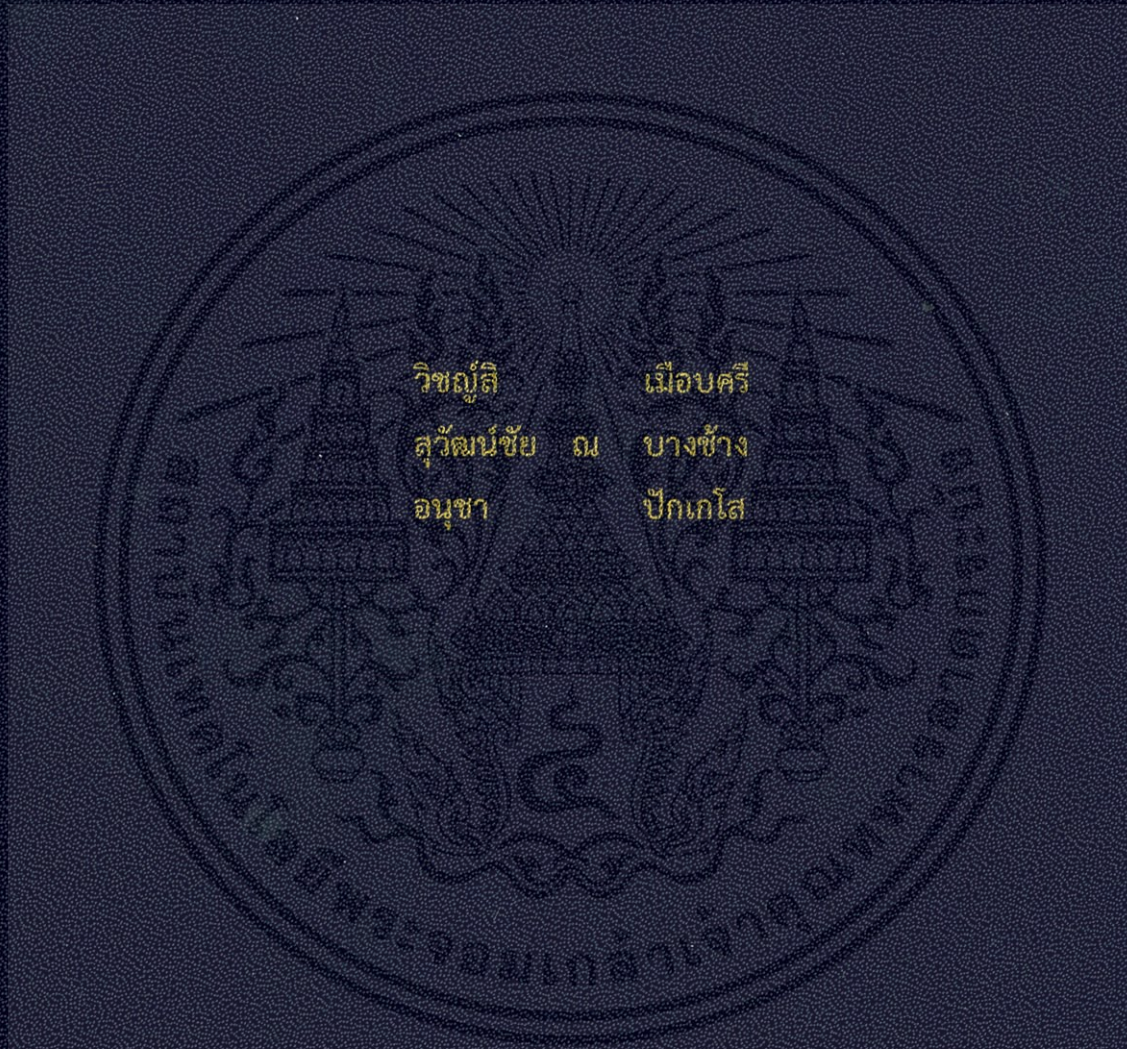


การควบคุมและการแสดงสภาวะแบบจำลองกระเช้าจอดรถด้วย PLC ผ่านโฮสต์
คอมพิวเตอร์

CONTROL AND MONITORING OF CAR PARK MODEL THROUGH
PLC BASED ON HOST COMPUTER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การควบคุมและการแสดงสภาวะแบบจำลองกระเช้าจอดรถด้วย PLC ผ่านโฮส
คอมพิวเตอร์

CONTROL AND MONITORING OF CAR PARK MODEL THROUGH
PLC BASED ON HOST COMPUTER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานนี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ปีการศึกษา 2556

CONTROL AND MONITORING OF CAR PARK MODEL THROUGH
PLC BASED ON HOST COMPUTER



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหารวมและข้อมูลอันใดเป็นซ้ำของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การควบคุมและการแสดงสถานะแบบจำลองกระเช้าจอดรถด้วย PLC ผ่าน
โฮสคอมพิวเตอร์
CONTROL AND MONITORING OF CAR PARK MODEL THROUGH
PLC BASED ON HOST COMPUTER

นักศึกษาผู้จัดทำ นายวิษณุสิ เมื่อศรี รหัสนักศึกษา 53011471
นายสุวัฒน์ชัย ณ บางช้าง รหัสนักศึกษา 53011801
นายอนุชา ปักเกโส รหัสนักศึกษา 53011841

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2556

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.สุพรรณ กุลพานิชย์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การควบคุมและการแสดงสภาวะแบบจำลองกระเช้าจอดรถด้วย PLC ผ่านโฮสคอมพิวเตอร์		
	CONTROL AND MONITORING OF CAR PARK MODEL THROUGH PLC BASED ON HOST COMPUTER		
นักศึกษาผู้จัดทำ	นายวิษณุลี	เมื่อบศรี	รหัสนักศึกษา 53011471
	นายสุวัฒน์ชัย	ณ บางช้าง	รหัสนักศึกษา 53011801
	นายอนุชา	ปีกเกโส	รหัสนักศึกษา 53011841
อาจารย์ที่ปรึกษา ปีการศึกษา	รศ.สุพรรณ	กุลพาณิชย์	2556

บทคัดย่อ

โครงการนี้ นำเสนอการควบคุมและการแสดงสภาวะระบบกระเช้าจอดรถจำนวน 7 ชุด ควบคุมการทำงานกระเช้าด้วย PLC โดยมีระบบขับเคลื่อนหมุนกระเช้าเป็นวงล้อ ผู้ควบคุมสามารถหยุดในตำแหน่งล่าสุดเพื่อเคลื่อนย้ายรถที่ต้องการเข้าจัดเก็บภายในกระเช้าวงล้อนั้น ระบบขับเคลื่อนหลักจะอาศัย DC MOTOR ขับชุดฟันเฟืองผ่านระบบโซ่ที่เป็นกลไกของเครื่องที่

การพัฒนาโปรแกรมการแสดงผลทั้งหมดอาศัยโปรแกรม VISUAL BASIC 2010 เป็นโปรแกรมหลักทำงานร่วมกับโปรแกรมฐานข้อมูล MICROSOFT ACCESS 2010 และเชื่อมโยงกับเครื่องควบคุม PLC ที่ควบคุมระบบขับเคลื่อนผ่านระบบสื่อสารข้อมูลแบบ HOST LINK ได้อย่างสมบูรณ์แบบ โครงการนี้สามารถพัฒนาต่อยอดเพื่อประโยชน์ในเชิงพาณิชย์ได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	CONTROL AND MONITORING OF CAR PARK MODEL THROUGH PLC BASED ON HOST COMPUTER	
Authors	Mr.Witchasi	Muebsri
	Mr.Suwatchai	Nabangchang
	Mr.Anucha	Pakkeso
Thesis Advisor	Assoc. Prof. Suphan	Kulphanit
Year	2013	

ABSTRACT

This project presents the methods for Control and Monitoring of Car Park model based on PLC through Host computer. This Car Park have 7 floors, that move like a cycle. User can manage the last position of parking floor to storage in the car park model. This model is operate by DC motor and gears set with chain system.

