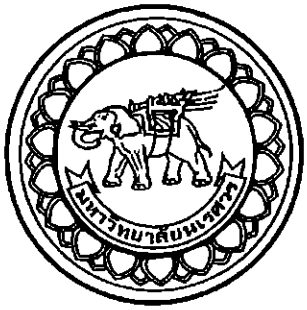


ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์

AUTOMATIC DOOR LOCK SYSTEM CONTROLLED VIA COMPUTER

นายเชาวลิต บุตรอ่ำ รหัส 53362624

นายหนึ่ง เอกปัตชา รหัส 53363188

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 19 พ.ค. 57
เลขทะเบียน 1656483
เลขเรียกหนังสือ ม5
มหาวิทยาลัยนเรศวร ร 741 ม 2556

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2556

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	**ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์**	
ผู้ดำเนินโครงงาน	นายเชาวลิต บุตรอ่ำ	รหัส 53362624
	นายหนึ่ง เอกปัตชา	รหัส 53363188
ที่ปรึกษาโครงงาน	ดร. มุฑิตา สงฆ์จันทร์	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2556	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

..... ที่ปรึกษาโครงงาน

(ดร. มุฑิตา สงฆ์จันทร์)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศุภวรรณ พลพิทักษ์ชัย)

..... กรรมการ

(อาจารย์ เศรษฐา ตั้งค้าวานิช)

ชื่อหัวข้อโครงงาน	ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์	
ผู้ดำเนินโครงงาน	นายเชาวลิต บุตรอ่ำ	รหัส 53362624
	นายหนึ่ง เอกปัตชา	รหัส 53363188
ที่ปรึกษาโครงงาน	ดร. มุฑิตา สงฆ์จันทร์	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2556	

.....

บทคัดย่อ

โครงงานนี้เป็นการสร้างระบบล็อกประตูแบบอัตโนมัติ โดยใช้ไมโครคอนโทรลเลอร์รับคำสั่งจากระบบควบคุมแล้วสั่งการตัวล็อกแบบแม่เหล็กไฟฟ้าให้ทำงาน ซึ่งสามารถควบคุมได้สองทาง คือ ควบคุมด้วยมือโดยใช้สวิตช์ปุ่มกด และควบคุมด้วยคอมพิวเตอร์โดยใช้หน้าต่างโปรแกรม Door lock system คำสั่งต่าง ๆ ที่เลือกใช้งานจะไปปรากฏผ่านหน้าจอแสดงผลแอลซีดีที่ติดอยู่บริเวณหน้าห้อง ทำให้ผู้ใช้งานหรือผู้มาติดต่อทราบสถานะการทำงานของระบบ

Project title	Automatic Door Lock System Controlled via Computer
Name	Mr. Chaowalit Boot-um ID. 53362624
	Mr. Nueng Eakpatcha ID. 53363188
Project advisor	Ms. Mutita Songjun, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2013

Abstract

This project is about constructing the model of the automatic door system controlled by microcontroller. The operation of the door lock system is designed in 2 mode. The first mode is manual system which can be controlled by manual switches. The other mode is controlled by computer via visual basic program. The status of the door lock can also be displayed on LCD screen installed in front of the door.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาเป็นอย่างยิ่งจาก ดร.มุฑิตา สงฆ์จันทร์ ซึ่งเป็นที่ปรึกษาโครงงานและให้ความกรุณาในการเอาใจใส่ในรายละเอียดในการสร้างชิ้นงานและตรวจทานปริญญานิพนธ์รวมถึงการให้คำแนะนำและให้คำปรึกษาในการแก้ไขปัญหาตลอดการดำเนินงาน คณะผู้ดำเนินโครงการขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบพระคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับผู้ดำเนินโครงงาน ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ศุภวรรณ พลพิทักษ์ชัย และอาจารย์เศรษฐา ตั้งค้าวานิช ที่ให้เกียรติเป็นกรรมการโครงงาน นอกจากนี้ยังขอกราบขอบพระคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ให้ความสะดวกในการยืมอุปกรณ์เครื่องมือวัดและเครื่องมือในการทำงานต่างๆ มาใช้งาน ขอขอบคุณเพื่อนทุกๆคนและบุคคลท่านอื่นๆที่มีส่วนร่วมในการให้คำแนะนำและการช่วยเหลือจนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณกองทุนกู้ยืมเพื่อการศึกษา (กยศ.) ที่มอบทุนแก่ผู้ดำเนินโครงงาน ให้ได้มีโอกาสศึกษาจนประสบความสำเร็จในระดับปริญญาตรี เหนือสิ่งใดคณะผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดามารดา ผู้มอบความรัก ความเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอขอบคุณทุกๆคนในครอบครัวของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นายเชาวลิต บุตรอ่ำ

นายหนึ่ง เอกปัตชา

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก

บทคัดย่อภาษาไทย ..... ข

บทคัดย่อภาษาอังกฤษ ..... ค

กิตติกรรมประกาศ ..... ง

สารบัญ ..... จ

สารบัญตาราง ..... ซ

สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

1.5 ขั้นตอนและแผนการดำเนินโครงงาน 3

1.6 งบประมาณที่ใช้ในการดำเนินโครงงาน 3

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 4

2.1 บ้านอัจฉริยะ (Smart Home) 4

2.2 ไมโครคอนโทรลเลอร์ 6

2.2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 6

2.2.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 6

2.2.3 การเขียนโปรแกรมไมโครคอนโทรลเลอร์ 7

2.2.4 รูปแบบการทำงานของขาไมโครคอนโทรลเลอร์หมายเลข P89V51RD2 8

2.3 แม่เหล็กล็อกประตู 11

2.3.1 ทฤษฎีสนามแม่เหล็กไฟฟ้า 11

สารบัญ (ต่อ)

หน้า

2.4 โปรแกรม Microsoft Visual Basic 13
2.4.1 เริ่มต้นใช้งานโปรแกรม Microsoft Visual Basic 14
2.5 รีเลย์ (Relay) 15
2.5.1 หน้าที่และหลักการทำงานของรีเลย์ 15
2.5.2 ชนิดของรีเลย์แบ่งตามลักษณะการใช้งาน 16
2.6 จอแสดงผลแอลซีดี 17
2.7 หม้อแปลงไฟฟ้า 18
2.7.1 ชนิดของหม้อแปลง 19
2.7.2 ข้อควรระวังในการใช้งานหม้อแปลง 21

บทที่ 3 วิธีการดำเนินโครงงาน 22
3.1 โครงสร้างของประตูจำลอง 23
3.2 วงจรควบคุมการทำงาน 24
3.2.1 วงจรไมโครคอนโทรลเลอร์ 25
3.3 ระบบสั่งการ 29
3.3.1 ระบบสั่งการควบคุมด้วยมือ (MANUAL) 30
3.3.2 ระบบสั่งการด้วยคอมพิวเตอร์ (COMPUTER) 33

บทที่ 4 ผลการทดสอบและผลการวิเคราะห์ 39
4.1 การทดสอบการทำงานแบบสั่งการจากระบบควบคุมด้วยมือ 39
4.2 การทดสอบการทำงานแบบสั่งการผ่านคอมพิวเตอร์ 50
4.3 การทดสอบแรงยึดของตัวล็อกแม่เหล็กที่ระดับแรงดันไฟฟ้า 12 โวลต์ 55
4.4 การทดสอบวัดพิกัดแรงดันไฟฟ้าของสายตัวนำเทียบกับระยะทาง 57
4.5 การทดสอบการใช้งานระบบควบคุมด้วยมือต่อเนื่องเป็นเวลา 48 ชั่วโมง 58
4.6 การทดสอบการใช้งานระบบควบคุมผ่านคอมพิวเตอร์ต่อเนื่องเป็นเวลา 8 ชั่วโมง 59

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลและข้อเสนอแนะ 60

5.1 สรุปผลการดำเนินโครงงาน 60

5.2 ปัญหาและการแก้ไข 61

5.3 ข้อเสนอแนะและแนวทางในการพัฒนาต่อไป 61

เอกสารอ้างอิง 62

ภาคผนวก ก โปรแกรมการทำงานของระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ 63

ภาคผนวก ข รายละเอียดของไมโครคอนโทรลเลอร์ MCS-51 หมายเลข P89V51RD2 83

ภาคผนวก ค รายละเอียดของไอซีหมายเลข Max 232 91

ภาคผนวก ง คู่มือการใช้งานระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ 94

ประวัติผู้ดำเนินโครงงาน 100

สารบัญตาราง

ตารางที่ หน้า

2.1 รายละเอียดการทำงานแต่ละขาของไมโครคอนโทรลเลอร์หมายเลข P89V51RD2 9

4.1 ผลทดสอบความสามารถในการรับน้ำหนักในการควบคุมระบบ 55

4.2 ผลทดสอบวัดพิกัดแรงดันไฟฟ้าของสายตัวนำเทียบกับระยะทางในการควบคุมระบบ..... 57

4.3 ผลทดสอบการใช้งานระบบควบคุมด้วยมือต่อเนื่องเป็นเวลา 48 ชั่วโมง 58

4.4 ผลทดสอบการใช้งานระบบควบคุมผ่านคอมพิวเตอร์ต่อเนื่องเป็นเวลา 8 ชั่วโมง..... 59

สารบัญรูป

รูปที่ หน้า

2.1 แสดงระบบบ้านอัตโนมัติ 5
2.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 7
2.3 รูปแบบการทำงานของขาไมโครคอนโทรลเลอร์หมายเลข P89V51RD2 9
2.4 ลักษณะของแม่เหล็กล็อกประตู 11
2.5 เส้นแรงแม่เหล็กของแท่งเหล็ก 11
2.6 สนามแม่เหล็กของโซลีนอยด์ 12
2.7 สนามแม่เหล็กของแม่เหล็กไฟฟ้าแท่งเหล็กอ่อน 13
2.8 แสดงหน้าต่างโปรแกรมและส่วนประกอบที่สำคัญ 14
2.9 แสดงหลักการทำงานเบื้องต้นของรีเลย์ 16
2.10 แสดงรูปร่างและสัญลักษณ์ของรีเลย์ 16
2.11 แสดงลักษณะจอแอลซีดี 18
2.12 แสดงลักษณะโครงสร้างของหม้อแปลงไฟฟ้า 18
2.13 แสดงโครงสร้างและสัญลักษณ์ต่าง ๆ ของหม้อแปลงไฟฟ้า 19
2.14 แสดงลักษณะหม้อแปลงชนิดแกนเหล็กและสัญลักษณ์หม้อแปลงชนิดแกนเหล็ก 20
2.15 แสดงลักษณะหม้อแปลงชนิดแกนเฟอร์ไรท์และสัญลักษณ์หม้อแปลงชนิดแกนเฟอร์ไรท์ 20
2.16 แสดงลักษณะหม้อแปลงชนิดแกนอากาศและสัญลักษณ์หม้อแปลงชนิดแกนอากาศ 21
3.1 แผนภาพแสดงภาพรวมของระบบ (System Overview Diagram) 22
3.2 แสดงลักษณะโครงสร้างของประตูจำลองด้านหน้า ด้านหลังและด้านข้าง 23
3.3 แสดงแผนภาพการต่อวงจรของระบบควบคุมการทำงาน 24
3.4 แสดงแผนภาพการต่อวงจรของระบบควบคุมการทำงาน 25
3.5 วงจรแปลงไฟและจ่ายไฟ 26
3.6 แสดงวงจรไมโครคอนโทรลเลอร์ P89V51RD2 27
3.7 แสดงวงจรติดต่อผ่านพอร์ตสื่อสารอนุกรม RS-232 28
3.8 แสดงลักษณะบอร์ดไมโครคอนโทรลเลอร์ P89V51RD2 29
3.9 แสดงลักษณะปุ่มกดเลือกระบบและรับคำสั่งการควบคุมด้วยมือ 29
3.10 แสดงปุ่มรับคำสั่งการควบคุมด้วยมือ 30

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.11 แผนภาพการทำงานของระบบควบคุมด้วยมือ 32
3.12 แสดงลักษณะไอคอนโปรแกรมรับคำสั่ง (Door Lock System) 33
3.13 แสดงโปรแกรมรับคำสั่งจากผู้ใช้ผ่านคอมพิวเตอร์ 34
3.14 แสดงการเข้าสู่ระบบด้วยชื่อผู้ใช้งานหรือรหัสผ่านไม่ถูกต้อง 34
3.15 แสดงการเข้าสู่ระบบด้วยชื่อผู้ใช้งานหรือรหัสผ่านถูกต้อง 35
3.16 แสดงหน้าต่างโปรแกรมและไอคอนการรับคำสั่ง 35
3.17 แสดงหน้าต่างโปรแกรมและไอคอนการรับคำสั่ง (UNLOCK) 37
3.18 แสดงหน้าต่างโปรแกรมและไอคอนการรับคำสั่ง (HELP) 37
3.19 แผนภาพการทำงานของโปรแกรมควบคุมผ่านคอมพิวเตอร์ 38
4.1 แสดงปุ่มกดเลือกระบบการทำงานติดตั้งอยู่ภายในห้องทำงาน 39
4.2 แสดงการทำงานของคำสั่ง (UNLOCK) 40
4.3 แสดงการทำงานของปุ่มกดคำสั่ง (IN) 41
4.4 แสดงการทำงานของปุ่มกดคำสั่ง (OUT) 43
4.5 แสดงการทำงานของปุ่มกดคำสั่ง (IN CLASS) 44
4.6 แสดงการทำงานของปุ่มกดคำสั่ง (MEETING) 45
4.7 แสดงการทำงานของปุ่มกดคำสั่ง (LUNCH) 46
4.8 แสดงการทำงานของปุ่มกดคำสั่ง (BE BACK SOON) 47
4.9 แสดงการทำงานของปุ่มกดคำสั่ง (PASSWORD) 48
4.10 แสดงปุ่มกดเลือกระบบการทำงานและไอคอนของโปรแกรมรับคำสั่ง 50
4.11 แสดงการทำงานของคำสั่ง (UNLOCK) ของโปรแกรมรับคำสั่ง 51
4.12 แสดงการทำงานของคำสั่ง (IN) ของโปรแกรมรับคำสั่ง 52
4.13 แสดงการทำงานของคำสั่ง (MEETING) ของโปรแกรมรับคำสั่ง 53
4.14 แสดงการทำงานของคำสั่ง (LUNCH) ของโปรแกรมรับคำสั่ง 54
4.15 แสดงการทดสอบการรับน้ำหนักของตัวล็อกแม่เหล็ก 56

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ความปลอดภัยเป็นสิ่งที่ทุกคนต้องการ ไม่ว่าจะทางร่างกายหรือทรัพย์สินเนื่องจากเหตุการณ์ต่างๆ ที่เกิดขึ้นในสังคม สะท้อนให้เห็นว่าทุกย่างก้าวเต็มไปด้วยความเสี่ยงและอันตราย ทำให้ในปัจจุบันมีการพัฒนาระบบที่พักอาศัยให้มีความสะดวกสบายและเพิ่มความปลอดภัย เช่น โครงการบ้านอัจฉริยะ (Smart home) ซึ่งใช้เทคโนโลยีมาควบคุมระบบของอุปกรณ์ไฟฟ้าต่างๆ ภายในบ้าน เพื่ออำนวยความสะดวกแก่ผู้อยู่อาศัย มีระบบการจัดการพลังงาน เพิ่มระบบรักษาความปลอดภัยแบบอัตโนมัติทั้งภายในและรอบตัวบ้าน ในส่วนของระบบรักษาความปลอดภัยมีการพัฒนาอุปกรณ์ต่างๆ ที่ช่วยให้เกิดความปลอดภัยมากยิ่งขึ้น เช่น กุญแจล็อกไฟฟ้า ระบบคีย์การ์ด การใช้รหัสผ่านประตู เป็นต้น การพัฒนานำแม่เหล็กไฟฟ้ามาช่วยรักษาความปลอดภัย โดยการทำเป็นระบบล็อกประตูอัตโนมัติ ซึ่งระบบสามารถรักษาความปลอดภัยในระดับสูงได้ ลักษณะการทำงานของระบบล็อกประตูอัตโนมัติ เมื่อต้องการให้ประตูล็อกทำได้โดยการจ่ายกระแสไฟฟ้าเข้าไปสร้างสนามแม่เหล็กไฟฟ้า แม่เหล็กจะดูดแผ่นโลหะติดกันทำให้ประตูปิดสนิท แต่ในทางกลับกันเมื่อต้องการให้ประตูเปิดออก ทำได้โดยการไม่จ่ายกระแสไฟฟ้าเข้าไปทำให้ไม่มีการสร้างสนามแม่เหล็กไฟฟ้าประตูจึงเปิดออก

การผลิตระบบล็อกประตูอัตโนมัติจำเป็นต้องใช้ต้นทุนในการผลิตค่อนข้างสูง ส่งผลให้ราคาในท้องตลาดสูงตามไปด้วย อีกทั้งการทำงานของระบบยังถูกตั้งมาจากบริษัทที่จำหน่าย ทำให้ยากต่อการควบคุมดูแล เมื่อเกิดเหตุขัดข้องผู้ใช้ต้องตามช่างมาซ่อมเพียงอย่างเดียวทำให้สิ้นเปลือง ในการศึกษาครั้งนี้จึงมีความต้องการในการปรับปรุงระบบควบคุมการทำงานของระบบล็อกประตูอัตโนมัติ ให้มีการทำงานผ่านคอมพิวเตอร์ที่ผู้ใช้งานสามารถควบคุมได้ด้วยตนเอง และลดต้นทุนการผลิตให้ราคาถูกลง เพื่อให้ผู้บริโภคลดค่าใช้จ่ายในขณะที่ความปลอดภัยเพิ่มมากขึ้น

1.2 วัตถุประสงค์ของโครงงาน

ประยุกต์ใช้ระบบไมโครคอนโทรลเลอร์ ในการควบคุมและสั่งการตัวล็อกแม่เหล็กไฟฟ้า (magnetic lock) ให้ล็อกและปลดล็อกประตูอัตโนมัติผ่านทางระบบคอมพิวเตอร์

1.3 ขอบเขตการศึกษาของโครงงาน

1. ระบบล็อกและปลดล็อกประตูสามารถทำงานในสถานที่มีแหล่งจ่ายไฟฟ้าเท่านั้น

2. การควบคุมระบบสามารถทำได้ภายในระยะทางไม่เกิน 10 เมตร จากคอมพิวเตอร์ถึงตำแหน่งที่ติดตั้งตัวล็อกแม่เหล็ก

3. การใช้งานปุ่มกดเลือกสถานะคำสั่งเพื่อทำการล็อกและปลดล็อกประตูทำได้จากภายในห้องและภายนอกห้อง โดยการปลดล็อกประตูจากภายนอกห้องจะมีการป้อนรหัสผ่านเพื่อตรวจสอบ และการใช้งานภายในห้องผู้ใช้สามารถเลือกสถานะคำสั่งในโปรแกรมรับคำสั่งผ่านคอมพิวเตอร์ได้

4. ในการทดลองการทำงานของระบบมีการสร้างประตูจำลองขึ้นมาแทนประตูที่ใช้งานปกติ ประตูจำลองมีขนาดกว้าง 70 เซนติเมตร สูง 100 เซนติเมตร หนา 5 เซนติเมตร

5. ลักษณะของประตูที่ใช้ในการติดตั้งตัวล็อกแม่เหล็กไฟฟ้าเป็นแบบประตูบานเดี่ยว ใช้งานได้กับประตูแบบไม้และประตูแบบอลูมิเนียม

6. ลักษณะห้องในการติดตั้งระบบล็อกและปลดล็อกประตูเป็นห้องทำงานส่วนบุคคลหรือที่พักอาศัยส่วนบุคคล

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ระบบล็อกและปลดล็อกประตูอัตโนมัติควบคุมผ่านระบบคอมพิวเตอร์และระบบควบคุมด้วยมือ

2. สามารถนำไปใช้เพื่อเพิ่มความปลอดภัยในการล็อกและปลดล็อกประตูภายในที่พักอาศัยและห้องทำงานได้

3. ลดภาระค่าใช้จ่ายในการจัดซื้อและออกแบบสร้างระบบรักษาความปลอดภัย ซึ่งมีราคาของระบบล็อกและปลดล็อดอัตโนมัติสูงตามท้องตลาด

4. ผู้มาติดต่อสามารถทราบตารางงานของผู้ใช้งานระบบได้ และทำให้สะดวกสบายในการติดต่อธุระกับผู้ใช้

5. สร้างความสะดวกสบายในการควบคุมการล็อกและปลดล็อกประตูในระยะไกล

1.5 ขั้นตอนและแผนการดำเนินโครงงาน

กิจกรรม	ปี 2556								
	เม.ย	พ.ค	มิ.ย	ก.ค	ส.ค	ก.ย	ต.ค	พ.ย	ธ.ค
1. ศึกษาและค้นคว้าหาข้อมูลที่เกี่ยวข้อง	←	→							
2. ออกแบบโครงสร้างและระบบการทำงาน		←	→						
3. สร้างประตูจำลองและติดตั้งอุปกรณ์			←	→					
4. เขียนโปรแกรมรองรับระบบการทำงาน					↔				
5. ทดสอบ และปรับปรุงแก้ไข						←	→	→	
6. สรุปผลการทดลองและจัดทำปริญญานิพนธ์								←	→

1.6 งบประมาณที่ใช้ในการดำเนินโครงงาน

1. ตัวล็อกแม่เหล็กไฟฟ้า	1,900 บาท
2. จอแสดงผลแอลซีดี (LCD)	200 บาท
3. แบบจำลองประตู	900 บาท
4. ค่าอุปกรณ์ไฟฟ้าและอิเล็กทรอนิกส์	500 บาท
5. ค่าวัสดุอื่นๆ	500 บาท
6. ค่าทำเล่มปริญญานิพนธ์	1,000 บาท
รวมเป็นเงินทั้งสิ้น (ห้าพันบาทถ้วน)	5,000 บาท

หมายเหตุ : ถัวเฉลี่ยทุกรายการ

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

ในบทนี้เป็นการนำเสนอหลักการและทฤษฎีที่ใช้เป็นองค์ประกอบของระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ แต่ละองค์ประกอบที่มีความสำคัญต่อการทำงานของระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ มีองค์ประกอบที่สำคัญดังนี้ ระบบควบคุมอุปกรณ์ล็อกและปลดล็อกแม่เหล็กไฟฟ้า ระบบแสดงผลสถานะภาพการใช้งาน ชุดอุปกรณ์ต่างๆในการทำงานของระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์

ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ ใช้ไมโครคอนโทรลเลอร์ควบคุมและสั่งการทำงานของระบบสามารถพบได้ทั่วไปในปัจจุบัน โดยมีอุปกรณ์ในการทำงานที่สำคัญดังนี้ คือ ตัวล็อกแม่เหล็กไฟฟ้า จอแสดงผลสถานะภาพการใช้งาน (Liquid Crystal Display) และโซลิดสเตทรีเลย์ (Solid State Relay) เป็นต้น

2.1 บ้านอัจฉริยะ (Smart Home)

คือการใช้เทคโนโลยีมาควบคุมอุปกรณ์ต่างๆภายในบ้าน เพื่ออำนวยความสะดวกแก่ผู้อยู่อาศัย มีระบบการจัดการพลังงานและระบบรักษาความปลอดภัยอัตโนมัติทั้งภายในและรอบตัวบ้าน ส่วนใหญ่จะควบคุมด้วยระบบคอมพิวเตอร์ เรียกว่าบ้านอัตโนมัติ (Home Automation) ซึ่งสามารถถูกจำแนกความสามารถและความซับซ้อนในการควบคุมออกเป็น

1. ระบบควบคุมไฟฟ้าแสงสว่าง เช่น เปิดปิด หรือปรับระดับความสว่าง

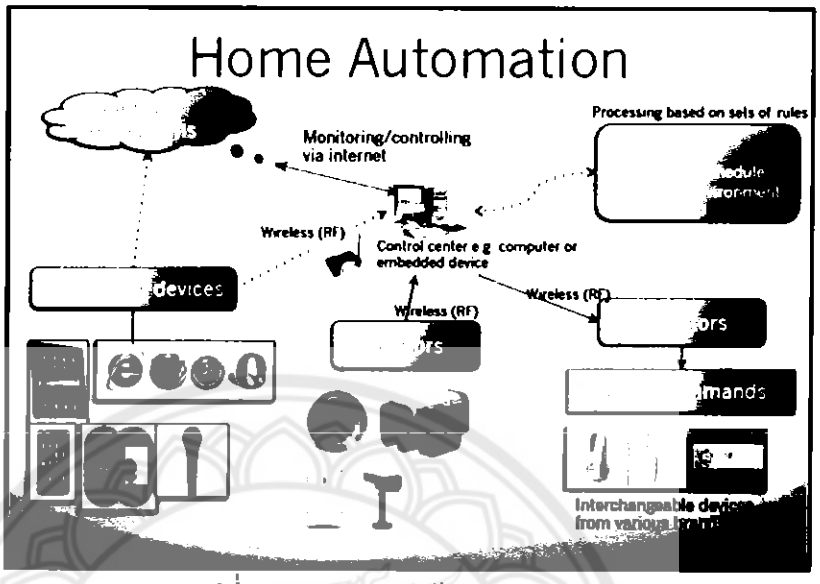
2. ระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน เช่น สั่งงานเครื่องปรับอากาศ หรือการเปิดปิดม่าน

3. ระบบความบันเทิงภายในบ้าน เช่น สั่งให้เปิดวิทยุออนไลน์จากอินเตอร์เน็ตทำงานในห้องที่ผู้ใช้อยู่และปิดเมื่อผู้ใช้ออกจากห้อง

4. ระบบบริหารพลังงาน และพลังงานสำรอง เช่น การปิดเปิด เครื่องใช้ไฟฟ้าต่างๆโดยขึ้นกับสิ่งแวดล้อม

5. ระบบสื่อสาร เช่น รับส่ง ข้อความหรือคำสั่งระหว่างผู้ใช้

6. ระบบรักษาความปลอดภัย เช่น เชื่อมต่อระบบกันขโมย ติดตั้งระบบกล้องกับบริษัทรักษาความปลอดภัย



รูปที่ 2.1 แสดงระบบบ้านอัตโนมัติ[10]

อุปกรณ์ที่ใช้ในระบบบ้านอัตโนมัติ สามารถถูกแยกได้เป็น 4 ส่วนหลัก คือ

1. เซ็นเซอร์ (Sensors) ประกอบไปด้วยอุปกรณ์ตรวจจับต่าง ๆ ภายในบ้าน เช่น อุณหภูมิ ความเคลื่อนไหว ความสว่าง กล้องวีดีโอ รวมทั้งอุปกรณ์ชนิดอื่นๆ

2. ตัวกระตุ้น (Actuators) คืออุปกรณ์ที่ปฏิบัติตัวต่อสิ่งแวดล้อมทางกายภาพ เช่น สวิตช์เปิดปิดไฟฟ้า อุปกรณ์ควบคุมระดับความสว่างหลอดไฟ (Dimmer) มอเตอร์ควบคุมระบบม่าน

3. ศูนย์อำนวยการ (Control center) คือระบบคอมพิวเตอร์ ซึ่งทำหน้าที่เป็นศูนย์กลางการรับรู้สภาพแวดล้อม ประมวลผลสถานการณ์และส่งสัญญาณควบคุมตัวกระตุ้นต่าง ๆ

4. อุปกรณ์ควบคุม (Controlling devices) อุปกรณ์ควบคุมระบบไม่ได้ถูกจำกัดเพียงแค่แผงควบคุมหรือรีโมทคอนโทรลเท่านั้น ชุดควบคุมสามารถอยู่ในรูปของโทรศัพท์มือถือ คอมพิวเตอร์พกพา (tablets, iPad, Galaxy tab) เวปเบราเซอร์ (web browsers) และระบบข้อความ

2.2 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microprocessor) คือ ชิพประมวลผลอย่างหนึ่งทำหน้าที่ประมวลผลตามโปรแกรมหรือชุดคำสั่ง โครงสร้างภายในจะเป็นวงจรรวมขนาดใหญ่ประกอบไปด้วยหน่วยคำนวณทางคณิตศาสตร์และลอจิก บัสข้อมูล บัสควบคุม บัสที่อยู่ พอร์ตขนาน พอร์ตอนุกรม รีจีสเตอร์ หน่วยความจำ วงจรนับ วงจรจับเวลาและวงจรอื่น ๆ รวมอยู่ภายในชิพหรือไอซี ไมโครคอนโทรลเลอร์ ซึ่งถูกออกแบบเพื่อใช้งานควบคุมสามารถติดต่อกับอุปกรณ์อินพุตและเอาท์พุตได้สะดวก ใช้งานได้ง่ายสามารถทำงานได้โดยใช้ชิพเดียว มีคำสั่งสนับสนุนในการเขียนโปรแกรมควบคุมและสามารถเข้าถึงข้อมูลระบบบิตได้ ตระกูลไมโครคอนโทรลเลอร์ที่ใช้ในโครงงานนี้เป็นตระกูล MCS-51

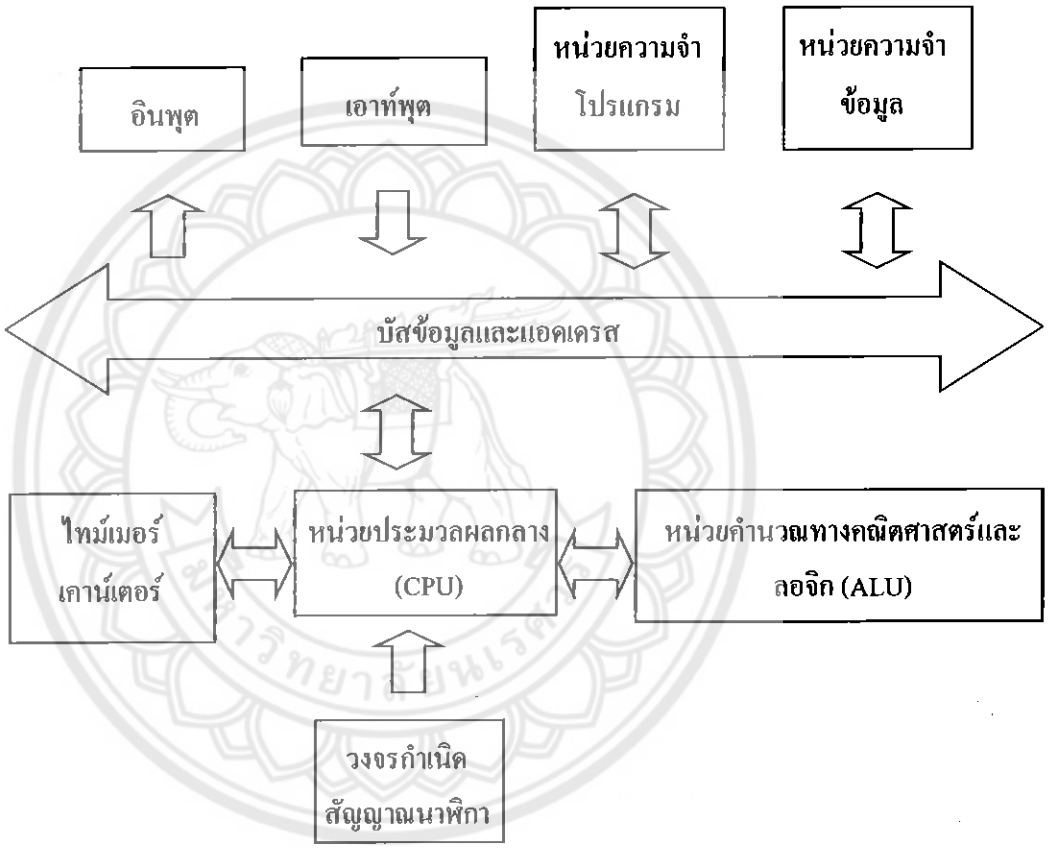
2.2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 เป็นไมโครคอนโทรลเลอร์ที่สนองความต้องการของผู้ใช้แบบสำเร็จในตัวไอซีตัวเดียว คือ สายสัญญาณอินพุตเอาท์พุตภายในตัวพอร์ตของอินพุตและเอาท์พุต บัฟเฟอร์ที่เชื่อมต่อกับวงจรภายนอก (interface) และสายสัญญาณควบคุมอื่น ๆ ที่ใช้สำหรับแยกสายสัญญาณข้อมูลกับสายสัญญาณกำหนดตำแหน่งหน่วยความจำ และยังมีชุดคำสั่งพิเศษเพื่อจัดการข้อมูลเพิ่มขึ้นอีก นอกจากนั้นยังมีวงจรนับเวลาและตั้งเวลาด้วย ข้อสำคัญคือมีการพัฒนาให้ไมโครคอนโทรลเลอร์มีหน่วยความจำเป็นแบบแฟลช (Flash Memory) ทำให้สามารถบันทึกโปรแกรมข้อมูลในหน่วยความจำโปรแกรมได้โดยไม่ต้องถอดตัวไมโครคอนโทรลเลอร์ออกจากวงจร เรียกว่า การโปรแกรมภายในวงจร (In-System Programming) และมีการติดต่อแบบเอสพีไอ (Serial Peripheral Interface) ทำให้การพัฒนาและการแก้ไขปรับปรุงโปรแกรมทำได้สะดวกมากยิ่งขึ้น

