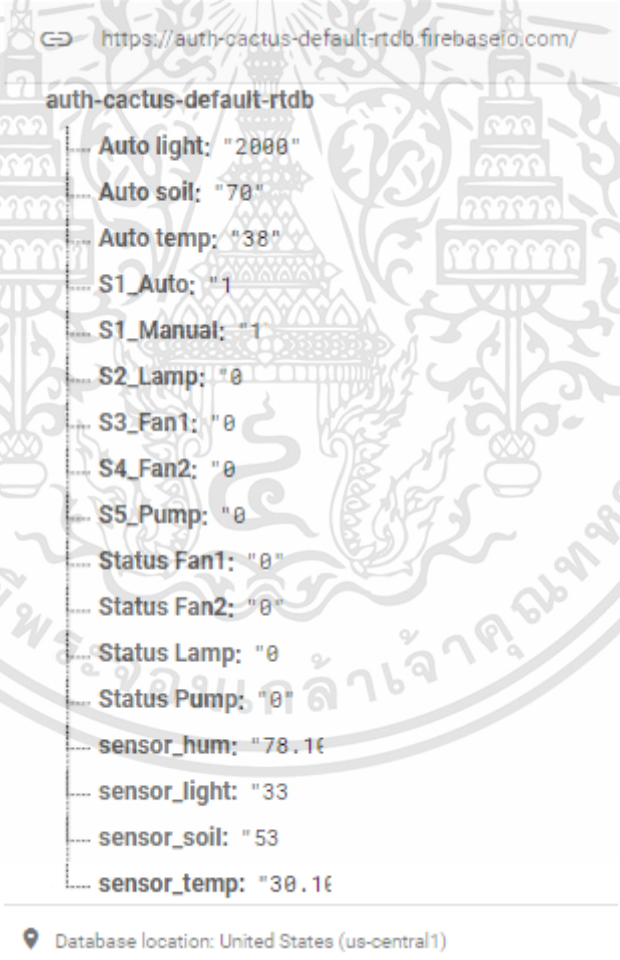
ในการทดสอบการรับส่งข้อมูลระหว่าง NodeMCU ESP32 กับ ฐานข้อมูล Firebase Realtime จะต้องมีการเปิดใช้งานโหมดการเชื่อมต่อ Wi-Fi ของ NodeMCU ESP32 เพื่อใช้ในติดต่อสื่อสารข้อมูลไปยังฐานข้อมูลแบบตามเวลาจริง ซึ่งครั้งนี้ผู้จัดทำได้ใช้การเปิดบริการอินเทอร์เน็ตไร้สายสาธารณะบนโครงข่ายโทรศัพท์มือถือ ย่านความถี่ 2.4 กิกะเฮิรตซ์ เป็นแหล่งปล่อยสัญญาณ Wi-Fi ให้กับ NodeMCU ESP32 และเขียนโปรแกรมระบุการใช้งานในโหมด STA กำหนดชื่อของบริการอินเทอร์เน็ตไร้สายสาธารณะบนโทรศัพท์มือถือ Wi-Fi และรหัสผ่านเพื่อใช้ในการเชื่อมต่อ โปรแกรมจะทำการค้นหาสัญญาณโดยจะแสดงสถานะ “.....” บนจอ Serial Monitor เมื่อสามารถเชื่อมต่อ Wi-Fi ได้จะแสดงข้อความ “WiFi connected” และแสดงหมายเลข IP Address ที่ NodeMCU ESP32 ได้รับมา ดังแสดงในรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Connecting to BOW
.....
WiFi connected
IP address:
192.168.43.236
```

รูปที่ 4.2 ผลการเชื่อมต่อ Wi-Fi ในโหมด STP

ผู้จัดทำได้ทดสอบการเชื่อมต่อกับฐานข้อมูล Firebase Realtime โดยสร้างโครงงานทดลองที่มีชื่อว่า Auth-cactus เพื่อทำการเชื่อมต่อกับแอปพลิเคชันบน Android โดยได้สร้างที่อยู่ของไฟล์ไว้สำหรับเชื่อมต่อข้อมูลระหว่างอุปกรณ์และแอปพลิเคชันและที่อยู่ของไฟล์สำหรับการส่งค่าจากเซนเซอร์ต่าง ๆ ไปแสดงบนแอปพลิเคชัน และในหน้าแสดงผลการทำงานของระบบ และ หน้าควบคุมการทำงานของอุปกรณ์ ซึ่งจะมีชื่อที่อยู่ของไฟล์แทนคำสั่ง ผลการทดสอบแสดงดังรูปที่ 4.3



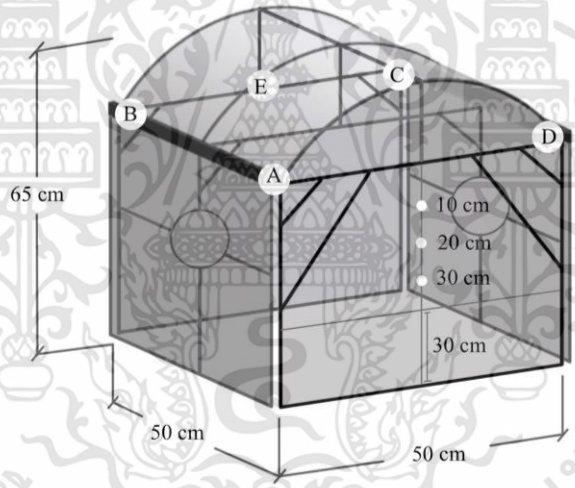
รูปที่ 4.3 ชื่อที่อยู่ของไฟล์ที่เชื่อมต่อระหว่างอุปกรณ์และ Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดสอบการเชื่อมต่อระหว่าง NodeMCU ESP32 กับ ฐานข้อมูล ดังแสดงในรูปที่ 4.3 สามารถสรุปได้ว่า NodeMCU ESP32 สามารถส่งคำสั่งการสร้างชื่อที่อยู่ของไฟล์ไปยังฐานข้อมูลได้ โดยจะเห็นได้ว่าค่าเซนเซอร์ต่าง ๆ จะแสดงในส่วนของชุดไฟล์คำสั่งที่สร้างไว้และมีการรายงานค่าแบบทันทีทันใดเมื่อมีการเปลี่ยนแปลง และยังสามารถรับคำสั่งการทำงานจากการสั่งงานผ่านแอปพลิเคชันผ่านฐานข้อมูลได้

4.3 ผลการทดสอบการใช้งานเซนเซอร์และอุปกรณ์ภายในโรงเรือน

ในการทดสอบการใช้งานเซนเซอร์และอุปกรณ์ภายในโรงเรือน ผู้จัดทำได้ทำการทดสอบและเก็บผลการทดสอบการติดตั้งเซนเซอร์วัดความเข้มแสงในตำแหน่ง A B C D และ E ภายในโรงเรือนต้นแบบ โดยทดสอบการอ่านค่าแสงจากหลอดไฟปลูกพืชภายในโรงเรือนแสดงดังรูปที่ 4.4 และผลที่วัดได้แสดงดังตารางที่ 4.1



รูปที่ 4.4 ตำแหน่งการทดสอบติดตั้งเซนเซอร์เพื่อวัดความเข้มแสงภายในโรงเรือน

ตารางที่ 4.1 ความเข้มแสงในระยะความสูงและตำแหน่งที่ต่างกันภายในโรงเรือน

ระยะความสูงจากหลอดไฟ (เซนติเมตร)	ความเข้มแสง (ลักซ์)				
	ตำแหน่ง A	ตำแหน่ง B	ตำแหน่ง C	ตำแหน่ง D	ตำแหน่ง E
10	360	353	378	370	896
15	620	626	623	630	1078
20	847	860	849	852	1564

จากข้อมูลข้างต้นทำให้ได้เลือกติดตั้งเซนเซอร์ความเข้มแสงที่ตำแหน่ง B ซึ่งคือตำแหน่งฝั่งซ้ายทางด้านหลังของโรงเรือนเพราะสามารถติดตั้งได้สะดวกและการจัดวางสายไฟทำได้ง่าย และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยระยะห่างจากหลอดไฟที่ควรจะห่างจากพืชกระบองเพชรอย่างน้อย 15 เซนติเมตร ถึง 20 เซนติเมตร เพื่อป้องกันไม่ให้ต้นกระบองเพชรได้รับแสงจากหลอดไฟมากเกินความจำเป็นซึ่งไม่เหมาะสมต่อการเจริญเติบโตและป้องกันความร้อนจากหลอดไฟเมื่อเปิดใช้งานต่อเนื่องเป็นระยะเวลายาวนาน และในการติดตั้งเซนเซอร์วัดอุณหภูมิได้ติดตั้งที่ตำแหน่ง C ซึ่งก็คือตำแหน่งฝั่งขวาทางด้านหลังของโรงเรือน โดยติดตั้งที่ระยะความสูงจากพื้นโรงเรือนเป็นระยะประมาณ 35 เซนติเมตร โดยตำแหน่งการติดตั้งเซนเซอร์ทั้ง 2 แสดงดังรูปที่ 4.5 วงกลมสีแดงแสดงตำแหน่งของเซนเซอร์วัดความเข้มแสงและวงกลมสีน้ำเงินแสดงตำแหน่งของเซนเซอร์วัดอุณหภูมิ

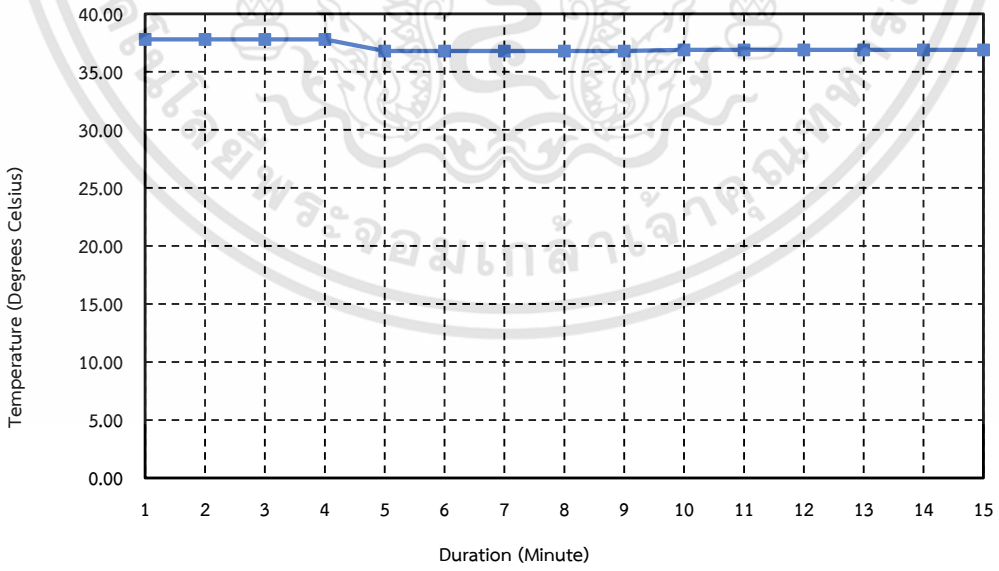
รูปที่ 4.5 ตำแหน่งการติดตั้งเซนเซอร์ภายในโรงเรือน

ผู้จัดทำได้ทดสอบการระบายอากาศภายในโรงเรือนขณะใช้งานที่สถานที่กลางแจ้งและมีแดดจัด โดยจัดเก็บผลการทดสอบการเปลี่ยนแปลงอุณหภูมิเมื่อเปิดพัดลมระบายอากาศทั้ง 2 ตัว โดยอุณหภูมิเริ่มต้นขณะทำการทดสอบที่วัดได้คือ 37.80 องศาเซลเซียส จากนั้นบันทึกผลการเปลี่ยนแปลงอุณหภูมิทุก ๆ 1 นาทีเป็นเวลา 15 นาที ซึ่งได้ผลการทดสอบดังตารางที่ 4.2 และกราฟการเปลี่ยนแปลงค่าอุณหภูมิเมื่อเปิดใช้งานพัดลมเป็นเวลา 15 นาที แสดงดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ค่าอุณหภูมิเมื่อเปิดใช้งานพัดลมระบายอากาศภายในโรงเรือน

ระยะเวลา (นาที)	อุณหภูมิ (องศาเซลเซียส)
1	37.80
2	37.80
3	37.80
4	37.80
5	36.80
6	36.80
7	36.80
8	36.80
9	36.80
10	36.90
11	36.90
12	36.90
13	36.90
14	36.90
15	36.90



รูปที่ 4.6 กราฟการเปลี่ยนแปลงค่าอุณหภูมิเมื่อเปิดพัดลมเป็นเวลา 15 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดสอบในตารางที่ 4.2 สังเกตได้ว่าจากการเปิดใช้งานพัดลมระบายอากาศ ณ อุณหภูมิเริ่มต้นที่ 37.8 องศาเซลเซียส สรุปได้ว่าเมื่อทดสอบการใช้งานโรงเรือนในสถานที่กลางแจ้งและมีแดดมากการเปิดพัดลมช่วยระบายอากาศทั้ง 2 ตัวภายในโรงเรือนจะยังไม่เพียงพอต่อการปรับระดับอุณหภูมิให้ลดลงได้และเนื่องจากการใช้งานกลางแจ้งจะมีความร้อนสะสมค่อนข้างสูงดังนั้นการนำไปใช้งานจริง ในอนาคตอาจจะมีการปรับเปลี่ยนขนาดของพัดลมหรือเปลี่ยนจากการใช้พัดลมระบายอากาศแบบธรรมดาเป็นพัดลมไอเย็นเป็นต้น และหากต้องการลดเวลาในการระบายอากาศสามารถทำได้จากการคำนวณปริมาตรอากาศภายในโรงเรือนเพื่อนำไปสู่การเลือกใช้ขนาดของพัดลมระบายอากาศที่เหมาะสม ซึ่งพัดลมที่เลือกใช้ควรจะมีอัตราการระบายอากาศที่เหมาะสมและเพียงพอต่อขนาดของโรงเรือนที่ใช้ โดยตารางที่ 4.3 แสดงตัวอย่างความสัมพันธ์ระหว่างปริมาตรของห้องกับอัตราการระบายอากาศของพัดลมที่เหมาะสม

ตารางที่ 4.3 ความสัมพันธ์ระหว่างปริมาตรห้องกับปริมาณลมระบายอากาศที่เหมาะสมของพัดลม

ปริมาตรห้อง (ลูกบาศก์เมตร)	ปริมาณลมระบายอากาศ = ปริมาตรห้อง (ลูกบาศก์เมตร) x 10 (ลูกบาศก์เมตรต่อชั่วโมง)
0.125	1.25
1.000	10
8.000	80
15.000	150

จากตารางที่ 4.3 แสดงความสัมพันธ์ระหว่างปริมาตรห้องและปริมาณลมระบายอากาศที่เหมาะสมซึ่งจะนำมาพิจารณาในการเลือกใช้พัดลมระบายอากาศโดยในโครงงานนี้ ผู้จัดทำได้สร้างโรงเรือนที่มีปริมาตรเท่ากับ 0.125 ลูกบาศก์เมตร ดังนั้นพัดลมที่เหมาะสมที่จะใช้ระบายอากาศในโรงเรือนควรมีปริมาณลมระบายอากาศมากกว่า 1.25 ลูกบาศก์เมตรต่อชั่วโมง ซึ่งขนาดของพัดลมระบายอากาศที่แตกต่างกันจะให้ปริมาณลมระบายอากาศที่แตกต่างกันไป ดังนั้นการเลือกใช้พัดลมแต่ละขนาดจึงมีความจำเป็นที่จะต้องพิจารณาประกอบกับปริมาตรของโรงเรือน โดยความสัมพันธ์ระหว่างขนาดของพัดลมและปริมาณลมระบายอากาศแสดงดังตารางที่ 4.4