All of the monitoring part in this program is develop by VISUAL BASIC 2010. And this program is including the database by MICROSOFT ACCESS 2010. For the completely communicate between PLC, model and computer are based on HOST LINK protocol. Finally, this project can be develop for benefit to the full commercial management program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความอนุเคราะห์จาก รองศาสตราจารย์ สุพรรณ กุลพาณิชย์ ที่ได้ให้คำปรึกษา, คำแนะนำรวมถึงอุปกรรมแก่ผู้วิจัยตลอดมา คณาจารย์ภาควิชาวิศวกรรมการวัด คุมทุกท่านที่ได้มอบคำแนะนำและกำลังใจในการทำวิจัยตลอดมา เพื่อนๆ พี่ๆ ที่คอยช่วยให้กำลังใจในการทำ วิจัยและความช่วยเหลือในด้านต่างๆ

คุณค่า ประโยชน์และคำชมเชยอันได้จากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแต่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	IX
สารบัญภาพ	X

บทที่ 1 บทนำ	1
1.1 ความเป็นมาและเหตุจูงใจของการวิจัย	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	2
2.1 กล่าวนำ	2
2.2 ทฤษฎีและหลักการของ Visual Basic 2010	2
2.2.1 แนวทางการเขียนโปรแกรมแบบ Object Oriented	2
2.2.2 แนวทางการเขียนโปรแกรมแบบ Event Driven	3
2.2.3 ตัวแปรในภาษา Visual Basic	4
2.2.3.1 ตัวแปรแบบ Number	4
2.2.4 การประกาศตัวแปร	5
2.2.5 การใช้งานตัวดำเนินการ	5
2.2.6 ตัวแปรแบบ Floating Point	6
2.2.7 ตัวแปรแบบ String	6
2.2.8 คอนโทรล Textbox : กล่องข้อความ	7
2.2.8.1 พร็อพเพอร์ตี้ที่สำคัญของ Textbox	8

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ขอสงวนสิทธิ์ในนามของมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกสิ่งนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

สารบัญ(ต่อ)

หน้า

2.2.8.2	เมธอดสำคัญของ Textbox	9
2.2.9	คอนโทรล Label : แถบแสดงข้อความ	9
2.2.9.1	พรีอเพอร์ตีสำคัญของ Label	9
2.2.10	คอนโทรล Button : ปุ่มคำสั่ง	9
2.2.10.1	พรีอเพอร์ตีสำคัญของ Button	9
2.2.10.2	เมธอดสำคัญของ Button	10
2.2.10.3	อีเวนต์สำคัญของ Button	10
2.2.11	คอนโทรล Message Box : กล่องข้อความ	10
2.2.12	ตัวแปร Date และการจัดการต่างๆ	10
2.2.13	ตัวแปร Boolean	11
2.3	จุดเด่นของ Visual Basic	11
2.3.1	สร้างแอปพลิเคชันได้ง่ายและรวดเร็ว	11
2.3.2	การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้	11
2.3.3	รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม	11
2.4	รูปแบบการพัฒนาแอปพลิเคชันด้วย Visual Basic	12
2.5	เริ่มสร้างแอปพลิเคชันด้วย Visual Basic	13
2.6	ฐานข้อมูล (Database)	13
2.6.1	องค์ประกอบของฐานข้อมูล	13
2.6.2	ชุดคำสั่งที่ใช้จัดการกับข้อมูล	15
2.6.3	โครงสร้างของภาษา	15
2.6.3.1	ตัวดำเนินการ (Operator)	16
2.6.3.2	ตัวดำเนินการด้านตรรกะ (Logical Operator)	16
2.6.3.3	กลุ่มฟังก์ชัน Aggregate	16
2.6.4	ลักษณะการใช้งานของกลุ่มคำสั่ง DML	16
2.6.4.1	คำสั่ง INSERT	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.6.4.2 คำสั่ง SELECT	17
2.6.4.3 คำสั่ง UPDATE	18
2.6.5 การเชื่อมโยงตาราง	18
2.6.5.1 การเชื่อมโยงตารางโดยใช้คำสั่ง INNER JOIN	18
2.6.5.2 การเชื่อมโยงโดยใช้คำสั่ง LEFT JOIN	19
2.6.5.3 การเชื่อมโยงคำสั่งโดยให้คำสั่ง RIGHT JOIN	19
2.6.6 เทคโนโลยี ActiveX Data Object	20
2.6.7 วิธีติดต่อกับฐานข้อมูลของ Visual Basics 10	20
2.6.7.1 การใช้งานดาต้าคอนโทรล (Data Control)	21
2.7 การติดต่อสื่อสารและส่งผ่านข้อมูล	21
2.7.1 การสื่อสารข้อมูลแบบอนุกรม	22
2.7.1.1 การส่งข้อมูลแบบซิมเพล็กซ์ และ ดูเพล็กซ์	22
2.7.1.2 โพรโตคอลของการสื่อสารแบบอนุกรม	23
2.7.1.3 มาตรฐานสัญญาณอนุกรมแบบ RS232C	24
2.7.1.4 พอร์ตสื่อสารอนุกรม	25
2.8 ตัวควบคุมเชิงตรรกะแบบโปรแกรมได้ (Programmable logic control:PLC)	26
2.8.1 นิยามของตัวควบคุมเชิงตรรกะแบบโปรแกรมได้	26
2.8.2 ส่วนประกอบของตัวควบคุมเชิงตรรกะแบบโปรแกรมได้รุ่น CP1L	27
2.8.3 กลุ่มคำสั่งพื้นฐาน	28
2.8.3.1 การใช้คำสั่ง LOAD, LOAD NOT	28
2.8.3.2 การใช้คำสั่ง AND, AND NOT	28
2.8.3.3 การใช้คำสั่ง OR, OR NOT	29
2.8.3.4 การใช้คำสั่ง OUT, OUT NOT	29
2.8.3.5 การใช้คำสั่ง TIMER : TIM	29
2.8.3.6 การใช้คำสั่ง MOVE : MOV(21)	30
2.8.3.7 การใช้คำสั่ง END	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายเท่านั้น มิได้อนุญาตให้เผยแพร่หรือขึ้นต้นการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีลิขสิทธิ์เป็นของสถาบันฯ และต้องอ้างอิงถึงชื่อของสถาบันฯ

สารบัญ(ต่อ)

หน้า

2.9	การเชื่อมต่อแบบ Host – Link	31
2.9.1	กระบวนการในการรับ – ส่งคำสั่ง Host – Link	32
2.9.2	รูปแบบในการส่งคำสั่ง Host - Link.....	34
2.9.3	ตัวอย่างการคำนวณหา FCS.....	35
บทที่ 3	การออกแบบโปรแกรมควบคุม	36
3.1	กล่าวนำ	36
3.2	การออกแบบโปรแกรมควบคุม	36
3.2.1	โปรแกรมการทำงานของระบบ	36
3.2.2	โปรแกรมตรวจสอบสถานะชั้นจอตรด (ว่าง/ไม่ว่าง)	38
3.2.3	โปรแกรมแสดงสถิติการเข้าจอดในกระเช้าจอตรดยนต์	39
3.3	การออกแบบแลตเตอร์ไดอะแกรม	40
3.3.1	กำหนดเงื่อนไขของการทำงาน	40
3.3.1.1	กำหนดอินพุต และ เอาท์พุต ของ PLC	40
3.4	กำหนดการหมุนของกระเช้าจอตรดทั้ง 7 กระเช้า	40
3.4.1	เขียนแลตเตอร์ไดอะแกรม ตามเงื่อนไขที่กำหนด	40
3.5	การออกแบบ Visual Basic	41
3.5.1	หน้าแอดมิน	41
3.5.2	หน้ายูชเชอร์	42
3.6	รายละเอียดของโปรแกรมทั้งหมด	43
บทที่ 4	ผลการทดลอง	44
4.1	ผลการทดลอง RUN โปรแกรมการควบคุมกระเช้าจอตรดด้วยคอมพิวเตอร์	44
4.1.1	หน้าโปรแกรมเข้าสู่ระบบ (Login)	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