2.2.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

วงจรภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบด้วยวงจรพอร์ตอินพุตหรือเอาท์พุตทั้งหมด 4 พอร์ต แต่ละพอร์ตจะเป็นแบบ 8 บิต หน่วยความจำโปรแกรมภายใน (EPROM, EEPROM และแฟลช) หน่วยความจำที่เป็นข้อมูล (RAM) ซึ่งรวมอยู่ในวงจรหลักในไมโครคอนโทรลเลอร์ตลอดจนวงจรการคำนวณทางคณิตศาสตร์และลอจิก (ALU) วงจรรีจิสเตอร์

การควบคุมไมโครคอนโทรลเลอร์นั้นกระทำผ่านกระบวนการควบคุมโดยโปรแกรมที่เขียนขึ้น เพื่อบอกถึงการทำงานของไมโครคอนโทรลเลอร์ โดยลักษณะที่ถูกระบุขึ้นโดยผู้เขียนโปรแกรมควบคุม ซึ่งควบคุมการทำงานทั้งหมดของไมโครคอนโทรลเลอร์ ในการกำหนดพอร์ตให้เป็นอินพุตและเอาท์พุต และยังสามารถกำหนดหน่วยความจำภายในซึ่งเป็นที่เก็บข้อมูล และเป็นที่พักข้อมูลตามความต้องการ โดยในการทำงานของไมโครคอนโทรเลอร์แต่ละคำสั่ง จะอ้างอิงเวลาจากสัญญาณนาฬิกาที่ส่งให้กับไมโครคอนโทรลเลอร์ โดยมีโครงสร้างการทำงานรูปที่ 2.2



รูปที่ 2.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์[1,6]

2.2.3 การเขียนโปรแกรมไมโครคอนโทรลเลอร์

ภาษาที่ใช้สำหรับการเขียนโปรแกรมบนไมโครคอนโทรลเลอร์แบ่งได้เช่นเดียวกับภาษาในการเขียนโปรแกรมบนคอมพิวเตอร์คือ ภาษาเครื่อง ภาษาแอสเซมบลี และภาษาซี

ภาษาเครื่องเป็นภาษาระดับต่ำสุดประกอบไปด้วยรหัสฐาน 2 คือ 0 กับ 1 เท่านั้น ซึ่งเป็นภาษาที่ไมโครคอนโทรลเลอร์เข้าใจแต่มนุษย์จะทำความเข้าใจได้ยาก เพราะต้องอาศัยการจดจำรหัสคำสั่งต่างๆ รวมถึงต้องเข้าใจโครงสร้างภายในของไมโครคอนโทรลเลอร์ด้วย จึงได้มีการคิดค้นสิ่ง

ที่เรียกว่า คอมไพล์เลอร์ (Compiler) ขึ้นมาเพื่อทำให้มนุษย์สามารถเขียนโปรแกรมด้วยภาษาระดับสูงที่มนุษย์เข้าใจได้ โดยคอมไพล์เลอร์ทำหน้าที่เปลี่ยนภาษาระดับสูงเหล่านั้นกลายเป็นภาษาเครื่องเอง

ภาษาแอสเซมบลี เป็นภาษาที่ใช้รหัสคำสั่งที่เป็นตัวอักษรภาษาอังกฤษมาแทนคำสั่งเลขฐาน 2 ในแบบของภาษาเครื่องทำให้ภาษาแอสเซมบลีกลายเป็นภาษาที่มนุษย์ทำความเข้าได้ง่ายขึ้น นอกจากนั้นแล้วยังเป็นภาษาที่ใช้ยังทำให้โปรแกรมมีการทำงานได้อย่างรวดเร็ว เพราะมีการสั่งงานไปที่ฮาร์ดแวร์โดยตรง ด้วยเหตุนี้ทำให้ผู้พัฒนาโปรแกรมจำเป็นจะต้องเข้าใจโครงสร้างภายในไมโครคอนโทรลเลอร์อย่างละเอียดด้วย ซึ่งกลายเป็นข้อด้อยของภาษาแอสเซมบลีไป คอมไพล์เลอร์ที่ทำหน้าที่แปลงภาษาแอสเซมบลีให้เป็นภาษาเครื่องเรียกว่า แอสเซมเบลอร์

ภาษาซีเป็นภาษาระดับสูงที่มีความใกล้เคียงกับภาษามนุษย์ ทำให้ทำความเข้าใจได้ง่าย นอกจากนั้นแล้วการเขียนโปรแกรมภาษาซีก็ไม่ต้องจำเป็นต้องเข้าใจโครงสร้างภายในของไมโครคอนโทรลเลอร์อย่างละเอียด เพียงแต่เข้าใจการเขียนโปรแกรมแบบโครงสร้างก็เพียงพอแล้ว ภาษาซีสามารถใช้ในการเข้าถึงโครงสร้างภายในของไมโครคอนโทรลเลอร์ได้โดยตรง ทำให้ได้โปรแกรมที่เขียนขึ้นทำงานได้รวดเร็ว ดังนั้นภาษาซี จึงเป็นที่นิยมแพร่หลายในการเขียนโปรแกรมไมโครคอนโทรลเลอร์คอมไพล์เลอร์ที่ใช้ในการแปลงภาษาซีเป็นภาษาเครื่องมีอยู่มากมาย เช่น คอมไพล์เลอร์ Keil uVision 4 เป็นต้น

2.2.4 รูปแบบการทำงานของขาไมโครคอนโทรลเลอร์หมายเลข P89V51RD2

ไมโครคอนโทรลเลอร์ที่ใช้งานในปัจจุบันมีอยู่ด้วยกันหลากหลายรุ่น และในแต่ละโครงสร้างอันได้แก่ หน่วยความจำภายใน จำนวนขา จำนวนพอร์ต ที่แตกต่างกัน ดังนั้นการเลือกไมโครโปรเซสเซอร์ไปใช้งาน จึงขึ้นอยู่กับความต้องการของผู้ใช้ หรือความเหมาะสมของงาน ในโครงสร้างนี้ผู้ดำเนินโครงงานเลือกใช้ไมโครคอนโทรลเลอร์หมายเลข P89V51RD2 ซึ่งมีคุณสมบัติดังนี้

1. เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
2. มีหน่วยความจำภายในแบบแฟลชขนาด 4 กิโลไบต์ หรือ 8 กิโลไบต์
3. สามารถเขียนและลบได้เป็นพันครั้ง
4. มีสายสัญญาณสำหรับอินพุตหรือเอาท์พุตได้ 32 เส้น (แบบ 2 ทิศทาง)
5. มีหน่วยความจำชั่วคราว (RAM) ภายในขนาด 128 กิโลไบต์ หรือ 256 กิโลไบต์

6. ใช้ความถี่สัญญาณนาฬิกาตั้งแต่ 0 เฮิรตซ์ จนถึง 24 เมกกะเฮิรตซ์

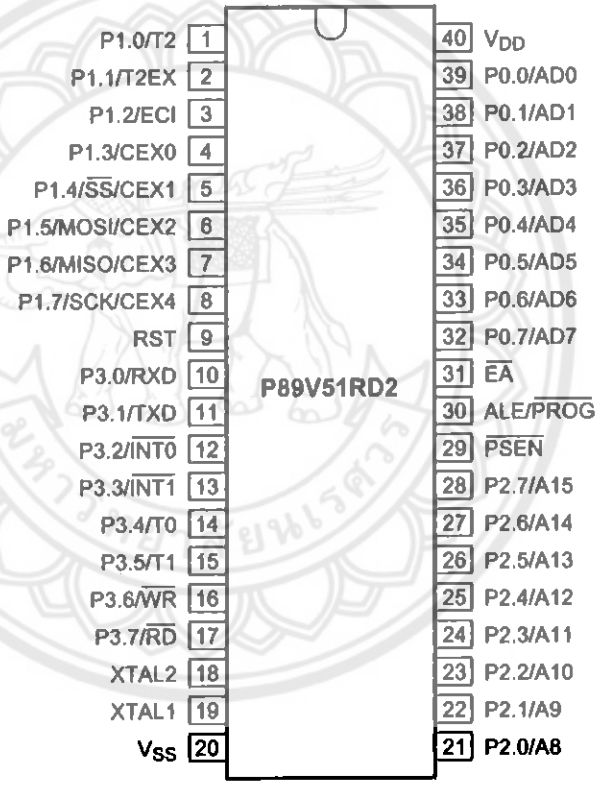
7. มีวงจรตั้งเวลาและนับเวลาขนาด 16 บิต จำนวน 2 ชุด หรือ 3 ชุด

8. มีวงจรรับสัญญาณอินเทอร์รัพต์ได้ไม่ต่ำกว่า 6 ชนิด

9. สามารถต่อขยายหน่วยความจำภายนอกได้สูงสุด 64 กิโลไบต์

10. มีวงจรสื่อสาร 2 ทางเต็มอัตรา (Full Duplex)

สำหรับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 หมายเลข P89V51RD2 แสดงการจัดขาของไมโครคอนโทรลเลอร์ในแบบ 40 ขา ดังรูปที่ 2.3 และมีรายละเอียดการทำงานดังตารางที่ 2.1



รูปที่ 2.3 รูปแบบการทำงานของขาไมโครคอนโทรลเลอร์หมายเลข P89V51RD2[1,6]

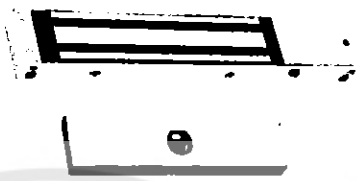
ตารางที่ 2.1 รายละเอียดการทำงานแต่ละขาของไมโครคอนโทรลเลอร์หมายเลข P89V51RD2[1]

ขา	หน้าที่การทำงาน
V_{DD}	เป็นขาสำหรับต่อไฟเลี้ยง 5 โวลต์
V_{SS}	สำหรับต่อลงกราวด์

XTAL1/XTAL2	ต่อกับตัวผลิตสัญญาณนาฬิกา
RST (Reset)	เป็นขาอินพุตเพื่อเริ่มต้นการทำงานใหม่ของไมโครคอนโทรลเลอร์ โดยการป้อนสัญญาณลอจิก 1
ALE/PROG (Address Latch Enable)	เป็นขาสัญญาณเอาท์พุตเพื่อแลตช์ค่าแอดเดรสตำแหน่งข้อมูล (Address Bus) ในการติดต่อกับหน่วยความจำภายนอก และเป็นขาสัญญาณเอาท์พุตเพื่อควบคุมการโปรแกรมให้กับไมโครคอนโทรลเลอร์
PSEN (Program Store Enable)	เป็นขาสัญญาณสโตรบ เมื่อต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ให้ส่งสัญญาณนี้ 2 ครั้งใน 1 พัลส์สัญญาณนาฬิกา
Port 0 (P0.0-P0.7)	เป็นขาอินพุตและเอาท์พุตให้กับอุปกรณ์ภายนอก แบบโอเพ่นเดรน (ไม่มีตัวต้านทานพูลอัพภายใน) ดังนั้นการใช้งานพอร์ต 0 จึงจำเป็นต้องต่อตัวต้านทานพูลอัพ ด้วย นอกจากนี้ยังทำหน้าที่เป็นขาแอดเดรสบัส (A0-A7) ในการติดต่อกับหน่วยความจำภายนอกและบัสข้อมูล (D0-D7) เพื่อรับข้อมูลการโปรแกรมให้กับไมโครคอนโทรลเลอร์
Port 1 (P1.0-P1.7)	เป็นขาอินพุตและเอาท์พุตต่อกับอุปกรณ์ภายนอก แบบมีตัวต้านทานพูลอัพภายใน
Port 2 (P2.0-P2.7)	เป็นขาอินพุตและเอาท์พุตต่อจากอุปกรณ์ภายนอก แบบมีตัวต้านทานพูลอัพภายใน และเป็นขาแอดเดรสบัส (A8-A15) ในการติดต่อกับหน่วยความจำภายนอก
P3.0/RXD	รับข้อมูลแบบอนุกรม
P3.1/TXD	ส่งข้อมูลแบบอนุกรม
P3.2/INT0	อินเทอร์รัพต์ภายนอกหมายเลข 0
P3.3/IMT1	อินเทอร์รัพต์ภายนอกหมายเลข 1
P3.4/T0	ตัวควบคุมเวลาการทำงาน/ตัวช่วยนับ ตัวที่ 1
P3.5/T1	ตัวควบคุมเวลาการทำงาน/ตัวช่วยนับ ตัวที่ 2
P3.6/WR	สัญญาณในการเขียนข้อมูลไปยังหน่วยความจำภายนอก
P3.7/RD	สัญญาณในการอ่านข้อมูลจากหน่วยความจำภายนอก

2.3 แม่เหล็กล็อกประตู

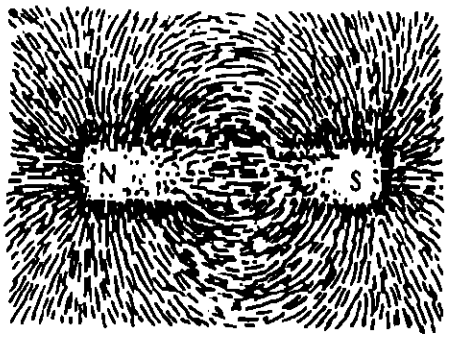
แม่เหล็กล็อกประตู คืออุปกรณ์ที่ถูกสร้างและพัฒนาขึ้นมาเพื่อช่วยรักษาความปลอดภัยในชีวิตประจำวันของมนุษย์ โดยอาศัยหลักการทางแม่เหล็กไฟฟ้ามาช่วยในการทำงานของตัวแม่เหล็กล็อกประตู



รูปที่ 2.4 ลักษณะของแม่เหล็กล็อกประตู[7]

2.3.1 ทฤษฎีสนามแม่เหล็กไฟฟ้า

กระแสไฟฟ้าที่ไหลผ่านเส้นลวดจะทำให้เกิดสนามแม่เหล็กรอบ ๆ เส้นลวด ลักษณะของสนามแม่เหล็กขึ้นอยู่กับรูปร่างของเส้นลวดและกระแสไฟฟ้าที่ไหลผ่าน สนามแม่เหล็กที่เกิดขึ้นนี้เรียกว่าแม่เหล็กไฟฟ้า ซึ่งใช้สร้างแม่เหล็กที่มีกำลังสูง และใช้สำหรับทำให้เกิดการเคลื่อนที่โดยกระแสไฟฟ้าแม่เหล็กไฟฟ้า (Electromagnets) หมายถึง อำนาจแม่เหล็กเกิดจากการไหลของกระแสไฟฟ้าผ่านวัตถุตัวนำ ซึ่งจะทำให้เกิดสนามแม่เหล็กรอบ ๆ ตัวนำนั้น เมื่อมีกระแสไฟฟ้าไหลผ่านเส้นลวดตัวนำ จะเกิดเส้นแรงแม่เหล็กขึ้นรอบ ๆ เส้นลวดตัวนำ แต่อำนาจแม่เหล็กที่เกิดขึ้นมีเพียงจำนวนเล็กน้อย ซึ่งไม่สามารถนำไปใช้ประโยชน์ได้การเพิ่มความเข้มของสนามแม่เหล็กทำได้โดยการนำเส้นลวดตัวนำมาพันเป็นขดลวด เส้นแรงแม่เหล็กที่เกิดในแต่ละส่วนของเส้นลวดตัวนำจะเสริมอำนาจกัน ทำให้มีความเข้มของสนามแม่เหล็กเพิ่มขึ้น



รูปที่ 2.5 เส้นแรงแม่เหล็กของแท่งเหล็ก[7]

2.3.1.1 ความเข้มของสนามแม่เหล็กไฟฟ้า จะขึ้นอยู่กับส่วนประกอบต่าง ๆ ดังนี้

1. จำนวนรอบของการพันเส้นลวดตัวนำ การพันจำนวนรอบของเส้นลวดตัวนำมากเกิดสนามแม่เหล็กมาก ในทางกลับกันถ้าพันจำนวนรอบน้อยการเกิดสนามแม่เหล็กก็น้อยตามไปด้วย

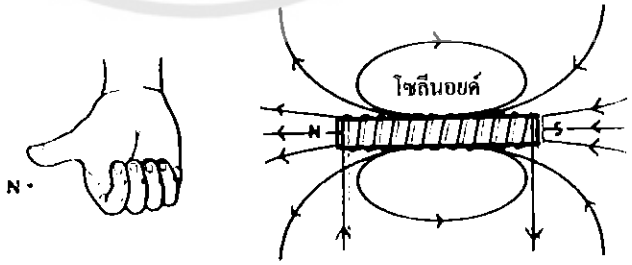
2. ปริมาณการไหลของกระแสไฟฟ้าผ่านเส้นลวดตัวนำกระแสไฟฟ้าไหลผ่านมากสนามแม่เหล็กเกิดขึ้นมาก และถ้ากระแสไฟฟ้าไหลผ่านน้อยสนามแม่เหล็กเกิดน้อย

3. ชนิดของวัสดุที่ใช้ทำแกนของแท่งแม่เหล็กไฟฟ้า วัสดุต่างชนิดกันจะให้ความเข้มของสนามแม่เหล็กต่างกัน เช่น แกนอากาศจะให้ความเข้มของสนามแม่เหล็กน้อยกว่าแกนที่ทำจากสารเฟอโรแมกเนติก (Ferromagnetic) หรือสารที่สามารถเกิดอำนาจแม่เหล็กได้ เช่น เหล็กเฟอร์ไรท์ เป็นต้น สารเหล่านี้จะช่วยเสริมอำนาจแม่เหล็กในขดลวดทำให้มีความเข้มของสนามแม่เหล็กมาก

4. ขนาดของแกนแท่งแม่เหล็กไฟฟ้า แกนที่มีขนาดใหญ่จะให้สนามแม่เหล็กมาก ส่วนแกนที่มีขนาดเล็กจะให้สนามแม่เหล็กน้อย

2.3.1.2 สนามแม่เหล็กของโซลีนอยด์

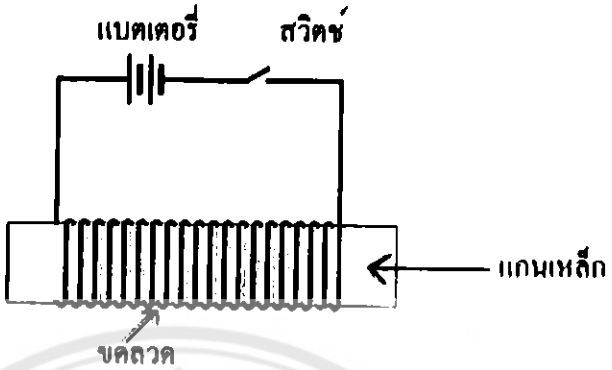
เมื่อนำลวดตัวนำที่มีฉนวนหุ้มมาขดเป็นวงกลมหลาย ๆ วงเรียงซ้อนกันเป็นรูปทรงกระบอกขดลวดที่ได้นี้ เรียกว่า โซลีนอยด์ (solenoid) เมื่อให้กระแสไฟฟ้าผ่านโซลีนอยด์จะมีสนามแม่เหล็กเกิดขึ้น การหาทิศของสนามแม่เหล็กใช้วิธีการกำมือขวาแบบเดียวกับการหาทิศของสนามแม่เหล็กของลวดตัวนำวงกลม ปลายขดลวดด้านที่สนามแม่เหล็กพุ่งออกจะเป็นขั้วเหนือ และอีกปลายหนึ่งซึ่งสนามแม่เหล็กพุ่งเข้าจะเป็นขั้วใต้



รูปที่ 2.6 สนามแม่เหล็กของโซลีนอยด์[7]

สนามแม่เหล็กที่เกิดจากโซลีนอยด์มีค่าสูงสุดที่บริเวณแกนกลางของโซลีนอยด์และขนาดของสนามแม่เหล็กนี้จะมีค่าเพิ่มขึ้น เมื่อกระแสไฟฟ้าเพิ่มหรือจำนวนรอบของขดลวดเพิ่ม ถ้าใส่แท่งเหล็กอ่อนไว้ที่แกนกลางของโซลีนอยด์ เมื่อกระแสไฟฟ้าผ่านโซลีนอยด์แท่งเหล็กอ่อนจะมีสมบัติ

เป็นแม่เหล็ก แม่เหล็กที่เกิดจากวิธีนี้เรียกว่า แม่เหล็กไฟฟ้า (electromagnet) สนามแม่เหล็กของแม่เหล็กไฟฟ้าจะเพิ่ม เมื่อกระแสไฟฟ้าเพิ่มและจำนวนรอบต่อความยาวของขดลวดเพิ่ม แต่เมื่อไม่มีกระแสไฟฟ้าแท่งเหล็กอ่อนจะหมดสภาพแม่เหล็กทันที



รูปที่ 2.7 สนามแม่เหล็กของแม่เหล็กไฟฟ้าแท่งเหล็กอ่อน[7]

หลักการของแม่เหล็กไฟฟ้านี้ถูกนำไปประยุกต์สร้างอุปกรณ์ไฟฟ้า เช่น สวิตช์อัตโนมัติ สวิตช์รีเลย์ และตัวล็อกแม่เหล็กไฟฟ้า เป็นต้น

2.4 โปรแกรม Microsoft Visual Basic

วิชัวร์เบสิก (Visual Basic) หรือ VB เป็นโปรแกรมภาษาคอมพิวเตอร์ตัวหนึ่ง ซึ่งใช้สำหรับสร้างหรือพัฒนาโปรแกรมใช้งานบนวินโดวส์ มีความสามารถในการทำงานคล้ายกับภาษาคอมพิวเตอร์อื่น ๆ เช่น C , Pascal , C++ , C# สร้างโดยบริษัทไมโครซอฟท์ ภาษานี้เป็นหนึ่งในภาษาโปรแกรมยอดนิยมสำหรับโปรแกรมที่ใช้ในด้านธุรกิจ ซึ่งเป็นภาษาคอมพิวเตอร์ระดับสูง ใช้งานง่าย เหมาะสำหรับผู้เริ่มต้น เพราะใช้คำในภาษาอังกฤษที่เข้าใจง่าย และเมื่อเป็นวิชัวร์ เบสิก ซึ่งใช้ลักษณะของการมองเห็นได้ วิชัวร์ที่เป็นการติดต่อกับผู้ใช้ด้วยกราฟิกหรือรูปภาพ (Graphical User Interface - GUI) จึงทำให้การพัฒนาโปรแกรมใช้งานได้สะดวกและรวดเร็วขึ้น ถึงแม้จะใช้งานง่าย แต่ก็มีความสามารถสูง เหมาะสำหรับการพัฒนาโปรแกรมใช้งานได้หลายด้าน เช่น งานคำนวณทั่วไป งานด้านฐานข้อมูล เกม ฯลฯ และยังได้พัฒนาต่อเป็นภาษา (VB.NET) อีกด้วย

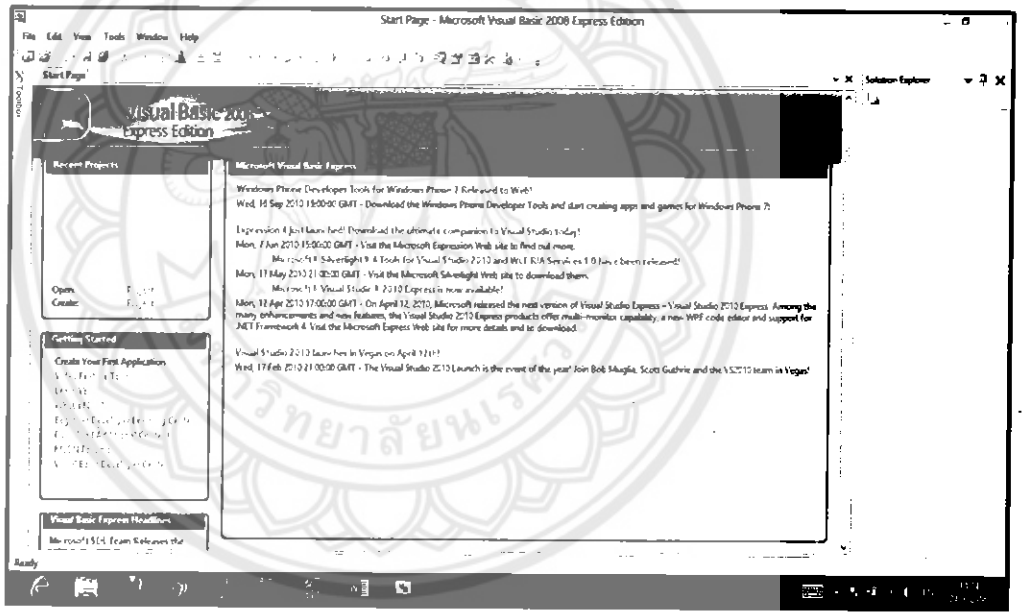
ความหมายของดอทเน็ต (.NET) เป็นกลุ่มของเทคโนโลยีทางซอฟต์แวร์ที่เชื่องโยงข้อมูล ข่าวสาร, คน, ระบบและอุปกรณ์ต่าง ๆ เข้าด้วยกัน ด้านการเขียนโปรแกรมดอทเน็ต หมายถึง เค้าโครง (Framework) คือ สภาพแวดล้อมที่สนับสนุนการพัฒนา และรันโปรแกรมในรูปแบบของเค้าโครงดอทเน็ต จะทำหน้าที่ควบคุมการรันโปรแกรม และให้บริการทรัพยากรต่าง ๆ แก่

โปรแกรมที่รัน เช่น การโหลดโปรแกรมขึ้นมาทำงาน การจัดการหน่วยความจำ การจัดเตรียมไลบรารีให้โปรแกรมเรียกใช้งาน

2.4.1 เริ่มต้นใช้งานโปรแกรม Microsoft Visual Basic

การเปิดใช้งานโปรแกรม สามารถทำได้ดังนี้

1. คลิกที่ปุ่มเริ่มทำงาน (Start)
2. เลือก All Program
3. เลือก Microsoft Visual Studio
4. เลือก Microsoft Visual Basic
5. รอสักครู่จะปรากฏหน้าจอโปรแกรมขึ้นมา



รูปที่ 2.8 แสดงหน้าต่างโปรแกรมและส่วนประกอบที่สำคัญ[4,8]

แถบหัวเรื่อง (Title bar) : เป็นแถบแสดงหัวเรื่องโดยปกติจะบอกชื่อของโปรแกรม

แถบเมนู (Menu bar) : เป็นแถบแสดงรายการคำสั่งต่าง ๆ ของโปรแกรม

แถบเครื่องมือ (Tool bar) : เป็นแถบเครื่องมือที่รวบรวมไอคอนต่าง ๆ ที่ใช้ประจำ

หน้าต่างเริ่มต้น (Start Page) : เป็นหน้าต่างเริ่มการทำงานของโปรแกรม ซึ่งจะมีส่วนที่แสดงรายชื่อโปรเจคที่ได้เคยสร้างไว้แล้วในกรณีที่เพิ่งเปิดใช้งานครั้งแรกจะไม่มีรายชื่อปรากฏอยู่

กล่องเครื่องมือ (Toolbox) : เป็นส่วนที่เก็บเครื่องมือต่าง ๆ ที่สามารถนำมาใช้งานในโปรเจคแต่ในกรณียังไม่มีเครื่องมือเนื่องจากเรายังไม่ได้สร้างโปรเจค (Project)

หน้าต่าง (Solution Explorer) : หน้าต่างที่แสดงรายชื่อและจัดการกับโปรเจค

แถบสถานะ (Status Bar) : เป็นแถบที่มีไว้สำหรับในการบอกสถานะของการทำงานของโปรแกรม

2.5 รีเลย์ (Relay)

รีเลย์ เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ ได้มากมาย

2.5.1 หน้าที่และหลักการทำงานของรีเลย์

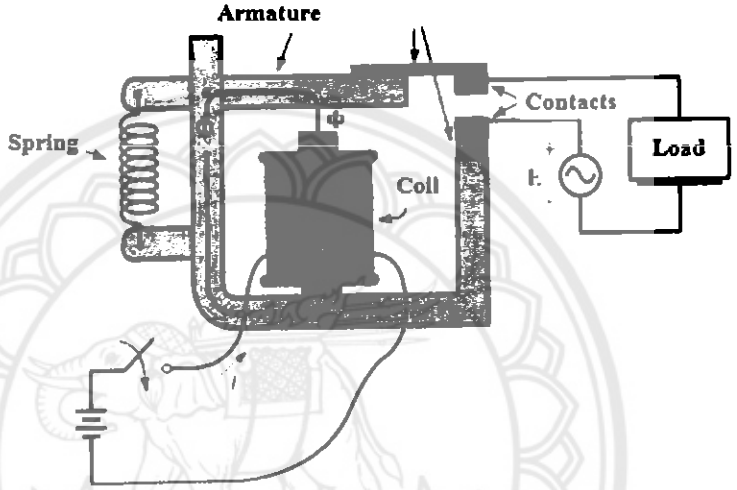
หน้าที่ของคอนแทคเตอร์ คือ การใช้กำลังไฟฟ้าจำนวนน้อยเพื่อไปควบคุมการตัดต่อกำลังไฟฟ้าจำนวนมาก คอนแทคเตอร์ ทำให้เราสามารถควบคุมกำลังไฟฟ้าในตำแหน่งอื่น ๆ ของระบบไฟฟ้าได้ สายไฟควบคุมให้รีเลย์กำลังหรือคอนแทคเตอร์ทำงานเป็นสายไฟฟ้าขนาดเล็กต่อเข้ากับสวิตช์ควบคุมและขดลวด (Coil) ของของคอนแทคเตอร์ กำลังไฟฟ้าที่ป้อนเข้าขดลวดอาจจะเป็นไฟฟ้ากระแสตรง หรือไฟฟ้ากระแสสลับก็ได้ขึ้นอยู่กับการออกแบบการใช้คอนแทคเตอร์ทำให้สามารถควบคุมวงจรจากระยะไกล (Remote) ได้ ซึ่งทำให้เกิดความปลอดภัยกับผู้ปฏิบัติงานในการควบคุมกำลังไฟฟ้า