ตารางที่ 4.4 ความสัมพันธ์ระหว่างขนาดของพัดลมและปริมาณลมระบายอากาศ

ขนาดพัดลมระบายอากาศ (เซนติเมตร)	ปริมาณลมระบายอากาศ (ลูกบาศก์เมตรต่อชั่วโมง)
12	230
20	580
25	920
30	1,152

จากตารางที่ 4.4 พบว่าปริมาณลมระบายอากาศจะขึ้นอยู่กับการเลือกใช้พัดลมระบายอากาศขนาดต่าง ๆ ซึ่งการตัดสินใจเลือกใช้จะขึ้นอยู่กับลักษณะการนำไปใช้งานและขนาดของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

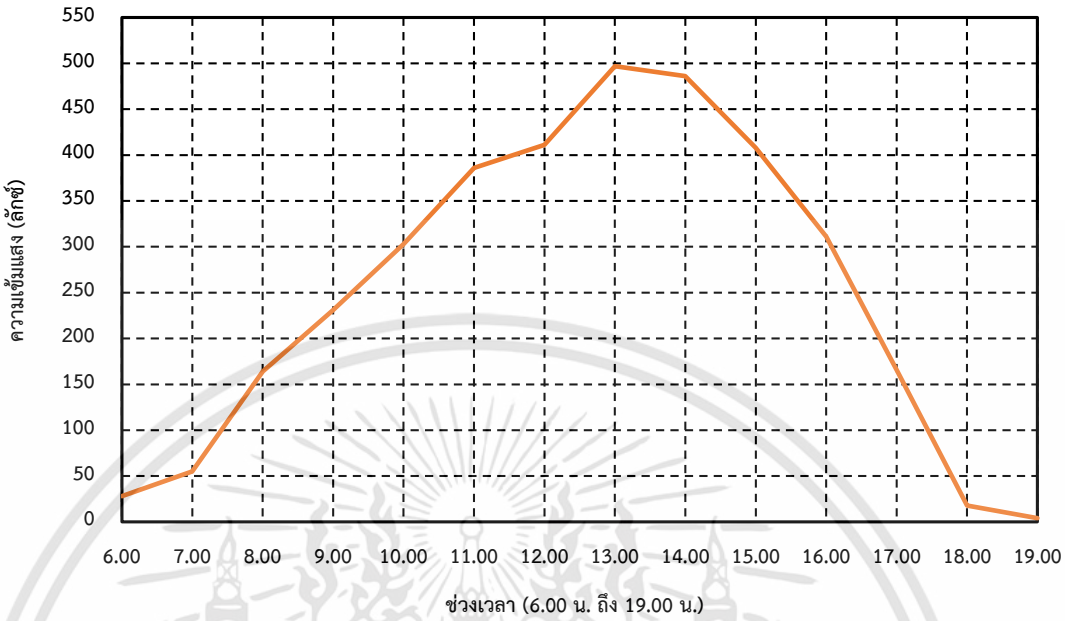
โรงเรือน โดยขนาดของโรงเรือนจะสามารถบอกถึงปริมาตรของอากาศภายในโรงเรือนได้ ดังนั้นการเลือกใช้พัดลมระบายอากาศหรือพัดลมไอเย็น จึงจำเป็นจะต้องพิจารณาให้เหมาะสมกับขนาดของโรงเรือน

จากนั้นผู้จัดทำได้ทดสอบการใช้งานเซนเซอร์วัดความเข้มแสงโดยทดสอบในสถานที่กลางแจ้ง ณ ช่วงเวลาที่แตกต่างกัน โดยได้บันทึกผลการทดสอบการอ่านค่าความเข้มแสงเป็นเวลาทุก ๆ 1 ชั่วโมงตั้งแต่ช่วงเช้าจนถึงกลางคืน ซึ่งผลของความเข้มแสง ณ ช่วงเวลาต่าง ๆ แสดงดังตารางที่ 4.5 และความสัมพันธ์ระหว่างความเข้มแสงกับช่วงเวลาต่าง ๆ แสดงดังรูปที่ 4.7

ตารางที่ 4.5 ความสัมพันธ์ระหว่างความเข้มแสงกับช่วงเวลา

ช่วงเวลา (6.00 น. ถึง 19.00 น.)	ความเข้มแสง (ลักซ์)
6.00 น.	28
7.00 น.	55
8.00 น.	164
9.00 น.	231
10.00 น.	303
11.00 น.	386
12.00 น.	411
13.00 น.	497
14.00 น.	486
15.00 น.	408
16.00 น.	311
17.00 น.	166
18.00 น.	18
19.00 น.	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

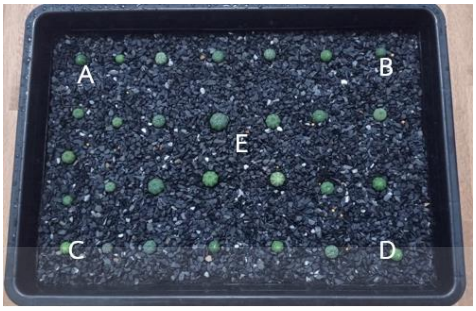


รูปที่ 4.7 ความสัมพันธ์ระหว่างความเข้มแสงกับช่วงเวลา

จากตารางที่ 4.5 จะสังเกตได้ว่าค่าความเข้มแสงที่เซนเซอร์อ่านได้ในช่วงเวลาที่แตกต่างกันนั้น จะได้ค่าเข้มแสงที่มีค่ามากที่สุดคือ 497 ลักซ์ ซึ่งอยู่ในช่วงเวลาประมาณ 13.00 น. เป็นช่วงเวลากลางวัน โดยปกติแล้วต้นกระบองเพชรจะได้รับแสงแดดมากที่สุดในช่วงเวลานี้ ทั้งนี้อาจจะขึ้นอยู่กับปัจจัยทางสภาพอากาศอื่น ๆ อาทิ วันที่มีเมฆมาก อาจทำให้ปริมาณแสงแดดนั้นแตกต่างกันไป โดยจากกราฟในรูปที่ 4.6 จะสังเกตได้ว่าความเข้มแสงมีค่าเพิ่มขึ้นอย่างต่อเนื่องตั้งแต่ช่วงเวลา 6.00 น. และมีค่ามากที่สุดในช่วงเวลา 13.00 น. จนเมื่อถึงช่วงเวลาประมาณ 15.00 น. ความเข้มแสงลดลงอย่างต่อเนื่อง และมีค่าน้อยที่สุดคือ 4 ลักซ์ ในช่วงเวลากลางคืนหลัง 18.00 น. ซึ่งเป็นช่วงเวลาหลังจากพระอาทิตย์ตก ดังนั้นการเปิดใช้งานหลอดไฟปลูกพืชในช่วงเวลาที่แสงไม่เพียงพอ อาทิ ช่วงเช้าตรู่ หรือช่วงกลางคืน จะช่วยทดแทนแสงแดดในเวลากลางวันได้และด้วยคุณสมบัติของหลอดไฟปลูกพืชที่จะให้แสงที่มีสีและความยาวคลื่นแตกต่างกันตามความจำเป็นของการเจริญเติบโตทั้งต้นและดอกของกระบองเพชรซึ่งจะช่วยให้ต้นกระบองเพชรสามารถเจริญเติบโตได้แม้ในวันที่แสงไม่เพียงพอก็ตาม

จากนั้นผู้จัดทำได้ทำการทดสอบการทำงานของเซ็นเซอร์วัดความชื้นในดินและเก็บผลการทดสอบการการอ่านค่าความชื้นในดินที่ละตำแหน่งทั้งหมด 5 ตำแหน่ง ได้แก่ ตำแหน่ง A B C D และ E ภายในกระถางปลูกกระบองเพชรแสดงดังรูปที่ 4.8 โดยทดสอบร่วมกับการทำงานของระบบให้น้ำเป็นเวลา 5 นาที และผลที่วัดได้แสดงดังตารางที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ตำแหน่งการวัดค่าความชื้นในดินภายในกระถางกระบองเพชร

ตารางที่ 4.6 ค่าความชื้นในดินทั้ง 5 ตำแหน่งเมื่อเปิดใช้งานระบบให้น้ำภายในโรงเรือน

เวลา (นาที)	ค่าความชื้นในดิน (เปอร์เซ็นต์)				
	ตำแหน่ง A	ตำแหน่ง B	ตำแหน่ง C	ตำแหน่ง D	ตำแหน่ง E
1	1	1	1	1	1
2	28	28	28	27	29
3	52	53	53	53	54
4	75	77	76	77	77
5	95	96	96	96	98

จากข้อมูลในตารางที่ 4.6 เมื่อเปิดใช้งานระบบให้น้ำเป็นเวลา 5 นาที และเก็บค่าความชื้นในดินที่ละตำแหน่งทั้งหมด 5 ตำแหน่ง เป็นเวลาทุก 1 นาที จะสังเกตได้ว่าค่าความชื้นในดินค่อยๆเพิ่มขึ้นอย่างต่อเนื่องจากเริ่มต้นก่อนให้น้ำค่าความชื้นในดินมีค่าเท่ากับ 1 เปอร์เซ็นต์ทั้ง 5 ตำแหน่ง และเมื่อเปิดใช้งานจนครบ 5 นาที ค่าความชื้นในดินเฉลี่ยทั้ง 5 ตำแหน่งมีค่าประมาณ 96 เปอร์เซ็นต์ โดยจากผลการทดสอบจะพบว่าตำ ณ ตำแหน่ง E ซึ่งจะอยู่บริเวณตรงกลางของกระถางกระบองเพชรจะมีค่าความชื้นในดินมากกว่าตำแหน่งอื่น ๆ เพียงเล็กน้อยโดยประมาณ 2 ถึง 3 เปอร์เซ็นต์ ซึ่งสามารถสรุปผลจากการทดสอบได้ว่าการใช้งานระบบให้น้ำโดยการติดตั้งชุดมินิสปริงเกอร์จำนวน 3 ตัว ภายในโรงเรือน มีความสามารถเพียงพอในการกระจายตัวของละอองน้ำให้ครอบคุลมบริเวณพื้นที่ทั้งหมดของกระถางกระบองเพชร ดังนั้นกระบองเพชรที่ปลูกในกระถางจะได้รับปริมาณน้ำที่เพียงพอในทุก ๆ ต้น และนอกจากนั้นผู้จัดทำได้เก็บผลค่าความชื้นในดินทั้ง 5 ตำแหน่งดังเดิมหลังจากที่ให้น้ำกระบองเพชรจนกระทั่งค่าความชื้นในดินมีค่าประมาณ 90 เปอร์เซ็นต์ เพื่อที่จะสังเกตการเปลี่ยนแปลงค่าความชื้นในดินที่ผ่านไปแต่ละวัน โดยได้เก็บผลค่าความชื้นในดิน ณ เวลา 18.00 น. เป็นระยะเวลา 5 วันหลังจากให้น้ำ ซึ่งผลที่วัดได้แสดงดังตารางที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ค่าความชื้นในดินทั้ง 5 ตำแหน่งหลังจากการให้น้ำผ่านไปเป็นระยะเวลา 5 วัน

วันที่	ค่าความชื้นในดิน (เปอร์เซ็นต์)				
	ตำแหน่ง A	ตำแหน่ง B	ตำแหน่ง C	ตำแหน่ง D	ตำแหน่ง E
1	75	76	75	75	78
2	49	48	48	49	50
3	23	21	23	23	23
4	5	4	4	5	6
5	1	1	1	1	1

จากตารางที่ 4.7 จะสังเกตได้ว่าเมื่อเวลาผ่านไป 4 วันหลังจากให้น้ำกระบองเพชรค่าความชื้นในดินจะมีค่าประมาณ 4 ถึง 6 เปอร์เซ็นต์ในแต่ละตำแหน่ง ซึ่งบ่งบอกได้ว่าดินเริ่มมีสภาวะแห้งและขาดน้ำ ถึงแม้ว่าจากกระบองเพชรเป็นพืชที่ไม่ได้ต้องการน้ำในปริมาณมากทุกวันก็ตาม แต่ค่าความชื้นในดินที่เหมาะสมต่อการเจริญเติบโตควรมีค่าไม่น้อยกว่า 10 เปอร์เซ็นต์ เพราะหากว่าน้อยกว่านี้อาจจะส่งผลให้กระบองเพชรเกิดการชะลอและหยุดการเจริญเติบโตได้ ดังนั้นแล้วผู้ปลูกควรจะให้น้ำกระบองเพชรในปริมาณที่เหมาะสมเป็นเวลา 2 ครั้งต่อสัปดาห์ ซึ่งการให้น้ำแต่ละครั้งควรจะเว้นระยะห่างประมาณ 3 ถึง 4 วัน ทั้งนี้การพิจารณาการให้น้ำในแต่ละครั้งอาจขึ้นอยู่กับปัจจัยสภาพแวดล้อมและสภาพอากาศในแต่ละวัน ซึ่งโดยปกติแล้วหากอยู่ในฤดูร้อนกระบองเพชรสามารถทนต่อสภาพอากาศได้ค่อนข้างดีและควรมีการให้น้ำ 2 ถึง 3 ครั้งต่อสัปดาห์ ส่วนในฤดูฝนและฤดูหนาวจะมีค่าความชื้นและปริมาณไอน้ำอิ่มตัวในอากาศมากกว่าฤดูร้อน จึงไม่ควรให้น้ำบ่อยหรือให้น้ำแก่กระบองเพชรในปริมาณที่มากเกินไปเพราะอาจจะส่งผลต่อการเจริญเติบโต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์มีเป้าหมายเพื่อศึกษาและคิดค้นอุปกรณ์เพื่อใช้สำหรับสังเกตการณ์ควบคุมโรงเรือนกระบองเพชรที่สามารถตรวจสอบค่าความเข้มแสง อุณหภูมิ ความชื้นในอากาศ และความชื้นในดินภายในโรงเรือน โดยการประมวลผลด้วย NodeMCU ESP32 ที่ทำงานร่วมกับเซนเซอร์วัดความเข้มแสง เซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ และเซนเซอร์วัดความชื้นในดิน ซึ่งอุปกรณ์ดังกล่าวจะทำงานร่วมกับพัดลมระบายอากาศ หลอดไฟปลูกพืช และปั๊มน้ำ โดยระบบจะส่งข้อมูลไปยังฐานข้อมูลแบบตามเวลาจริงเพื่อแสดงค่าจากเซนเซอร์ต่าง ๆ และสามารถควบคุมอุปกรณ์ในโรงเรือนโดยการสั่งการผ่านแอปพลิเคชัน