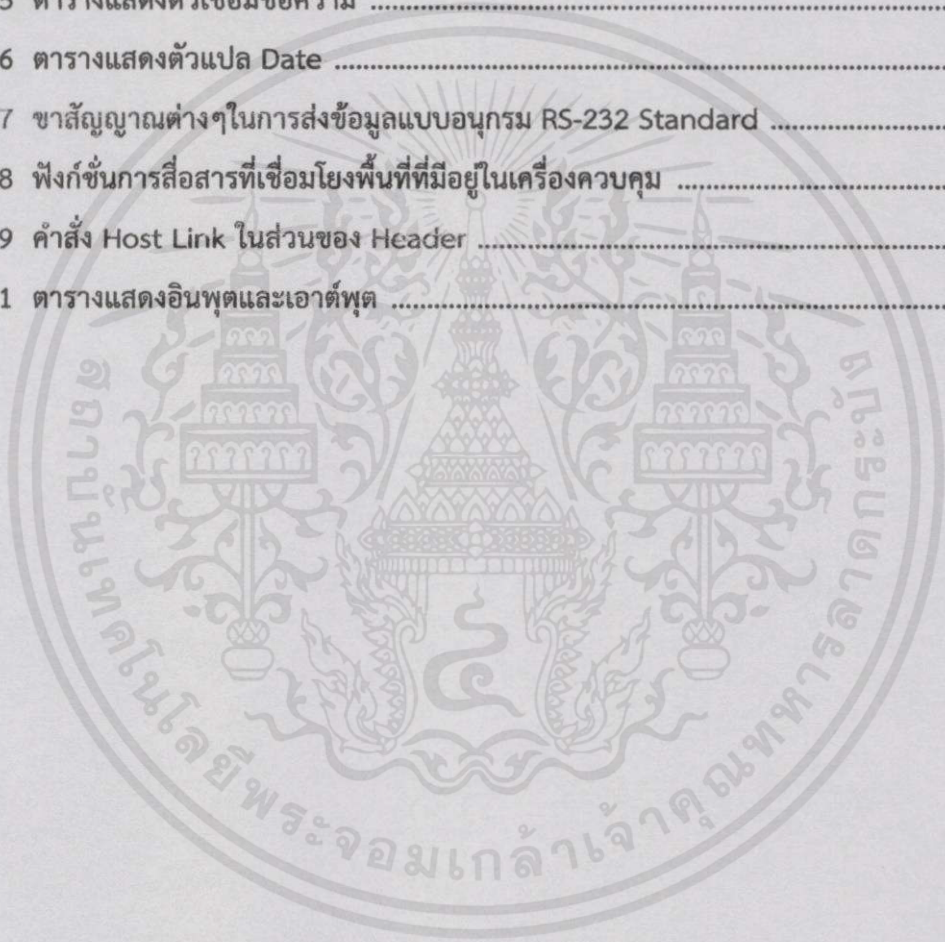
หน้า

4.1.2 หน้า ADMIN สำหรับผู้ดูแลระบบ	46
4.1.3 หน้า USER	49
4.2 สรุปผลการทดลอง	51
บทที่ 5 สรุปและวิจารณ์โครงงาน	52
5.1 สรุปโครงงาน	52
5.2 สรุปโครงงานโดยรวม	52
5.3 ปัญหาที่พบและอุปสรรคในการทำงาน	52
บรรณานุกรม	53
ภาคผนวก	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 2.1 ชนิดของตัวแปรในโปรแกรม Visual Basic 2010	4
ตารางที่ 2.2 ตารางแสดงตัวดำเนินการใน Visual Basic 2010	5
ตารางที่ 2.3 ตารางแสดงข้อมูลชนิด float	6
ตารางที่ 2.4 ตารางแสดงตัวแปรชนิด Char และ String	7
ตารางที่ 2.5 ตารางแสดงตัวเชื่อมข้อความ	7
ตารางที่ 2.6 ตารางแสดงตัวแปร Date	11
ตารางที่ 2.7 ขาสัญญานต่าง ๆ ในการส่งข้อมูลแบบอนุกรม RS-232 Standard	26
ตารางที่ 2.8 ฟังก์ชันการสื่อสารที่เชื่อมโยงพื้นที่ที่มีอยู่ในเครื่องควบคุม	31
ตารางที่ 2.9 คำสั่ง Host Link ในส่วนของ Header	33
ตารางที่ 3.1 ตารางแสดงอินพุตและเอาต์พุต	40