หลักการทำงานเบื้องต้นของรีเลย์แสดงดังรูปที่ 2.7 การทำงานเริ่มจากปิดสวิตช์เพื่อป้อนกระแสให้กับขดลวด โดยทั่วไปจะเป็นขดลวดพันรอบแกนเหล็กทำให้เกิดสนามแม่เหล็กไปดูดเหล็กอ่อนที่เรียกว่า อาร์เมเจอร์ (Armature) ให้ต่ำลงมา ที่ปลายของอาร์เมเจอร์ด้านหนึ่งมักยึดติดกับสปริง (Spring) และปลายอีกด้านหนึ่งยึดติดกับหน้าสัมผัสการเคลื่อนที่อาร์เมเจอร์ จึงทำให้เป็นการควบคุมการเคลื่อนที่ของหน้าสัมผัส ให้แยกจากกันหรือแตะกับหน้าสัมผัสอีกอันหนึ่งซึ่งยึดติดอยู่กับที่ เมื่อเปิดสวิตช์อาร์เมเจอร์ก็จะกลับสู่ตำแหน่งเดิม เราสามารถนำหลักการนี้ไปควบคุมโหลด (Load) หรือวงจรอิเล็กทรอนิกส์ต่าง ๆ ได้ตามต้องการ

2.5.2 รีเลย์แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power relay) หรือมักเรียกกันว่า คอนแทคเตอร์ (Contactor or Magnetic contactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา

2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทคเตอร์ขนาดใหญ่ รีเลย์ควบคุมบางทีเรียกกันง่าย ๆ ว่า รีเลย์



รูปที่ 2.9 แสดงหลักการทำงานเบื้องต้นของรีเลย์[5]

(ก) รูปร่างของรีเลย์ที่มีตัวถังเป็นพลาสติกใสป้องกันฝุ่น

(ข) สัญลักษณ์แบบลวดพัน

(ค) สัญลักษณ์แบบตัวเหนี่ยวนำพันแกนเหล็ก

รูปที่ 2.10 แสดงรูปร่างและสัญลักษณ์ของรีเลย์[5]

2.6 จอแสดงผลแอลซีดี

เทคโนโลยีมอนิเตอร์ แอลซีดี ย่อมาจากหน้าจอแสดงผลผลึกเหลว (Liquid Crystal Display) ซึ่งเป็นจอแสดงผลแบบดิจิตอล (Digital) โดยภาพที่ปรากฏขึ้นเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพ (Black Light) ผ่านชั้นกรองแสง (Polarized filter) แล้ววิ่งไปยังคริสตัลเหลวที่เรียงตัวด้วยกัน 3 เซลล์ คือ แสงสีแดง แสงสีเขียว และแสงสีน้ำเงิน กลายเป็นพิกเซล (Pixel) ที่สว่างสดใสเกิดขึ้น

เทคโนโลยีที่พัฒนามาใช้กับ แอลซีดี นั้นแบ่งออกเป็น 2 ประเภทคือ

1. โพสซิธีฟเมทริกต์ (Super-Twisted Nematic) เป็นเทคโนโลยีแบบเก่าที่ให้ความคมชัดและความสว่างน้อยกว่า ใช้ในจอโทรศัพท์มือถือทั่วไป

2. แอกซิธีฟเมทริกต์ (Thin Film Transistors) สามารถแสดงภาพได้คมชัดและสว่างกว่าแบบแรก ใช้ในจอมอนิเตอร์หรือโน้ตบุ๊ก

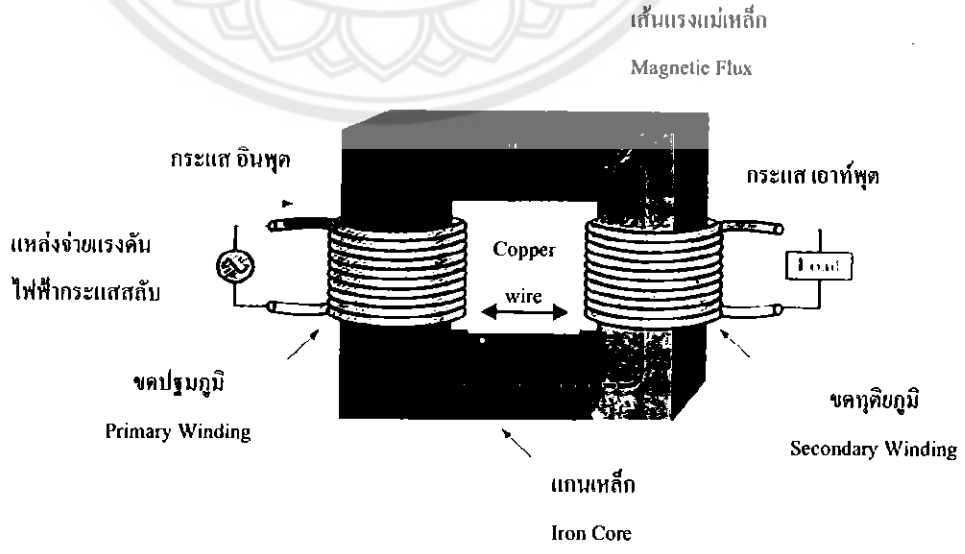
สทิวเท็ด นีมาติก (Twisted Nematic) คือสารประเภทนี้จะมีการจัดโครงสร้างโมเลกุลเป็นเกลียว แต่ถ้าเราผ่านกระแสไฟฟ้าเข้าไปมันก็จะคลายตัวออกเป็นเส้นตรง เราใช้ปรากฏการณ์นี้เป็นตัวกำหนดว่าจะให้แสงผ่านได้หรือไม่ได้ สทิวเท็ด นีมาติกผลึกเหลวชนิดนี้จะให้เราสามารถเปลี่ยนทิศทางการสั่นของคลื่นแสงได้ 90 องศา ถึง 150 องศา คือเปลี่ยนจากแนวตั้งให้กลายเป็นแนวนอนหรือเปลี่ยนกลับกันจากแนวนอนให้เป็นแนวตั้งก็ได้ ด้วยจุดนี้เองทำให้ค่าตอบสนองสัญญาณเทียบกับเวลา (Response Time) มีค่าสูง

หลักการพื้นฐานคือการบังคับให้หยดของผลึกเหลว (Liquid crystal) ซึ่งมีแผ่นแก้วกักเอาไว้ให้ไปปิดรูช่องแสง ซึ่งแสงถูกฉายมาจากด้านหลังของหน้าจอ ก่อให้เกิดการแสดงผลเป็นตัวอักษร หรือตัวเลขในรูปแบบต่าง ๆ ได้ตามต้องการ จุดเด่นของหน้าจอแอลซีดีขาวดำ หรือเรียกอีกอย่างว่าหน้าจอแบบโมโนโครม์ คือการทำงานที่ไม่อาศัยปืนยิงอิเล็กตรอน จึงช่วยให้ด้านลึกของจอภาพมีขนาดสั้นกว่ามอนิเตอร์แบบซีดีที (CDT) ถึง 3 เท่าและด้วยรูปร่างที่แนบราบทางด้านหน้าและด้านหลัง ขนาดเล็กกะทัดรัดและน้ำหนักเบาและประหยัดพลังงานไฟฟ้า สำหรับการแสดงในโครงงานนี้ถูกออกแบบให้แสดงบนจอแสดงผลแอลซีดีแบบ 16 ตัวอักษร 2 บรรทัด ซึ่งแสดงตัวอย่างของแอลซีดีดังรูปที่ 2.11

รูปที่ 2.11 แสดงลักษณะจอแอลซีดี[1]

2.7 หม้อแปลงไฟฟ้า

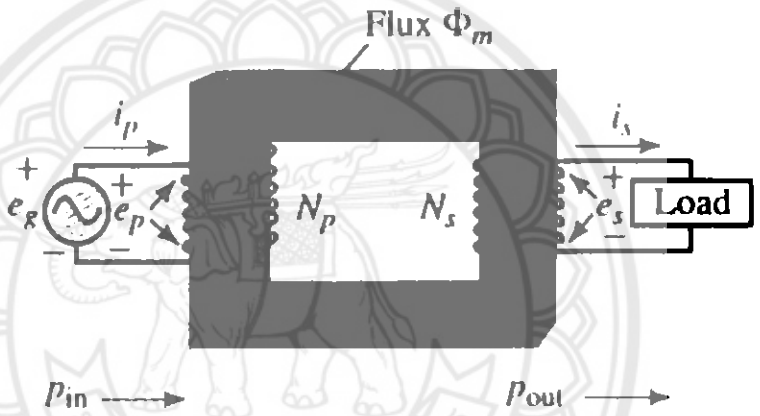
การทำงานของหม้อแปลงไฟฟ้านั้น อาศัยหลักการความสัมพันธ์ระหว่างกระแสไฟฟ้ากับเส้นแรงแม่เหล็กในการสร้างแรงเคลื่อนเหนี่ยวนำให้กับตัวนำ คือ เมื่อมีกระแสไหลผ่านขดลวดตัวนำ จะทำให้เกิดเส้นแรงแม่เหล็กรอบ ๆ ตัวนำนั้น และถ้ากระแสที่ป้อนมีขนาดและทิศทางที่เปลี่ยนแปลงไปมา จะทำให้สนามแม่เหล็กที่เกิดขึ้นมีการเปลี่ยนแปลงตามไปด้วย ถ้าสนามแม่เหล็กที่มีการเปลี่ยนแปลงดังกล่าวตัดผ่านตัวนำ ก็จะเกิดแรงเคลื่อนเหนี่ยวนำขึ้นที่ตัวนำนั้น โดยขนาดของแรงเคลื่อนเหนี่ยวนำจะสัมพันธ์กับความเข้มของสนามแม่เหล็ก และความเร็วในการตัดผ่านตัวนำของสนามแม่เหล็ก



รูปที่ 2.12 แสดงลักษณะโครงสร้างของหม้อแปลงไฟฟ้า[9]

พิจารณาจากรูป จะเห็นว่าโครงสร้างของหม้อแปลงจะประกอบไปด้วยขดลวด 2 ขดพันรอบแกนที่เป็นสื่อกลางของเส้นแรงแม่เหล็ก ซึ่งอาจเป็นแกนเหล็ก แกนเฟอไรท์ หรือแกนอากาศ ขดลวดที่เราจ่ายไฟเข้าไปเราเรียกว่า ขดปฐมภูมิ (Primary Winding) และขดลวดอีกขดที่ต่อเข้ากับโหลดเราเรียกว่า ขดทุติยภูมิ (Secondary Winding)

เมื่อเราจ่ายกระแสไฟฟ้าสลับให้กับขดปฐมภูมิก็จะทำให้เกิดสนามแม่เหล็กที่เปลี่ยนแปลงไปมา โดยเส้นแรงแม่เหล็กดังกล่าวก็จะวิ่งไปมาตามแกน และไปตัดกับขดทุติยภูมิ ทำให้เกิดแรงดันเหนี่ยวนำขึ้นที่ขดทุติยภูมิที่ต่ออยู่กับโหลด โดยแรงเคลื่อนเหนี่ยวนำที่เกิดขึ้นนั้นจะมีความสัมพันธ์กับการเปลี่ยนแปลงของสนามแม่เหล็กและจำนวนรอบของขดลวด

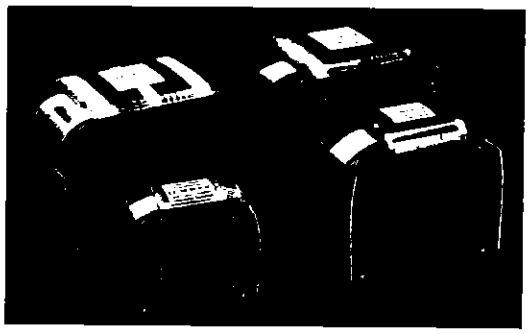


รูปที่ 2.13 แสดงโครงสร้างและสัญลักษณ์ต่าง ๆ ของหม้อแปลงไฟฟ้า[9]

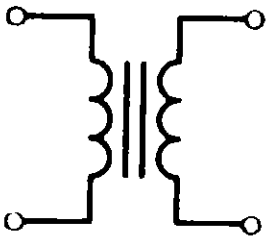
2.7.1 ชนิดของหม้อแปลง

เราสามารถแบ่งชนิดของหม้อแปลงไฟฟ้า ตามแกนของหม้อแปลงได้ 3 แบบ คือ

1. หม้อแปลงชนิดแกนเหล็ก (Iron Core Transformer) หม้อแปลงแบบนี้จะใช้แผ่นเหล็กอ่อนหลาย ๆ แผ่นส่วนใหญ่จะใช้รูปทรงตัว E กับ ตัว I ประกอบกันเป็นแกนซึ่งส่วนใหญ่จะใช้ในงานทั่วไปที่มีความถี่ไม่สูงนัก เช่น หม้อแปลงในงานส่งกำลังไฟฟ้า หรือหม้อแปลงแปลงแรงดันไฟฟ้าตามบ้านเป็นแรงดันต่ำ ๆ ตามที่ต้องการ หม้อแปลงชนิดนี้จะมีประสิทธิภาพสูงที่สุด



(ก)



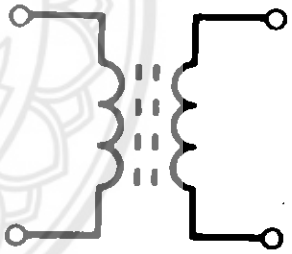
(ข)

รูปที่ 2.14 (ก) แสดงลักษณะหม้อแปลงชนิดแกนเหล็ก[9]

(ข) แสดงสัญลักษณ์หม้อแปลงชนิดแกนเหล็ก

2. หม้อแปลงชนิดแกนเฟอร์ไรท์ (Ferrite Core Transformer) หม้อแปลงชนิดนี้ส่วนใหญ่จะใช้ในงานที่มีความถี่สูง เช่นในเครื่องรับ เครื่องส่ง วิทยุ หรือในวงจรสวิตชิ่ง เพราะไม่สามารถใช้หม้อแปลงชนิดแกนเหล็กได้

(ก)

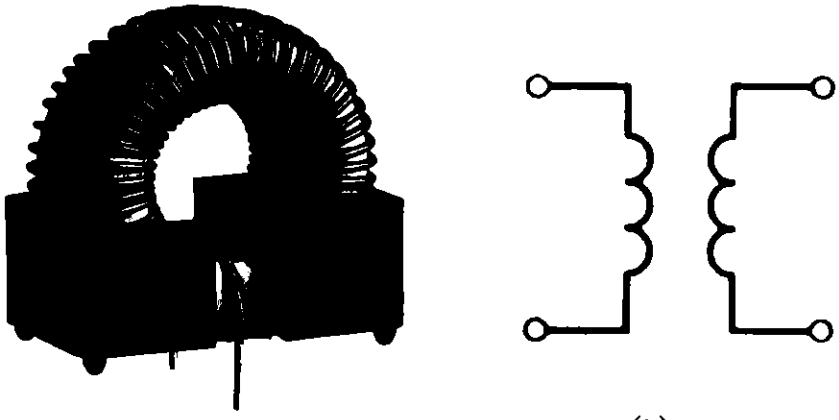


(ข)

รูปที่ 2.15 (ก) แสดงลักษณะหม้อแปลงชนิดแกนเฟอร์ไรท์[9]

(ข) แสดงสัญลักษณ์หม้อแปลงชนิดแกนเฟอร์ไรท์

3. หม้อแปลงชนิดแกนอากาศ (Air Core Transformer) หม้อแปลงชนิดนี้จะใช้ในงานความถี่สูงมาก ๆ เช่นในเครื่องรับ เครื่องส่งวิทยุความถี่สูง เพราะไม่สามารถใช้หม้อแปลงชนิดอื่นได้เนื่องจากจะเกิดความสูญเสียอย่างมาก



(ก) (ข)

รูปที่ 2.16 (ก) แสดงลักษณะหม้อแปลงชนิดแกนอากาศ[9]

(ข) แสดงสัญลักษณ์หม้อแปลงชนิดแกนอากาศ

2.7.2 ข้อควรระวังในการใช้งานหม้อแปลง

1. เลือกชนิดหม้อแปลงให้เหมาะสมกับความถี่ที่ใช้งาน

2. การใช้งานหม้อแปลงควรคำนึงถึงอัตราการทนกำลังของหม้อแปลงด้วย มิฉะนั้นจะทำให้หม้อแปลงไหม้ได้ เนื่องจากมีกระแสไหลสูงเกินไป

3. หม้อแปลงเป็นอุปกรณ์ที่ใช้งานกับไฟกระแสสลับ จึงไม่ควรป้อนไฟกระแสตรงเข้าที่ขั้วหม้อแปลงเพราะอาจจะทำให้หม้อแปลงไหม้ได้

4. ถ้าต่อใช้งานหม้อแปลงในลักษณะ ออโต้ทรานส์ฟอร์เมอร์ (Auto Transformer) ควรระวังถูกไฟฟ้าดูดด้วย เนื่องจากไม่มีการแยกการเชื่อมต่อทางไฟฟ้าจากแหล่งจ่ายไฟบ้าน เหมือนกับหม้อแปลงที่ใช้งานในลักษณะปกติ

5. หม้อแปลงเป็นอุปกรณ์ที่ทำงานด้วยสนามแม่เหล็ก ในขณะใช้งานจึงควรระวังไม่นำไปใกล้อุปกรณ์ที่มีผลต่อสนามแม่เหล็ก เช่น แผ่นดิสก์ เทปเสียง หรือ จอภาพโทรทัศน์

บทที่ 3
วิธีการดำเนินโครงงาน

ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ในกรณีศึกษานี้ เป็นการควบคุมด้วยไมโครคอนโทรลเลอร์ตระกูล MCS-51 และใช้งานผ่านคอมพิวเตอร์ ซึ่งหลังจากได้ศึกษาทฤษฎีที่มีความเกี่ยวข้องและหลักการทำงานของอุปกรณ์ที่ต้องการนำมาใช้งาน ผู้ปฏิบัติงานจึงออกแบบและจัดหาอุปกรณ์ในการทำโครงงานนี้ โดยมีการแบ่งขั้นตอนการทำงานระบบแสดงดังรูปที่ 3.1 และแยกการทำงานออกเป็น 4 ส่วนหลักๆ คือ ส่วนระบบสั่งการ ส่วนระบบควบคุมการทำงาน ส่วนระบบล็อกและปลดล็อกประตูและระบบจอแสดงผลสถานะของผู้ใช้งาน

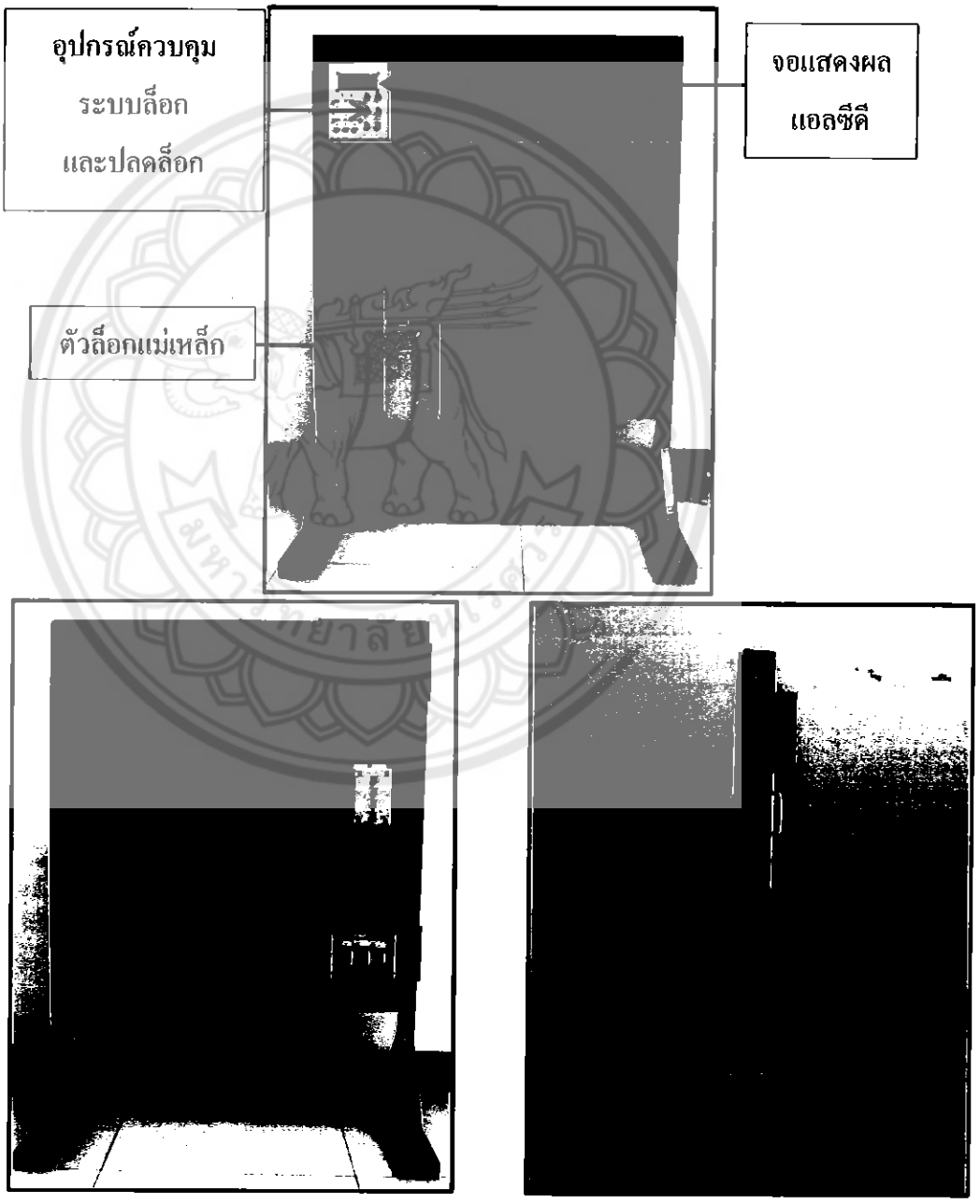
รูปที่ 3.1 แผนภาพแสดงภาพรวมของระบบ (System Overview Diagram)

ในการดำเนินงานเพื่อสร้างระบบการทำงานของโครงงานนี้สามารถ แบ่งออกเป็น 3 ส่วน ได้แก่

1. โครงสร้างของประตูจำลอง
2. วงจรควบคุมการทำงานของระบบ
3. ระบบรับคำสั่ง

3.1 โครงสร้างของประตูจำลอง

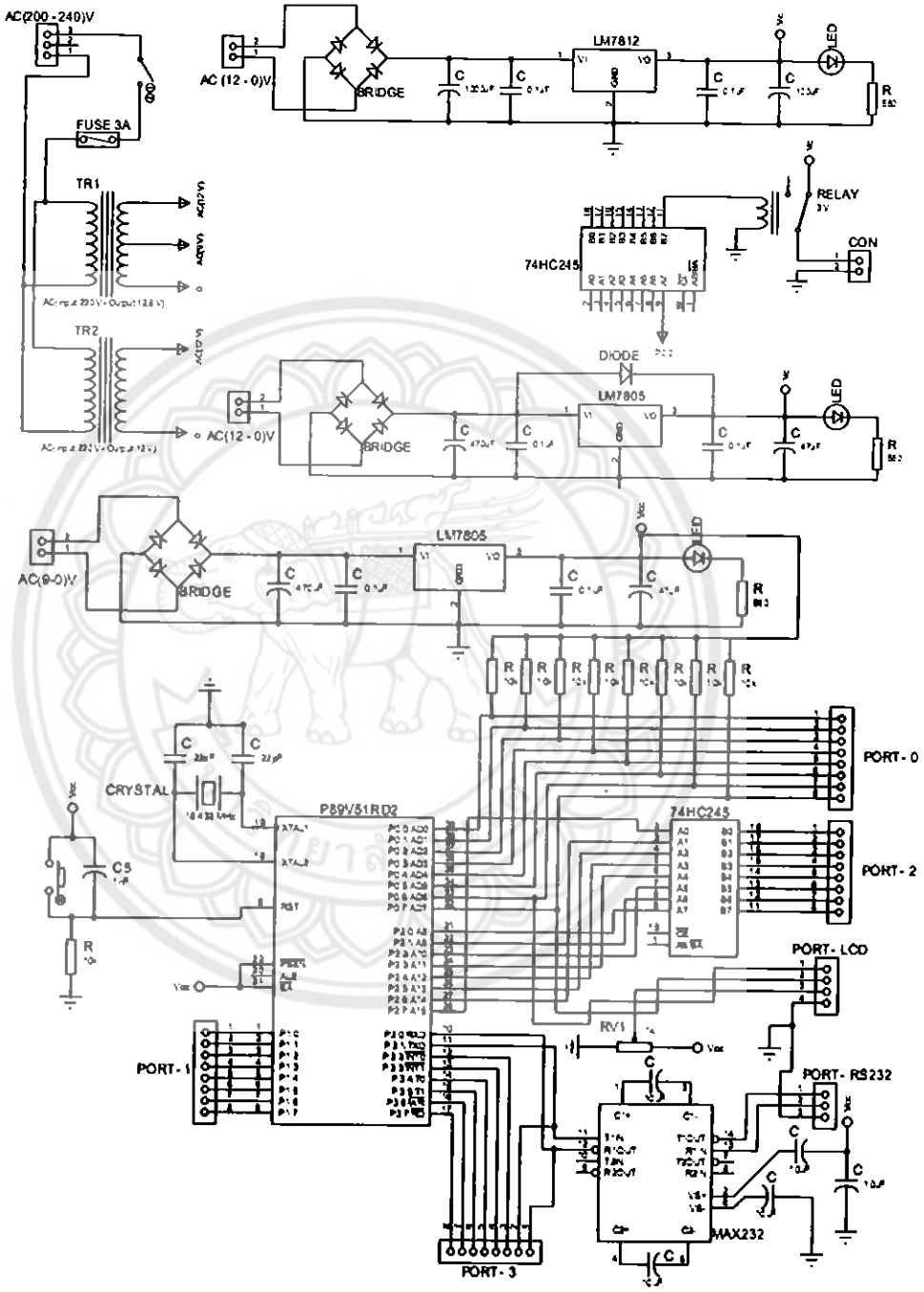
โครงสร้างของประตูจำลองที่สร้างขึ้นจะมีการติดตั้งอุปกรณ์ที่ใช้ควบคุมระบบล็อกและปลดล็อกประตู ระบบจอแสดงผลการทำงานและตัวล็อกแม่เหล็ก โครงสร้างของประตูจำลองด้านหน้า ด้านหลังและด้านข้างขนาดของประตูจำลองกว้าง 70 เซนติเมตร สูง 100 เซนติเมตร หนา 5 เซนติเมตรแสดงดังรูปที่ 3.2



รูปที่ 3.2 แสดงลักษณะโครงสร้างของประตูจำลองด้านหน้า ด้านหลังและด้านข้าง

3.2 วงจรควบคุมการทำงาน

แผนภาพการต่อวงจรควบคุมการทำงานของระบบแสดงดังรูปที่ 3.3 และรูปที่ 3.4



รูปที่ 3.3 แสดงแผนภาพการต่อวงจรของระบบควบคุมการทำงาน[2]

รูปที่ 3.4 แสดงแผนภาพการต่อวงจรของระบบควบคุมการทำงาน

3.2.1 วงจรไมโครคอนโทรลเลอร์

วงจรของไมโครคอนโทรลเลอร์แบ่งออกเป็น 3 ส่วนหลักๆ คือ

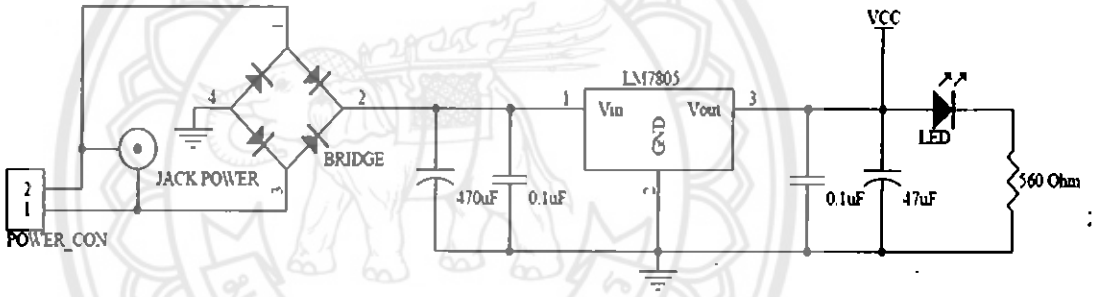
1. วงจรแปลงไฟ และ จ่ายไฟ
2. วงจรตัวไมโครคอนโทรลเลอร์
3. วงจรการติดต่อสื่อสารผ่านพอร์ตอนุกรม (RS-232)

อุปกรณ์ที่ใช้ทั้งหมดของวงจรมีดังต่อไปนี้

1. ไมโคคอนโทรลเลอร์หมายเลข P89V51RD2 + ซ็อกเก็ต 40 ขา	1 ตัว
2. ไอซีหมายเลข MAX232 + ซ็อกเก็ต 16 ขา	1 ตัว
3. คริสตัลความถี่ 18.432 เมกกะเฮิรตซ์	1 ตัว
4. ไอซีหมายเลข LM7805	1 ตัว
5. บริดจ์ไดโอด	1 ตัว
6. ตัวเก็บประจุขนาด 10 ไมโครฟารัด	4 ตัว
7. ตัวเก็บประจุขนาด 47 ไมโครฟารัด	1 ตัว
8. ตัวเก็บประจุขนาด 470 ไมโครฟารัด	1 ตัว
9. ตัวเก็บประจุขนาด 22 พิโกฟารัด	2 ตัว
10. ตัวต้านทานขนาด 10 กิโลโอห์ม	1 ตัว

11. ตัวต้านทานขนาด 560 โอห์ม	1 ตัว
12. หลอดไฟ (LED)	1 หลอด
13. สวิตช์	1 ตัว
14. คอนเนคเตอร์ 8 ขา (สำหรับพอร์ตไมโครคอนโทรลเลอร์)	4 ตัว
15. คอนเนคเตอร์ 4 ขา (สำหรับพอร์ต RS-232)	1 ตัว
16. คอนเนคเตอร์ 2 ขา (สำหรับ Input Power)	1 ตัว
17. บอร์ดอเนกประสงค์แบบไข่ปลา	1 บอร์ด