ผู้จัดทำได้ออกแบบและจัดทำอุปกรณ์สำหรับสังเกตการณ์ควบคุมโรงเรือนกระบองเพชร โดยได้ทำการทดสอบการทำงานของชุดคำสั่งควบคุมอุปกรณ์ตามเงื่อนไขการทำงานในโหมดอัตโนมัติ โดยได้ทดสอบการทำงานในส่วนของอุปกรณ์ร่วมกับการทำงานควบคุมผ่านแอปพลิเคชัน ซึ่งในส่วนของการทดสอบการทำงานของอุปกรณ์ที่แสดงในรูป 4.1 อุปกรณ์สามารถทำงานร่วมกันได้ โดยการเปิดใช้งานหลอดไฟปลูกพืชจะเป็นไปตามค่าเกณฑ์ที่กำหนดเมื่อค่าแสงมีค่าน้อยกว่า 2,000 ลักซ์ หลอดไฟจะเปิดทำงานโดยอัตโนมัติ เช่นเดียวกับการควบคุมพัดลมระบายอากาศ เมื่ออุณหภูมิมีค่ามากกว่า 38 องศาเซลเซียส พัดลมระบายอากาศทั้ง 2 ตัวจะงานโดยอัตโนมัติ และในส่วนของการควบคุมระบบให้น้ำจะเป็นไปตามค่าเกณฑ์ที่ผู้ใช้กำหนดโดยเมื่อค่าความชื้นในดินมีค่าน้อยกว่า 70 เปอร์เซ็นต์ปั๊มน้ำจะเปิดทำงานโดยอัตโนมัติ และผู้จัดทำได้ทดสอบการเชื่อมต่อระหว่าง NodeMCU ESP32 กับฐานข้อมูล ซึ่งผลการทดสอบที่แสดงในรูปที่ 4.3 พบว่า NodeMCU ESP32 สามารถเชื่อมต่อและสื่อสารรับส่งคำสั่งต่าง ๆ ไปยัง ฐานข้อมูลแบบตามเวลาจริงได้ โดยในส่วนนี้จะมีการอัปเดตค่าของเซนเซอร์แบบทันทีทันใดเมื่อมีการเปลี่ยนแปลง และในส่วนของการทดสอบการทำงานของแอปพลิเคชัน ผู้จัดทำได้ทดสอบการเชื่อมต่อระหว่างฐานข้อมูล Firebase Realtime กับแอปพลิเคชันโดยผลการทดสอบในส่วนดังกล่าวพบว่าแอปพลิเคชันสามารถสั่งการควบคุมอุปกรณ์ผ่านการส่งคำสั่งควบคุมไปยังฐานข้อมูลและ NodeMCU ESP32 แบบตามเวลาจริงและสามารถแสดงค่าเซนเซอร์ที่เก็บในฐานข้อมูลผ่านแอปพลิเคชันได้แบบตามเวลาจริง นอกจากนี้ผู้จัดทำได้ทำการบันทึกผลค่าของเซนเซอร์และการทำงานของอุปกรณ์ต่าง ๆ ภายในโรงเรือนที่ทดสอบการใช้งาน ณ สถานที่กลางแจ้ง โดยมีการบันทึกผลสองส่วนด้วยกันส่วนแรกคือการบันทึกผลค่าอุณหภูมิและความชื้นในอากาศเมื่อเปิดใช้งานพัดลมระบายอากาศภายในโรงเรือน ซึ่งจากผลการทดสอบที่แสดงในตารางที่ 4.2 ปรากฏว่า การเปิดใช้งานพัดลมระบายอากาศทั้งสองตัวภายในโรงเรือนจะยังไม่เพียงพอต่อการปรับระดับอุณหภูมิให้ลดลงได้เนื่องจากการใช้งานกลางแจ้งมีความร้อนสะสมค่อนข้างสูง และส่วนที่สองการบันทึกค่าความสัมพันธ์ระหว่างความเข้มแสงกับช่วงเวลา ซึ่งผลการทดสอบที่แสดงในตารางที่ 4.5 และจากความสัมพันธ์ที่แสดงในรูปที่ 4.6 พบว่าค่าความเข้มแสงที่เซนเซอร์อ่านได้จะมีค่ามากที่สุดคือ 497 ลักซ์ ซึ่งอยู่ในช่วงเวลาประมาณ 13.00 น. ซึ่งเป็นช่วงเวลากลางวันที่กระบองเพชรจะได้รับแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากที่สุดในช่วงเวลานี้ สามารถสรุปได้ว่าค่าความเข้มแสงจะมีค่าเพิ่มขึ้นอย่างต่อเนื่องและลดลงจนมีค่าน้อยที่สุดคือ 4 ลักซ์ ในช่วงเวลาประมาณ 6.00 น. ดังนั้นการให้แสงจากหลอดไฟปลูกพืชในช่วงเวลาที่แสงแดดไม่เพียงพอต่อการเจริญเติบโตจึงมีความสำคัญอย่างยิ่งเพราะจะช่วยให้ต้นกระบองเพชรสามารถเจริญเติบโตได้แม้ในวันที่แสงไม่เพียงพอก็ตาม และในส่วนสุดท้ายคือการทดสอบการทำงานของเซนเซอร์วัดความชื้นในดินและเก็บผลค่าความชื้นในดิน ซึ่งผลการทดสอบการเปิดใช้งานระบบให้น้ำแสดงในตารางที่ 4.7 ซึ่งจะพบว่าหลังเปิดใช้งานระบบให้น้ำจนค่าความชื้นในดินมีค่ามากสุดเท่ากับ 90 เปอร์เซ็นต์จากนั้นสังเกตผลที่ได้จะพบว่าค่าความชื้นในดินที่ผ่านไปเป็นเวลา 5 วันหลังจากให้น้ำ มีค่าลดอย่างต่อเนื่อง จนเมื่อผ่านไป 4 วันค่าความชื้นมีค่าประมาณ 4 ถึง 6 เปอร์เซ็นต์ ซึ่งเป็นเวลาที่เหมาะสมที่จะให้น้ำแก่ต้นกระบองเพชร เนื่องจากกระบองเพชรควรได้รับน้ำในปริมาณที่เพียงพอ 2 ครั้งต่อสัปดาห์ และไม่ควรปล่อยให้ค่าความชื้นในดินของกระบองเพชรมีค่าต่ำกว่า 10 เปอร์เซ็นต์เป็นเวลาติดต่อกันยาวนาน เพราะการให้น้ำในปริมาณมากเกินไปและการขาดน้ำเป็นเวลานานอาจจะทำให้กระบองเพชรเกินการชะลอและหยุดการเจริญเติบโตในที่สุด

5.2 ข้อเสนอแนะ

ในขณะนี้การเปิดใช้งานพัดลมระบายอากาศทั้ง 2 ตัวจะยังไม่เพียงพอต่อการช่วยปรับระดับอุณหภูมิในโรงเรือนให้ลดลง ดังนั้นการนำไปใช้งานจริงในอนาคตอาจจะมีการปรับเปลี่ยนขนาดของพัดลมหรือเปลี่ยนจากพัดลมระบายอากาศแบบธรรมดาเป็นพัดลมไอเย็น เพื่อเพิ่มประสิทธิภาพในการระบายอากาศภายในโรงเรือน ในส่วนของแอปพลิเคชันอาจจะมีการเพิ่มฟังก์ชันการตั้งค่าเวลาในการเปิดปิดของอุปกรณ์ และเพิ่มฟังก์ชันสำหรับวิเคราะห์และรายงานสภาพอากาศเพื่อช่วยประกอบการตัดสินใจของผู้ใช้ในการเปิดปิดอุปกรณ์ และนอกจากนี้ยังสามารถนำไปประยุกต์ร่วมกับโรงเรือนที่มีขนาดใหญ่มากยิ่งขึ้น เพื่ออำนวยความสะดวกให้กับเกษตรกรที่เพาะพันธุ์ต้นกระบองเพชร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] วิภา ผิวจันทร์, ขวัญภิรมย์ พรมมา, ชัญญานุช ศรีถาน. “ทฤษฎีที่เกี่ยวข้องในการศึกษาเรื่องกระบองเพชรจิ๋ว.” [ออนไลน์]. เข้าถึงได้จาก : https://sites.google.com/site/mookyuykik/bth-thi-2-thvsdi-thi-keiywkhxng. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[2] สวนคุณยายน้องข้าวปั้น. “ข้อมูลความชื้น กรด-ด่างและแสง ในต้นแคคตัส.” [ออนไลน์]. เข้าถึงได้จาก : http://gmagarden.blogspot.com/2017/06/blog-post_11.html. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[3] สวัสดีเกษตร. “รูปทรงโรงเรือนทรงหลังคาโค้ง.” [ออนไลน์]. เข้าถึงได้จาก : https://www.hellofarming.com. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[4] บริษัท บิลค์ เอเชีย จำกัด. “เหล็กกล่องกัลป์วาไนซ์. ” [ออนไลน์]. เข้าถึงได้จาก : https://www.builk.com/yello. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[5] บริษัท วีซีเอสเอเชีย จำกัด. “เหล็กแบน.” [ออนไลน์]. เข้าถึงได้จาก : http://www.vcsasiasteel.com/index.php/th/steel/steel-plate/steel-flat-bars.html. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[6] TATA TISCON. “เหล็กเส้นกลม.” [ออนไลน์]. เข้าถึงได้จาก : https://www.wazzadu.com/article/3737. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[7] kongsawat . “พลาสติกคลุมใสโรงเรือน.” [ออนไลน์]. เข้าถึงได้จาก : http://kongsawat.com/products/greenhouse. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[8] Ioxhop. “ESP32 เบื้องต้น บทที่ 1 แนะนำ ESP32.” [ออนไลน์]. เข้าถึงได้จาก : https://www.ioxhop.com/article/62/esp32. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[9] Ioxhop. “ESP32 เบื้องต้น บทที่ 8 การใช้งานเซนเซอร์ต่าง ๆ.” [ออนไลน์]. เข้าถึงได้จาก : https://www.ioxhop.com/article/69/esp32. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[10] Arduitronics. “Soil Moisture Sensor.” [ออนไลน์.] เข้าถึงได้จาก : https://www.arduitronics.com/product/104/soil-moisture-sensor. (วันที่สืบค้นข้อมูล : 20 มกราคม 2565)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[11] My Arduino. “โมดูล Relay 3V.” [ออนไลน์]. เข้าถึงได้จาก : https://www.myarduino.net/category/151/relay-3v. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[12] AEE-Grow Light. “LED Grow Light คืออะไร.” [ออนไลน์]. เข้าถึงได้จาก : https://www.aee-growlight.com/led-grow-light. (วันที่สืบค้นข้อมูล : 28 สิงหาคม 2564)

[13] W.WACHARA. “พัดลมระบายอากาศทำงานอย่างไร.” [ออนไลน์]. เข้าถึงได้จาก : http://ventilation-smartseo123.blogspot.com/2014/01/ventilating-fan.html. (วันที่สืบค้นข้อมูล : 10 กุมภาพันธ์ 2565)

[14] Thai Water. “ปั๊มน้ำรุ่น Green-01 DC 12 โวลต์.” [ออนไลน์]. เข้าถึงได้จาก : https://www.thaiwatersystem.com/product/ -green-01no-pressure-switch. (วันที่สืบค้นข้อมูล : 10 กุมภาพันธ์ 2565)

[15] Cybertice. “Switching Power Supply 12V 5A” [ออนไลน์]. เข้าถึงได้จาก : https://www.cybertice.com/product/462/switching-power-supply%A2-12v.5a. (วันที่สืบค้นข้อมูล : 10 กุมภาพันธ์ 2565)

[16] Thai Water. “ท่อ PE ขนาด 3 หุน.” [ออนไลน์]. เข้าถึงได้จาก : https://www.thaiwatersystem.com/product/-PE/3. (วันที่สืบค้นข้อมูล : 10 กุมภาพันธ์ 2565)

[17] ชุติภรณ์ วิเศษชาติ. “Android คือ อะไร” [ออนไลน์.]. เข้าถึงได้จาก : https://sites.google.com/a/thantong.ac.th/rabb-ptibati-kar-android/android-khux-xari. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[18] Palm’s. “เริ่มต้นสร้าง Android Application พื้นฐานด้วย Android Studio” [ออนไลน์.]. เข้าถึงได้จาก : https://medium.com/@palmz/เริ่มต้นสร้าง-android-application-พื้นฐานด้วย-android-studio-lab-3sb04-3fda43b07a1. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[19] Android Developers. “User Interface & Navigation” [ออนไลน์.]. เข้าถึงได้จาก : https://developer.android.com/guide/topics/ui. (วันที่สืบค้นข้อมูล : 20 สิงหาคม 2564)

[20] “XML คืออะไร” [ออนไลน์.]. เข้าถึงได้จาก : https://www.mindphp.com/คู่มือ/73-คืออะไร/2129-xml-คืออะไร.html. (วันที่สืบค้นข้อมูล : 21 สิงหาคม 2564)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[21] mr.win - TC Admin. “Java คืออะไร” [ออนไลน์.]. เข้าถึงได้จาก : https://www.mindphp.com/คู่มือ/73-คืออะไร/2129-xml-คืออะไร.html. (วันที่สืบค้นข้อมูล : 22 สิงหาคม 2564)

[22] “Firebase Basic [EP1] : Firebase คืออะไร ?” [ออนไลน์.]. เข้าถึงได้จาก : http://www.arduino-makerzone.com/article/54/firebase-basic-ep1-firebase-คืออะไร. (วันที่สืบค้นข้อมูล : 25 สิงหาคม 2564)

[23] Jedsada Saengow. “[Firebase] คืออะไร มาดูวิธีสร้าง Project และทำความรู้จักกับ Firebase” [ออนไลน์.]. เข้าถึงได้จาก : https://medium.com/jed-ng/firebase-คืออะไร-มาดูวิธีสร้าง-project-มาดูวิธีสร้าง-firebase-d48bfac67b14. (วันที่สืบค้นข้อมูล : 25 สิงหาคม 2564)

[24] “วิธีใช้งานโปรแกรม Arduino IDE เบื้องต้น.” [ออนไลน์]. เข้าถึงได้จาก : https://poundxi.com/วิธีใช้งานโปรแกรม-arduino-ide-เบื้องต้น/. (วันที่สืบค้นข้อมูล : 25 สิงหาคม 2564)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คำสั่งควบคุมการทำงานของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <DHT.h>
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>
#include <BH1750FVI.h>
#include <NTPtimeESP32.h>

NTPtime NTPch("ch.pool.ntp.org");   // Server NTP
strDateTime dateTime;

// Set wifi //
#define WIFI_STA_NAME "BOW"
#define WIFI_STA_PASS "0948154588"
// DHT sensor//
#define DHTPIN 15
#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
// Soil sensor //
#define SoilSensorPin 25
#define Pin 34
 //Firebase//
#define FIREBASE_HOST "auth-cactus-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "VUem0yW35eM3PWnOCTL72Nu7vduwjWrarXVQcX3e"

// Set Relay 3 Chanal//
int Lamp = 19;
int FAN1 = 18;
int FAN2 = 5;
int Pump = 13;

// Set soil sensor //
int SoilPercentValue1 = 0;
int CurrentSoilValue1 = 0;
int MapReadSoilValue1 = 0;

int SoilPercentValue2 = 0;
int CurrentSoilValue2 = 0;
int MapReadSoilValue2 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int sum;
float avg;

// Set Treshold from app//
int      A;
int      B;
int      C;

// Set mode //
int     valauto;
int     val;
int     val1;
int     val2;
int     val3;
int     val4;


//*********************************
DHT dht(DHTPIN, DHTTYPE);
BH1750FVI LightSensor(BH1750FVI::k_DevModeContLowRes);

//*********************************

void setup() {

  Serial.begin(115200);
  pinMode(Lamp,OUTPUT);
  pinMode(FAN1,OUTPUT);
  pinMode(FAN2,OUTPUT);
  pinMode(Pump,OUTPUT);
  digitalWrite(Lamp,HIGH); //ให้รีเลย์ไม่ทำงานก่อนเสมอ
  digitalWrite(FAN1,HIGH); //ให้รีเลย์ไม่ทำงานก่อนเสมอ
  digitalWrite(FAN2,HIGH); //ให้รีเลย์ไม่ทำงานก่อนเสมอ
  digitalWrite(Pump,HIGH); //ให้รีเลย์ไม่ทำงานก่อนเสมอ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(WIFI_STA_NAME);

WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");


}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);


//Set Status value//

String s1 = String(valauto);
String s2 = String(val);
String s3 = String(val1);
String s4 = String(val2);
String s5 = String(val3);
String s6 = String(A);
String s7 = String(B);
String s8 = String(C);
String s9 = String(val4);

Firebase.setString("S1_Auto",s1);
Firebase.setString("S1_Manual",s2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Firebase.setString("S2_Lamp",s3);
  Firebase.setString("S3_Fan1",s4);