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่		หน้า
2.1	ตัวอย่างออบเจกต์กับผลิตภัณฑ์	3
2.2	การติดต่อสื่อสารและส่งผ่านข้อมูล	21
2.3	การส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์ (Half duplex) และ แบบฟูลดูเพล็กซ์ (Full duplex)	23
2.4	แบบมาตรฐานการสื่อสารข้อมูลแบบอะซิงโครนัส	24
2.5	ลักษณะสมบัติทางไฟฟ้าของ RS-232	24
2.6	พอร์ต RS-232 ชนิด DB9	25
2.7	พอร์ต RS-232 ชนิด DB25	25
2.8	ไดอะแกรมแสดงการประยุกต์ใช้งานของระบบ PLC	27
2.9	โครงสร้างภายนอกของตัวควบคุมเชิงตรรกะแบบโปรแกรมได้ รุ่น CP1L	27
2.10	ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง LD และ LD NOT	28
2.11	ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง AND และ AND NOT	28
2.12	ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง OR และ OR NOT	29
2.13	ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง OUT และ OUT NOT	39
2.14	ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง TIMER	29
2.15	ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง MOVE	30
2.16	กระบวนการรับ - ส่งคำสั่ง Host - Link	32
2.17	รูปแบบคำสั่งของ Host - Link	34
3.1	Flow chart แสดงขั้นตอนการทำงานของระบบกระเช้าจอดรถ	37
3.2	Flow chart แสดงการตรวจสอบสถานะชั้นจอดรถ	38
3.3	Flow chart โปรแกรมแสดงสถิติการเข้าจอดในกระเช้าจอดรถยนต์	39
3.4	หน้าจอโปรแกรม CX-Programmer	41
3.5	Shortcut Key ไว้สำหรับเขียนแลตเตอร์ตามเงื่อนไขต่างๆ	41
3.6	หน้าจอที่เป็นส่วนของ ADMIN	42
3.7	หน้าจอที่เป็นส่วนของ USER	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่	หน้า
4.1 ส่วนของหน้าจอ Login	44
4.2 การแสดง Forget Password	45
4.3 หน้าต่างตรวจสอบรหัสผ่าน	45
4.4 หน้าหลักของ ADMIN	44
4.5 ช่องสำหรับกรอกข้อมูลพนักงาน	44
4.6 แสดงข้อมูลพนักงานในระบบฐานข้อมูล	47
4.7 ข้อมูลในฐานข้อมูลที่แสดงการให้บริการลูกค้า	47
4.8 หน้าจอแสดงผลประกอบการ	48
4.9 กำหนดอัตราค่าบริการ	48
4.10 ส่วนของ USER	49
4.11 เมื่อมีรถเข้ามาจอดในชั้นที่ 1 หลังจากนั้นก็จะแสดงสถานะเป็นสีแดง	49
4.12 การดึงฐานข้อมูลเลขทะเบียนมาแสดงผล	50
4.13 แสดงการคำนวณค่าใช้จ่าย	50
4.14 ลูกค้าชำระบริการและนำรถออก	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันการใช้ชีวิตของคนส่วนใหญ่เริ่มเปลี่ยนแปลงจากสมัยก่อนที่เป็นยุคเกษตรกรรม ต่างคนต่างใช้ชีวิตในภูมิลาเนาที่เกิดมา แต่ในปัจจุบันโลกได้เปลี่ยนแปลงได้เข้าสู่ยุคการใช้ชีวิตแบบสังคมเมืองมากขึ้น คนจากภูมิภาคต่างๆ พยายามจะมาสร้างความมั่งคั่งและความสะดวกสบายในตัวเมืองกันมากขึ้นซึ่งเมื่อเกิดการอพยพเข้ามาใช้ชีวิตในเมืองกันมากขึ้นปัญหาหลายๆอย่างก็ได้ตามมา อาทิเช่น การจราจร ระบบสาธารณสุข อาชญากรรม รวมถึงปัญหาการจัดสรรพื้นที่อยู่อาศัย โดยการแก้ปัญหาความหนาแน่นของประชากรและการจัดสรรพื้นที่อยู่อาศัยถือเป็นเรื่องที่ต้องจำเป็นจะต้องแก้ไข เนื่องจากในอนาคตประชากรก็จะอพยพเข้ามาสู่เมืองมากขึ้นเรื่อยๆ โดยส่วนมากแต่ละครอบครัวจะมียานพาหนะเป็นของตนเองอย่างน้อยครัวเรือนละ 1 คัน ยกตัวอย่างในกรุงเทพมหานคร มีประชากรประมาณ 2,750,000 ครัวเรือน มีรถยนต์ประมาณ 7,000,000 คัน ซึ่งพื้นที่ของกรุงเทพมหานครไม่สามารถรองรับได้พอ เพราะเหตุเช่นนี้จึงเป็นที่มาของการดำเนินโครงการขึ้นนี้ขึ้นมา โดยการบริหารจัดการที่จอดรถนี้ต้องการที่จะลดปัญหาพื้นที่ไม่เพียงพอต่อความต้องการของผู้อาศัย โดยการใช้พื้นที่ในแนวตั้งมาแก้ปัญหา ซึ่งใช้ซอฟต์แวร์และโครงสร้างชิ้นงานที่ทางผู้จัดทำได้สร้างขึ้นมา โดยซอฟต์แวร์นี้พัฒนาโดยโปรแกรม Visual Basic 2010 ร่วมกับ PLC ในการควบคุม การแสดงผลรวมถึงการจัดการฐานข้อมูลสำหรับผู้ใช้งาน

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาการทำงานของ Proximity sensors (ชนิด Capacitive)
2. ศึกษาการทำงานและการเขียนคำสั่งบน PLC ด้วย Ladder diagram
3. ศึกษาการออกแบบและการใช้โปรแกรม Visual Basic 2010
4. ศึกษาการเชื่อมต่ออุปกรณ์แบบ Host link
5. ศึกษาการจัดการฐานข้อมูลด้วย Microsoft Access 2010

1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถควบคุมกระเช้าจราจรผ่านคอมพิวเตอร์ได้
2. สามารถแสดงผลของกระเช้าจราจรแบบเรียลไทม์ได้
3. สามารถจัดการข้อมูลการจราจรในกระเช้าจราจร

1.4 ขั้นตอนการศึกษา

1. ศึกษาระบบการทำงานของกระเช้าจราจร
2. ศึกษาการเขียน Ladder Diagram บน PLC
3. ศึกษาการเขียนภาษาเบสิกบน Visual Basic 2010
4. ศึกษาการเชื่อมต่ออุปกรณ์ระหว่าง PLC และโปรแกรม Visual Basic 2010

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามคัดลอกส่งมอบให้แก่บุคคลอื่นโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 กล่าวนำ

การควบคุมกระเช้าจอตrolley นั้นจะเป็นการนำเอาคอมพิวเตอร์เข้ามาใช้ในการควบคุม โดยทำการเขียนโปรแกรมสั่งงานเพื่อกำหนดเงื่อนไขและการทำงานผ่าน Visual Basic 2010 ในการเลื่อนขั้นรวมถึงการเข้าสู่ระบบ (login) ซึ่งโปรแกรมที่ได้พัฒนาขึ้นมาจาก Visual Basic 2010 นั้นจะรับคำสั่งสัญญาณมาจาก PLC ที่รับค่ามาจาก Proximity Sensor ในกระเช้าจอตrolley มาประมวลผลผ่านทางพอร์ตอนุกรม RS-232 ตัวโปรแกรมจะทำการประมวลผลและแสดงสถานะปัจจุบันของกระเช้าจอตrolley แล้วจึงทำตามเงื่อนไขต่างๆ ในโปรแกรมเพื่อขับเคลื่อนกระเช้าจอตrolley ต่อไป

2.2 ทฤษฎีและหลักการของ Visual Basic 2010

ปัจจุบันการพัฒนาซอฟต์แวร์ให้ทำงานบนระบบปฏิบัติการ Windows บนระบบอินเทอร์เน็ต หรือบนโทรศัพท์มือถือ สามารถใช้แนวคิดของการเขียนโปรแกรมแบบ Visual Programming ได้เป็นอย่างดี แนวทางการเขียนโปรแกรมหาดังกล่าวมีความแตกต่างจากแนวทางการเขียนโปรแกรมแบบดั้งเดิมเช่นการเขียนโปรแกรมภาษา C หรือ Java เป็นอย่างมาก สิ่งที่มีความสำคัญที่สุดสำหรับการเขียนโปรแกรมแบบ Visual Programming คือ เครื่องมือช่วยพัฒนาแอปพลิเคชัน ในปัจจุบันมีให้ใช้งานมากมายหลายตัว แต่ละตัวก็มีความสามารถในการสนับสนุนกระบวนการพัฒนาซอฟต์แวร์ในขั้นตอนต่างๆ เริ่มจากการออกแบบแอปพลิเคชัน การเขียนโปรแกรมตัวแปลภาษา เครื่องมือทดสอบ เครื่องมือที่ใช้ติดต่อฐานข้อมูล เป็นต้น การเขียนโปรแกรมแบบ Visual Programming มี 2 แนวทางที่เป็นที่นิยม คือ การเขียนโปรแกรมแบบ Object Oriented และ การเขียนโปรแกรมแบบ Event Driven

2.2.1 แนวทางการเขียนโปรแกรมแบบ Object Oriented

แนวทางการเขียนโปรแกรมแบบ Object Oriented สามารถเรียกได้อีกชื่อหนึ่งว่าการเขียนโปรแกรมเชิงวัตถุ การจะทำความเข้าใจกับแนวทางการเขียนโปรแกรมเชิงวัตถุนั้น ต้องทำความเข้าใจกับคำว่า ออบเจกต์ (Object) คุณสมบัติเฉพาะตัว (Property) และความสามารถที่เรียกใช้งานได้ (Method) เสียก่อน