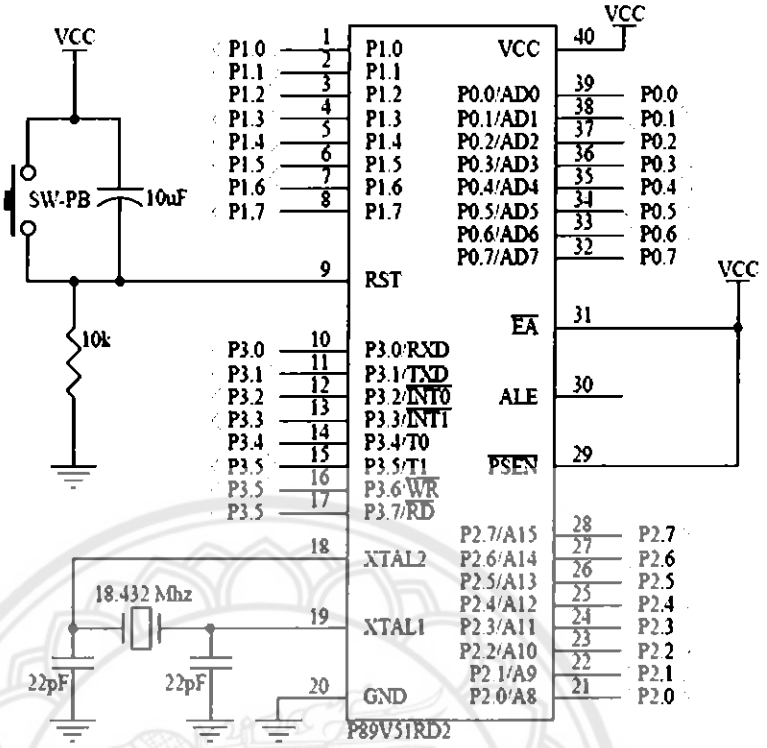
วงจรแปลงไฟ และจ่ายไฟ นั้นสามารถต่อได้ดังรูปวงจรแสดงดังรูปที่ 3.5 จะเห็นได้ว่าใช้ไอซีหมายเลข LM7805 ซึ่งทำหน้าที่แปลงไฟจาก 9-12 โวลต์ ให้เป็นไฟกระแสตรง 5 โวลต์ สามารถจ่ายกระแสได้สูงสุด 1 แอมแปร์



รูปที่ 3.5 วงจรแปลงไฟและจ่ายไฟ[3,6]

ตามรูปที่ 3.5 บริดจ์ไดโอดทำหน้าที่แปลงไฟให้เป็นไฟบวกทำให้เราสามารถจ่ายไฟเข้าวงจรนี้เป็นกระแสสลับ หรือ ไฟกระแสตรงที่ 9-12 โวลต์ ได้โดยไม่จำเป็นต้องคำนึงถึงขั้วของการต่อไฟสำหรับเพาเวอร์ซัพพลายจ่ายไฟกระแสตรงที่ 5 โวลต์ อยู่แล้ว ก็สามารถใช้เพาเวอร์ซัพพลายนั้นกับไมโครคอนโทรลเลอร์ได้โดยตรง โดยไม่ต้องต่อเข้าวงจรแปลงไฟ

วงจรไมโครคอนโทรลเลอร์สามารถต่อได้ตามวงจรในรูปที่ 3.6 โดยมีไฟเลี้ยง 5 โวลต์จ่ายไฟให้กับวงจรดังกล่าว จะเห็นได้ว่าไมโครคอนโทรลเลอร์มีพอร์ตอินพุต/เอาท์พุต แบบขนาด 8 บิตอยู่ทั้งหมด 4 พอร์ต (พอร์ต 0 – พอร์ต 3) แต่ละพอร์ตสามารถทำงานเป็นพอร์ตอินพุต หรือเอาท์พุตก็ได้



รูปที่ 3.6 แสดงวงจรไมโครคอนโทรลเลอร์ P89V51RD2[3,6]

ขา EA (External Access Enable) ขาที่ 31 ใช้เลือกการทำงานของไมโครคอนโทรลเลอร์ จะใช้หน่วยความจำโปรแกรมภายนอกหรือหน่วยความจำโปรแกรมภายใน กรณีเลือกใช้หน่วยความจำโปรแกรมภายนอกขา EA ต้องเป็นลอจิก "0" ส่วนในกรณีเลือกใช้หน่วยความจำโปรแกรมภายในขา EA ต้องเป็นลอจิก "1" สำหรับไมโครคอนโทรลเลอร์ P89V51RD2 นั้นมีหน่วยความจำโปรแกรมภายในแบบแฟลช ขนาด 64 กิโลไบต์ ดังนั้นเราจึงเลือกให้ทำงานจากหน่วยความจำโปรแกรมภายใน ซึ่งต้องต่อกับไฟ 5 โวลต์ ให้เป็นลอจิก "1"

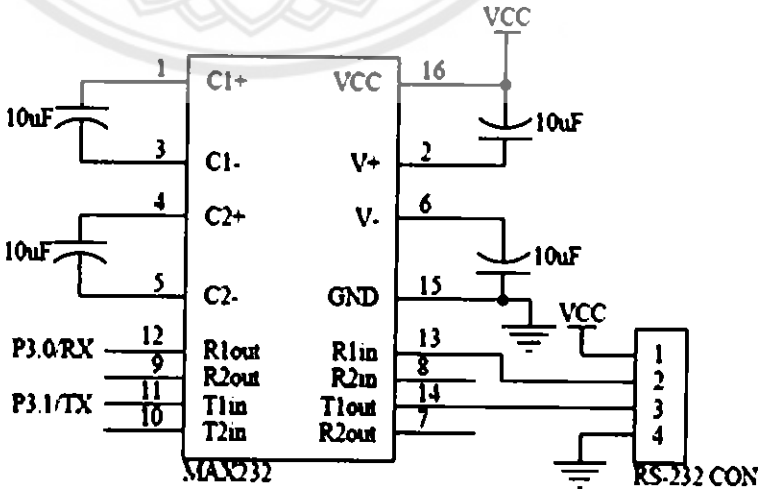
ขา RST (RESET) ขาที่ 9 ไว้สำหรับรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยถ้าขา RST นี้มีสถานะเป็นลอจิกสูง นานกว่าช่วงเวลา 2 แมชชีนไซเคิล จะเป็นการรีเซตการทำงานของไมโครคอนโทรลเลอร์ ดังนั้นเราจึงต่อขา RST ของไมโครคอนโทรลเลอร์เข้ากับสวิตซ์ และตัวเก็บประจุขนาด 10 ไมโครฟารัด

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาให้กับตัวไมโครคอนโทรลเลอร์ ที่เลือกใช้คริสตัลความถี่ 18.432 เมกกะเฮิรตซ์ เนื่องจากเป็นความถี่ที่สามารถนำไปสร้างสัญญาณนาฬิกาให้กับการติดต่อสื่อสารผ่านพอร์ตอนุกรม RS-232

ขา ALE (Address Latch Enable) ขาที่ 30 เป็นขาที่ใช้ควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการต่อใช้งานหน่วยความจำภายนอก แต่เนื่องจากเรายังไม่ได้ต่อใช้งานหน่วยความจำภายนอก ดังนั้นเราจึงปล่อยลอยขา ALE ไว้

ขา PSEN (Program Store Enable) ขาที่ 29 ใช้ส่งสัญญาณเพื่อร้องขอการติดต่อกับหน่วยความจำโปรแกรมภายนอก สำหรับไมโครคอนโทรลเลอร์บางรุ่นขา PSEN นี้ใช้ในการควบคุมสภาวะการทำงานของไมโครคอนโทรลเลอร์สำหรับการโหลดโปรแกรมลงบนไมโครคอนโทรลเลอร์ด้วย

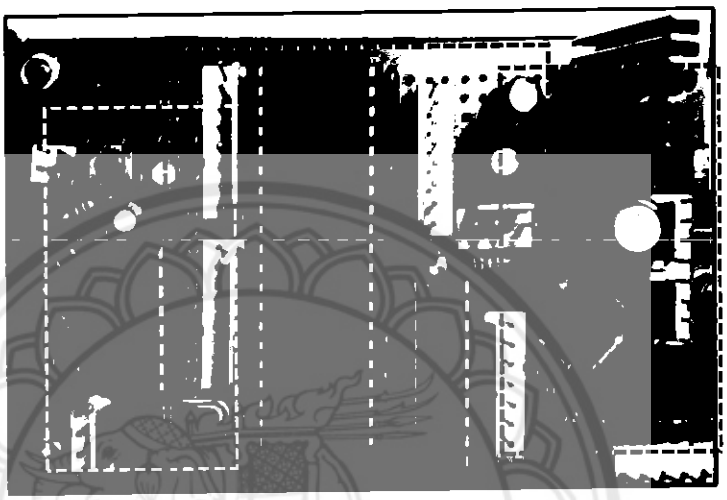
ไมโครคอนโทรลเลอร์ P89V51RD2 มีพอร์ตอนุกรม ซึ่งนอกจากจะใช้สำหรับรับส่งข้อมูลตามปกติแล้ว พอร์ตอนุกรมของ P89V51RD2 ยังสามารถใช้ดาวน์โหลดโปรแกรมลงหน่วยความจำโปรแกรม หรือที่เรียกว่าการโหลดโปรแกรมแบบ ISP ได้อีกด้วยพอร์ตอนุกรมของ P89V51RD2 อยู่ที่พอร์ต P3.0 (RXD) ขาที่ 10 และ P3.1 (TXD) ขาที่ 11 สัญญาณที่ออกมาจากพอร์ตของไมโครคอนโทรลเลอร์นั้นเป็นสัญญาณระดับทีทีแอล (TTL) ซึ่งมีระดับสัญญาณอยู่ที่ 0-5 โวลต์ แต่ในการติดต่อสื่อสารแบบอนุกรมตามมาตรฐาน RS-232 นั้นสัญญาณลอจิก "0" ต้องมีระดับสัญญาณอยู่ที่ 3-15 โวลต์ และลอจิก "1" ต้องมีระดับสัญญาณอยู่ที่ (-3)-(-15) โวลต์ ดังนั้นในการใช้งานสื่อสารตามมาตรฐาน RS-232 เราจึงต้องใช้วงจรสื่อสารผ่านพอร์ตอนุกรมซึ่งมีไอซี MAX232 เป็นตัวปรับระดับสัญญาณจากระดับทีทีแอล (TTL) ให้เป็นไปตามมาตรฐานของ RS-232 โดยสามารถต่อวงจรได้แสดงดังรูปที่ 3.7



รูปที่ 3.7 แสดงวงจรติดต่อผ่านพอร์ตสื่อสารอนุกรม RS-232[3,6]

การสร้างบอร์ดไมโครคอนโทรลเลอร์มีลักษณะการจัดเรียงตำแหน่งของอุปกรณ์ได้หลายแบบแต่จะมีส่วนหลักๆ แบ่งออกเป็น 3 ส่วน แสดงดังรูปที่ 3.8

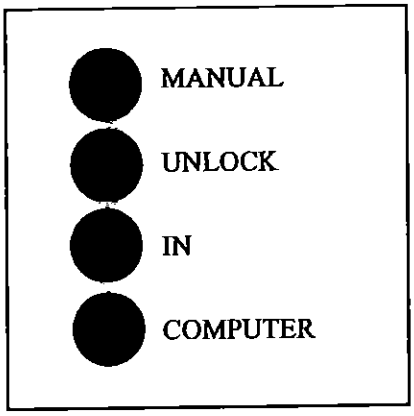
- วงจรแปลงไฟเลี้ยงและจ่ายไฟบริเวณในกรอบสีเขียว
- วงจรตัวไมโครคอนโทรลเลอร์บริเวณในกรอบสีแดง
- วงจรติดต่อสื่อสารผ่านพอร์ตอนุกรม (RS-232) บริเวณในกรอบสีเหลือง



รูปที่ 3.8 แสดงลักษณะบอร์ดไมโครคอนโทรลเลอร์ P89V51RD2

3.3 ระบบสั่งการ

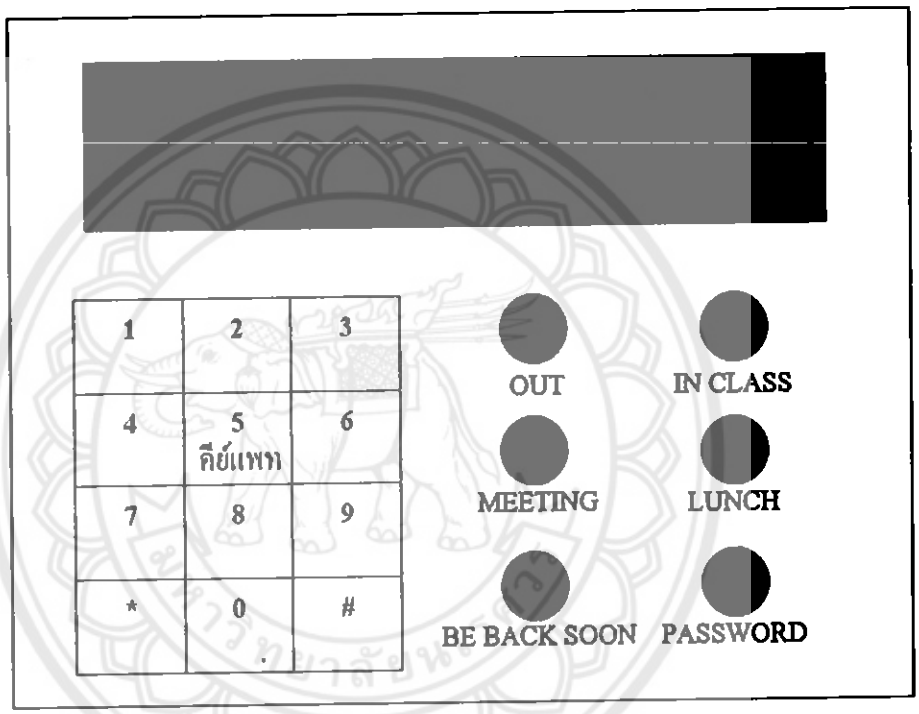
ในการควบคุมการทำงานของระบบล็อกและปลดล็อกประตู การใช้งานสามารถเลือกระบบควบคุมได้ 2 ระบบ คือ ระบบสั่งการควบคุมด้วยมือและระบบสั่งการด้วยคอมพิวเตอร์ ซึ่งผู้ใช้งานเป็นผู้กำหนดระบบของการทำงาน โดยมีปุ่มรับคำสั่งอยู่ภายในห้องของผู้ใช้งานแสดงดังรูปที่ 3.9



รูปที่ 3.9 แสดงลักษณะปุ่มกดเลือกระบบและรับคำสั่งการควบคุมด้วยมือ

3.3.1 ระบบสั่งการควบคุมด้วยมือ (MANUAL)

การควบคุมการทำงานของระบบล็อกและปลดล็อกประตูนั้น การใช้โหมดควบคุมด้วยมือเพื่อใช้เป็นคำสั่งให้กับส่วนประมวลผลโดยตรง ทำให้การทำงานของระบบล็อกและปลดล็อกประตูสะดวกขึ้นและกรณีที่ไม่ได้ใช้งานคอมพิวเตอร์ผู้ใช้ยังสามารถควบคุมระบบล็อกและปลดล็อกประตูได้ ลักษณะรูปแบบของปุ่มกดรับคำสั่งที่อยู่ภายในห้องบริเวณข้างประตูผู้ใช้แสดงดังรูปที่ 3.9 และรูปแบบของปุ่มกดรับคำสั่งที่อยู่ภายนอกห้องติดบริเวณข้างประตูของผู้ใช้แสดงดังรูปที่ 3.10



รูปที่ 3.10 แสดงปุ่มรับคำสั่งการควบคุมด้วยมือ

ลักษณะการใช้งานปุ่มกดต่างๆในการควบคุมด้วยมือมีรายละเอียดดังนี้

กดปุ่ม (UNLOCK) การเลือกใช้คำสั่ง (UNLOCK) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการปลดล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (UNLOCK) แปลว่า อนุญาตให้เข้าพบ

กดปุ่ม (IN) การเลือกใช้คำสั่ง (IN) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตูและหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ

(IN) แปลว่า อยู่ในห้อง

กดปุ่ม (OUT) การเลือกใช้คำสั่ง (OUT) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (OUT) แปลว่า ไปข้างนอกไม่อยู่ในห้อง

กดปุ่ม (IN CLASS) การเลือกใช้คำสั่ง (IN CLASS) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (IN CLASS) แปลว่า ไปห้องเรียน(สอน)

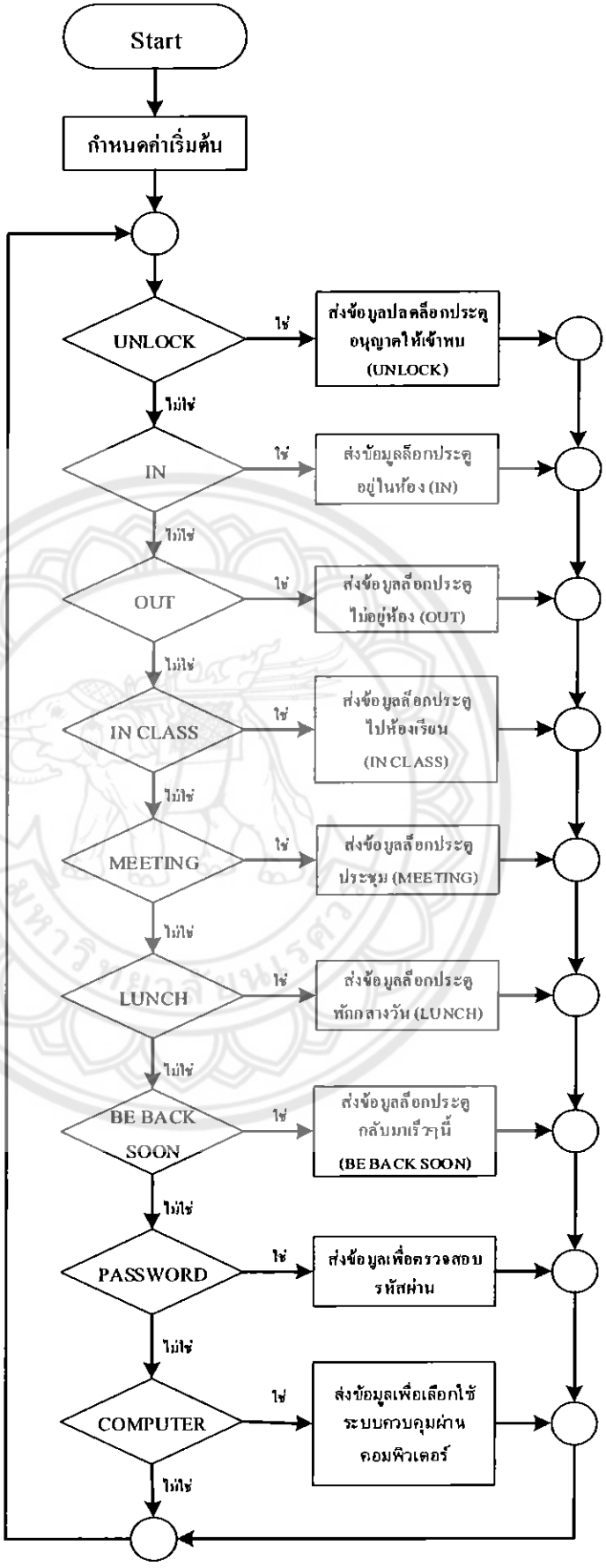
กดปุ่ม (MEETING) การเลือกใช้คำสั่ง (MEETING) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (MEETING) แปลว่า ประชุม

กดปุ่ม (LUNCH) การเลือกใช้คำสั่ง (LUNCH) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (LUNCH) แปลว่า พักกลางวัน

กดปุ่ม (BE BACK SOON) การเลือกใช้คำสั่ง (BE BACK SOON) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (BE BACK SOON) แปลว่า กลับมาเร็วๆนี้

กดปุ่ม (PASSWORD) การเลือกใช้คำสั่ง (PASSWORD) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์จะทำการตรวจสอบรหัสผ่านจากการกดรหัสบนคีย์แพทและแสดงผลผ่านหน้าจอแอลซีดี ซึ่งการกดรหัสผ่านครบ 4 ตัวแล้วให้กด # เพื่อยืนยันรหัสผ่าน

ระบบควบคุมด้วยมือ สามารถแบ่งขั้นตอนการทำงานของโปรแกรมได้แสดงดังรูปที่ 3.11



รูปที่ 3.11 แผนภาพการทำงานของระบบควบคุมด้วยมือ

ระบบจะสั่งการให้ตัวล็อกแม่เหล็กทำงานหรือไม่ทำงาน จะสามารถสังเกตได้จากหลอดแอลอีดีที่ตัวล็อกแม่เหล็ก โดยลักษณะการแสดงผลของหลอดแอลอีดีเป็นดังนี้

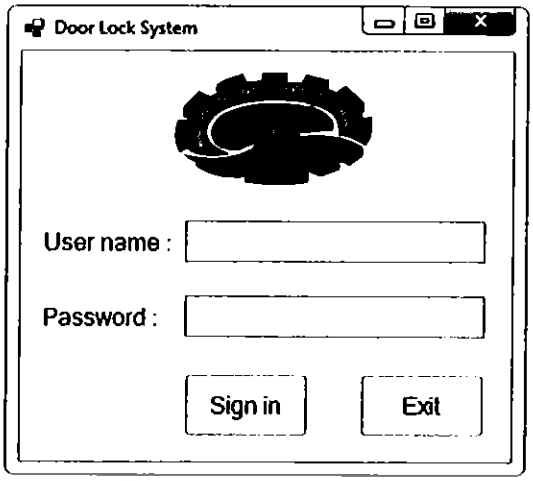
- หลอดแอลอีดีแสดงสีเขียว แผ่นโลหะติดกับตัวล็อกแม่เหล็กสนิททำให้ประตูล็อก
- หลอดแอลอีดีแสดงสีแดง แผ่นโลหะติดกับตัวล็อกแม่เหล็กไม่สนิททำให้ประตูล็อกแต่สามารถปลดล็อก เมื่อใช้แรงดึงประตู
- หลอดแอลอีดีไม่แสดงผล แผ่นโลหะไม่ติดกับตัวล็อกแม่เหล็กทำให้ประตูปลดล็อก

3.3.2 ระบบสั่งการด้วยคอมพิวเตอร์ (COMPUTER)

ในการควบคุมการทำงานของระบบล็อกและปลดล็อกประตูนั้น จะใช้หน้าต่างรับคำสั่งโปรแกรมที่สร้างขึ้นจากโปรแกรมวิชัวร์เบสิก (Visual Basic) โดยที่ผู้ใช้สามารถเลือกสถานะในการทำงานได้จากโปรแกรมรับคำสั่งจากผู้ใช้ผ่านทางคอมพิวเตอร์ที่สร้างขึ้น เพื่อใช้ในการสื่อสารระหว่างพอร์ตอนุกรมของคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ โดยใช้ชุดแปลงสัญญาณพอร์ตอนุกรม RS-232 เป็นตัวกลางส่งคำสั่งควบคุมไปยังระบบล็อกและปลดล็อกประตู ในการสื่อสารระหว่างพอร์ตอนุกรม RS-232

ส่วนของโปรแกรมรับคำสั่งจากผู้ใช้ผ่านทางคอมพิวเตอร์แสดงดังรูปที่ 3.12 โปรแกรมจะตรวจสอบการเข้าสู่ระบบของผู้ใช้งาน โดยตรวจสอบจากชื่อผู้ใช้งานและรหัสผ่านที่ตั้งไว้แสดงดังรูปที่ 3.13 หากใส่ชื่อผู้ใช้งานหรือรหัสผ่านไม่ถูกต้องจะมีหน้าต่างแสดงข้อความเตือน เพื่อให้ผู้ใช้ตรวจสอบและเข้าสู่ระบบอีกครั้งแสดงดังรูปที่ 3.14 หากใส่ชื่อผู้ใช้งานหรือรหัสผ่านถูกต้องจะสามารถเข้าสู่ระบบได้แสดงดังรูปที่ 3.15

รูปที่ 3.12 แสดงลักษณะไอคอนโปรแกรมรับคำสั่ง(Door Lock System)

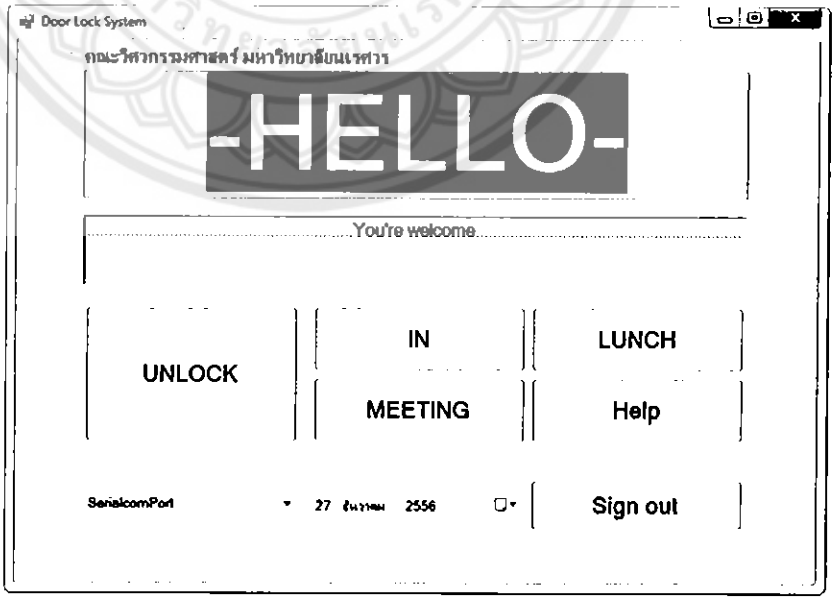


รูปที่ 3.13 แสดงโปรแกรมรับคำสั่งจากผู้ใช้ผ่านคอมพิวเตอร์

รูปที่ 3.14 แสดงการเข้าสู่ระบบด้วยชื่อผู้ใช้งานหรือรหัสผ่านไม่ถูกต้อง

รูปที่ 3.15 แสดงการเข้าสู่ระบบด้วยชื่อผู้ใช้งานหรือรหัสผ่านถูกต้อง

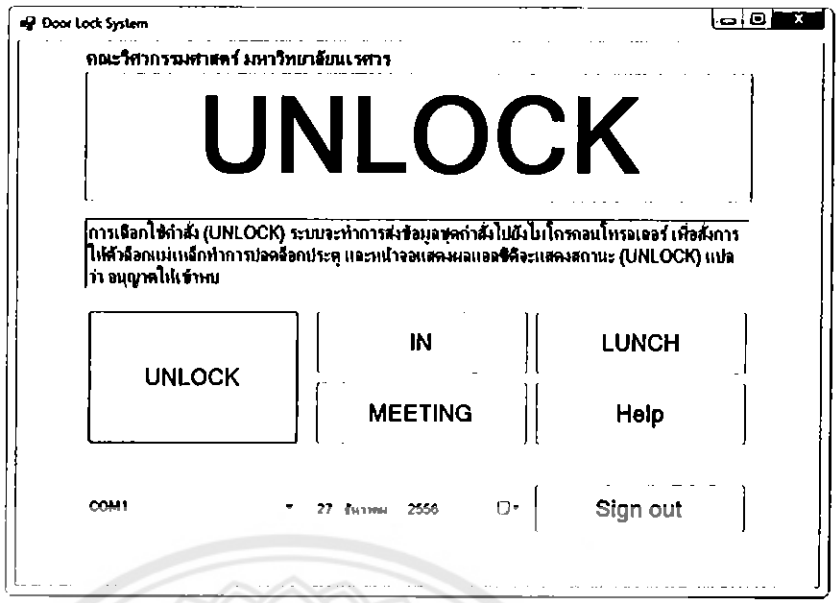
โปรแกรมรับคำสั่งส่งข้อมูลคำสั่งการทำงานไปยังระบบล็อกและปลดล็อกประตู การคลิกเลือกไอคอนในโปรแกรมรับคำสั่งสามารถเลือกใช้ได้ตามความสะดวก แสดงดังรูปที่ 3.16



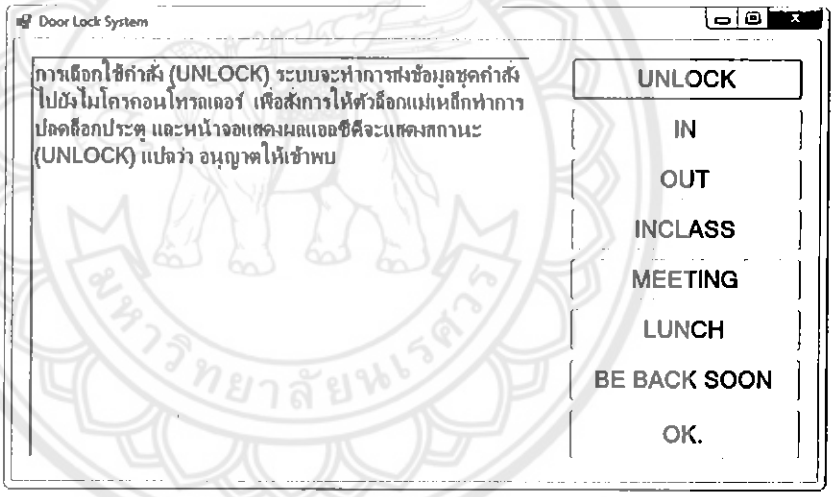
รูปที่ 3.16 แสดงหน้าต่างโปรแกรมและไอคอนการรับคำสั่ง

ลักษณะไอคอนและการใช้งานปุ่มต่างๆในโปรแกรมรับคำสั่งมีรายละเอียดดังนี้

กดปุ่ม (UNLOCK)	การเลือกใช้คำสั่ง (UNLOCK) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการปลดล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (UNLOCK) แปลว่า อนุญาตให้เข้าพบ แสดงดังรูปที่ 3.17
กดปุ่ม (IN)	การเลือกใช้คำสั่ง (IN) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (IN) แปลว่า อยู่ในห้อง
กดปุ่ม (MEETING)	การเลือกใช้คำสั่ง (MEETING) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (MEETING) แปลว่า ประชุม
กดปุ่ม (LUNCH)	การเลือกใช้คำสั่ง (LUNCH) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (LUNCH) แปลว่า พักกลางวัน
กดปุ่ม (HELP)	การเลือกใช้คำสั่ง (HELP) โปรแกรมจะแสดงหน้าต่างขึ้นมา เพื่อช่วยอธิบายลักษณะการทำงานของปุ่มไอคอนรับคำสั่งต่างๆ ภายในหน้าโปรแกรมรับคำสั่ง แสดงดังรูปที่ 3.18
กด (Serial comport)	การเลือกพอร์ต USB 2.0 เพื่อกำหนดใช้เป็นพอร์ตในการติดต่อ สื่อสารผ่านพอร์ตอนุกรม
กดปุ่ม (Sign out)	การเลือกใช้คำสั่ง (Sign out) โปรแกรมจะแสดงหน้าขึ้นมา เพื่อให้ยืนยันการออกจะระบบ ใช่ หรือ ไม่ใช่