  Firebase.setString("S4_Fan2",s5);
  Firebase.setString("S5_Pump",s9);
  Firebase.setString("Auto light",s6);
  Firebase.setString("Auto temp",s7);
  Firebase.setString("Auto soil",s8);


  dht.begin();
  LightSensor.begin();
}
void hand()

{
val=Firebase.getString("S1_Manual").toInt();
 if (val==0)
 {
  val1=Firebase.getString("S2_Lamp").toInt();                    //Reading the
value of the varialble Status from the firebase
  if(val1==1)                                                    // If, the Status is 1, turn on
the Relay3
     {
      digitalWrite(Lamp,LOW);
      //Serial.println("light : ON");
      Firebase.setString("Status Lamp","1");
    }
 else if(val1==0)                                               // If, the Status is 0, turn Off
the Relay3
    {
      digitalWrite(Lamp,HIGH);
      //Serial.println("light : OFF");
      Firebase.setString("Status Lamp","0");
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
val2=Firebase.getString("S3_Fan1").toInt();                    //Reading the value of
the varialble Status from the firebase

  if(val2==1)                                                  // If, the Status is 1, turn on
the Relay4
     {
      digitalWrite(FAN1,LOW);
      //Serial.println("FAN1 : ON");
      Firebase.setString("Status Fan1","1");
    }
    else if(val2==0)                                           // If, the Status is 0, turn Off
the Relay4
    {
      digitalWrite(FAN1,HIGH);
      //Serial.println("FAN1 : OFF");
      Firebase.setString("Status Fan1","0");
    }

val3=Firebase.getString("S4_Fan2").toInt();                    //Reading the value of
the varialble Status from the firebase

  if(val3==1)                                                  // If, the Status is 1, turn on
the Relay3
     {
      digitalWrite(FAN2,LOW);
      //Serial.println("FAN2 : ON");
      Firebase.setString("Status Fan2","1");
    }
    else if(val3==0)                                           // If, the Status is 0, turn Off
the Relay3
    {
      digitalWrite(FAN2,HIGH);
      //Serial.println("FAN2 : OFF");
      Firebase.setString("Status Fan2","0");
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
val4=Firebase.getString("S5_Pump").toInt();                    //Reading the value
of the varialble Status from the firebase

  if(val4==1)                                                  // If, the Status is 1, turn on
the Relay3
     {
      digitalWrite(Pump,LOW);
      //Serial.println("Pump : ON");
      Firebase.setString("Status Pump","1");
    }
    else if(val4==0)                                           // If, the Status is 0, turn Off
the Relay3
    {
      digitalWrite(Pump,HIGH);
      //Serial.println("Pump : OFF");
      Firebase.setString("Status Pump","0");
    }

 else if(val==1)
 {
  //Serial.println("Mode Manual :OFF");
  }
  }
}
void automation()

 {
  float h = dht.readHumidity();                                // Reading temperature
or humidity takes about 250 milliseconds!
  float t = dht.readTemperature();                             // Read temperature
as Celsius (the default)
  uint16_t lux = LightSensor.GetLightIntensity();

 CurrentSoilValue2 = analogRead(Pin);
  MapReadSoilValue2 = map(CurrentSoilValue2, 800, 4000, 100, 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
valauto=Firebase.getString("S1_Auto").toInt();

A =  Firebase.getString("Auto light").toInt();
B =  Firebase.getString("Auto temp").toInt();
C =  Firebase.getString("Auto soil").toInt();

if(valauto==1)
{

    if ( lux < A)
      {
      digitalWrite(Lamp,LOW);
      Firebase.setString("Status Lamp","1");

      }
     else {
      digitalWrite(Lamp,HIGH);
      Firebase.setString("Status Lamp","0");
      }

       if ( t > B)
       {
       digitalWrite(FAN1,LOW);
       digitalWrite(FAN2,LOW);
       Firebase.setString("Status Fan1","1");
       Firebase.setString("Status Fan2","1");
       }
      else {
       digitalWrite(FAN1,HIGH);
       digitalWrite(FAN2,HIGH);
       Firebase.setString("Status Fan1","0");
       Firebase.setString("Status Fan2","0");
       }
       if ( MapReadSoilValue2 < C)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          {
          digitalWrite(Pump,LOW);
          Firebase.setString("Status Pump","1");
          }
        else {
          digitalWrite(Pump,HIGH);
          Firebase.setString("Status Pump","0");
          }
  }
    else if (valauto==0)
    {
     //Serial.println("Mode Auto :off");
     }
  }
void readsensor(void){
  float h = dht.readHumidity();                          // Reading temperature
or humidity takes about 250 milliseconds!
  float t = dht.readTemperature();                       // Read temperature
as Celsius (the default)
  uint16_t lux = LightSensor.GetLightIntensity();

  CurrentSoilValue2 = analogRead(Pin);
  MapReadSoilValue2 = map(CurrentSoilValue2, 0, 4095, 100, 0);

  String fireHumid = String(h);
  String fireTemp = String(t);
  String fireLight = String(lux);                  //convert integer humidity to
string humidity
  String fireSoil = String(MapReadSoilValue2);

 Firebase.setString("sensor_hum", fireHumid);            //setup path and send
readings
 Firebase.setString("sensor_temp", fireTemp);            //setup path and send
readings
 Firebase.setString("sensor_light", fireLight);          //setup path and send readings
 Firebase.setString("sensor_soil", fireSoil);            //setup path and send readings
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void loop(){
  //float h = dht.readHumidity();                    // Reading temperature or
humidity takes about 250 milliseconds!