ออบเจกต์ (Object) คือสิ่งที่สมมติขึ้นมาแทนวัตถุอย่างใดอย่างหนึ่ง เช่น ออบเจกต์รถยนต์ ออบเจกต์โทรศัพท์มือถือ ออบเจกต์บ้าน เป็นต้น ซึ่งออบเจกต์จะต้องมีลักษณะที่บ่งบอกคุณสมบัติเฉพาะตัว (Property) และความสามารถที่เรียกใช้งานได้ (Method)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 ตัวอย่างออบเจกต์กับผลิตภัณฑ์

จากรูปจะเห็นว่าออบเจกต์แต่ละชนิดมีPropertyและMethod ที่ใช้แยกแยะความแตกต่างระหว่างออบเจกต์คนละชนิดกัน เช่นออบเจกต์รถยนต์กับออบเจกต์โทรศัพท์มือถือมีProperty ที่แตกต่างกัน แต่ถ้าหากว่าเป็นออบเจกต์ชนิดเดียวกัน เช่น iPhone กับ Samsung Galaxy นั้นเป็นออบเจกต์ชนิดเดียวกันคือออบเจกต์สมาร์ทโฟน มี Method และ Property เหมือนกัน แตกต่างกันที่ค่าของ Property ไม่เหมือนกัน

2.2.2 แนวทางการเขียนโปรแกรมแบบ Event Driven

เอกสารฉบับนี้ใช้แนวทางการเขียนโปรแกรมแบบ Event Driven เป็นหลัก ปกติหากต้องเขียนโปรแกรมด้วยภาษา C หรือ Java จะต้องทำความเข้าใจกับขั้นตอนการทำงานเป็นอย่างดี ก่อน แล้วจึงเขียนโปรแกรมให้ครอบคลุมขั้นตอนการทำงานเหล่านั้น ซึ่งการทำแบบนี้จะได้ผลดีถ้าโปรแกรมนั้นมีขนาดเล็กและการทำงานไม่ซับซ้อน โดยเรียกแนวความคิดการเขียนโปรแกรมแบบนี้ว่า Structure Programming หรือการเขียนโปรแกรมเชิงโครงสร้าง แต่ถ้าโปรแกรมมีขนาดใหญ่และมีการทำงานที่ซับซ้อนมาก หรือเป็นระบบงานที่มีความหลากหลาย การใช้แนวความคิดการเขียนโปรแกรมแบบ Structure Programming จะทำให้เขียนโปรแกรมมีความยุ่งยาก การแก้ไขทำได้ยาก โดยเฉพาะเมื่อโปรแกรมนั้นทำงานบนระบบปฏิบัติการ Windows ดังนั้นไมโครซอฟต์จึงสร้างแนวคิดที่เรียกว่า Event Driven Programming หรือแนวความคิดการเขียนโปรแกรมแบบเหตุการณ์พาไป สำหรับการเขียนโปรแกรมบนระบบปฏิบัติการ Windows ซึ่งเป็นแนวคิดในการเขียนโปรแกรมที่แตกต่างจากการเขียนโปรแกรมเชิงโครงสร้าง

Event Driven Programming จะคิดว่า ถ้าเกิดเหตุการณ์แต่ละอย่างแล้ว จะจัดการอย่างไร เช่น แอปพลิเคชันมีปุ่มกดอยู่ 2 ปุ่ม ถ้าผู้ใช้งานคลิกปุ่ม <แสดงวันเดือนปี และเวลา> ก็จะเป็นเหตุการณ์หนึ่งโดยมีการแสดงวันประจำสัปดาห์ แต่ถ้าผู้ใช้งานกดปุ่ม <จบการทำงาน> ก็จะเป็นเหตุการณ์อย่างหนึ่งก็จะถามยืนยันว่าต้องการจะจบการทำงานของโปรแกรมหรือไม่

การเขียนโปรแกรมแบบเชิงโครงสร้าง มักจะเกิดปัญหาในการปรับปรุงแก้ไขโปรแกรม เพราะต้องคิดใหม่ทั้งหมด และมีผลกระทบกับทุกส่วน โดยเฉพาะโปรแกรมที่มีขนาดใหญ่และซับซ้อน ซึ่งเป็นข้อได้เปรียบของการโปรแกรมแบบ Event Driven โดย Visual Studio สนับสนุนการเขียนโปรแกรมแบบ Event Driven กับระบบปฏิบัติการ Windows มาโดยตลอดทำให้ได้รับความนิยมใช้งาน และสำหรับ Visual Studio 2010 Ultimate นั้นสามารถทำงานร่วมกับเครื่องมือพัฒนาแอปพลิเคชันตัวอื่น เช่น Delphi หรือ NetBean ได้อีกด้วย

ตัวอย่างเหตุการณ์

1. ในปุ่ม (Button) จะมีเหตุการณ์ เช่น
 - Click เหตุการณ์คลิก
 - Key press เหตุการณ์กดปุ่ม
2. ในกล่องข้อความ (Textbox) เช่น
 - Text Changed เหตุการณ์การเปลี่ยนแปลงค่าในกล่องข้อความ
 - Key press เหตุการณ์กดปุ่ม

2.2.3 ตัวแปรในภาษา Visual Basic

ข้อมูลที่ใช้ในการเขียนโปรแกรม Visual Basic 2010 มีหลายประเภท ในเบื้องต้นนี้สามารถแบ่งข้อมูลออกเป็นกลุ่มได้ดังนี้

Numeric เป็นข้อมูลตัวเลข ทั้งจำนวนเต็มและทศนิยม

String เป็นข้อมูลอักขระ (ตัวอักษรตัวเดียว) และข้อความ

Date Time เป็นข้อมูลที่อยู่ในรูปของวันเดือนปีและเวลา หรือมีทั้งสองแบบผสมกัน

Boolean เป็นข้อมูลที่มีค่าเป็นจริง (True) หรือเป็นเท็จ (False)

2.2.3.1 ตัวแปรแบบ Number

ชนิดของข้อมูลเลขจำนวนเต็มมีการแบ่งชนิดข้อมูล 8 ชนิด โดยมีรายละเอียดดังตาราง

ตารางที่ 2.1 ชนิดของตัวแปรในโปรแกรม Visual Basic 2010

ชนิดข้อมูล	คำอธิบาย	ขอบเขตข้อมูล
byte	8-bit signed integer	-128 ถึง 127 (2-7 ถึง 27-1)
Short	16-bit signed integer	-32,768 ถึง 32,767 (2-15 ถึง 215-1)
Integer	32-bit signed integer	-2,147,483,648 ถึง 2,147,483,647 (2-31 ถึง 231-1)
Long	64-bit signed integer	-9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807 (2-63 ถึง 263-1)
Byte	8-bit unsigned integer	0 ถึง 255 (0 ถึง 28-1)
Short	16-bit unsigned integer	0 ถึง 65,535 (0 ถึง 216-1)
integer	32-bit unsigned integer	0 ถึง 4,294,967,295 (0 ถึง 232-1)
long	64-bit unsigned integer	0 ถึง 18,446,744,073,709,551,615 (0 ถึง 264-1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ป 0 ถึง 264-1
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งไป

การใช้ชนิดของตัวแปรควรเลือกให้เหมาะสมกับขนาดของข้อมูลที่ใช้งานไม่กำหนดค่าเกินขอบเขตที่ตัวแปรชนิดนั้นจะรับได้ การประกาศตัวแปรให้มีขนาดใหญ่ครอบคลุมขนาดของข้อมูลจะมีผลเสียเพราะใช้พื้นที่เก็บข้อมูลเยอะกว่า

2.2.4 การประกาศตัวแปร

การประกาศตัวแปรเป็นการจองเนื้อที่ในหน่วยความจำสำหรับเก็บข้อมูลเพื่อใช้อ้างอิงในอนาคต การประกาศตัวแปรในภาษา Visual basic สามารถประกาศได้ดังนี้