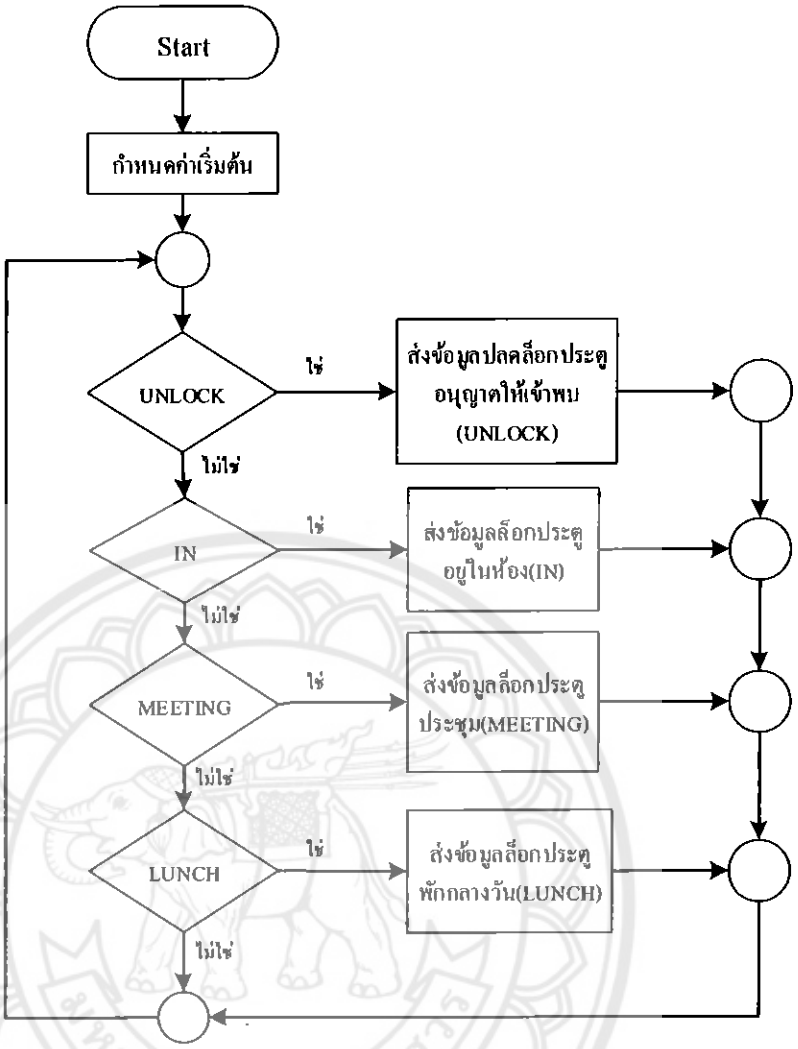


รูปที่ 3.17 แสดงหน้าต่างโปรแกรมและไอคอนการรับคำสั่ง (UNLOCK)



รูปที่ 3.18 แสดงหน้าต่างโปรแกรมและไอคอนการรับคำสั่ง (HELP)

โปรแกรมควบคุมมีการกำหนดค่าเริ่มต้นของโปรแกรมเป็นการกำหนดความเร็วการส่งข้อมูลที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ที่ควบคุม การแบ่งขั้นตอนการทำงานของโปรแกรมดังรูปที่ 3.19



รูปที่ 3.19 แผนภาพการทำงานของโปรแกรมควบคุมผ่านคอมพิวเตอร์

บทที่ 4

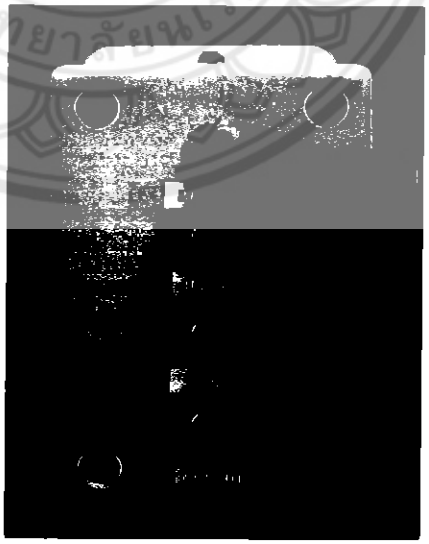
ผลการทดสอบและผลการวิเคราะห์

หลังจากศึกษาทฤษฎี หลักการทำงานและลงมือสร้างโครงสร้างจำลองระบบล็อกและปลดล็อกประตูแล้ว ในบทนี้จะเป็นการทดสอบการทำงานของระบบล็อกและปลดล็อกประตู โดยแบ่งการทดสอบเป็น 6 หัวข้อ ดังนี้

1. การทดสอบการทำงานแบบสั่งการจากระบบควบคุมด้วยมือ
2. การทดสอบการทำงานแบบสั่งการจากระบบควบคุมผ่านคอมพิวเตอร์
3. การทดสอบแรงยึดของตัวล็อกแม่เหล็กที่ระดับแรงดันไฟฟ้า 12 โวลต์
4. การทดสอบวัดพิกัดแรงดันไฟฟ้าของสายตัวนำเทียบกับระยะทาง
5. การทดสอบการใช้งานระบบควบคุมด้วยมือต่อเนื่องเป็นเวลา 48 ชั่วโมง
6. การทดสอบการใช้งานระบบควบคุมผ่านคอมพิวเตอร์ต่อเนื่องเป็นเวลา 8 ชั่วโมง

4.1 การทดสอบการทำงานแบบสั่งการจากระบบควบคุมด้วยมือ

การทดสอบการทำงานแบบสั่งการจากระบบควบคุมด้วยมือ ทดสอบการทำงานของปุ่มกดเลือกสถานะ การประมวลผลคำสั่งและการแสดงผลผ่านหน้าจอแอลซีดีแสดงดังรูปที่ 4.1

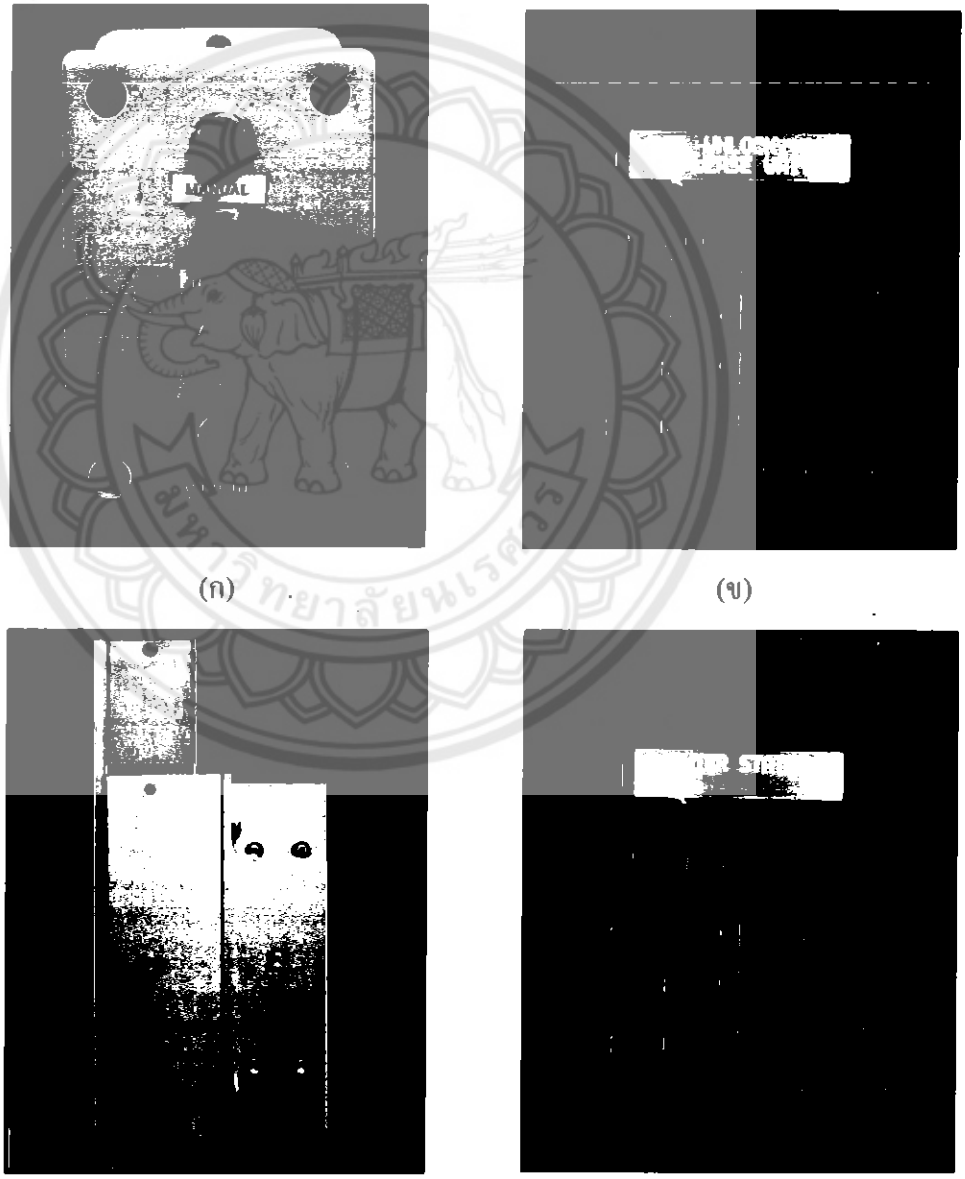


รูปที่ 4.1 แสดงปุ่มกดเลือกระบบการทำงานติดตั้งอยู่ภายในห้องทำงาน

ปุ่มกดเลือกสถานะคำสั่งของระบบควบคุมด้วยมือ สามารถแบ่งออกเป็น 8 สถานะคำสั่ง ได้แก่

1. กดปุ่ม (UNLOCK)
2. กดปุ่ม (IN)
3. กดปุ่ม (OUT)
4. กดปุ่ม (IN CLASS)
5. กดปุ่ม (MEETING)
6. กดปุ่ม (LUNCH)
7. กดปุ่ม (BE BACK SOON)
8. กดปุ่ม (PASSWORD)

ระบบควบคุมด้วยมือ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (UNLOCK) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.2



รูปที่ 4.2 แสดงการทำงานของคำสั่ง (UNLOCK)

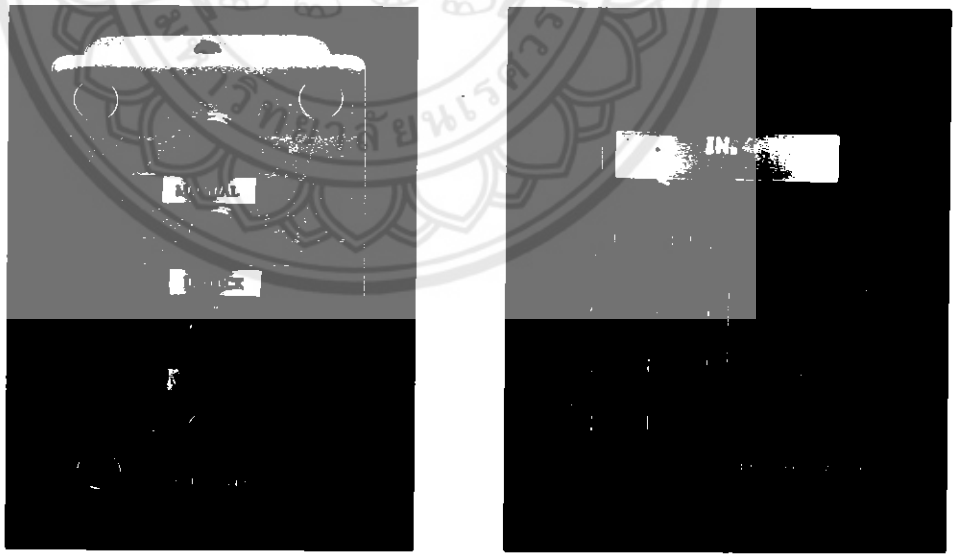
(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (UNLOCK) ที่ติดตั้งอยู่ภายในห้องทำงาน

(ข) การทำงานของจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (UNLOCK) จอแสดงผลแอลซีดีจะแสดงข้อความ (UNLOCK) ขึ้นมาเพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะคำสั่งการทำงานของระบบ

(ค) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (UNLOCK) ตัวล็อกแม่เหล็กจะปลดล็อกประตูและอาจจะสังเกตได้จากหลอดแอลอีดีที่บริเวณตัวล็อกแม่เหล็กไม่แสดงผล จากนั้นระบบจะมีการหน่วงเวลาไว้ 5 วินาทีในการปลดล็อกประตู เมื่อเวลาครบ 5 วินาทีตัวล็อกแม่เหล็กจะทำงานในสถานะล็อกตามปกติและอาจจะสังเกตได้จากหลอดแอลอีดีของตัวล็อกแม่เหล็กจะแสดงสีเขียว

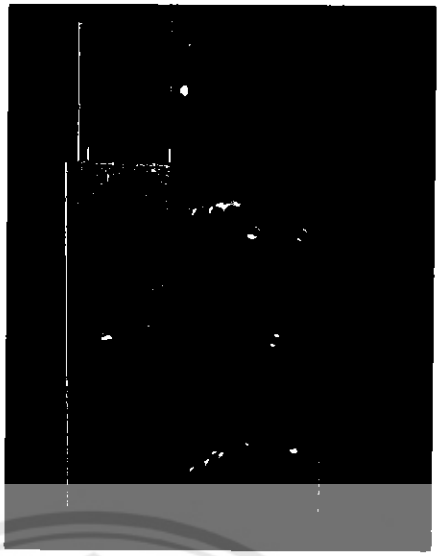
(ง) หลังจากระบบทำงานตามสถานะคำสั่ง (UNLOCK) แล้ว ตัวล็อกแม่เหล็กทำงานสถานะล็อกตามปกติและหน้าจอแสดงผลจะแสดงข้อความ (ENTER STATUS) เพื่อให้ผู้ใช้เลือกสถานะคำสั่งอื่นๆ ต่อไป

ระบบควบคุมด้วยมือ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (IN) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.3



(ก) (ข)

รูปที่ 4.3 แสดงการทำงานของปุ่มกดคำสั่ง (IN)



(ค)

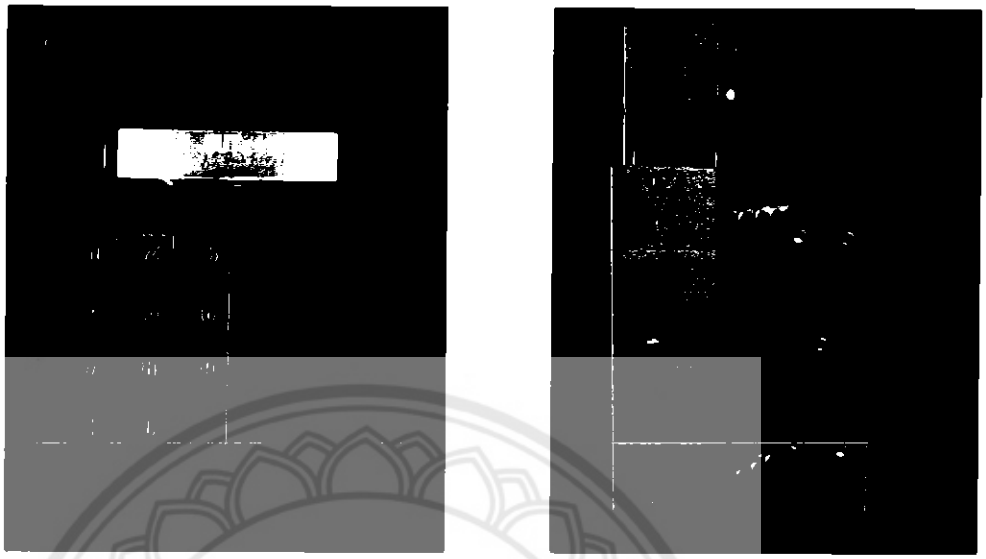
รูปที่ 4.3(ต่อ) แสดงการทำงานของปุ่มกดคำสั่ง (IN)

(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (IN) ที่ติดตั้งอยู่ภายในห้องทำงาน

(ข) การทำงานของจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (IN) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (IN) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะคำสั่งการทำงานของระบบ

(ค) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (IN) ตัวล็อกแม่เหล็กจะล็อกประตู และอาจจะสังเกตได้จากหลอดแอลอีดีของตัวล็อกแม่เหล็กจะแสดงสีเขียว

ระบบควบคุมด้วยมือ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (OUT) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.4



(ก) (ข)

รูปที่ 4.4 แสดงการทำงานของปุ่มกดคำสั่ง (OUT)

(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (OUT) ที่ติดตั้งอยู่ภายนอกห้อง ระบบจะแสดงการทำงานของหน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (OUT) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (OUT) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะการทำงานของระบบ

(ข) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (OUT) ตัวล็อกแม่เหล็กจะล็อกประตู และอาจจะสังเกตได้จากหลอดแอลอีดีของตัวล็อกแม่เหล็กจะแสดงสีเขียว

ระบบควบคุมด้วยมือ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (IN CLASS) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.5

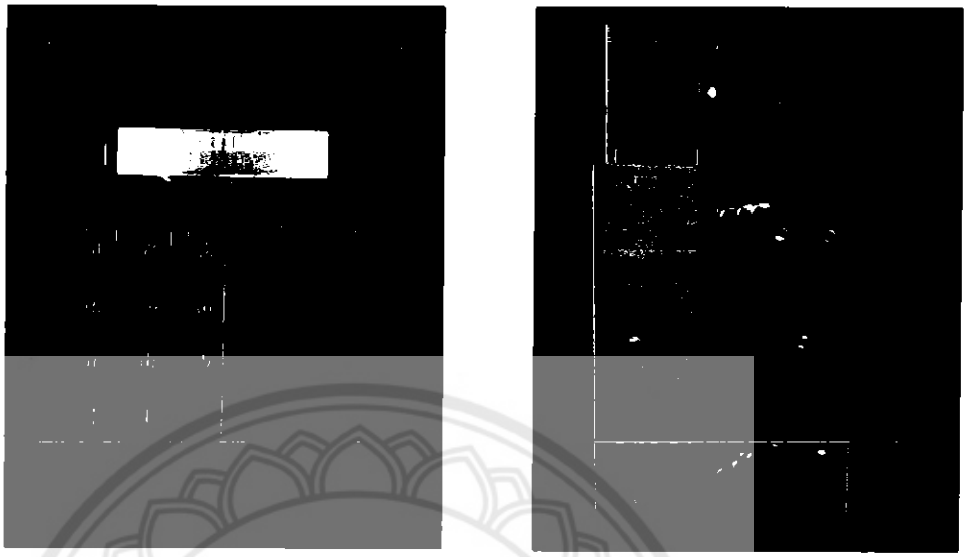
(ก) (ข)

รูปที่ 4.5 แสดงการทำงานของปุ่มกดคำสั่ง (IN CLASS)

(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (IN CLASS) ที่ติดตั้งอยู่ภายนอกห้อง ระบบจะแสดงการทำงานของหน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (IN CLASS) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (IN CLASS) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะการทำงานของระบบ

(ข) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (IN CLASS) ตัวล็อกแม่เหล็กจะล็อกประตู และอาจจะสังเกตได้จากหลอดแอลอีดีของตัวล็อกแม่เหล็กจะแสดงสีเขียว

ระบบควบคุมด้วยมือ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (MEETING) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.6



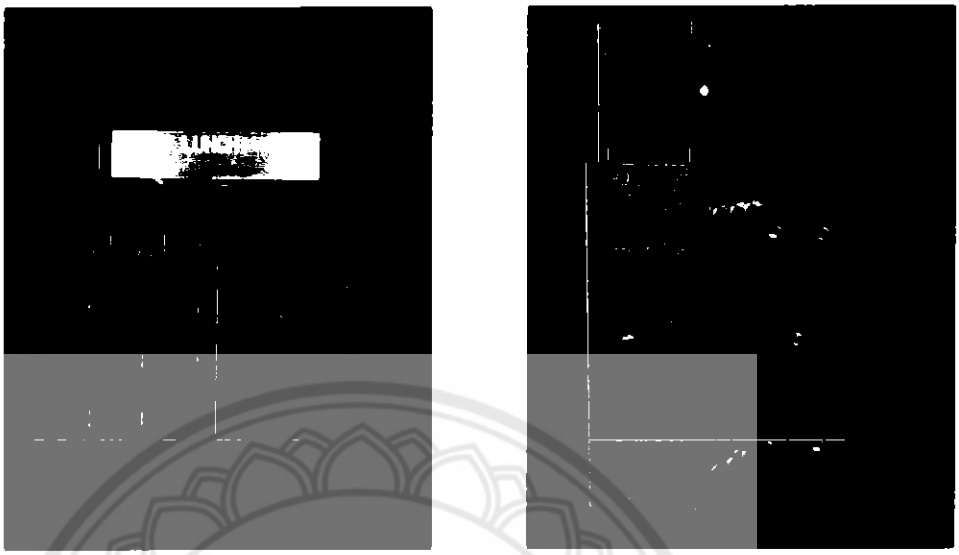
(ก) (ข)

รูปที่ 4.6 แสดงการทำงานของปุ่มกดคำสั่ง (MEETING)

(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (MEETING) ที่ติดตั้งอยู่ภายนอกห้อง ระบบจะแสดงการทำงานของหน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (MEETING) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (MEETING) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะการทำงานของระบบ

(ข) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (MEETING) ตัวล็อกแม่เหล็กจะล็อกประตู และอาจจะสังเกตได้จากหลอดแอลอีดีของตัวล็อกแม่เหล็กจะแสดงสีเขียว

ระบบควบคุมด้วยมือ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (LUNCH) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.7



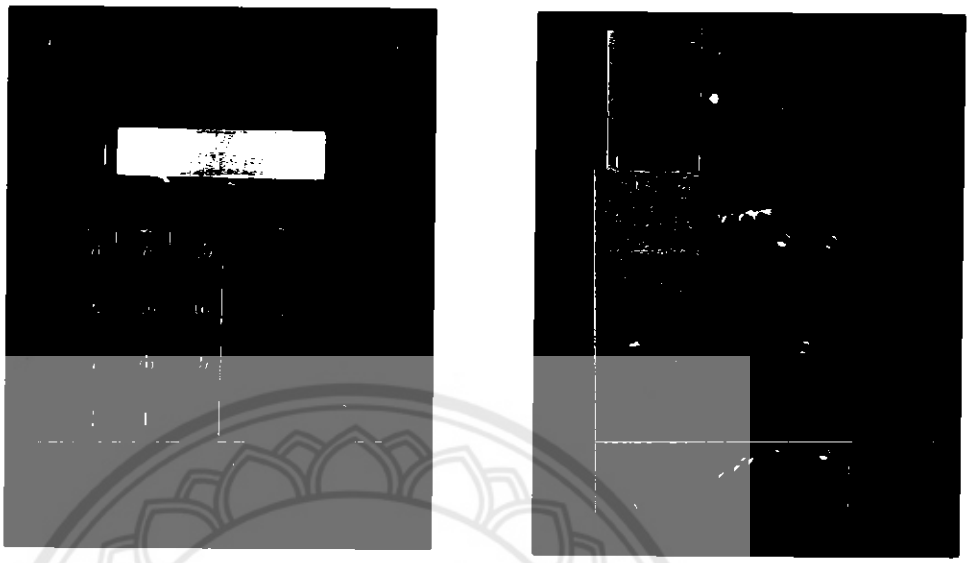
(ก) (ข)

รูปที่ 4.7 แสดงการทำงานของปุ่มกดคำสั่ง (LUNCH)

(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (LUNCH) ที่ติดตั้งอยู่ภายนอกห้อง ระบบจะแสดงการทำงานของหน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (LUNCH) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (LUNCH) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะการทำงานของระบบ

(ข) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (LUNCH) ตัวล็อกแม่เหล็กจะล็อกประตู และอาจจะสังเกตได้จากหลอดแอลอีดีของตัวล็อกแม่เหล็กจะแสดงสีเขียว

ระบบควบคุมด้วยมือ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (BE BACK SOON) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.8



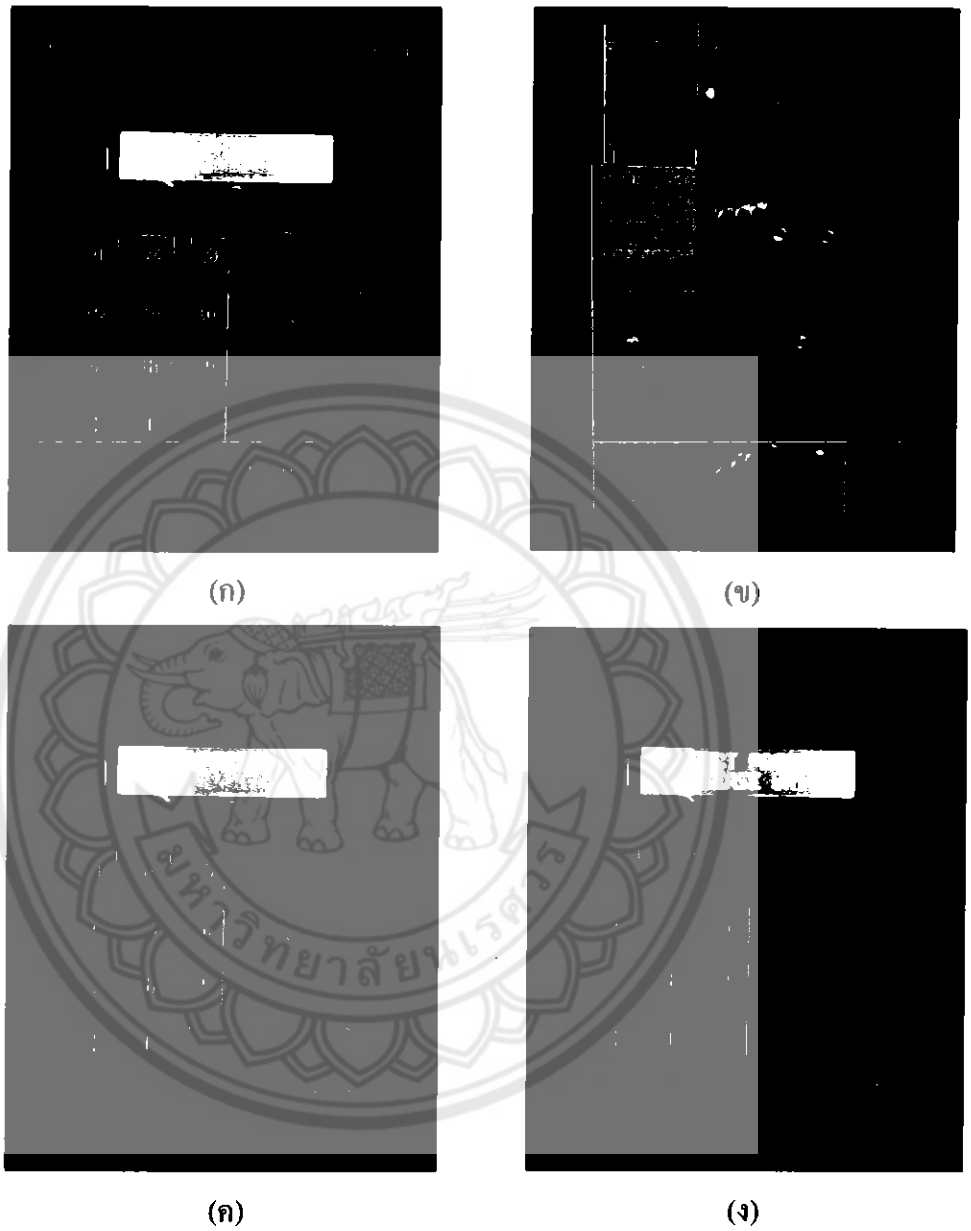
(ก) (ข)

รูปที่ 4.8 แสดงการทำงานของปุ่มกดคำสั่ง (BE BACK SOON)

(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (BE BACK SOON) ที่ติดตั้งอยู่ภายนอกห้อง ระบบจะแสดงการทำงานของหน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (BE BACK SOON) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (BE BACK SOON) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะการทำงานของระบบ

(ข) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (BE BACK SOON) ตัวล็อกแม่เหล็กจะล็อกประตูและ อาจจะสังเกตได้จากหลอดแอลอีดีของตัวล็อกแม่เหล็กจะแสดงสีเขียว

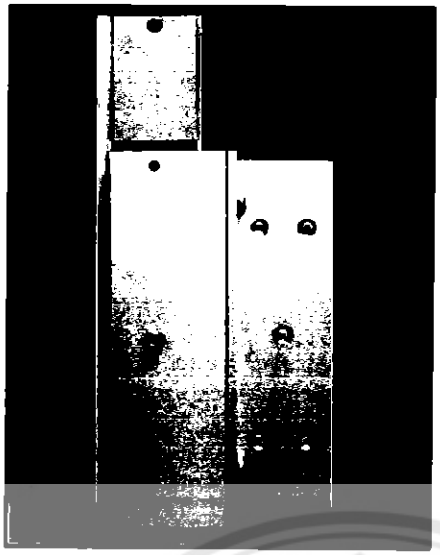
ระบบควบคุมด้วยมือ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (PASSWORD) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.9



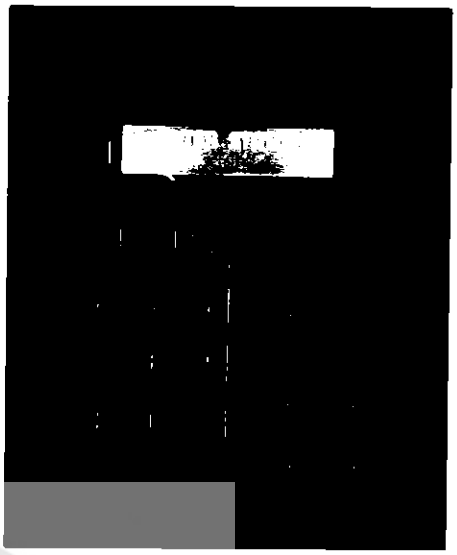
(ก) (ข)

(ค) (ง)

รูปที่ 4.9 แสดงการทำงานของปุ่มกดคำสั่ง (PASSWORD)



(จ)



(ฉ)

รูปที่ 4.9(ต่อ) แสดงการทำงานของปุ่มกดคำสั่ง (PASSWORD)

(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (PASSWORD) ที่ติดตั้งอยู่ภายนอกห้องทำงาน หน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะจะแสดงข้อความ (ENTER PASSWORD)

(ข) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (PASSWORD) ตัวล็อกแม่เหล็กทำงานในสถานะล็อกและอาจจะสังเกตได้จากหลอดหลอดอีดี แสดงผลสีเขียว

(ค) กดตัวเลข 4 หลักป้อนที่คีย์แพท เพื่อตรวจรหัสผ่านกรณีที่ผู้ใช้ต้องการเข้าห้อง รหัสผ่านมี 4 หลัก เมื่อใส่รหัสผ่านครบทั้ง 4 หลักแล้วกด # เพื่อยืนยันการใส่รหัสผ่าน