  //float t = dht.readTemperature();                  // Read temperature as
Celsius (the default)
  //uint16_t lux = LightSensor.GetLightIntensity();
    readsensor();
    automation();
    hand();

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คำสั่งสำหรับการเข้าสู่ระบบบนแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
package com.example.testapplication;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.res.Resources;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;
import java.util.regex.Pattern;

public class LoginActivity<onActivityResult> extends AppCompatActivity implements
View.OnClickListener {

    private TextView forgotPassword;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    private EditText editTextemail,editTextpassword;
    private Button signIn;

    private FirebaseAuth mAuth;
    private ProgressBar progressBar;
    private GoogleSignInClient mGoogleSignInClient;
    private final static int RC_SIGN_IN = 123;

    @Override
    protected void onStart() {
        super.onStart();

        FirebaseUser user = mAuth.getCurrentUser();
        if(user!=null){
            Intent intent = new Intent(getApplicationContext(),Display.class);
            startActivity(intent);
        }
    }

    @SuppressLint("WrongViewCast")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //for changing status bar icon color
        if(Build.VERSION.SDK_INT>= Build.VERSION_CODES.LOLLIPOP){

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
            getWindow().setStatusBarColor(getResources().getColor(R.color.cream));
            getWindow().setNavigationBarColor(getResources().getColor(R.color.cream));
        }
        setContentView(R.layout.activity_login);

        signIn =(Button) findViewById(R.id.cirLoginButton);
        signIn.setOnClickListener((View.OnClickListener) this);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        editTextemail = (EditText) findViewById(R.id.editTextEmail);
        editTextpassword = (EditText) findViewById(R.id.editTextPassword);

        progressBar = (ProgressBar) findViewById(R.id.progressBar);

        mAuth = FirebaseAuth.getInstance();

        forgotPassword = (TextView) findViewById(R.id.forgotPassword);
        forgotPassword.setOnClickListener(this);

        createRequest();

        findViewById(R.id.google_button).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                signIn();
            }
        });

        }

    private void createRequest() {

        // Configure Google Sign In
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build();

        // Build a GoogleSignInClient with the options specified by gso.
        mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

    }

    private void signIn() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Intent signInIntent = mGoogleSignInClient.getSignInIntent();
        //noinspection deprecation
        startActivityForResult(signInIntent, RC_SIGN_IN);
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        // Result returned from launching the Intent from
GoogleSignInApi.getSignInIntent(...);
        if (requestCode == RC_SIGN_IN) {
            Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data);
            try {
                // Google Sign In was successful, authenticate with Firebase
                GoogleSignInAccount account = task.getResult(ApiException.class);
                firebaseAuthWithGoogle(account);
            } catch (ApiException e) {
                // Google Sign In failed, update UI appropriately
                Toast.makeText(this, e.getMessage(),Toast.LENGTH_SHORT).show();
            }
        }
    }

    private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {

        AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(),
null);
        mAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            // Sign in success, update UI with the signed-in user's information
                            FirebaseUser user = mAuth.getCurrentUser();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Intent intent = new Intent(getApplicationContext(),Display.class);
                    startActivity(intent);

                } else {
                    Toast.makeText(LoginActivity.this, "Sorry auth failed",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }


    public void onLoginClick(View View){
        startActivity(new Intent(this,RegisterActivity.class));
        overridePendingTransition(R.anim.slide_in_right,R.anim.stay);

    }

    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.cirLoginButton:
                userLogin();
                break;

            case R.id.forgotPassword:
                startActivity(new Intent(this, ForgotPassword.class));
                break;

        }
    }

    private void userLogin() {
        String email = editTextemail.getText().toString().trim();
        String password = editTextpassword.getText().toString().trim();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(email.isEmpty()){
            editTextemail.setError("Email is required!");
            editTextemail.requestFocus();
            return;
        }

        if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
            editTextemail.setError("Please enter a valid email!");
            editTextemail.requestFocus();
            return;
        }

        if (password.isEmpty()){
            editTextpassword.setError("Password is required!");
            editTextpassword.requestFocus();
            return;
        }

        if (password.length() < 6){
            editTextpassword.setError("Min password length is 6 characters!");
            editTextpassword.requestFocus();
            return;
        }

        progressBar.setVisibility(View.VISIBLE);

        mAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()){
                    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

                    if(user.isEmailVerified()) {

                        //redirect to user profile
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    startActivity(new Intent(LoginActivity.this, Display.class));
                    finish();

                }else {
                    user.sendEmailVerification();
                    Toast.makeText(LoginActivity.this,"Check your email to verify your
account!", Toast.LENGTH_LONG).show();
                }

            }else{
                Toast.makeText(LoginActivity.this,"Failed to login! Please check your
credentials", Toast.LENGTH_LONG).show();
            }

        }
    });

}
int counter =0;

@Override
public void onBackPressed() {
    counter++;
    if (counter == 1000)
        super.onBackPressed();
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

คำสั่งสำหรับลงทะเบียนผู้ใช้บนแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
package com.example.testapplication;

import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.FirebaseDatabase;

public class RegisterActivity extends AppCompatActivity implements
View.OnClickListener {

    private TextView banner, registerUser;
    private EditText editTextname, editTextemail, editTextmobile, editTextpassword;
    private ProgressBar progressBar;

    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_register);
        changeStatusBarColor();

        mAuth = FirebaseAuth.getInstance();

        banner = (TextView) findViewById(R.id.banner);
        banner.setOnClickListener(this);

        registerUser = (Button) findViewById(R.id.cirRegisterButton);
        registerUser.setOnClickListener(this);

        editTextname = (EditText) findViewById(R.id.editTextName);
        editTextemail = (EditText) findViewById(R.id.editTextEmail);
        editTextmobile = (EditText) findViewById(R.id.editTextMobile);
        editTextpassword = (EditText) findViewById(R.id.editTextPassword);

        progressBar = (ProgressBar) findViewById(R.id.progressBar);

    }

    private void changeStatusBarColor() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            Window window = getWindow();

window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGR
OUNDS);
//              window.setStatusBarColor(Color.TRANSPARENT);
            window.setStatusBarColor(getResources().getColor(R.color.colorAccent));

getWindow().setNavigationBarColor(getResources().getColor(R.color.colorAccent));
        }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public void onLoginClick(View view){
    startActivity(new Intent(this,LoginActivity.class));
    overridePendingTransition(R.anim.slide_in_left,android.R.anim.slide_out_right);

}

@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.banner:
            startActivity(new Intent(this,LoginActivity.class));
            break;
        case R.id.cirRegisterButton:
            registerUser();
            break;

    }

}

private void registerUser() {
    String email= editTextemail.getText().toString().trim();
    String password= editTextpassword.getText().toString().trim();
    String name= editTextname.getText().toString().trim();
    String mobilenumber= editTextmobile.getText().toString().trim();

    if(name.isEmpty()){
        editTextname.setError("Name is required!");
        editTextname.requestFocus();
        return;
    }

    if(email.isEmpty()){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    editTextemail.setError("Email is required!");
    editTextemail.requestFocus();
    return;
}

if(mobilenumber.isEmpty()){
    editTextmobile.setError("Mobile Number is required!");
    editTextmobile.requestFocus();
    return;
}

if(password.isEmpty()){
    editTextpassword.setError("Password is required!");
    editTextpassword.requestFocus();
    return;
}

if(password.length() < 6){
    editTextpassword.setError("Min password length should be 6 characters!");
    editTextpassword.requestFocus();
    return;
}

progressBar.setVisibility(View.VISIBLE);
mAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                if (task.isSuccessful()) {
                    User user = new User(name, mobilenumber, email);

                    FirebaseDatabase.getInstance().getReference("User")
                            .child(name)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        .setValue(user).addOnCompleteListener(new
OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {

                        if (task.isSuccessful()) {
                            Toast.makeText(RegisterActivity.this, "User has been
registeres successfully!", Toast.LENGTH_LONG).show();
                            progressBar.setVisibility(View.GONE);
                            //redirect to login layout!
                        } else {
                            Toast.makeText(RegisterActivity.this, "Failed to register!
Try again!", Toast.LENGTH_LONG).show();
                            progressBar.setVisibility(View.GONE);
                        }
                    }
                });
            }else{
                Toast.makeText(RegisterActivity.this, "Failed to register!",
Toast.LENGTH_LONG).show();
                progressBar.setVisibility(View.GONE);

            }
        }
    });

}
int counter =0;

@Override
public void onBackPressed() {
    counter++;
    if (counter == 1)
        startActivity(new Intent(getApplicationContext(),LoginActivity.class));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        super.onBackPressed();
    }

}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

คำสั่งกรณีผู้ใช้ลืมรหัสผ่านบนแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
package com.example.testapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Build;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;

import java.util.regex.Pattern;

public class ForgotPassword extends AppCompatActivity {

    private EditText emailEditText;
    private Button resetPasswordButton;
    private ProgressBar progressBar;

    FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //for changing status bar icon color
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        setContentView(R.layout.activity_forgot_password);
        changeStatusBarColor();

        emailEditText = (EditText) findViewById(R.id.Email);
        resetPasswordButton=(Button) findViewById(R.id.resetPassword);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);

        auth = FirebaseAuth.getInstance();

        resetPasswordButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                resetPassword();
            }
        });

    }

    private void changeStatusBarColor() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            Window window = getWindow();

window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGR
OUNDS);
//            window.setStatusBarColor(Color.TRANSPARENT);
            window.setStatusBarColor(getResources().getColor(R.color.colorAccent));
        }
    }

    private void resetPassword(){
        String email = emailEditText.getText().toString().trim();

        if (email.isEmpty()){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            emailEditText.setError("Email is required!");
            emailEditText.requestFocus();
            return;
        }

        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
            emailEditText.setError("Please provide valid email!");
            emailEditText.requestFocus();
            return;
        }

        progressBar.setVisibility(View.VISIBLE);
        auth.sendPasswordResetEmail(email).addOnCompleteListener(new
OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {

                if (task.isSuccessful()){
                    Toast.makeText(ForgotPassword.this,"Check your email to reset your
password!",Toast.LENGTH_LONG).show();

                }else{
                    Toast.makeText(ForgotPassword.this,"Try again! Something wrong
happened!",Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

คำสั่งแสดงสถานะการทำงานของอุปกรณ์บนแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
package com.example.testapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import org.jetbrains.annotations.NotNull;

import java.util.Objects;

public class Display extends AppCompatActivity {

    BottomNavigationView bottomNavigationView;

    private TextView temp;
    private TextView light;
    private TextView Humid;
    private TextView soil;
    private ImageView Led;
    private ImageView led_status;
    private ImageView fan1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    private ImageView fan1_status;
    private ImageView fan2;
    private ImageView fan2_status;
    private ImageView pumps;
    private ImageView pump_status;