Dim student ID as Integer
 Dim คำสั่งเพื่อใช้ในการประกาศตัวแปร
 student ID ชื่อตัวแปรที่ใช้เก็บค่า
 As ตัวเชื่อมบอกว่าเป็นตัวแปรชนิดใด
 Integer ชนิดของตัวแปร

2.2.5 การใช้งานตัวดำเนินการ

ในคำสั่งของภาษา Visual Basic จะพบกับการใช้ตัวดำเนินการ (Operator) ซึ่งเป็นตัวสำคัญของคำสั่ง ตัวดำเนินการจะคำนวณ หรือหาผลลัพธ์ที่เกิดขึ้นจากการกระทำในแต่ละตัว เช่น $a + b$ คือ มีตัวดำเนินการ + ทำการบวกค่าของตัวแปร a กับ b เข้าด้วยกัน โดยคำสั่งที่มีตัวดำเนินการ จะประกอบไปด้วยข้อมูล 2 ส่วนโดยมีตัวดำเนินการคั่นกลาง ข้อมูลที่นำมาใช้กับตัวดำเนินการเรียกว่า Operand ตัวดำเนินการที่พบได้บ่อยที่สุดคือตัวดำเนินการทางคณิตศาสตร์ ผลลัพธ์ที่ได้จากการคำนวณนั้นเกิดจากข้อมูลทั้ง 2 ตัวดำเนินการ ตัวดำเนินการทางคณิตศาสตร์นั้นมีรายละเอียดดังตาราง

ตารางที่ 2.2 ตารางแสดงตัวดำเนินการใน Visual Basic 2010

ตัวดำเนินการ	คำอธิบาย	ตัวอย่าง
+	บวก	$a + b$ ผลลัพธ์คือ a บวก b
-	ลบ	$a - b$ ผลลัพธ์คือ a ลบ b
*	คูณ	$a * b$ ผลลัพธ์คือ a คูณ b
/	หาร	a / b ผลลัพธ์คือ a หาร b
Mod	หาเศษจากการหาร	$a \text{ Mod } b$ ผลลัพธ์คือเศษจาก a หาร b เช่น $9 \text{ Mod } 2$ เท่ากับ 1
\	ผลหารที่เป็นจำนวนเต็ม	$a \setminus b$ ผลลัพธ์คือ เลขจำนวนเต็มที่ได
^	ยกกำลัง	$a ^ b$ ผลลัพธ์คือ a ยกกำลัง b
-	นิเสธ	-a ผลลัพธ์คือ นิเสธของ a เช่น $a = -9$ เพราะฉะนั้น $-a = 9$

2.2.6 ตัวแปรแบบ Floating Point

ข้อมูลชนิดเลขทศนิยม สามารถแบ่งย่อยได้เป็น 2 ชนิด

ตารางที่ 2.3 ตารางแสดงข้อมูลชนิด float

ชนิดข้อมูล	คำอธิบาย	ขอบเขตข้อมูล
Single	32-bit single-precision floating point	-3.4028235 × 10 ³⁸ ถึง -1.401298 × 10 ⁻⁴⁵ (สำหรับค่าลบ) 1.401298 × 10 ⁻⁴⁵ ถึง 3.4028235 × 10 ³⁸ (สำหรับค่าบวก) Significant Figure เป็น 7
Double	64-bit double-precision floating point	-1.79769313486232 × 10 ³⁰⁸ ถึง -4.94065645841247 × 10 ⁻³²⁴ (สำหรับค่าลบ) 4.94065645841247 × 10 ⁻³²⁴ ถึง 1.79769313486232 × 10 ³⁰⁸ (สำหรับค่าบวก) Significant Figure เป็น 15

ลักษณะของการเขียนตัวเลขทศนิยมนั้นอาจจะเขียนในเชิงวิทยาศาสตร์คือใช้ตัว “E” หรือ “e” ก็ได้ เพื่ออ้างถึง 10 ยกกำลัง เช่น 8.75E+4 (หรือ 8.75e+4) ก็จะมีหมายถึง 8.75×10⁴ หรือเท่ากับ 87,500 ส่วน 3.465E-2 ก็จะมีหมายถึง 3.45×10⁻² ซึ่งเท่ากับ 0.0345

จะเห็นว่าเลขทศนิยมที่ต้องการใช้ และระดับความละเอียดที่ต้องการสามารถเลือกได้ในกรณีที่ต้องการความละเอียดในการคำนวณสูง เช่น งานด้านวิทยาศาสตร์ วิศวกรรมศาสตร์ และการออกแบบนั้น จะใช้เลขทศนิยมชนิด Double

2.2.7 ตัวแปรแบบ String

ข้อมูลชนิดตัวอักษรและข้อความและข้อความนั้น Visual Basic มีชนิดข้อมูลอยู่ 2 แบบ คือ Char และ String

ในกรณีที่ข้อมูลนั้นเก็บตัวอักษรตัวเดียว จะใช้พื้นที่เก็บ 16 บิต หรือรู้จักกันทั่วไปในชื่อของ Unicode Character หรือ Char ซึ่งสามารถเก็บตัวอักษรได้ทุกตัวในโลกนี้ แต่สำหรับข้อความนั้นจะใช้ข้อมูลชนิด String ในการจัดเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ตารางแสดงตัวแปรชนิด Char และ String

ชนิดข้อมูล	ขนาดหน่วยความจำ	ขอบเขตข้อมูล
Char	2 byte	เป็นตัวอักษรหนึ่งตัว จะเก็บแบบ Unicode
String	แปรผันตามความยาวของข้อมูล	เป็นข้อความมีความยาวตั้งแต่ 0 ถึง 231 ตัวอักษร หรือราวสองพันล้านตัวอักษร

String concatenation คือตัวดำเนินการเชื่อมต่อข้อความใน Visual Basic 2010 มีตัวดำเนินการที่ใช้ในการเชื่อมต่อข้อความอยู่ 2 ตัว ได้แก่

ตารางที่ 2.5 ตารางแสดงตัวเชื่อมข้อความ

ตัวดำเนินการ	คำอธิบาย	ความหมาย/ตัวอย่าง
&	เชื่อมต่อข้อความ	“สวัสดี” & “VB2010” ผลลัพธ์คือ “สวัสดี VB2010”
+	เชื่อมต่อข้อความ	“สวัสดี” + “VB2010” ผลลัพธ์คือ “สวัสดี VB2010”

แม้ว่าเราสามารถใช้ตัวดำเนินการ & กับ + ได้เหมือนกัน แต่ในที่นี้นิยม & มากกว่าเพื่อไม่ให้สับสนกับ + ซึ่งมักจะใช้งานกับการบวก

อีกเหตุผลหนึ่งที่ & มีความเหมาะสมมากกว่า + เพราะก่อนที่ & จะนำข้อความมาเชื่อมต่อกันนั้น มันจะแปลงข้อมูลใน Operand ทั้ง 2 ตัว ให้เป็นข้อความก่อน แม้ว่าก่อนหน้านั้นจะเป็นข้อมูลชนิดใดก็ตาม

แต่ถ้าเป็น + เราต้องแปลงให้เป็นข้อมูล String เสียก่อน ไม่เช่นนั้น Visual Studio 2010 จะฟ้องความผิดพลาดขึ้นมา