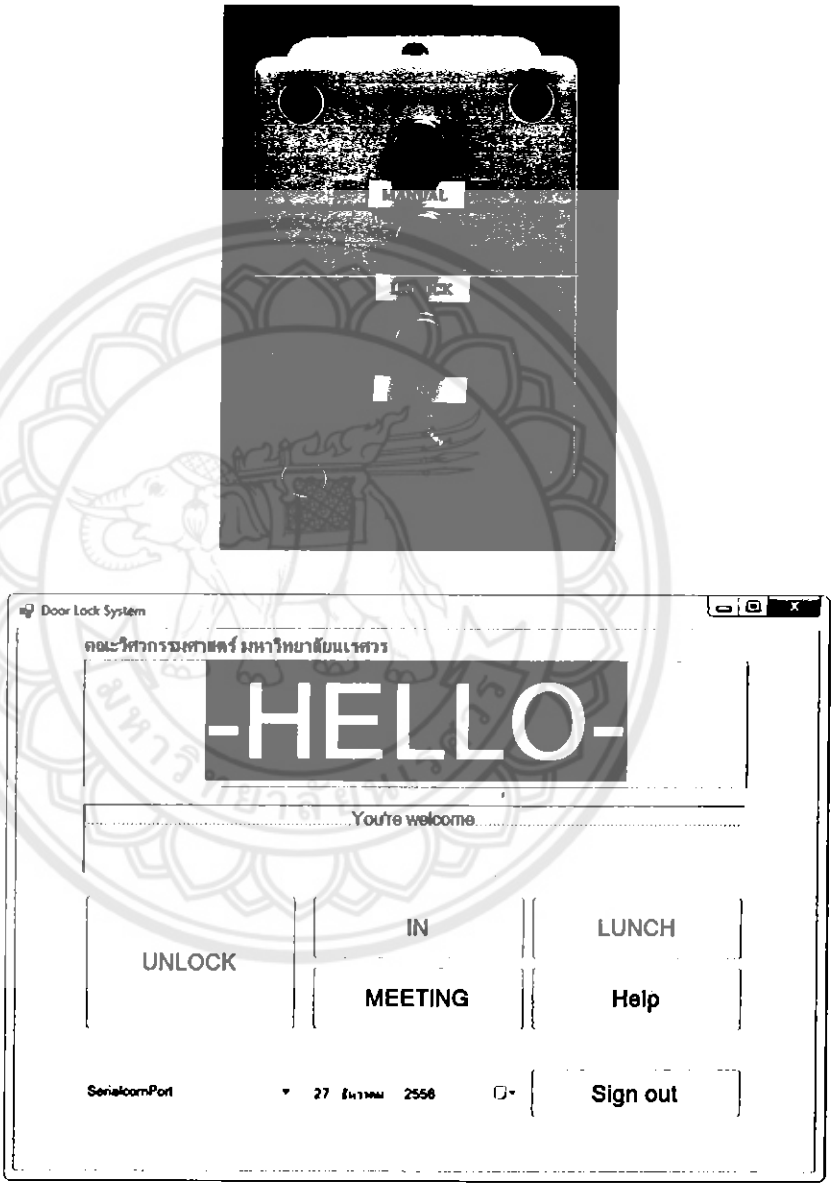
(ง) การทำงานของหน้าจอแสดงผลหลังจากตรวจสอบรหัสผ่านถูกต้อง จะแสดงข้อความ (CORRECT PASSWORD) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบ

(จ) การทำงานของตัวล็อกแม่เหล็กหลังจากตรวจสอบรหัสผ่านถูกต้อง ตัวล็อกแม่เหล็กปลดล็อกประตูและหน่วงเวลาไว้ 5 วินาทีสังเกตจากหลอดหลอดอีดีไม่แสดงผล จากนั้นระบบจะสั่งให้ตัวล็อกแม่เหล็กทำงานสถานะล็อกตามปกติ

(ฉ) หลังจากระบบทำงานตามสถานะคำสั่ง (PASSWORD) แล้ว ตัวล็อกแม่เหล็กทำงานสถานะล็อกตามปกติและหน้าจอแสดงผลจะแสดงข้อความ (ENTER STATUS) เพื่อให้ผู้ใช้เลือกสถานะคำสั่งอื่นๆ ต่อไป

4.2 การทดสอบการทำงานแบบสั่งการผ่านคอมพิวเตอร์

ในการทดสอบการทำงานแบบสั่งการผ่านคอมพิวเตอร์ จะทดสอบการทำงานของไอคอนเลือกสถานะในโปรแกรมรับคำสั่ง การประมวลผลคำสั่งและการแสดงผลออกทางหน้าจอแอลซีดี แสดงดังรูปที่ 4.10

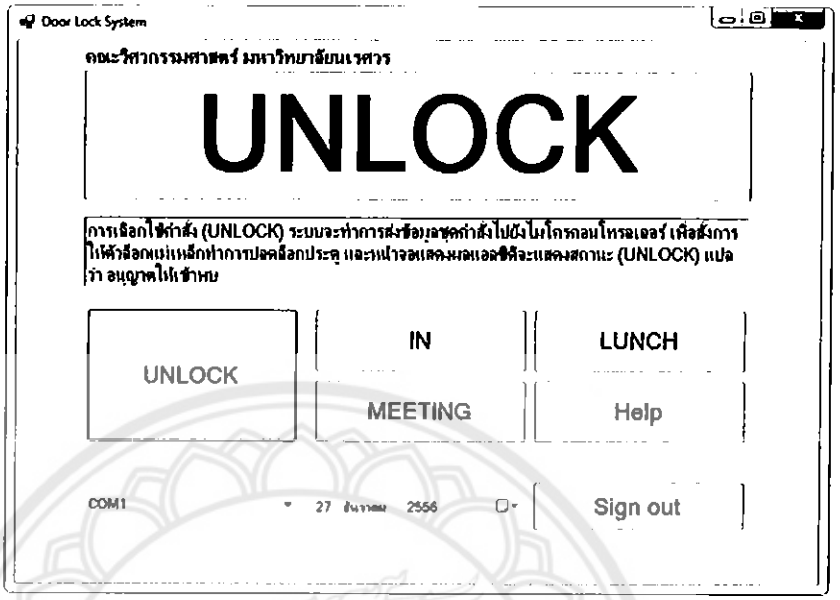


รูปที่ 4.10 แสดงปุ่มกดเลือกระบบการทำงานและไอคอนของโปรแกรมรับคำสั่ง

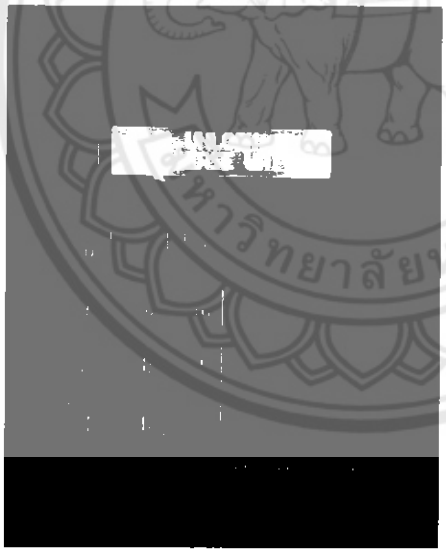
ปุ่มกดเลือกสถานะของระบบควบคุมผ่านคอมพิวเตอร์ สามารถแบ่งออกเป็น 4 คำสั่ง ได้แก่

1. กดปุ่ม (UNLOCK)
2. กดปุ่ม (IN)
3. กดปุ่ม (MEETING)
4. กดปุ่ม (LUNCH)

ระบบควบคุมผ่านคอมพิวเตอร์ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (UNLOCK) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.11



(ก)



(ข)

(ค)

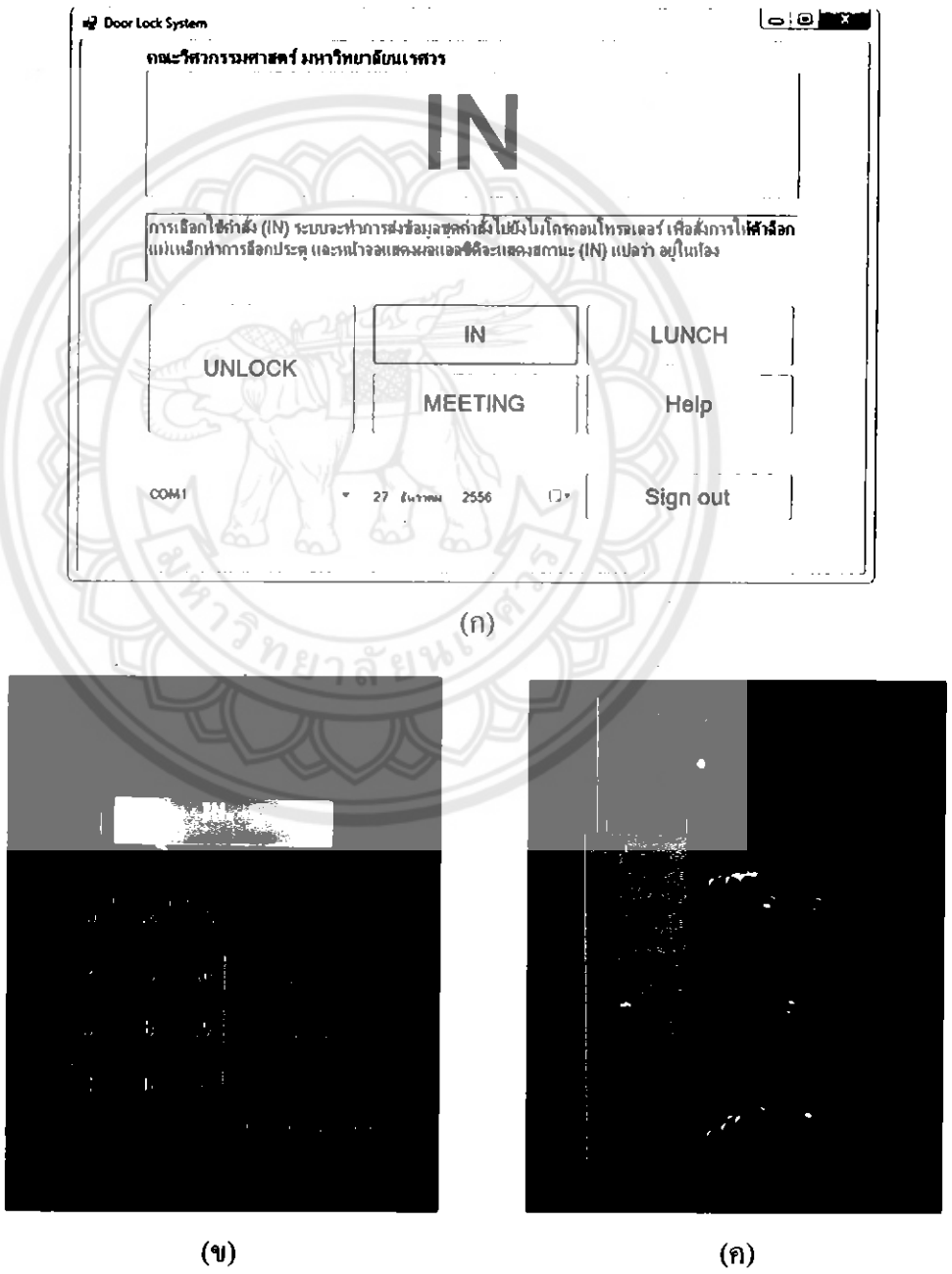
รูปที่ 4.11 แสดงการทำงานของคำสั่ง (UNLOCK) ของโปรแกรมรับคำสั่ง

(ก) เมื่อกดปุ่มเลือกสถานะคำสั่ง (UNLOCK) และยืนยันคำสั่ง

(ข) การทำงานของหน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (UNLOCK) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (UNLOCK) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะ

(ค) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (UNLOCK) ตัวล็อกแม่เหล็กจะปลดล็อกสังเกตจากหลอดหลอดอีดีไม่แสดงผล และหน่วงเวลาไว้ 5 วินาที แล้วตัวล็อกแม่เหล็กจะทำงานสถานะล็อก หลอดแอลอีดีของตัวล็อกแม่เหล็กแสดงสีเขียว

ระบบควบคุมผ่านคอมพิวเตอร์ เมื่อผู้ใช้กดปุ่มสถานะคำสั่ง (IN) มีขั้นตอนการทำงานแสดงดัง รูปที่ 4.12



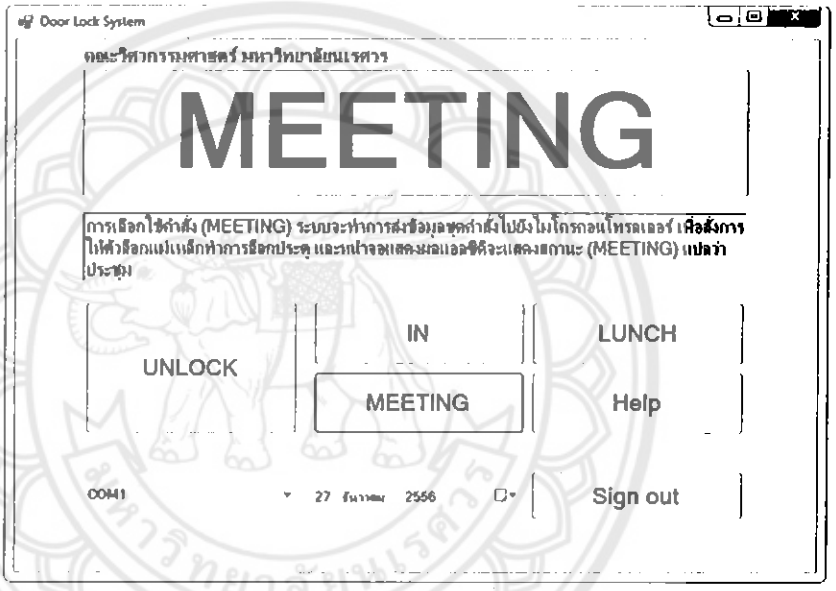
รูปที่ 4.12 แสดงการทำงานของคำสั่ง (IN) ของโปรแกรมรับคำสั่ง

(ก) เมื่อกดปุ่มเลือกสถานะคำสั่ง (IN) และยืนยันคำสั่ง

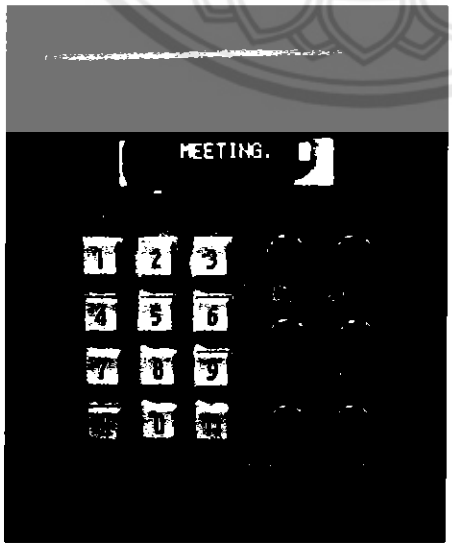
(ข) การทำงานของหน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (IN) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (IN) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะ

(ค) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (IN) ตัวล็อกแม่เหล็กจะทำงานสถานะล็อก สังเกตจากหลอดหลอดอีดีแสดงผลสีเขียว

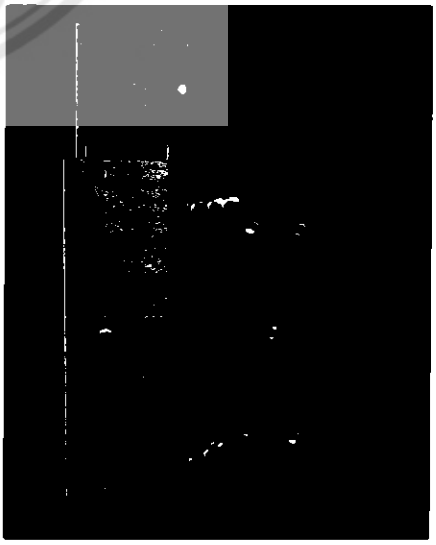
ระบบควบคุมผ่านคอมพิวเตอร์ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (MEETING) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.13



(ก)



(ข)



(ค)

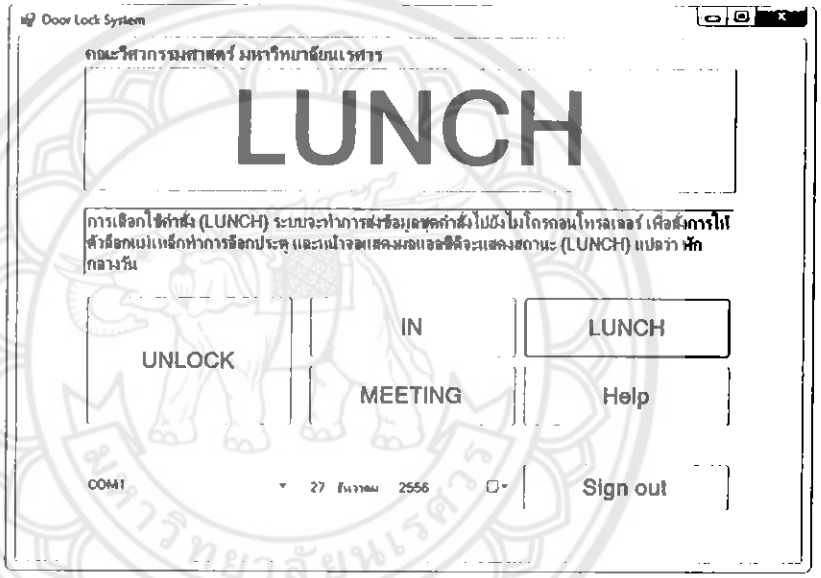
รูปที่ 4.13 แสดงการทำงานของคำสั่ง (MEETING) ของโปรแกรมรับคำสั่ง

(ก) เมื่อกดปุ่มเลือกสถานะคำสั่ง (MEETING) และยืนยันคำสั่ง

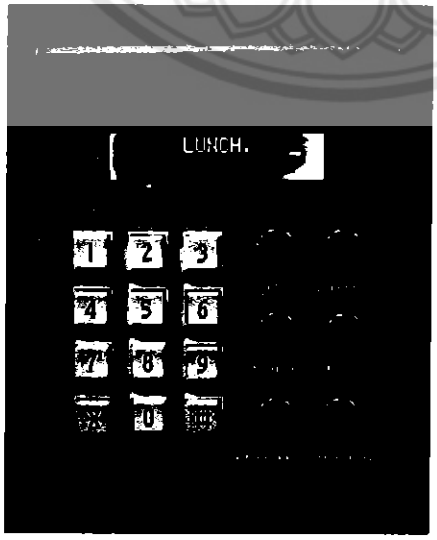
(ข) การทำงานของหน้าจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (MEETING) หน้าจอแสดงผลแอลซีดีจะแสดงข้อความ (MEETING) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบ

(ค) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลือกสถานะคำสั่ง (MEETING) ตัวล็อกแม่เหล็กจะทำงานสถานะล็อก สังเกตจากหลอดหลอดอีดีแสดงผลสีเขียว

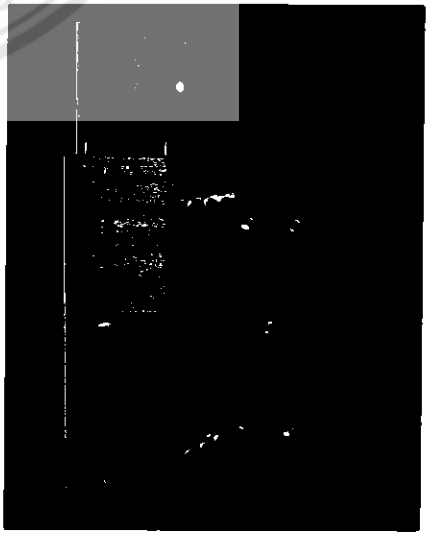
ระบบควบคุมผ่านคอมพิวเตอร์ เมื่อผู้ใช้กดปุ่มเลือกสถานะคำสั่ง (LUNCH) มีขั้นตอนการทำงานแสดงดังรูปที่ 4.14



(ก)



(ข)



(ค)

รูปที่ 4.14 แสดงการทำงานของคำสั่ง (LUNCH) ของโปรแกรมรับคำสั่ง

(ก) เมื่อมีการกดปุ่มเลือกสถานะคำสั่ง (LUNCH) และยืนยันคำสั่ง

(ข) การทำงานของจอแสดงผลหลังจากกดปุ่มเลือกสถานะคำสั่ง (LUNCH) จอแสดงผลแอลซีดีจะแสดงข้อความ (LUNCH) เพื่อให้ผู้ใช้หรือผู้มาติดต่อทราบสถานะ

(ค) การทำงานของตัวล็อกแม่เหล็กหลังจากกดปุ่มเลิกสถานะคำสั่ง (LUNCH) ตัวล็อกแม่เหล็กจะทำงานสถานะล็อก สังเกตจากหลอดหลอดอีดีแสดงผลสีเขียว

4.3 การทดสอบแรงยึดของตัวล็อกแม่เหล็กที่ระดับแรงดันไฟฟ้า 12 โวลต์

การทดสอบความสามารถในการรับแรงน้ำหนักในการควบคุมระบบ ขณะที่ตัวล็อกแม่เหล็กออกแรงยึดติดกับประตู โดยการทดสอบน้ำหนักแสดงดังตารางที่ 4.1 และแสดงดังรูปที่ 4.15

ตารางที่ 4.1 แสดงผลทดสอบความสามารถในการรับน้ำหนักในการควบคุมระบบ

จำนวนน้ำหนัก (กิโลกรัม)	จำนวนครั้งที่ทำการทดสอบ		
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3
30	Y	Y	Y
60	Y	Y	Y
90	Y	Y	Y
120	Y	Y	Y
150	Y	Y	Y
180	Y	Y	Y
210	Y	Y	Y
240	Y	Y	Y
280	Y	Y	Y
300	N	N	N

หมายเหตุ Y = YES สามารถรับน้ำหนักได้

N = NO ไม่สามารถรับน้ำหนักได้

รูปที่ 4.15 แสดงการทดสอบการรับน้ำหนักของตัวล็อกแม่เหล็ก

สรุปผลการทดสอบ

ในการควบคุมตัวล็อกแม่เหล็กที่ระดับแรงดันไฟฟ้า 12 โวลต์ ตัวล็อกแม่เหล็กสามารถรับน้ำหนักได้ไม่เกิน 280 กิโลกรัม เนื่องจากคุณสมบัติของตัวล็อกแม่เหล็กได้กำหนดไว้ที่ 600 ปอนด์ หรือคิดเป็น 280 กิโลกรัม

4.4 การทดสอบวัดพิกัดแรงดันไฟฟ้าของสายตัวนำเทียบกับระยะทาง

การทดสอบวัดพิกัดแรงดันไฟฟ้าของสายตัวนำเทียบกับระยะทางในการควบคุมระบบขณะที่ตัวล็อกแม่เหล็กออกแรงยึดติดกับประตู ทดสอบโดยการเพิ่มระยะทางของการควบคุมให้มีความยาวของสายตัวนำเพิ่มมากขึ้นแสดงดังตารางที่ 4.2

ตารางที่ 4.2 แสดงผลทดสอบวัดพิกัดแรงดันไฟฟ้าของสายตัวนำเทียบกับระยะทางในการควบคุมระบบ

ระยะทาง (เมตร)	จำนวนครั้งที่ทำการทดสอบ			ค่าเฉลี่ย (โวลต์)
	ครั้งที่ 1 (โวลต์)	ครั้งที่ 2 (โวลต์)	ครั้งที่ 3 (โวลต์)	
1	12.14	12.14	12.14	12.140
2	12.14	12.13	12.14	12.137
3	12.13	12.14	12.13	12.133
4	12.13	12.13	12.13	12.130
5	12.12	12.13	12.13	12.127
6	12.13	12.12	12.12	12.123
7	12.12	12.12	12.12	12.120
8	12.11	12.11	12.11	12.110
9	12.10	12.10	12.11	12.103
10	12.08	12.10	12.09	12.090

สรุปผลการทดสอบ

ในการควบคุมตัวล็อกแม่เหล็กที่ระดับแรงดันไฟฟ้าเฉลี่ย 12 โวลต์ ของสายตัวนำเทียบกับระยะทางความยาวต่าง ๆ ดังตารางที่ 4.2 จะเห็นได้ว่าเมื่อความยาวของสายตัวนำมีความยาวที่เพิ่มมากขึ้นจะส่งผลต่อระดับแรงดันไฟฟ้า ทำให้ระดับแรงดันไฟฟ้ามีค่าลดน้อยลง เนื่องจากเกิดการสูญเสียภายในสายตัวนำ

4.5 การทดสอบการใช้งานระบบควบคุมด้วยมือต่อเนื่องเป็นเวลา 48 ชั่วโมง

การทดสอบการใช้งานของระบบควบคุมด้วยมือเป็นเวลาต่อเนื่อง ในการควบคุมระบบขณะที่ตัวล็อกแม่เหล็กออกแรงยึดติดกับประตูและมีการเลือกใช้สถานะคำสั่งต่างๆ มีการทดสอบโดยใช้งานระบบตามระยะเวลาที่กำหนด ตรวจสอบการทำงานเมื่อเลือกใช้สถานะคำสั่งแสดงดังตารางที่ 4.3

ตารางที่ 4.3 แสดงการทดสอบการใช้งานระบบควบคุมด้วยมือต่อเนื่องเป็นเวลา 48 ชั่วโมง

สถานะคำสั่ง	เวลา(ชั่วโมง)								
	0	6	12	18	24	30	36	42	48
UNLOCK	Y	Y	Y	Y	Y	Y	Y	Y	Y
IN	Y	Y	Y	Y	Y	Y	Y	Y	Y
OUT	Y	Y	Y	Y	Y	Y	Y	Y	Y
IN CLASS	Y	Y	Y	Y	Y	Y	Y	Y	Y
MEETING	Y	Y	Y	Y	Y	Y	Y	Y	Y
LUNCH	Y	Y	Y	Y	Y	Y	Y	Y	Y
BE BACK SOON	Y	Y	Y	Y	Y	Y	Y	Y	Y
PASSWORD	Y	Y	Y	Y	Y	Y	Y	Y	Y

หมายเหตุ Y = YES สามารถทำงานตามสถานะคำสั่งได้

สรุปผลการทดสอบ

ในการทดสอบการใช้งานระบบควบคุมด้วยมือต่อเนื่องเป็นเวลา 48 ชั่วโมง ระบบสามารถทำงานตามสถานะคำสั่งต่างๆ ได้ตามปกติในช่วงเวลาที่แสดงดังตารางที่ 4.3 เนื่องจากในวงจรควบคุมการทำงานมีหม้อแปลงไฟฟ้า 2 ลูกแยกกันจ่ายกระแสไฟฟ้า ทำให้เกิดความร้อนเล็กน้อย ระบบมีพัดลมเพื่อช่วยระบายความร้อนและการต่อวงจรใช้ฉนวนห่อหุ้มอย่างดี ดังนั้นจึงทำให้ระบบสามารถทำงานได้อย่างปกติและต่อเนื่องเป็นระยะเวลานานได้

4.6 การทดสอบการใช้งานระบบควบคุมผ่านคอมพิวเตอร์ต่อเนื่องเป็นเวลา 8 ชั่วโมง

การทดสอบการใช้งานของระบบควบคุมผ่านคอมพิวเตอร์เป็นเวลาต่อเนื่อง การควบคุมระบบขณะที่ตัวล็อกแม่เหล็กออกแรงยึดติดกับประตูและมีการเลือกใช้สถานะคำสั่งต่างๆ การทดสอบโดยใช้งานระบบตามระยะเวลาที่กำหนด ตรวจสอบการทำงานเมื่อเลือกใช้สถานะคำสั่ง แสดงดังตารางที่ 4.4

ตารางที่ 4.4 แสดงการทดสอบการใช้งานระบบควบคุมผ่านคอมพิวเตอร์ต่อเนื่องเป็นเวลา 8 ชั่วโมง

เวลา (ชั่วโมง)	สถานะคำสั่ง			
	UNLOCK	IN	MEETING	LUNCH
0	Y	Y	Y	Y
1	Y	Y	Y	Y
2	Y	Y	Y	Y
3	Y	Y	Y	Y
4	Y	Y	Y	Y
5	Y	Y	Y	Y
6	Y	Y	Y	Y
7	Y	Y	Y	Y
8	Y	Y	Y	Y

หมายเหตุ Y = YES สามารถทำงานตามสถานะคำสั่งได้

สรุปผลการทดสอบ

ในการทดสอบการใช้งานระบบควบคุมผ่านคอมพิวเตอร์ต่อเนื่องเป็นเวลา 8 ชั่วโมง ระบบสามารถทำงานตามสถานะคำสั่งต่างๆ ได้ตามปกติในช่วงเวลาที่แสดงดังตารางที่ 4.4 เนื่องจากในวงจรควบคุมการทำงานมีหม้อแปลงไฟฟ้า 2 ลูกแยกกันจ่ายกระแสไฟฟ้า ทำให้เกิดความร้อนเล็กน้อย ระบบมีพัดลมเพื่อช่วยระบายความร้อนและการต่อวงจรใช้ฉนวนห่อหุ้มอย่างดี ดังนั้นจึงทำให้ระบบสามารถทำงานได้อย่างปกติและต่อเนื่องเป็นระยะเวลานานได้

บทที่ 5
สรุปผลและข้อเสนอแนะ

จากการศึกษา ออกแบบ ประกอบ ทดสอบ และทำการปรับปรุงชิ้นงานจนเป็นระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ โดยใช้ระยะเวลาดำเนินโครงงาน 2 ภาคการศึกษา ทำให้ทราบถึงปัญหาที่เกิดขึ้นในการทำโครงงาน สำหรับบทนี้จะเป็นการสรุปผลที่ได้จากการทดสอบในโครงงาน พร้อมเสนอแนะแนวทางในการนำโครงงานนี้ไปพัฒนาหรือต่อยอดให้มีประสิทธิภาพมากยิ่งขึ้น

5.1 สรุปผลการดำเนินโครงงาน

จากการทดสอบระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ พบว่าระบบสามารถควบคุมตัวล็อกแม่เหล็กและแสดงผลสถานะคำสั่งผ่านหน้าจอแอลซีดีได้ ดังนี้

1. ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ สามารถทำงานแบบสั่งการจากระบบควบคุมด้วยมือและแสดงผลสถานะคำสั่งให้ผู้ใช้และผู้มาติดต่อทราบได้

2. ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ สามารถทำงานแบบสั่งการจากระบบควบคุมผ่านคอมพิวเตอร์ได้และแสดงผลสถานะคำสั่งให้ผู้ใช้และผู้มาติดต่อทราบได้

3. ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ สามารถรับน้ำหนักได้ไม่เกิน 280 กิโลกรัม (600 ปอนด์)

4. ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ สามารถควบคุมได้ภายในระยะทางไม่เกิน 9 เมตรจากคอมพิวเตอร์ถึงตำแหน่งที่ติดตั้งตัวล็อกแม่เหล็ก จากที่กำหนดขอบเขตไว้ 10 เมตร

5. ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ สามารถเชื่อมต่อกับคอมพิวเตอร์ได้ทั้ง 2 แบบได้แก่ คอมพิวเตอร์แบบพกพาและคอมพิวเตอร์แบบพีซี บนระบบปฏิบัติการแบบ 64 บิต

และในส่วนของประตูนั้นต้องมีการสร้างโครงประตูจำลองขึ้นมา เพื่อให้สามารถติดตั้งอุปกรณ์และสามารถเคลื่อนย้ายได้สะดวกมากยิ่งขึ้น ซึ่งโดยรวมของการทำงานของระบบนั้นโครงสร้างประตูจำลอง สามารถทำงานตามที่คำสั่งได้กำหนดได้

5.2 ปัญหาและการแก้ไข

1. การติดตั้งตัวล็อกแม่เหล็กกับประตูทำได้ยากและไม่สามารถเคลื่อนย้ายได้ แก้ไขโดยการใช้โครงสร้างประตูจำลองแทนประตูทั่วไป

2. การใช้กระแสไฟฟ้าในระบบสูง ทำให้ต้องใช้หม้อแปลงขนาดใหญ่ น้ำหนักมาก และเกิดความร้อนสูง แก้ไขโดยการเปลี่ยนมาใช้หม้อแปลง 2 ลูก

3. การเชื่อมต่อกับคอมพิวเตอร์เพื่อควบคุมระบบ คอมพิวเตอร์ที่ใช้สั่งการโปรแกรมรับคำสั่ง (Door Lock System) โปรแกรมรันได้เฉพาะบนระบบปฏิบัติการแบบ 64 บิต แก้ไขโดยให้คอมพิวเตอร์ติงตั้งโปรแกรม (Framework 4.0 .NET) ขึ้นไป ทำให้ใช้ได้บนระบบปฏิบัติการแบบ 32 บิตได้

4. การใช้งานระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ สามารถทำงานในสถานะที่มีแหล่งจ่ายไฟฟ้าเท่านั้น แก้ไขโดยในกรณีที่แหล่งจ่ายไฟฟ้าไม่สามารถจ่ายไฟฟ้าได้ สามารถนำแบตเตอรี่กระแสสูงมาเป็นแหล่งจ่ายไฟฟ้าแทนได้

5. การใช้งานระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์เป็นระยะเวลานาน ทำให้เกิดความร้อนและการทำงานผิดพลาดของระบบได้ แก้ไขโดยการติดระบบระบายความร้อนและพัดลมให้กับวงจรควบคุมการทำงานเพื่อให้ช่วยอากาศถ่ายเทสะดวก

5.3 ข้อเสนอแนะและแนวทางในการพัฒนาต่อไป

จากปัญหาที่พบในระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ ส่งผลให้ศักยภาพการทำงานของระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์น้อยลง จึงต้องมีการศึกษาแนวทางวิธีในการแก้ไขปัญหาที่เกิดขึ้นเพื่อการพัฒนาระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ ซึ่งเป็นการพัฒนาในลักษณะดังต่อไปนี้

1. พัฒนาโดยการเพิ่มวงจรขยายกระแสและเลือกใช้หม้อแปลงไฟฟ้าที่มีขนาดเล็กและจ่ายกระแสไฟฟ้าสูงมาใช้งาน

2. พัฒนาโดยการเพิ่มอุปกรณ์ให้สามารถใช้งานระบบควบคุมด้วยการระบุข้อมูลสิ่งต่างๆ โดยใช้คลื่นความถี่วิทยุ (Radio Frequency IDentification) ได้

3. พัฒนาโดยการเพิ่มแหล่งจ่ายไฟฟ้าในกรณีที่แหล่งจ่ายไฟฟ้าไม่สามารถจ่ายไฟฟ้าได้ โดยการใช้แบตเตอรี่ที่จ่ายกระแสสูงได้

เอกสารอ้างอิง

[1] ทีมงานสมาร์ทเลิร์นนิ่ง. **เริ่มต้นเรียนรู้ไมโครคอนโทรลเลอร์ MCS-51 ด้วยภาษา C พร้อมโครงงาน.**กรุงเทพฯ: สมาร์ทเลิร์นนิ่ง, 2555.