    DatabaseReference dref;
    DatabaseReference dref2;
    DatabaseReference dref3;
    DatabaseReference dref4;
    DatabaseReference dled;
    DatabaseReference dfan1;
    DatabaseReference dfan2;
    DatabaseReference dpump;

    String statusTemp;
    String statuslight;
    String statusHumi;
    String statussoil;
    String status_Led;
    String status_fan1;
    String status_fan2;
    String status_pump;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if(Build.VERSION.SDK_INT>= Build.VERSION_CODES.LOLLIPOP){

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
            getWindow().setStatusBarColor(getResources().getColor(R.color.colorAccent));
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        setContentView(R.layout.activity_display);

        bottomNavigationView =findViewById(R.id.bottomNavigationView);
        bottomNavigationView.setSelectedItemId(R.id.Display);

        //noinspection deprecation
        bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull @NotNull MenuItem
item) {

                switch (item.getItemId()){
                    case R.id.Display:
                        return true;
                    case R.id.Controll:
                        startActivity(new Intent(getApplicationContext(),control.class));
                        overridePendingTransition(0,0);
                        return true;
                    case R.id.Guide:
                        startActivity(new Intent(getApplicationContext(),Guidebook.class));
                        overridePendingTransition(0,0);
                        return true;
                    case R.id.Logoutitem:
                        startActivity(new Intent(getApplicationContext(),ProfileActivity.class));
                        overridePendingTransition(0,0);
                        return true;
                }

                return false;

            }
        });
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp = (TextView) findViewById(R.id.Temp);
light = (TextView) findViewById(R.id.light);
Humid = (TextView) findViewById(R.id.airHu);
soil = (TextView) findViewById(R.id.soil);
Led = (ImageView) findViewById(R.id.Led);
led_status = (ImageView) findViewById(R.id.Led_status);
fan1 = (ImageView) findViewById(R.id.fan1);
fan1_status = (ImageView) findViewById(R.id.fan1_status);
fan2 = (ImageView) findViewById(R.id.fan2);
fan2_status = (ImageView) findViewById(R.id.fan2_status);
pumps = (ImageView) findViewById(R.id.pumps);
pump_status = (ImageView) findViewById(R.id.pump_status);

dref = FirebaseDatabase.getInstance().getReference();
dref.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        statusTemp = snapshot.child("sensor_temp").getValue().toString();
        temp.setText(statusTemp);
    }

    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});

dref2 = FirebaseDatabase.getInstance().getReference();
dref2.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        statuslight = snapshot.child("sensor_light").getValue().toString();
        light.setText(statuslight);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});

dref3 = FirebaseDatabase.getInstance().getReference();
dref3.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        statusHumi = snapshot.child("sensor_hum").getValue().toString();
        Humid.setText(statusHumi);
    }

    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});

dref4 = FirebaseDatabase.getInstance().getReference();
dref4.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        statussoil = snapshot.child("sensor_soil").getValue().toString();
        soil.setText(statussoil);
    }

    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dled = FirebaseDatabase.getInstance().getReference();
dled.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        status_Led = snapshot.child("Status Lamp").getValue().toString();
        if (status_Led.equals("1")) {
            Led.setImageResource(R.drawable.led_light);
            led_status.setImageResource(R.drawable.green);

        } else if (status_Led.equals("0")) {
            Led.setImageResource(R.drawable.led_lightoff);
            led_status.setImageResource(R.drawable.red);
        }
    }

    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});

dfan1 = FirebaseDatabase.getInstance().getReference();
dfan1.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        status_fan1 = snapshot.child("Status Fan1").getValue().toString();
        if (status_fan1.equals("1")) {
            fan1.setImageResource(R.drawable.fan_2on);
            fan1_status.setImageResource(R.drawable.green);

        } else if (status_fan1.equals("0")) {
            fan1.setImageResource(R.drawable.fan_2off);
            fan1_status.setImageResource(R.drawable.red);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    }

    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});

dfan2 = FirebaseDatabase.getInstance().getReference();
dfan2.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        status_fan2 = snapshot.child("Status Fan2").getValue().toString();
        if (status_fan2.equals("1")) {
            fan2.setImageResource(R.drawable.fan_1on);
            fan2_status.setImageResource(R.drawable.green);

        } else if (status_fan2.equals("0")) {
            fan2.setImageResource(R.drawable.fan_1off);
            fan2_status.setImageResource(R.drawable.red);
        }
    }

    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});

dpump = FirebaseDatabase.getInstance().getReference();
dpump.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                status_pump = snapshot.child("Status Pump").getValue().toString();
                if (status_pump.equals("1")) {
                    pumps.setImageResource(R.drawable.pumpon);
                    pump_status.setImageResource(R.drawable.green);

                } else if (status_pump.equals("0")) {
                    pumps.setImageResource(R.drawable.pumpoff);
                    pump_status.setImageResource(R.drawable.red);
                }
            }

            @Override
            public void onCancelled(@NonNull @NotNull DatabaseError error) {

            }
        });

    }
    int counter =0;

    @Override
    public void onBackPressed() {
        counter++;
        if (counter == 1000)
            super.onBackPressed();
    }

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ

คำสั่งควบคุมระบบการทำงานของอุปกรณ์บนแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
package com.example.testapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.drawable.TransitionDrawable;
import android.media.Image;
import android.os.Build;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.Switch;
import android.widget.TextView;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import org.jetbrains.annotations.NotNull;

public class control extends AppCompatActivity {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BottomNavigationView bottomNavigationView;

boolean check = false;

private boolean lampcheck = true;
private boolean fan2check = true;
private boolean fan1check = true;
private boolean pumpcheck = true;
private boolean lamp_see = true;
private boolean fan_see = true;
private boolean pump_see = true;

private TextView seeled;
private TextView seefan;
private TextView seepump;
private TextView paraligh;
private TextView paratemp;
private TextView parasoil;

private SeekBar light_seek;
private SeekBar fan_seek;
private SeekBar pump_seek;

ImageView Lamptran;
ImageView fan22;
ImageView fan11;
ImageView pumptran1;

@SuppressLint("UseSwitchCompatOrMaterialCode")
Switch Sw_automode;
@SuppressLint("UseSwitchCompatOrMaterialCode")
Switch Sw_manmode;
@SuppressLint("UseSwitchCompatOrMaterialCode")
Switch switch_lamp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
@SuppressLint("UseSwitchCompatOrMaterialCode")
Switch switch_fan2;
@SuppressLint("UseSwitchCompatOrMaterialCode")
Switch switch_fan1;
@SuppressLint("UseSwitchCompatOrMaterialCode")
Switch switch_pump;

DatabaseReference dseek_led;
DatabaseReference dseek_fan;
DatabaseReference dseek_pump;
DatabaseReference dlight;
DatabaseReference dtemp;
DatabaseReference dsoil;
DatabaseReference dlamp;
DatabaseReference dfan2;
DatabaseReference dfan1;
DatabaseReference dpump;
DatabaseReference dswauto;
DatabaseReference dswmanual;

String sta_seekled;
String sta_seekfan;
String sta_seekpump;
String statuslight;
String statustemp;
String statussoil;
String sta_lamp;
String sta_fan2;
String sta_fan1;
String sta_pump;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(Build.VERSION.SDK_INT>= Build.VERSION_CODES.LOLLIPOP){

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
            getWindow().setStatusBarColor(getResources().getColor(R.color.colorAccent));
        }
        setContentView(R.layout.activity_control);

        bottomNavigationView = findViewById(R.id.bottomNavigationView);
        bottomNavigationView.setSelectedItemId(R.id.Controll);

        //noinspection deprecation
        bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull @NotNull MenuItem
item) {

                switch (item.getItemId()){
                    case R.id.Display:
                        startActivity(new Intent(getApplicationContext(),Display.class));
                        overridePendingTransition(0,0);
                        return true;
                    case R.id.Controll:
                        return true;
                    case R.id.Guide:
                        startActivity(new Intent(getApplicationContext(),Guidebook.class));
                        overridePendingTransition(0,0);
                        return true;
                    case R.id.Logoutitem:
                        startActivity(new Intent(getApplicationContext(),ProfileActivity.class));
                        overridePendingTransition(0,0);
                        return true;
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return false;

        }

    });

    light_seek = (SeekBar) findViewById(R.id.seek_light);
    fan_seek = (SeekBar) findViewById(R.id.seek_temp);
    pump_seek = (SeekBar) findViewById(R.id.seek_soil);

    seeled = (TextView) findViewById(R.id.textLed);
    seefan = (TextView) findViewById(R.id.textfanopen);
    seepump = (TextView) findViewById(R.id.textpumpopen);
    paraligh = (TextView) findViewById(R.id.textled1);
    paratemp = (TextView) findViewById(R.id.textfan1);
    parasoil = (TextView) findViewById(R.id.textpump);

    Sw_automode = (Switch) findViewById(R.id.switch_auto);
    Sw_manmode = (Switch) findViewById(R.id.switch_ManualMode);
    switch_lamp = (Switch) findViewById(R.id.switchlight);
    switch_fan2 = (Switch) findViewById(R.id.switchfan2);
    switch_fan1 = (Switch) findViewById(R.id.switchfan1);
    switch_pump = (Switch) findViewById(R.id.switchpump);

    Lamptran = (ImageView) findViewById(R.id.imageView6);
    fan11 = (ImageView) findViewById(R.id.fan1);
    fan22 = (ImageView) findViewById(R.id.fan2);
    pumptran1 = (ImageView) findViewById(R.id.pumptran);

    //seek light
    light_seek.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()
{
        @SuppressLint("SetTextI18n")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {
            seeled.setText(progress + " Lux");
            sta_seekled = String.valueOf(progress);
            FirebaseDatabase database = FirebaseDatabase.getInstance();
            dseek_led = database.getReference().child("Auto light");
            dseek_led.setValue(sta_seekled);

        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });

    dseek_led = FirebaseDatabase.getInstance().getReference();
    dseek_led.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
            String a = snapshot.child("Auto light").getValue(String.class);

            int i = Integer.parseInt(a.replaceAll("[\\D]", ""));
            if (lamp_see) {
                light_seek.setProgress(i);
                lamp_see = false;
            }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        @Override
        public void onCancelled(@NonNull @NotNull DatabaseError error) {

        }
    });

    //seek fan
    fan_seek.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @SuppressLint("SetTextI18n")
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {
            seefan.setText(progress + " °C");
            sta_seekfan = String.valueOf(progress);
            FirebaseDatabase database = FirebaseDatabase.getInstance();
            dseek_fan = database.getReference().child("Auto temp");
            dseek_fan.setValue(sta_seekfan);

        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });

    dseek_fan = FirebaseDatabase.getInstance().getReference();
    dseek_fan.addValueEventListener(new ValueEventListener() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            @Override
            public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
                String a = snapshot.child("Auto temp").getValue(String.class);

                int i = Integer.parseInt(a.replaceAll("[\\D]", ""));
                if (fan_see) {
                    fan_seek.setProgress(i);
                    fan_see = false;
                }
            }

            @Override
            public void onCancelled(@NonNull @NotNull DatabaseError error) {

            }
        });

        //seek pump
        pump_seek.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
            @SuppressLint("SetTextI18n")
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {
                seepump.setText(progress + " %");
                sta_seekpump = String.valueOf(progress);
                FirebaseDatabase database = FirebaseDatabase.getInstance();
                dseek_pump = database.getReference().child("Auto soil");
                dseek_pump.setValue(sta_seekpump);

            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }
});

dseek_pump = FirebaseDatabase.getInstance().getReference();
dseek_pump.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        String a = snapshot.child("Auto soil").getValue(String.class);

        int i = Integer.parseInt(a.replaceAll("[\\D]", ""));
        if (pump_see) {
            pump_seek.setProgress(i);
            pump_see = false;
        }
    }

    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});

//parameter
dlight = FirebaseDatabase.getInstance().getReference();
dlight.addValueEventListener(new ValueEventListener() {
    @SuppressLint("SetTextI18n")
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            statuslight = snapshot.child("sensor_light").getValue(String.class);
            paraligh.setText("Light: " + statuslight + " Lux");