2.2.8 คอนโทรล Textbox : กล่องข้อความ

Textbox เป็นคอนโทรลที่ใช้ในการแสดงข้อความ หรือให้กรอกข้อความเข้าไป
 เอกสารนี้เป็นภายใน ซึ่งสามารถแสดงข้อความได้ทั้งข้อความที่มีบรรทัดเดียวและหลายบรรทัด รวมทั้งกำหนดว่าจะ
 ให้ผู้ใช้สามารถแก้ไขหรือเพิ่มเติมได้หรือไม่
 ไม่ว่าจะวิธีใดก็ตามก็ให้ผู้ใช้สามารถแก้ไขหรือเพิ่มเติมได้หรือแก้ไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.8.1 พร็อพเพอร์ตี้สำคัญของ Textbox

Read Only	จะกำหนดว่า Textbox นั้นสามารถแก้ไขข้อความได้หรือไม่ ปกติจะกำหนดเป็น False คือให้แก้ไขได้
Textbox Text	เป็นข้อความที่จะกำหนดให้แสดง
Fore Color	เป็นสีของตัวอักษรที่แสดงข้อความใน Textbox (ปกติเป็นสีดำ)
Back Color	เป็นสีพื้นหลังของ Textbox
Multiline	เป็นการกำหนดว่าถ้าข้อความที่จะแสดงผลมีความยาวเกิน 1 บรรทัด จะยอมให้ขึ้นบรรทัดใหม่หรือไม่ ปกติจะกำหนดให้เป็น False คือไม่ยอม แต่ถ้าต้องการให้แสดงข้อความได้หลายบรรทัด ให้กำหนดให้เป็น True
MaxLength	กำหนดความยาวตัวอักษรมากที่สุดที่ยอมให้กรอกได้ใน Textbox (แต่ห้ามเกิน 32,767 ตัวอักษร)
Word Wrap	เป็นการกำหนดให้ตัดข้อความให้พอดีที่จะแสดงเฉพาะใน Textbox
Character Casing	เป็นการกำหนดว่าจะแปลงข้อความใน Textbox ให้เป็นตัวอักษรพิมพ์เล็ก/พิมพ์ใหญ่ทั้งหมดหรือไม่
PasswordChar	เป็นการกำหนดรูปแบบตัวอักษรที่แสดงออกมา เมื่อกำหนดให้ Textbox ทำหน้าที่รับการป้อนรหัสผ่าน โดยปกติจะใช้เครื่องหมายดอกจัน (*)
Lines	เป็นพร็อพเพอร์ตี้ในลักษณะอาร์เรย์ที่จะเก็บข้อความแต่ละบรรทัด ซึ่งสามารถกำหนดข้อความให้กับแต่ละบรรทัดได้ทั้งช่วง Design Time และ Run Time พร็อพเพอร์ตี้นี้เป็นแบบอ่านค่าได้อย่างเดียว
Can Undo	เป็นพร็อพเพอร์ตี้ที่ยอมให้ Undo หลังจากแก้ไขข้อความใน Textbox
Selected Text	เป็นข้อความที่ได้จากการคลิกลากแถบเลือกข้อความ
Selection Start	เป็นตำแหน่งเริ่มต้นที่ได้จากการคลิกลากแถบเลือกข้อความ
Selection Length	เป็นความยาวของตัวอักษรที่ได้จากการคลิกลากแถบ เลือกข้อความ
Scrollbar	จะเป็นการกำหนดให้มี Scroll Bar หรือไม่ เมื่อความยาวข้อความที่กรอกเข้าไป ยาวกว่าขนาดของ Textbox ซึ่งสามารถเลือกได้ว่า จะให้มี Scrollbar ในแนวนอนหรือแนวตั้ง หรือทั้งคู่ โดยก่อนจะกำหนดพร็อพเพอร์ตี้นี้ได้ต้องกำหนด Multiline = True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.8.2 เมธอดสำคัญของ Textbox มีดังนี้

Clear	เคลียร์ข้อความใน Textbox
Undo	นำข้อความก่อนการแก้ไขล่าสุดกลับมาแสดงใน Textbox อีกครั้ง โดยต้องกำหนด Can Undo = True อีเวนต์สำคัญของ Textbox
Text Changed	เป็นอีเวนต์ที่เกิดขึ้นเมื่อข้อความใน Textbox เปลี่ยนแปลงไปจากเดิม

2.2.9 คอนโทรล Label : แถบแสดงข้อความ

Label เป็นแถบอักษร หรือป้ายตัวอักษรที่กำหนดข้อความลงไปได้ใน ขณะที่ใช้งาน ผู้ใช้งานจะแก้ไขข้อความนี้ไม่ได้ ซึ่งมักจะใช้ควบคู่กับคอนโทรลตัวอื่นๆ เพื่อให้ผู้ใช้งานอ่านข้อความ ช่างในเพียงอย่างเดียว

2.2.9.1 พร็อพเพอร์ตี้สำคัญของ Label

Text	เป็นข้อความที่แสดงในแถบตัวอักษร
Text Align	เป็นการจัดวางแนวของข้อความ
Auto Size	ปกติกำหนดให้เป็น False คือขนาดคงที่ ถ้ากำหนดเป็น True จะทำให้ Label เปลี่ยนขนาดได้ตามความยาวของตัวอักษรใน Label
Border Style	รูปแบบของเส้นขอบปกติจะไม่มี (None) แต่สามารถกำหนดได้เป็น Fixed Single หรือ Fixed3D
Font	รูปแบบฟอนต์ที่ใช้แสดงข้อความ
Image	เป็นรูปภาพที่จะแสดงใน Label
Fore Color	เป็นสีของตัวอักษรที่แสดงข้อความ
Back Color	เป็นสีของพื้นหลัง Use Mnemonic ถ้ากำหนดเป็น True ตัวอักษรหลัง เครื่องหมาย & จะมีการขีดเส้นใต้เพื่อใช้เป็นคีย์ลัด

2.2.10 คอนโทรล Button : ปุ่มคำสั่ง

Button เป็นคอนโทรลที่พบได้บ่อยที่สุด มีลักษณะเป็นปุ่มที่สามารถกดปุ่ม <Enter> หรือคลิกเมาส์ เพื่อบอกให้แอปพลิเคชันทราบว่าได้เลือกตัวเลือกนั้นโดยปกติจะใช้ Button แทนคำสั่ง 1 คำสั่ง ซึ่งการทำงานในคำสั่งนั้นจะเรียบง่าย หรือซับซ้อนก็ขึ้นอยู่กับโค้ดที่เขียนไว้ในอีเวนต์ Click ของคอนโทรลนี้

2.2.10.1 พร็อพเพอร์ตี้สำคัญของ Button

Text	เป็นข้อความที่จะแสดงบน Button
Text Align	เป็นการจัดวางแนวของข้อความบนปุ่ม (ทำได้ 9 รูปแบบเช่นเดียวกับ คอนโทรล Label)
Auto Size	ปกติกำหนดเป็น False ถ้ากำหนดเป็น True ขนาดของปุ่มจะถูกปรับขยายอัตโนมัติให้แสดงข้อความหรือรูปภาพได้พอดี
Image	เป็นรูปที่กำหนดให้กับปุ่มนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้อัปโหลดหรือเผยแพร่เอกสารนี้ไปยังช่องทางใดๆ ทุกครั้งที่มีการนำไปใช้

Image Align	เป็นการจัดตำแหน่งของรูปภาพที่แสดงบนปุ่ม (ทำได้ 9 รูปแบบ เช่นเดียวกับ Text Align)
Cursor	เป็นการกำหนดรูปแบบเคอร์เซอร์ของเมาส์ เมื่อเลื่อนเมาส์มาวางเหนือคอนโทรล
FlatStyle	เป็นลักษณะการแสดงผล ปกติกำหนดเป็น Standard คือปุ่มทั่วไป Popup คือเมื่อวางเมาส์เหนือคอนโทรลจึงจะนูนขึ้นมาเหมือนปุ่มทั่วไป (คล้ายกับคอนโทรลที่อยู่บน Toolbox)
2.2.10.2 เมธอดสำคัญของ Button	
Focus	เป็นการกำหนดให้ปุ่มนั้นถูก Focus สำหรับการกดปุ่ม <Enter>