[2] ทีมงานสมาร์ทเลิร์นนิ่ง. **ออกแบบและจำลองการทำงานของวงจรด้วย Proteus.** กรุงเทพฯ: สมาร์ทเลิร์นนิ่ง, 2552.

[3] ทีมงานอิเล็กทรอนิกส์แฟนคลับ. **รวมวงจร Project 1.** กรุงเทพฯ, 2552.

[4] สุภชัย สมพานิช. **คู่มือเรียนและใช้งาน Visual Basic.** บริษัท สวัสดีไอที จำกัด, กรุงเทพฯ: มิถุนายน 2556.

[5] วิธีการต่อใช้งานรีเลย์ 5 ขา และ 6 ขา. สืบค้นเมื่อ 19 พฤศจิกายน 2556 จาก http://www.thaiedurobot.com/article-th-86634-วิธีการต่อใช้งานรีเลย์+5+ขา+และ+6+ขา.html.

[6] ทดลองเล่นไมโครคอนโทรลเลอร์ MCS-51 ด้วย P89V51RD2. สืบค้นเมื่อ 19 พฤศจิกายน 2556 จาก http://www.mind-tek.net/8051.php.

[7] สนามแม่เหล็กไฟฟ้า (Electromagnetic Field: EMF). สืบค้นเมื่อ 19 พฤศจิกายน 2556 จาก http://eng.rmutsb.ac.th/eetc/research_file/research_20120402211054.pdf

[8] แนะนำโปรแกรม Microsoft Visual Basic. สืบค้นเมื่อ 19 พฤศจิกายน 2556 จาก http://krucom99.wordpress.com/สื่อการเรียนการสอน/เขียนโปรแกรม-visual-basic

[9] หม้อแปลงไฟฟ้า. วิทยาลัยสารพัดช่างกำแพงเพชร. สืบค้นเมื่อ 19 พฤศจิกายน 2556 จาก http://kpp.ac.th/elearning/elearning3/book-08.html.

[10] Smart home บ้านอัจฉริยะ. สืบค้นเมื่อ 19 พฤศจิกายน 2556 จาก http://www.it24hrs.com/2012/smart-room-smart-room-automation/.

ภาคผนวก ก

โปรแกรมการทำงานของระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์

1. การเขียนด้วยภาษา C ในโปรแกรม Kiel (การเขียนโปรแกรมให้กับไมโครคอนโทรลเลอร์)

```
#include<reg51.h>
#include<stdio.h>
#include<intrins.h>
unsigned char a[]={"ENTER PASSWORD"};        //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ a
unsigned char b[]={"CORRECT PASSWORD"};      //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ b
unsigned char c[]={"WRONG PASSWORD"};        //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ c
unsigned char e2[]={"   PLEASE WAIT. "};     //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ e2
unsigned char e3[]={"  ENTER STATUS. "};     //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ e3
unsigned char password[]={3,1,8,8};          //ประกาศตัวแปรอาร์เรย์เก็บรหัสผ่าน 4 หลัก
unsigned char input[4];                      //ประกาศตัวแปรอาร์เรย์เก็บรหัสอินพุต 4 หลัก
unsigned char i;                             //ประกาศตัวแปร index
sbit output=P0^0;                            //กำหนดให้ตัวแปร output แทนบิต P0.0
sbit sw8=P0^1;                               //กำหนดให้ตัวแปร sw8 แทนบิต P0.1
sbit RS=P0^6;                                //กำหนดให้ตัวแปร RS แทนบิต P0.6
sbit E=P0^7;                                 //กำหนดให้ตัวแปร E แทนบิต P0.7
sbit sw0=P1^0;                               //กำหนดให้ตัวแปร sw0 แทนบิต P1.0
sbit sw1=P1^1;                               //กำหนดให้ตัวแปร sw1 แทนบิต P1.1
sbit sw2=P1^2;                               //กำหนดให้ตัวแปร sw2 แทนบิต P1.2
sbit sw3=P1^3;                               //กำหนดให้ตัวแปร sw3 แทนบิต P1.3
sbit sw4=P1^4;                               //กำหนดให้ตัวแปร sw4 แทนบิต P1.4
sbit sw5=P1^5;                               //กำหนดให้ตัวแปร sw5 แทนบิต P1.5
sbit sw6=P1^6;                               //กำหนดให้ตัวแปร sw6 แทนบิต P1.6
sbit sw7=P1^7;                               //กำหนดให้ตัวแปร sw7 แทนบิต P1.7
sbit col1=P3^0;                              //กำหนดให้ตัวแปร col1 แทนบิต P3.0
sbit col2=P3^1;                              //กำหนดให้ตัวแปร col2 แทนบิต P3.1
sbit col3=P3^2;                              //กำหนดให้ตัวแปร col3 แทนบิต P3.2
sbit row1=P3^3;                              //กำหนดให้ตัวแปร row1 แทนบิต P3.3
sbit row2=P3^4;                              //กำหนดให้ตัวแปร row2 แทนบิต P3.4
sbit row3=P3^5;                              //กำหนดให้ตัวแปร row3 แทนบิต P3.5
sbit row4=P3^6;                              //กำหนดให้ตัวแปร row4 แทนบิต P3.6
```

```
void keypad_scan(unsigned int j);//ประกาศสร้างฟังก์ชัน keypad_scan แบบมีการรับค่าตัวแปรเข้า
void delay(unsigned int z);                                    //ประกาศสร้างฟังก์ชัน delay
void enable(void);                          //ประกาศสร้างฟังก์ชัน enable ไว้สั่งให้ LCD ทำงาน
void clear_screen(void);                //ประกาศสร้างฟังก์ชัน clear_screen ไว้เคลียร์หน้าจอ LCD
void func_1(void);                                             //ประกาศสร้างฟังก์ชัน func_1
void func_2(void);                                             //ประกาศสร้างฟังก์ชัน func_2
void func_3(void);                                             //ประกาศสร้างฟังก์ชัน func_3
void func_4(void);                                             //ประกาศสร้างฟังก์ชัน func_4
void func_5(void);                                             //ประกาศสร้างฟังก์ชัน func_5
void func_6(void);                                             //ประกาศสร้างฟังก์ชัน func_6
void func_7(void);                                             //ประกาศสร้างฟังก์ชัน func_7
void func_8(void);                                             //ประกาศสร้างฟังก์ชัน func_8
void keypad_scan(unsigned int j)                          //รายละเอียดฟังก์ชัน keypad_scan
{
while(row1&row2&row3&row4)                  //ให้วนอยู่ในลูปนี้ในขณะที่ปุ่มใดๆยังไม่ถูกกด
{
col1=0;                                                        //สแกนคอลัมน์ที่ 1
if(col1==0&&row1==0)                  //ตรวจสอบว่าคอลัมน์ 1 และแถว 1 เป็น 0 พร้อมกัน ?
{
P2='1';enable();                                               //แสดงตัวเลข 1 ที่ LCD
input[j]=1;                                                    //เก็บค่ารหัสอินพุต = 1
break;                                                         //ออกจากการสแกน keypad
}
else if(col1==0&&row2==1)             //ตรวจสอบว่าคอลัมน์ 1 และแถว 2 เป็น 0 พร้อมกัน ?
{
P2='4';enable();                                               //แสดงตัวเลข 4 ที่ LCD
input[j]=4;                                                    //เก็บค่ารหัสอินพุต = 4
 break;                                                        //ออกจากการสแกน keypad
 }
 else if(col1==0&&row3==0)            //ตรวจสอบว่าคอลัมน์ 1 และแถว 3 เป็น 0 พร้อมกัน ?
 {
 P2='7';enable();                                              //แสดงตัวเลข 7 ที่ LCD
```

```
input[j]=7;                                          //เก็บค่ารหัสอินพุต = 7
break;                                               //ออกจากการสแกน keypad
}
P3=0xff;                                             //สั่งให้บิตสแกนทั้ง 7 บิตมีลอจิก 1
col2=0;                                              //สแกนคอลัมน์ที่ 2
if(col2==0&&row1==0)                 //ตรวจสอบว่าคอลัมน์ 2 และแถว 1 เป็น 0 พร้อมกัน ?
{
P2='2';enable();                                     //แสดงตัวเลข 2 ที่ LCD
input[j]=2;                                          //เก็บค่ารหัสอินพุต = 2
break;                                               //ออกจากการสแกน keypad
}
else if(col2==0&&row2==0)            //ตรวจสอบว่าคอลัมน์ 2 และแถว 2 เป็น 0 พร้อมกัน ?
{
P2='5';enable();                                     //แสดงตัวเลข 5 ที่ LCD
input[j]=5;                                          //เก็บค่ารหัสอินพุต = 5
break;                                               //ออกจากการสแกน  keypad
}
else if(col2==0&&row3==0)            //ตรวจสอบว่าคอลัมน์ 2 และแถว 3 เป็น 0 พร้อมกัน ?
{
P2='8';enable();                                     //แสดงตัวเลข 8 ที่ LCD
input[j]=8;                                          //เก็บค่ารหัสอินพุต = 8
break;                                               //ออกจากการสแกน keypad
}
else if(col2==0&&row4==0)            //ตรวจสอบว่าคอลัมน์ 2 และแถว 4 เป็น 0 พร้อมกัน ?
{
P2='0';enable();                                     //แสดงตัวเลข 0 ที่ LCD
input[j]=0;                                          //เก็บค่ารหัสอินพุต = 0
break;                                               //ออกจากการสแกน keypad
}
 P3=0xff;                                            //สั่งให้บิตสแกนทั้ง 7 บิตมีลอจิก 1
col3=0;                                              //สแกนคอลัมน์ที่ 3
if(col3==0&&row1==0)                 //ตรวจสอบว่าคอลัมน์ 3 และแถว 1 เป็น 0 พร้อมกัน ?
```

```
{
P2='3';enable();                                    //แสดงตัวเลข 3 ที่ LCD
input[j]=3;                                         //เก็บค่ารหัสอินพุต = 3
break;                                              //ออกจากการสแกน keypad
}
else if(col3==0&&row2==0)          //ตรวจสอบว่าคอลัมน์ 3 และแถว 2 เป็น 0 พร้อมกัน ?
{
P2='6';enable();                                    //แสดงตัวเลข 6 ที่ LCD
input[j]=6;                                         //เก็บค่ารหัสอินพุต = 6
break;                                              //ออกจากการสแกน keypad
}
else if(col3==0&&row3==0)          //ตรวจสอบว่าคอลัมน์ 3 และแถว 3 เป็น 0 พร้อมกัน ?
{
P2='9';enable();                                    //แสดงตัวเลข 9 ที่ LCD
input[j]=9;                                         //เก็บค่ารหัสอินพุต = 9
break;                                              //ออกจากการสแกน keypad
}
P3=0xff;                                            //สั่งให้บิตสแกนทั้ง 7 บิตมีลอจิก 1
}
while(!row1|!row2|!row3|!row4);                     //วนลูปรอการปล่อยคีย์ที่ถูกกด
}
void clear_screen(void)                             //รายละเอียดฟังก์ชัน clear_screen
{
RS=0;                                   //สั่งให้ LCD Module ทราบว่าต่อไปจะเป็นคำสั่ง
P2=0x01;enable();                                   //คำสั่งเคลียร์หน้าจอ LCD
RS=1;                                   //ปิดการส่งคำสั่ง ต่อไปจะเป็นการส่งข้อมูล
}
 void enable(void)                                  //รายเอียดฟังก์ชัน enable
{
unsigned int x;                         //ประกาศสร้างตัวแปร x ไว้เพื่อหน่วงเวลาสร้างพัลส์
E=1;                                    //ส่งสัญญาณลอจิก 1 ไปที่ขา E ของ LCD Module
for(x=0;x<200;x++);                                 //หน่วงเวลาอย่างน้อย 2 ms
```

```
 E=0;                                          //ส่งสัญญาณลอจิก 0 ไปที่ขา E ของ LCD Module
 for(x=0;x<200;x++);                                          //หน่วงเวลาอย่างน้อย 2 ms
 }
void delay(unsigned int count)                                          //รายละเอียดฟังก์ชัน delay
{  unsigned int i;                          //ประกาศตัวแปร x เพื่อเก็บค่าได้ตั้งแต่ 0 ถึง 65,535
while(count)
{ i=230;while(i>0)i--;
count--;
}
}
void func_1(void)                                          //รายละเอียดฟังก์ชัน func_1
{
unsigned char e1[]={"    UNLOCK.   "};          //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ e1
 clear_screen();                                          //เรียกใช้งานฟังก์ชัน clear_screen
 for(i=0;i<14;i++)                                          //ให้มีการวนอยู่ในลูปนี้ 14 รอบ
 {
 P2=e1[i];enable();                                  //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
 delay(30);                                          //เรียกใช้งานฟังก์ชัน delay
 }
 delay(500);                                          //เรียกใช้งานฟังก์ชัน delay
 RS=0;                                  //สั่งให้ LCD Module ทราบว่าต่อไปจะเป็นคำสั่ง
 P2=0xC0;enable();                  //คำสั่งให้ LCD เริ่มต้นแสดงข้อความบรรทัดล่างซ้ายสุด
 RS=1;                                  //ปิดการส่งคำสั่ง ต่อไปจะเป็นการส่งข้อมูล
 for(i=0;i<16;i++)                                          //ให้มีการวนอยู่ในลูปนี้ 16 รอบ
 {
 P2=e2[i];enable();                                  //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
 delay(30);                                          //เรียกใช้งานฟังก์ชัน delay
 }
 output=1;                                          //Output มีค่าลอจิก 1
 delay(5000);                                          //เรียกใช้งานฟังก์ชัน delay
 }
```

```
void func_2(void)                                   //รายละเอียดฟังก์ชัน func_2
{
 unsigned char e[]={"     IN.     "};          //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ e2
 clear_screen();                                    //เรียกใช้งานฟังก์ชัน clear_screen
 for(i=0;i<14;i++)                                  //ให้มีการวนอยู่ในลูปนี้ 14 รอบ
 {
 P2=e[i];enable();                          //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
 delay(30);                                         //เรียกใช้งานฟังก์ชัน delay
 }
 output=0     ;                                     //Output มีค่าลอจิก 0
 }
 void func_3(void)                                  //รายละเอียดฟังก์ชัน func_3
 {
 unsigned char f[]={"     OUT.     "};          //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ f
 clear_screen();                                    //เรียกใช้งานฟังก์ชัน clear_screen
 for(i=0;i<14;i++)                                  //ให้มีการวนอยู่ในลูปนี้ 14 รอบ
 {
 P2=f[i];enable();                          //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
 delay(30);                                         //เรียกใช้งานฟังก์ชัน delay
 }
 output=0;                                          //Output มีค่าลอจิก 0
 }
 void func_4(void)                                  //รายละเอียดฟังก์ชัน func_4
 {
 unsigned char g[]={"   IN CLASS.  "};          //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ g
 clear_screen();                                    //เรียกใช้งานฟังก์ชัน clear_screen
 for(i=0;i<14;i++)                                  //ให้มีการวนอยู่ในลูปนี้ 14 รอบ
 {
 P2=g[i];enable();                          //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
 delay(30);                                         //เรียกใช้งานฟังก์ชัน delay
 }
 output=0;                                          //Output มีค่าลอจิก 0
```

```
}
void func_5(void)                                   //รายละเอียดฟังก์ชัน func_5
{
unsigned char h[]={"    MEETING.    "};          //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ h
clear_screen();                                     //เรียกใช้งานฟังก์ชัน clear_screen
for(i=0;i<14;i++)                                   //ให้มีการวนอยู่ในลูปนี้ 14 รอบ
{
P2=h[i];enable();                                   //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(30);                                          //เรียกใช้งานฟังก์ชัน delay
}
output=0;                                           //Output มีค่าลอจิก 0
}
void func_6(void)                                   //รายละเอียดฟังก์ชัน func_6
{
unsigned char k[]={"    LUNCH.    "};           //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ k
clear_screen();                                     //เรียกใช้งานฟังก์ชัน clear_screen
for(i=0;i<14;i++)                                   //ให้มีการวนอยู่ในลูปนี้ 14 รอบ
{
P2=k[i];enable();                                   //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(30);                                          //เรียกใช้งานฟังก์ชัน delay
}
output=0;                                           //Output มีค่าลอจิก 0
}
void func_7(void)                                   //รายละเอียดฟังก์ชัน func_7
{
unsigned char l[]={" BE BACK SOON. "};          //ประกาศเรียกตัวแปรอาร์เรย์เก็บข้อความที่ l
clear_screen();                                     //เรียกใช้งานฟังก์ชัน clear_screen
for(i=0;i<15;i++)                                   //ให้มีการวนอยู่ในลูปนี้ 15 รอบ
{
P2=l[i];enable();                                   //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(30);                                          //เรียกใช้งานฟังก์ชัน delay
}
```

```
 output=0;                                                    //Output มีค่าลอจิก 0
 }
 void func_8(void)                                            //รายละเอียดฟังก์ชัน func_8
 {
 while(1)                                                     //ให้มีการวนอยู่ในลูปนี้ไม่รู้จบ
 {
 clear_screen();                                              //เรียกใช้งานฟังก์ชัน clear_screen
 for(i=0;i<14;i++)                                            //ให้มีการวนอยู่ในลูปนี้ 14 รอบ
 {
 P2=a[i];enable();                                            //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
 delay(30);                                                   //เรียกใช้งานฟังก์ชัน delay
 }
 RS=0;                                                        //สั่งให้ LCD Module ทราบว่าต่อไปจะเป็นคำสั่ง
 P2=0xC0;enable();                                            //คำสั่งให้ LCD เริ่มต้นแสดงข้อความบรรทัดล่างซ้าย
 RS=1;                                                        //ปิดการส่งคำสั่ง ต่อไปจะเป็นการส่งข้อมูล
 keypad_scan(0);                                              //เรียกใช้งานฟังก์ชัน scan_keypad รับการกดรหัสหลักที่ 1
 delay(20);                                                   //เรียกใช้งานฟังก์ชัน delay
 keypad_scan(1);                                              //เรียกใช้งานฟังก์ชัน scan_keypad รับการกดรหัสหลักที่ 2
 delay(20);                                                   //เรียกใช้งานฟังก์ชัน delay
 keypad_scan(2);                                              //เรียกใช้งานฟังก์ชัน scan_keypad รับการกดรหัสหลักที่ 3
 delay(20);                                                   //เรียกใช้งานฟังก์ชัน delay
 keypad_scan(3);                                              //เรียกใช้งานฟังก์ชัน scan_keypad รับการกดรหัสหลักที่ 4
 delay(20);                                                   //เรียกใช้งานฟังก์ชัน delay
 P3=0xff;                                                     //สั่งให้บิตสแกนทั้ง 7 บิตมีลอจิก 1
 col3=0;                                                      //สแกนคอลัมน์ที่ 3
 while(col3|row4);                                            //รอการกดปุ่ม "#" ซึ่งคือการกดปุ่ม Enter
if(input[0]==password[0]&&input[1]==password[1]&&input[2]==password[2]&&input[3]==
password[3])                                                  //ถ้ารหัสที่กดทั้งหมดถูกต้อง
 {
 clear_screen();                                              //เรียกใช้งานฟังก์ชัน clear_screen
 for(i=0;i<16;i++)                                            //ให้มีการวนอยู่ในลูปนี้ 16 รอบ
 {
```

```
P2=b[i];enable();                          //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(30);                                 //เรียกใช้งานฟังก์ชัน delay
}
delay(500);                                //เรียกใช้งานฟังก์ชัน delay
RS=0;                                      //สั่งให้ LCD Module ทราบว่าต่อไปจะเป็นคำสั่ง
P2=0xC0;enable();                          //คำสั่งให้ LCD เริ่มต้นแสดงข้อความบรรทัดล่างซ้าย
RS=1;                                      //ปิดการส่งคำสั่ง ต่อไปจะเป็นการส่งข้อมูล
for(i=0;i<16;i++)                          //ให้มีการวนอยู่ในลูปนี้ 16 รอบ
{
P2=e2[i];enable();                         //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(30);                                 //เรียกใช้งานฟังก์ชัน delay
}
output=1;                                  //Output มีค่าลอจิก 1
delay(5000);                               //เรียกใช้งานฟังก์ชัน delay
break;                                     //ออกจากคำสั่ง
}
else
{
clear_screen();                            //เรียกใช้งานฟังก์ชัน clear_screen
for(i=0;i<14;i++)                          //ให้มีการวนอยู่ในลูปนี้ 14 รอบ
{
P2=c[i];enable();                          //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(70);                                 //เรียกใช้งานฟังก์ชัน delay
}
}
}
}
void serial_inter()                        //รายละเอียดฟังก์ชันการติดต่อสื่อสารผ่านพอร์ตอนุกรม
{
SCON = 0x50;                                                    //0101 0000
TMOD = 0x20;                                                    //timer 1 Mode 2
TH1 = 0xFB;                                                     //Baud rate = 9600
```

```
 EA = 1;
 ES = 1;
 TR1 = 1;
 TI = 1;
}
 void main(void)                                    //รายละเอียดเมนโปรแกรม
 {
 RS=0;                                  //สั่งให้ LCD Module ทราบว่าต่อไปจะเป็นคำสั่ง
 P2=0x38;enable();                          //คำสั่งตั้งค่าเริ่มต้นการแสดงผลของ LCD
 P2=0x0F;enable();          //คำสั่งให้ LCD เปิดการแสดงข้อความและเคอร์เซอร์กระพริบ
 P2=0x04;enable();       //คำสั่งให้เมื่อแสดงข้อความ เคอร์เซอร์เลื่อนไปทางขวาอัตโนมัติ
 RS=1;                                                  //ปิดการส่งคำสั่ง
 clear_screen();                                //เรียกใช้งานฟังก์ชัน clear_screen
 for(i=0;i<16;i++)                               //ให้มีการวนอยู่ในลูปนี้ 16 รอบ
 {
 P2=e3[i];enable();                      //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
 delay(30);                                          //เรียกใช้งานฟังก์ชัน delay
 }
 while(1)                                           //ให้มีการวนอยู่ในลูปนี้ไม่รู้จบ
 {
 if(sw0==0)                                        //ตรวจสอบ sw0 เป็น 0 หรือไม่
 {
 func_1();                                          //เรียกใช้การทำงานของ func_1
 output=0;                                              //Output มีค่าลอจิก 0
 clear_screen();                                //เรียกใช้งานฟังก์ชัน clear_screen
 for(i=0;i<16;i++)                               //ให้มีการวนอยู่ในลูปนี้ 16 รอบ
 {
 P2=e3[i];enable();                      //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
 delay(30);                                          //เรียกใช้งานฟังก์ชัน delay
 }
 }
 else if(sw1==0)                                    //ตรวจสอบ sw1 เป็น 0 หรือไม่
```

```
{
func_2();                                   //เรียกใช้การทำงานของ func_2
}
else if(sw2==0)                             //ตรวจสอบ sw2 เป็น 0 หรือไม่
{
func_3();                                   //เรียกใช้การทำงานของ func_3
}
else if(sw3==0)                             //ตรวจสอบ sw3 เป็น 0 หรือไม่
{
func_4();                                   //เรียกใช้การทำงานของ func_4
}
else if(sw4==0)                             //ตรวจสอบ sw4 เป็น 0 หรือไม่
{
func_5();                                   //เรียกใช้การทำงานของ func_5
}
else if(sw5==0)                             //ตรวจสอบ sw5 เป็น 0 หรือไม่
{
func_6();                                   //เรียกใช้การทำงานของ func_6
}
else if(sw6==0)                             //ตรวจสอบ sw6 เป็น 0 หรือไม่
{
func_7();                                   //เรียกใช้การทำงานของ func_7
}
else if(sw7==0)                             //ตรวจสอบ sw7 เป็น 0 หรือไม่
{
 func_8();                                  //เรียกใช้การทำงานของ func_1
output=0;                                   //Output มีค่าลอจิก 0
clear_screen();                             //เรียกใช้งานฟังก์ชัน clear_screen
for(i=0;i<16;i++)                           //ให้มีการวนอยู่ในลูปนี้ 16 รอบ
{
P2=e3[i];enable();                          //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(30);                                  //เรียกใช้งานฟังก์ชัน delay
```

```
}
}
else if(sw8==0)                          //ตรวจสอบ sw8 เป็น 0 หรือไม่
{
clear_screen();                          //เรียกใช้งานฟังก์ชัน clear_screen
for(i=0;i<16;i++)                        //ให้มีการวนอยู่ในลูปนี้ 16 รอบ
{
P2=e3[i];enable();                       //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(30);                               //เรียกใช้งานฟังก์ชัน delay
}
while(1)                                 //ให้มีการวนอยู่ในลูปนี้ไม่รู้จบ
{
char v;                                  //ประกาศตัวแปร v
serial_inter();                          //เรียกใช้ฟังก์ชันติดต่อสื่อสารผ่านพอร์ตอนุกรม
printf("\nEnter char:");
v=getchar();                             //ตัวแปร v เท่ากับ getchar()
switch(v)                                //ตัวแปรที่ต้องการตรวจสอบ
{
case'1':                                 //ตรวจสอบ case1 เป็น (ASKII 31H) หรือไม่
func_1();                                //เรียกใช้การทำงานของ func_1
output=0;                                //Output มีค่าลอจิก 0
clear_screen();                          //เรียกใช้งานฟังก์ชัน clear_screen
for(i=0;i<16;i++)                        //ให้มีการวนอยู่ในลูปนี้ 16 รอบ
{
P2=e3[i];enable();                       //ส่งข้อมูลตัวอักษรที่ index ชี้อยู่ให้กับ LCD
delay(30);                               //เรียกใช้งานฟังก์ชัน delay
}
break;                                   //ออกจากคำสั่ง
case'2':                                 //ตรวจสอบ case2 เป็น (ASKII 32H) หรือไม่
func_2();                                //เรียกใช้การทำงานของ func_2
break;                                   //ออกจากคำสั่ง
case'3':                                 //ตรวจสอบ case3 เป็น (ASKII 33H) หรือไม่
```

```
    func_5();                                   //เรียกใช้การทำงานของ func_5
    break;                                      //ออกจากคำสั่ง
    case'4':                                    //ตรวจสอบ case4 เป็น (ASKII 34H) หรือไม่
    func_6();                                   //เรียกใช้การทำงานของ func_6
    break;                                      //ออกจากคำสั่ง
     printf("\nEnter char:");
}
}
}
}
}
```

2. เขียนด้วยภาษา Visual Basic (โปรแกรมรับคำสั่ง Door Lock System)

2.1 Form 1

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
If TextBox1.Text = "" Or TextBox2.Text = "" Then
MessageBox.Show("คุณกรอกข้อมูลไม่ครบ กรุณาตรวจสอบข้อมูลของคุณอีกครั้ง", "ข้อความจาก
ระบบ", MessageBoxButtons.OK, MessageBoxIcon.Error)
ElseIf TextBox1.Text = "admin" And TextBox2.Text = "3188" Then
MessageBox.Show("ยินดีต้อนรับเข้าสู่ระบบ", "ข้อความจากระบบ", MessageBoxButtons.OK,
MessageBoxIcon.Information)
Me.Hide()
TextBox1.Clear()
TextBox2.Clear()
Form2.Show()
Else
MessageBox.Show("ชื่อผู้ใช้งาน หรือ รหัสผ่าน ไม่ถูกต้อง", "ข้อความจากระบบ",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
Me.Close()
End Sub
Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs) Handles
TextBox2.TextChanged
If (TextBox2.Text.Length > 4) Then
ErrorProvider1.SetError(TextBox2, "กรุณาป้อนรหัสผ่านไม่เกิน 4 ตัวอักษร !!!")
Else
ErrorProvider1.SetError(TextBox2, "")
End If
```

```
End Sub

End Class
```

2.2 Form 2

```
Imports System

Imports System.IO

Imports System.IO.Ports

Imports System.Threading

Imports System.ComponentModel

Public Class Form2

Dim myport As Array

Delegate Sub SetTextCallback(ByVal [text] As String)

Shared _continue As Boolean

Shared _serialPort As SerialPort

Private Sub Form2_Load(sender As System.Object, e As System.EventArgs) Handles

MyBase.Load

myport = IO.Ports.SerialPort.GetPortNames()

ComboBox1.Items.AddRange(myport)

TextBox1.Text = "-HELLO-"

RichTextBox1.Text = ".....You're

welcome....."