        }

        @Override
        public void onCancelled(@NonNull @NotNull DatabaseError error) {

        }
    });

    dtemp = FirebaseDatabase.getInstance().getReference();
    dtemp.addValueEventListener(new ValueEventListener() {
        @SuppressLint("SetTextI18n")
        @Override
        public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
            statustemp = snapshot.child("sensor_temp").getValue(String.class);
            paratemp.setText("Temp: " + statustemp + " °C");

        }

        @Override
        public void onCancelled(@NonNull @NotNull DatabaseError error) {

        }
    });

    dsoil = FirebaseDatabase.getInstance().getReference();
    dsoil.addValueEventListener(new ValueEventListener() {
        @SuppressLint("SetTextI18n")
        @Override
        public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
            statussoil = snapshot.child("sensor_soil").getValue(String.class);
            parasoil.setText("Soil: " + statussoil + " %");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }

            @Override
            public void onCancelled(@NonNull @NotNull DatabaseError error) {

            }
        });

        //Manaul Lamp
        switch_lamp.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                TransitionDrawable drawable = (TransitionDrawable)
Lamptran.getDrawable();

                if (isChecked) {
                    FirebaseDatabase database = FirebaseDatabase.getInstance();
                    DatabaseReference lamp = database.getReference("S2_Lamp");
                    lamp.setValue("1");
                    drawable.startTransition(200);
                } else {
                    FirebaseDatabase database = FirebaseDatabase.getInstance();
                    DatabaseReference lamp = database.getReference("S2_Lamp");
                    lamp.setValue("0");
                    drawable.reverseTransition(200);

                }
            }
        });

        dlamp = FirebaseDatabase.getInstance().getReference();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dlamp.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
                sta_lamp = snapshot.child("S2_Lamp").getValue(String.class);
                if(lampcheck){
                    if (sta_lamp.equals("1")) {
                        switch_lamp.setChecked(true);

                    } else if (sta_lamp.equals("0")) {
                        switch_lamp.setChecked(false);
                    }
                    lampcheck = false;
                }
            }

            @Override
            public void onCancelled(@NonNull @NotNull DatabaseError error) {

            }
        });

        //Manual Fan1
        switch_fan1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                TransitionDrawable drawable = (TransitionDrawable) fan11.getDrawable();

                if (isChecked) {
                    FirebaseDatabase database = FirebaseDatabase.getInstance();
                    DatabaseReference fan1 = database.getReference("S3_Fan1");
                    fan1.setValue("1");
                    drawable.startTransition(200);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        } else {
            FirebaseDatabase database = FirebaseDatabase.getInstance();
            DatabaseReference fan1 = database.getReference("S3_Fan1");
            fan1.setValue("0");
            drawable.reverseTransition(200);

        }
    }
});

dfan1 = FirebaseDatabase.getInstance().getReference();
dfan1.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        sta_fan1 = snapshot.child("S3_Fan1").getValue(String.class);
        if(fan1check){
            if (sta_fan1.equals("1")) {
                switch_fan1.setChecked(true);

            } else if (sta_fan1.equals("0")) {
                switch_fan1.setChecked(false);
            }
            fan1check = false;
        }
    }

    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {

    }
});

//Manual Fan2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        switch_fan2.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                TransitionDrawable drawable = (TransitionDrawable) fan22.getDrawable();

                if (isChecked) {
                    FirebaseDatabase database = FirebaseDatabase.getInstance();
                    DatabaseReference fan2 = database.getReference("S4_Fan2");
                    fan2.setValue("1");
                    drawable.startTransition(200);
                } else {
                    FirebaseDatabase database = FirebaseDatabase.getInstance();
                    DatabaseReference fan2 = database.getReference("S4_Fan2");
                    fan2.setValue("0");
                    drawable.reverseTransition(200);

                }
            }
        });

        dfan2 = FirebaseDatabase.getInstance().getReference();
        dfan2.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
                sta_fan2 = snapshot.child("S4_Fan2").getValue(String.class);
                if(fan2check){
                    if (sta_fan2.equals("1")) {
                        switch_fan2.setChecked(true);

                    } else if (sta_fan2.equals("0")) {
                        switch_fan2.setChecked(false);
                    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                fan2check = false;
            }
        }

        @Override
        public void onCancelled(@NonNull @NotNull DatabaseError error) {

        }
    });

    //Manual Pump
    switch_pump.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            TransitionDrawable drawable = (TransitionDrawable)
pumptran1.getDrawable();

            if (isChecked) {
                FirebaseDatabase database = FirebaseDatabase.getInstance();
                DatabaseReference pump1 = database.getReference("S5_Pump");
                pump1.setValue("1");
                drawable.startTransition(200);
            } else {
                FirebaseDatabase database = FirebaseDatabase.getInstance();
                DatabaseReference pump1 = database.getReference("S5_Pump");
                pump1.setValue("0");
                drawable.reverseTransition(200);

            }
        }
    });
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dpump = FirebaseDatabase.getInstance().getReference();
        dpump.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
                sta_pump = snapshot.child("S5_Pump").getValue(String.class);
                if(pumpcheck){
                    if (sta_pump.equals("1")) {
                        switch_pump.setChecked(true);

                    } else if (sta_pump.equals("0")) {
                        switch_pump.setChecked(false);
                    }
                    pumpcheck = false;
                }
            }

            @Override
            public void onCancelled(@NonNull @NotNull DatabaseError error) {

            }
        });

        //Sw Auto
        Sw_automode.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                if (isChecked) {
                    FirebaseDatabase database = FirebaseDatabase.getInstance();
                    DatabaseReference myref = database.getReference("S1_Auto");
                    myref.setValue("1");
                } else {
                    FirebaseDatabase database = FirebaseDatabase.getInstance();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DatabaseReference myref = database.getReference("S1_Auto");
                myref.setValue("0");
            }
        }
    });

    dswauto = FirebaseDatabase.getInstance().getReference();
    dswauto.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
            String value = snapshot.child("S1_Auto").getValue(String.class);
            if (value.equals("1")) {
                Sw_automode.setChecked(true);
            } else if (value.equals("0")) {
                Sw_automode.setChecked(false);
            }
        }

        @Override
        public void onCancelled(@NonNull @NotNull DatabaseError error) {

        }
    });

    //Sw Manual
    Sw_manmode.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            if (isChecked) {
                FirebaseDatabase database = FirebaseDatabase.getInstance();
                DatabaseReference myref = database.getReference("S1_Manual");
                myref.setValue("0");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            } else {
                FirebaseDatabase database = FirebaseDatabase.getInstance();
                DatabaseReference myref = database.getReference("S1_Manual");
                myref.setValue("1");
            }
        }
    });

    //Sw Manual
    dswmanual = FirebaseDatabase.getInstance().getReference();
    dswmanual.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
            String value = snapshot.child("S1_Manual").getValue(String.class);
            if (value.equals("0")) {
                Sw_manmode.setChecked(true);
            } else if (value.equals("1")) {
                Sw_manmode.setChecked(false);
            }
        }
        @Override
        public void onCancelled(@NonNull @NotNull DatabaseError error) {

        }
    });

    return;

}


int counter =0;

@Override
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    public void onBackPressed() {
        counter++;
        if (counter == 1)
            startActivity(new Intent(getApplicationContext(),Display.class));
            super.onBackPressed();
    }

}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ช

คำสั่งสำหรับแสดงคู่มือการใช้งานหน้าควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
package com.example.testapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;

import com.google.android.material.bottomnavigation.BottomNavigationView;

import org.jetbrains.annotations.NotNull;

public class Guidebook extends AppCompatActivity {

    BottomNavigationView bottomNavigationView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if(Build.VERSION.SDK_INT>= Build.VERSION_CODES.LOLLIPOP){

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
            getWindow().setStatusBarColor(getResources().getColor(R.color.colorAccent));
        }

        setContentView(R.layout.activity_guidebook);

        bottomNavigationView = findViewById(R.id.bottomNavigationView);
        bottomNavigationView.setSelectedItemId(R.id.Guide);

        bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            @Override
            public boolean onNavigationItemSelected(@NonNull @NotNull MenuItem
item) {

                switch (item.getItemId()){
                    case R.id.Display:
                        startActivity(new Intent(getApplicationContext(),Display.class));
                        overridePendingTransition(0,0);
                        return true;
                    case R.id.Controll:
                        startActivity(new Intent(getApplicationContext(),control.class));
                        overridePendingTransition(0,0);
                        return true;
                    case R.id.Guide:
                        return true;
                    case R.id.Logoutitem:
                        startActivity(new Intent(getApplicationContext(),ProfileActivity.class));
                        overridePendingTransition(0,0);
                        return true;
                }

                return false;
            }
        });


    }
    int counter =0;

    @Override
    public void onBackPressed() {
        counter++;
        if (counter == 1000)
            super.onBackPressed();
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ซ

คำสั่งสำหรับออกจากระบบบนแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
package com.example.testapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.NavigationUI;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.firebase.auth.FirebaseAuth;

import org.jetbrains.annotations.NotNull;

public class ProfileActivity extends AppCompatActivity {

    private Button logout;
    FirebaseAuth mAuth;

    GoogleSignInClient mGoogle;
    public static int RC_SIGN_IN = 1;

    BottomNavigationView bottomNavigationView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        super.onCreate(savedInstanceState);
        if(Build.VERSION.SDK_INT>= Build.VERSION_CODES.LOLLIPOP){

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
            getWindow().setStatusBarColor(getResources().getColor(R.color.colorAccent));
        }

        setContentView(R.layout.activity_profile);

        bottomNavigationView = findViewById(R.id.bottomNavigationView);
        bottomNavigationView.setSelectedItemId(R.id.Logoutitem);

        //noinspection deprecation
        bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull @NotNull MenuItem
item) {

            switch (item.getItemId()){
                case R.id.Display:
                    startActivity(new Intent(getApplicationContext(),Display.class));
                    overridePendingTransition(0,0);
                    return true;
                case R.id.Controll:
                    startActivity(new Intent(getApplicationContext(),control.class));
                    overridePendingTransition(0,0);
                    return true;
                case R.id.Guide:
                    startActivity(new Intent(getApplicationContext(),Guidebook.class));
                    overridePendingTransition(0,0);
                    return true;
                case R.id.Logoutitem:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                return true;
            }

            return false;

        }
    });

    mAuth = FirebaseAuth.getInstance();

    GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestIdToken(getString(R.string.default_web_client_id))
            .requestEmail()
            .build();

    // Build a GoogleSignInClient with the options specified by gso.
    mGoogle = GoogleSignIn.getClient(ProfileActivity.this, gso);

    logout = (Button) findViewById(R.id.Logout);

    logout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            mGoogle.signOut();
            mAuth.signOut();

            FirebaseAuth.getInstance().signOut();
            startActivity(new Intent(ProfileActivity.this, LoginActivity.class));
            finish();
        }
    });
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int counter =0;

    @Override
    public void onBackPressed() {
        counter++;
        if (counter == 1)
            startActivity(new Intent(getApplicationContext(),Display.class));
        super.onBackPressed();
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้