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

TextBox1.Text = "UNLOCK"

RichTextBox1.Text = "การเลือกใช้คำสั่ง (UNLOCK) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง

ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการปลดล็อกประตู และหน้าจอแสดงผล

แอลซีดีจะแสดงสถานะ (UNLOCK) แปลว่า อนุญาตให้เข้าพบ"

SerialPort1.Open()
```

```
SerialPort1.Write(1)
SerialPort1.Close()
MessageBox.Show("คุณได้เลือกคำสั่ง UNLOCK เรียบร้อยแล้ว", "ยืนยันคำสั่ง",
MessageBoxButtons.OK, MessageBoxIcon.None)
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
TextBox1.Text = "IN"
RichTextBox1.Text = "การเลือกใช้คำสั่ง (IN) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดี
จะแสดงสถานะ (IN) แปลว่า อยู่ในห้อง"
SerialPort1.Open()
SerialPort1.Write(2)
SerialPort1.Close()
MessageBox.Show("คุณได้เลือกคำสั่ง IN เรียบร้อยแล้ว", "ยืนยันคำสั่ง", MessageBoxButtons.OK,
MessageBoxIcon.None)
End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
TextBox1.Text = "MEETING"
RichTextBox1.Text = "การเลือกใช้คำสั่ง (MEETING) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดี
จะแสดงสถานะ (MEETING) แปลว่า ประชุม"
SerialPort1.Open()
SerialPort1.Write(3)
SerialPort1.Close()
MessageBox.Show("คุณได้เลือกคำสั่ง MEETING เรียบร้อยแล้ว", "ยืนยันคำสั่ง",
MessageBoxButtons.OK, MessageBoxIcon.None)
End Sub
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
```

```
TextBox1.Text = "LUNCH"
RichTextBox1.Text = "การเลือกใช้คำสั่ง (LUNCH) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดี
จะแสดงสถานะ (LUNCH) แปลว่า พักกลางวัน"
SerialPort1.Open()
SerialPort1.Write(4)
SerialPort1.Close()
MessageBox.Show("คุณได้เลือกคำสั่ง LUNCH เรียบร้อยแล้ว", "ยืนยันคำสั่ง",
MessageBoxButtons.OK, MessageBoxIcon.None)
End Sub
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
TextBox1.Text = "Help"
RichTextBox1.Text = ""
Form3.Show()
End Sub
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
TextBox1.Text = "คุณกำลังออกจากระบบ"
RichTextBox1.Text = ""
Form4.Show()
Me.Close()
End Sub
Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
SerialPort1.PortName = ComboBox1.Text
SerialPort1.BaudRate = 9600
SerialPort1.DataBits = 8
SerialPort1.Parity = IO.Ports.Parity.None
SerialPort1.StopBits = IO.Ports.StopBits.One
```

```
SerialPort1.Handshake = IO.Ports.Handshake.None
SerialPort1.Encoding = System.Text.Encoding.Default
End Sub
End Class
```

2.3 Form 3

```
Public Class Form3
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
RichTextBox1.Text = "การเลือกใช้คำสั่ง (UNLOCK) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการปลดล็อกประตู และหน้าจอแสดงผล
แอลซีดีจะแสดงสถานะ (UNLOCK) แปลว่า อนุญาตให้เข้าพบ"
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
RichTextBox1.Text = "การเลือกใช้คำสั่ง (IN) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดี
จะแสดงสถานะ (IN) แปลว่า อยู่ในห้อง"
End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
RichTextBox1.Text = "การเลือกใช้คำสั่ง (OUT) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดี
จะแสดงสถานะ (OUT) แปลว่า ไปข้างนอกไม่อยู่ในห้อง"
End Sub
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
RichTextBox1.Text = "การเลือกใช้คำสั่ง (IN CLASS) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดี
จะแสดงสถานะ (IN CLASS) แปลว่า ไปห้องเรียน(สอน)"
End Sub
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
RichTextBox1.Text = "การเลือกใช้คำสั่ง (MEETING) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดี
จะแสดงสถานะ (MEETING) แปลว่า ประชุม"
```

```
End Sub
Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
RichTextBox1.Text = "การเลือกใช้คำสั่ง (LUNCH) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยัง
ไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดี
จะแสดงสถานะ (LUNCH) แปลว่า พักกลางวัน"
End Sub
Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
RichTextBox1.Text = "การเลือกใช้คำสั่ง (BE BACK SOON) ระบบจะทำการส่งข้อมูลชุดคำสั่งไป
ยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอล
ซีดีจะแสดงสถานะ (BE BACK SOON) แปลว่า กลับมาเร็ว ๆ นี้"
End Sub
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
Me.Close()
End Sub
End Class
```

2.4 Form 4

```
Public Class Form4
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Me.Close()
Form1.Show()
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
Form2.Show()
Me.Close()
End Sub
End Class
```

ภาคผนวก ข

รายละเอียดของไมโครคอนโทรลเลอร์ MCS-51 หมายเลข **P89V51RD2**

P89V51RD2

8-bit 80C51 5 V low power 64 kB Flash microcontroller with 1 kB RAM

Rev. 01 — 01 March 2004 Product data

1. General description

The P89V51RD2 is an 80C51 microcontroller with 64 kB Flash and 1024 bytes of data RAM.

A key feature of the P89V51RD2 is its X2 mode option. The design engineer can choose to run the application with the conventional 80C51 clock rate (12 clocks per machine cycle) or select the X2 mode (6 clocks per machine cycle) to achieve twice the throughput at the same clock frequency. Another way to benefit from this feature is to keep the same performance by reducing the clock frequency by half, thus dramatically reducing the EMI.

The Flash program memory supports both parallel programming and in serial In-System Programming (ISP). Parallel programming mode offers gang-programming at high speed, reducing programming costs and time to market. ISP allows a device to be reprogrammed in the end product under software control. The capability to field/update the application firmware makes a wide range of applications possible.

The P89V51RD2 is also In-Application Programmable (IAP), allowing the Flash program memory to be reconfigured even while the application is running.

2. Features

- 80C51 Central Processing Unit
- 5 V Operating voltage from 0 to 40 MHz
- 64 kB of on-chip Flash program memory with ISP (In-System Programming) and IAP (In-Application Programming)
- Supports 12-clock (default) or 6-clock mode selection via software or ISP
- SPI (Serial Peripheral Interface) and enhanced UART
- PCA (Programmable Counter Array) with PWM and Capture/Compare functions
- Four 8-bit I/O ports with three high-current Port 1 pins (16 mA each)
- Three 16-bit timers/counters
- Programmable Watchdog timer (WDT)
- Eight interrupt sources with four priority levels
- Second DPTR register
- Low EMI mode (ALE inhibit)
- TTL- and CMOS-compatible logic levels

PHILIPS

- Brown-out detection
- Low power modes
  - Power-down mode with external interrupt wake-up
  - Idle mode
- PDIP40, PLCC44 and TQFP44 packages

3. Ordering information

Table 1: Ordering information

Type number	Package		Version
	Name	Description	
P89V51RD2FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89V51RD2FBC	TQFP44	plastic thin quad flat package; 44 leads	SOT376-1
P89V51RD2BN	PDIP40	plastic dual in-line package; 40 leads	SOT129-1

3.1 Ordering options

Table 2: Ordering options

Type number	Temperature range	Frequency
P89V51RD2FA	−40 °C to +85 °C	0 to 40 MHz
P89V51RD2FBC	−40 °C to +85 °C	
P89V51RD2BN	0 °C to +70 °C	

4. Block diagram

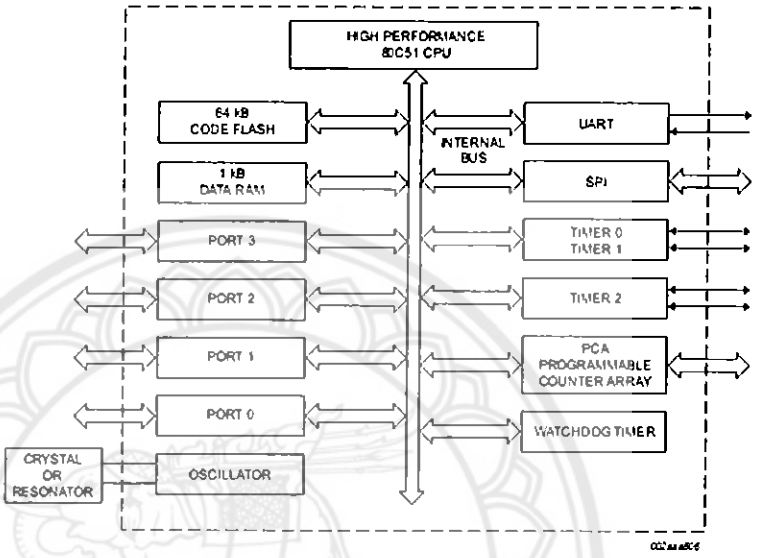


Fig 1. P89V51RD2 block diagram.

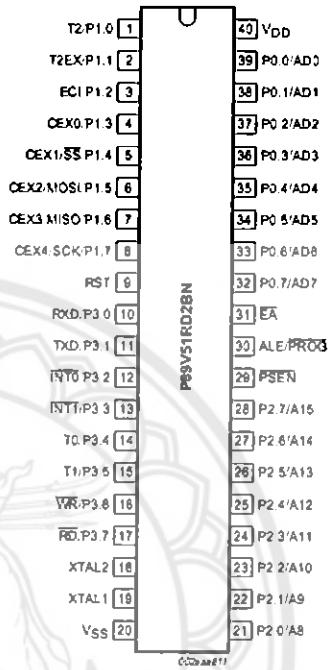


Fig 3. PDIP40 pin configuration.

© Koninklijke Philips Electronics N.V. 2004. All rights reserved.

5.2 Pin description

Table 3: P89V51RD2 pin description

Symbol	Pin			Type	Description
	DIP40	TQFP44	PLCC44		
P0.0 to P0.7	39-32	37-30	43-36	I/O	**Port 0:** Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have '1's written to them float, and in this state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external code and data memory. In this application, it uses strong internal pull-ups when transitioning to '1's. Port 0 also receives the code bytes during the external host mode programming, and outputs the code bytes during the external host mode verification. External pull-ups are required during program verification or as a general purpose I/O port.
P1.0 to P1.7	1-8	40-44, 1-3	2-9	I/O with internal pull-up	**Port 1:** Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 pins are pulled high by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 1 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups. P1.5, P1.6, P1.7 have high current drive of 16 mA. Port 1 also receives the low-order address bytes during the external host mode programming and verification.
P1.0	1	40	2	I/O	**T2:** External count input to Timer/Counter 2 or Clock-out from Timer/Counter 2
P1.1	2	41	3	I	**T2EX:** Timer/Counter 2 capture/reload trigger and direction control
P1.2	3	42	4	I	**ECI:** External clock input. This signal is the external clock input for the PCA.
P1.3	4	43	5	I/O	**CEX0:** Capture/compare external I/O for PCA Module 0. Each capture/compare module connects to a Port 1 pin for external I/O. When not used by the PCA, this pin can handle standard I/O.
P1.4	5	44	6	I/O	**SS:** Slave port select input for SPI **CEX1:** Capture/compare external I/O for PCA Module 1
P1.5	6	1	7	I/O	**MOSI:** Master Output Slave Input for SPI **CEX2:** Capture/compare external I/O for PCA Module 2
P1.6	7	2	8	I/O	**MISO:** Master Input Slave Output for SPI **CEX3:** Capture/compare external I/O for PCA Module 3
P1.7	8	3	9	I/O	**SCK:** Master Output Slave Input for SPI **CEX4:** Capture/compare external I/O for PCA Module 4

© Koninklijke Philips Electronics N.V. 2004. All rights reserved.

Table 3: P89V51RD2 pin description...*continued*

Symbol	Pin			Type	Description
	DIP40	TQFP44	PLCC44		
P2.0 to P2.7	21-28	18-25	24-31	I/O with internal pull-up	**Port 2**: Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins are pulled HIGH by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 2 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups. Port 2 sends the high-order address byte during fetches from external program memory and during accesses to external Data Memory that use 16-bit address (MOVX@DPTR). In this application, it uses strong internal pull-ups when transitioning to '1's. Port 2 also receives some control signals and a partial of high-order address bits during the external host mode programming and verification.
P3.0 to P3.7	10-17	5, 7-13	11, 13-19	I/O with internal pull-up	**Port 3**: Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins are pulled HIGH by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 3 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups. Port 3 also receives some control signals and a partial of high-order address bits during the external host mode programming and verification.
P3.0	10	5	11	I	**RXD**: serial input port
P3.1	11	7	13	O	**TXD**: serial output port
P3.2	12	8	14	I	**$\overline{INT0}$**: external interrupt 0 input
P3.3	13	9	15	I	**$\overline{INT1}$**: external interrupt 1 input
P3.4	14	10	16	I	**T0**: external count input to Timer/Counter 0
P3.5	15	11	17	I	**T1**: external count input to Timer/Counter 1
P3.6	16	12	18	O	**$\overline{WR}$**: external data memory write strobe
P3.7	17	13	19	O	**$\overline{RD}$**: external data memory read strobe
$\overline{PSEN}$	29	26	32	I/O	**Program Store Enable**: $\overline{PSEN}$ is the read strobe for external program memory. When the device is executing from internal program memory, $\overline{PSEN}$ is inactive (HIGH). When the device is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory. A forced HIGH-to-LOW input transition on the $\overline{PSEN}$ pin while the RST input is continually held HIGH for more than 10 machine cycles will cause the device to enter external host mode programming.
RST	9	4	10	I	**Reset**: While the oscillator is running, a HIGH logic state on this pin for two machine cycles will reset the device. If the $\overline{PSEN}$ pin is driven by a HIGH-to-LOW input transition while the RST input pin is held HIGH, the device will enter the external host mode, otherwise the device will enter the normal operation mode.

Table 3: P89V51RD2 pin description...*continued*

Symbol	Pin			Type	Description
	DIP40	TQFP44	PLCC44		
$\overline{EA}$	31	29	35	I	**External Access Enable:** $\overline{EA}$ must be connected to V_{SS} in order to enable the device to fetch code from the external program memory. $\overline{EA}$ must be strapped to V_{DD} for internal program execution. However, Security lock level 4 will disable $\overline{EA}$, and program execution is only possible from internal program memory. The $\overline{EA}$ pin can tolerate a high voltage of 12 V.
ALE/ $\overline{PROG}$	30	27	33	I/O	**Address Latch Enable:** ALE is the output signal for latching the low byte of the address during an access to external memory. This pin is also the programming pulse input ($\overline{PROG}$) for flash programming. Normally the ALE[1] is emitted at a constant rate of 1/6 the crystal frequency[2] and can be used for external timing and clocking. One ALE pulse is skipped during each access to external data memory. However, if AO is set to '1', ALE is disabled.
NC	-	6, 17, 28, 39	1, 12, 23, 34	I/O	**No Connect**
XTAL1	19	15	21	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	14	20	O	**Crystal 2:** Output from the inverting oscillator amplifier.
V_{DD}	40	38	44	I	**Power supply**
V_{SS}	20	16	22	I	**Ground**

[1] ALE loading issue: When ALE pin experiences higher loading (>30 pF) during the reset, the microcontroller may accidentally enter into modes other than normal working mode. The solution is to add a pull-up resistor of 3 kΩ to 50 kΩ to V_{DD}, e.g., for ALE pin.

[2] For 6-clock mode, ALE is emitted at 1/3 of crystal frequency.

ภาคผนวก ค

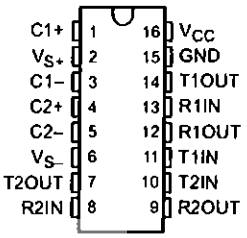
รายละเอียดของไอซีหมายเลข Max 232

MAX232, MAX232I
DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
- Applications
  - TIA/EIA-232-F
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
	SOIC (D)	Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
–40°C to 85°C	PDIP (N)	Tube	MAX232IN	MAX232IN
	SOIC (D)	Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

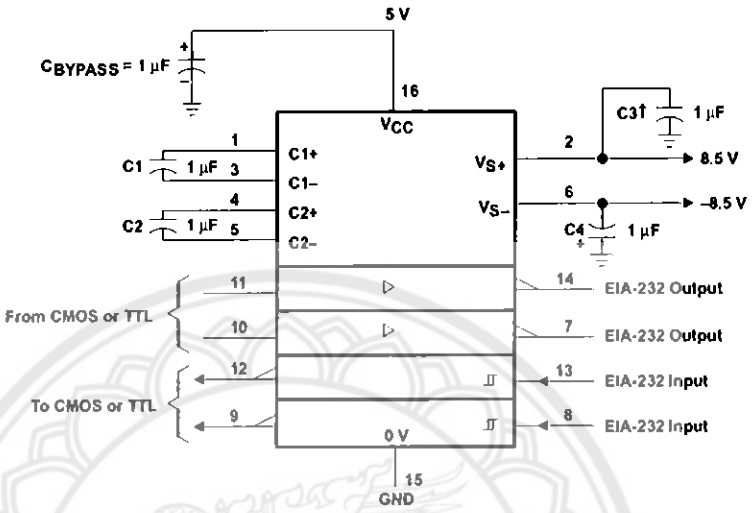
Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2002, Texas Instruments Incorporated

APPLICATION INFORMATION



† C3 can be connected to V_{CC} or GND.

Figure 4. Typical Operating Circuit

ภาคผนวก ง

คู่มือการใช้งานระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์

คู่มือการใช้งาน

ระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์

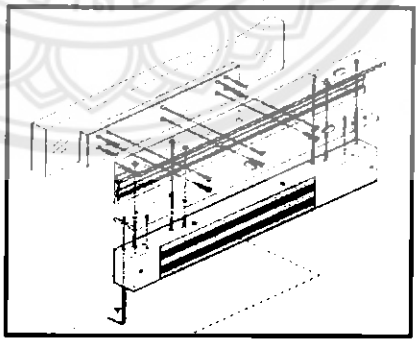
คู่มือฉบับนี้จัดทำขึ้นเพื่ออธิบายการใช้งานและวิธีการติดตั้งอุปกรณ์ต่างๆที่ใช้ในการทำงานของระบบล็อกประตูอัตโนมัติควบคุมผ่านคอมพิวเตอร์ ซึ่งมีระบบควบคุมการทำงาน 2 ระบบ คือ ระบบควบคุมด้วยมือและระบบควบคุมผ่านคอมพิวเตอร์ โดยมีรายละเอียดการใช้งานและวิธีการติดตั้งอุปกรณ์ต่างๆ ดังต่อไปนี้

1. วิธีการติดตั้งตัวล็อกแม่เหล็กไฟฟ้า
2. วิธีการต่อวงจรควบคุมการทำงาน
3. การติดตั้งโปรแกรมรับคำสั่งจากคอมพิวเตอร์ (Door Lock System)
4. รายละเอียดของปุ่มสถานะคำสั่งต่างๆ

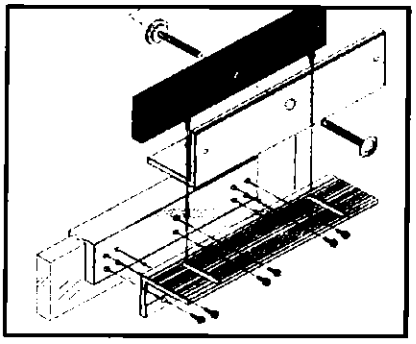
1. วิธีการติดตั้งตัวล็อกแม่เหล็กไฟฟ้า

ตำแหน่งการติดตั้งตัวล็อกแม่เหล็กไฟฟ้าที่เหมาะสมบนประตูแบบบานเดี่ยว มีลักษณะตำแหน่งการติดตั้งแสดงดังนี้

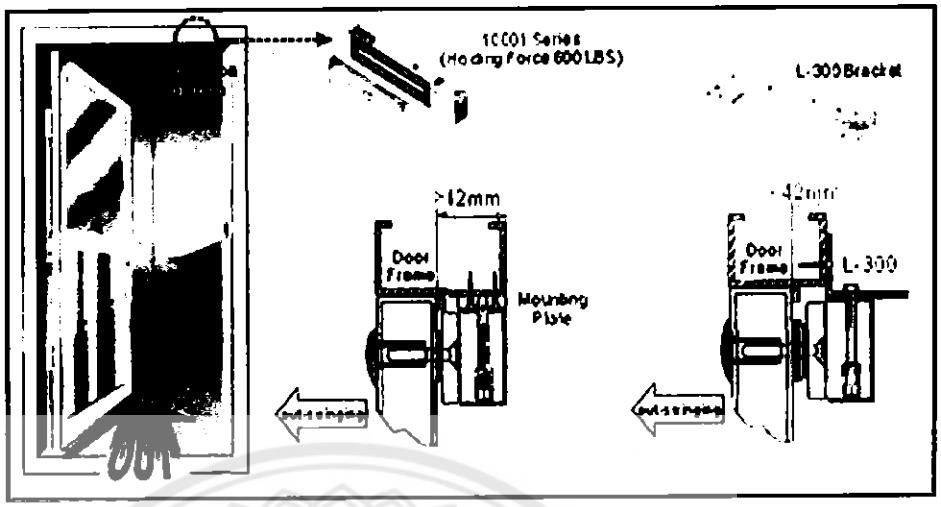
- ประกอบชิ้นส่วนชุดแม็กแนติกกับตัว



- ตามด้วย ชุด L - Z กับขอบประตู



- การติดตั้งตัวล็อกแม่เหล็กกับประตู

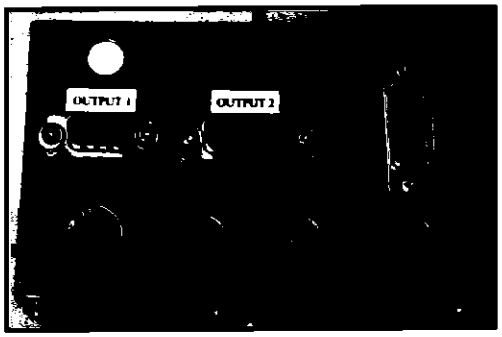


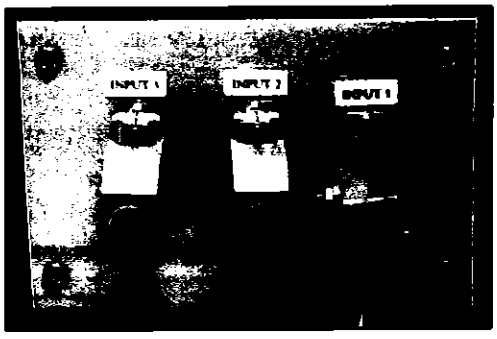
2. วิธีการต่อวงจรควบคุมการทำงาน

การทำงานของระบบมีการต่ออุปกรณ์ต่างๆ เข้าด้วยกัน เพื่อให้ระบบสามารถทำงานได้ ซึ่งอุปกรณ์เหล่านั้นประกอบด้วย หน้าจอแสดงผลแอลซีดี คีย์แพท สวิตซ์คำสั่ง คอมพิวเตอร์และวงจรควบคุมการทำงาน โดยมีวิธีการต่อดังนี้

- การต่อวงจรควบคุมการทำงานเข้ากับคอมพิวเตอร์และแหล่งจ่ายไฟฟ้า

- การต่อวงจรควบคุมการทำงานเข้ากับหน้าจอแสดงผลแอลซีดี คีย์แพท สวิตซ์คำสั่ง และตัวล็อกแม่เหล็กไฟฟ้า

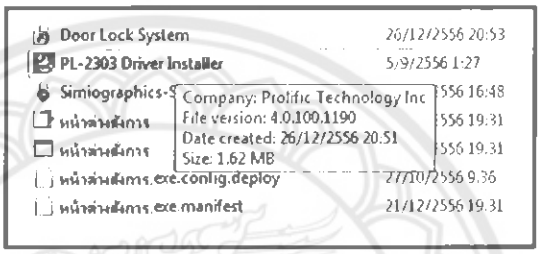




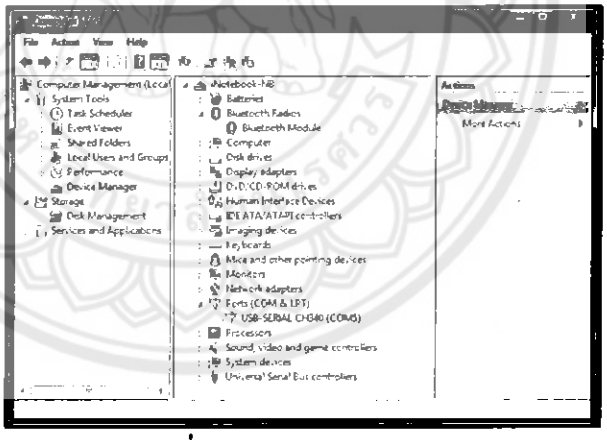
3. การติดตั้งโปรแกรมรับคำสั่งจากคอมพิวเตอร์ (Door Lock System)

การใช้งานระบบควบคุมผ่านคอมพิวเตอร์ ต้องมีการติดต่อสื่อสารผ่านพอร์ตอนุกรม ทำให้ต้องมีการติดตั้งโปรแกรมแปลงสัญญาณจากพอร์ต USB เป็นพอร์ตอนุกรม (RS-232) ในกรณีที่ผู้ใช้งานสั่งการจากเครื่องคอมพิวเตอร์แบบพกพา (Notebook) และเครื่องคอมพิวเตอร์แบบพีซีไม่ต้องมีการติดตั้งโปรแกรมแปลงสัญญาณจากพอร์ต USB เป็นพอร์ตอนุกรม (RS-232) เนื่องจากมีพอร์ตอนุกรมอยู่แล้วสามารถต่อใช้งานได้เลย

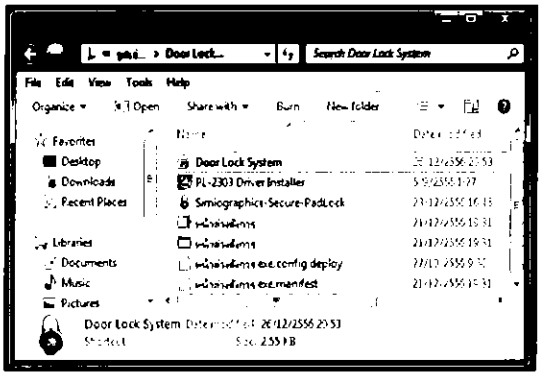
- โปรแกรมแปลงสัญญาณจากพอร์ต USB เป็นพอร์ตอนุกรม (RS-232)



- ตรวจสอบการเชื่อมต่อหลังจากติดตั้งโปรแกรมแปลงสัญญาณจากพอร์ต USB เป็นพอร์ตอนุกรม (RS-232)



- การใช้งานโปรแกรมรับคำสั่งสามารถคลิกเลือกไอคอนโปรแกรม (Door Lock System) ผู้ใช้งานสามรถล็อกอินเข้าใช้ระบบได้เลย



4. รายละเอียดของปุ่มสถานะคำสั่งต่างๆ

ลักษณะการใช้งานปุ่มกดต่างๆในการควบคุมด้วยมือมีรายละเอียดดังนี้

กดปุ่ม (UNLOCK)	การเลือกใช้คำสั่ง (UNLOCK) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการปลดล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (UNLOCK) แปลว่า อนุญาตให้เข้าพบ
กดปุ่ม (IN)	การเลือกใช้คำสั่ง (IN) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตูและหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (IN) แปลว่า อยู่ในห้อง
กดปุ่ม (OUT)	การเลือกใช้คำสั่ง (OUT) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (OUT) แปลว่า ไปข้างนอกไม่อยู่ในห้อง
กดปุ่ม (IN CLASS)	การเลือกใช้คำสั่ง (IN CLASS) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (IN CLASS) แปลว่า ไปห้องเรียน(สอน)
กดปุ่ม (MEETING)	การเลือกใช้คำสั่ง (MEETING) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (MEETING) แปลว่า ประชุม
กดปุ่ม (LUNCH)	การเลือกใช้คำสั่ง (LUNCH) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (LUNCH) แปลว่า พักกลางวัน
กดปุ่ม (BE BACK SOON)	การเลือกใช้คำสั่ง (BE BACK SOON) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (BE BACK SOON) แปลว่า กลับมาเร็วๆนี้
กดปุ่ม (PASSWORD)	การเลือกใช้คำสั่ง (PASSWORD) ระบบจะทำการส่งข้อมูล

ชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์จะทำการตรวจสอบรหัสผ่านจากการกดรหัสบนคีย์แพทและแสดงผลผ่านหน้าจอแอลซีดี ซึ่งการกดรหัสผ่านครบ 4 ตัวแล้วให้กด # เพื่อยืนยันรหัสผ่าน

ลักษณะไอคอนและการใช้งานปุ่มต่างๆในโปรแกรมรับคำสั่งมีรายละเอียดดังนี้

กดปุ่ม (UNLOCK) การเลือกใช้คำสั่ง (UNLOCK) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการปลด ล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (UNLOCK) แปลว่า อนุญาตให้เข้าพบ แสดงดังรูปที่ 3.17

กดปุ่ม (IN) การเลือกใช้คำสั่ง (IN) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (IN) แปลว่า อยู่ในห้อง

กดปุ่ม (MEETING) การเลือกใช้คำสั่ง (MEETING) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (MEETING) แปลว่า ประชุม

กดปุ่ม (LUNCH) การเลือกใช้คำสั่ง (LUNCH) ระบบจะทำการส่งข้อมูลชุดคำสั่งไปยังไมโครคอนโทรลเลอร์ เพื่อสั่งการให้ตัวล็อกแม่เหล็กทำการล็อกประตู และหน้าจอแสดงผลแอลซีดีจะแสดงสถานะ (LUNCH) แปลว่า พักกลางวัน

กดปุ่ม (HELP) การเลือกใช้คำสั่ง (HELP) โปรแกรมจะแสดงหน้าต่างขึ้นมาเพื่อช่วยอธิบายลักษณะการทำงานของปุ่มไอคอนรับคำสั่งต่างๆ ภายในหน้าโปรแกรมรับคำสั่ง แสดงดังรูปที่ 3.18

กด (Serial comport) การเลือกพอร์ต USB 2.0 เพื่อกำหนดใช้เป็นพอร์ตในการติดต่อ สื่อสารผ่านพอร์ตอนุกรม

กดปุ่ม (Sign out) การเลือกใช้คำสั่ง (Sign out) โปรแกรมจะแสดงหน้าขึ้นมาเพื่อให้ยืนยันการออกจะระบบ ใช่ หรือ ไม่ใช่