2.2.10.3 อีเวนต์สำคัญของ Button

Click	เป็นอีเวนต์ที่เกิดขึ้นเมื่อคลิกปุ่มกดนั้น
-------	---

2.2.11 คอนโทรล Message Box : กล่องข้อความ

ในการโต้ตอบกับผู้ใช้งานที่ คอนโทรลที่ถูกใช้บ่อยที่สุดคือ Message Box ซึ่ง MessageBox นั้นจะใช้เมื่อต้องการบอกข้อมูลแก่ผู้ใช้งาน แล้วให้ผู้ใช้งานเลือกกระทำอย่างใดอย่างหนึ่ง ตัวอย่างเช่น การยืนยันการปิดโปรแกรม หรือเตือนผู้ใช้ให้ทราบเมื่อเกิดเหตุการณ์ที่ผิดพลาด เป็นต้น

พารามิเตอร์ที่ใช้ในงานในเมธอด Show มีดังนี้

Text	เป็นข้อความที่ปรากฏใน MessageBox ซึ่งจะเป็ข้อความที่ต้องการบอกแก่ผู้ใช้งาน
Caption	เป็นข้อความบนแถบด้านบนของ MessageBox
Button	เป็นปุ่มที่จะให้ผู้ใช้งานเลือกกระทำหลังจากอ่านข้อความแล้ว
Icon	เป็นไอคอนที่แสดงประกอบกับข้อความเพื่อแสดงถึงอารมณ์ของข้อความ
DefaultButton	เป็นการกำหนดปุ่ม Default (ถ้ากดปุ่ม <Enter> จะเป็นการคลิกปุ่มนี้) ว่าเป็นปุ่มที่เท่าไร (นับจากซ้ายมาขวา)

2.2.12 ตัวแปร Date และการจัดการต่างๆ

ข้อมูลเกี่ยวกับวัน เดือน ปี และเวลาใน Visual Basic นั้น ในเบื้องต้นให้เข้าใจว่าเป็นข้อมูลชนิด Date ไว้ก่อน ซึ่งจะมีข้อมูลทั้งส่วนที่เป็นวัน เดือน ปี และส่วนที่เป็นเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 ตารางแสดงตัวแปร Date

ชนิดข้อมูล	ขนาดหน่วยความจำ	ขอบเขตข้อมูล
Date	8 Byte	วันเวลา ตั้งแต่ 00:00:00 ของวันที่ 1 มกราคม ค.ศ. 0 ถึง 23:59:59 ของวันที่ 31 ธันวาคม ค.ศ. 9999

2.2.13 ตัวแปร Boolean

ข้อมูลชนิดตรรกะ หรือ Boolean ใช้เก็บข้อมูลด้านตรรกศาสตร์ซึ่งก็คือ ค่าจริง (True) และเท็จ (False) เท่านั้น ซึ่งในภาษาโปรแกรมบางภาษา True หรือ False อาจแทนด้วย 1 และ 0 หรือ 0 กับ -1 ก็มี แต่ใน Visual Basic 2010 นั้น True ก็คือ True และ False ก็คือ False

2.3 จุดเด่นของ Visual Basic

2.3.1 สร้างแอปพลิเคชันได้ง่ายและรวดเร็ว

โปรแกรม Visual Basic ได้รับการยอมรับให้เป็นเครื่องมือที่ช่วยในการสร้างแอปพลิเคชันให้เป็นไปได้อย่างรวดเร็วและง่ายดาย เพื่อลดเวลาในการสร้างแอปพลิเคชันให้สั้นลง ซึ่งเรียกรูปแบบอันนี้ว่า Rapid Application Development หรือ RAD เพราะได้ทำการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำๆ ออกไปเหลือเฉพาะที่ต้องโฟกัสเกี่ยวกับปัญหาที่เกิดขึ้นของงานจริงๆ แล้วเขียนโปรแกรมจัดการปัญหานั้นๆ ส่วนเรื่องอื่นๆ จะให้ Visual Basic จัดการ

2.3.2 การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้

ถ้าหากมีโอกาสได้เขียนโปรแกรมด้วย Visual Basic แล้วจะเห็นว่าภาษา Basic ในตัวของ Visual Basic นั้นอ่านง่าย คืออ่านแล้วใกล้เคียงกับภาษาที่เราใช้งานกันปกติ ซึ่งทำให้สื่อความหมายเข้าใจได้ง่ายกว่าภาษาของโปรแกรมอื่นๆ ทำให้ผู้ที่เริ่มต้นในการเขียนโปรแกรมสามารถทำความเข้าใจกับการเขียนโปรแกรมได้อย่างรวดเร็ว

2.3.3 รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

Visual Basic มีเครื่องมือช่วยในการเขียนโปรแกรมที่ไม่ยุ่งยาก เพราะจะมีเครื่องมือที่ช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยากและจะตรวจสอบโดยอัตโนมัติว่าโปรแกรมที่เขียนนั้นมีความถูกต้องตามหลักของภาษาหรือไม่ ซึ่งจะมีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์นั้นลดลงได้มาก

นอกจากจะมีเครื่องมือช่วยในการเขียนโปรแกรมแล้ว ยังมีเครื่องมือที่ใช้ทดสอบแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online Help) ให้อ้างอิง และขอความช่วยเหลือในจุดที่เราสงสัยได้

เครื่องมือทั้งหมดที่กล่าวมาข้างต้นจะถูกจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกันเรียกย่อๆว่า IDE ซึ่งย่อมาจาก Integrate Development Environment ทำให้เรียกใช้งานได้สะดวก ตั้งแต่เขียนโปรแกรม, ทดสอบ, แก้ไข, สร้างชุดติดตั้ง รวมทั้งระบบการช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆเข้าไปได้ หรือถอดเครื่องมือที่ไม่จำเป็นต้องใช้ออกเพื่อประหยัดพื้นที่ฮาร์ดดิสก์ได้เช่นกัน

2.4 รูปแบบการพัฒนาแอปพลิเคชันด้วย Visual Basic

เมื่อมองเห็นว่าโปรแกรม Visual Basic สามารถช่วยให้เราทำการสร้างแอปพลิเคชันบน Windows ได้ง่าย และรวดเร็วแล้ว ยังมีรูปแบบหนึ่งที่โปรแกรม Visual Basic สร้างขึ้นมาได้อีก คือ

1. พัฒนาแอปพลิเคชันกับ ActiveX Control

เทคโนโลยีที่มีชื่อว่า ActiveX เป็นตัวอยู่เบื้องหลังความสำเร็จของ Visual Basic ซึ่งช่วยลดงานที่ซ้ำซ้อนของการเขียนโปรแกรม

ตัวอย่างเช่น การเขียนโปรแกรมเพื่อรับค่าของข้อมูลจากผู้ใช้แต่เดิมจะต้องมาเขียนโปรแกรมเพื่อวาดหน้าจอ, เขียนโปรแกรมวาดรูปของปุ่ม และช่องรับข้อความรวมทั้งเขียนโปรแกรมเพื่อจัดการกับข้อมูลที่ผู้ใช้ป้อนเข้ามา แต่ใน ActiveX จะทำให้เราสนใจเฉพาะการจัดการกับส่วนของข้อมูลที่ผู้ใช้ป้อนเข้ามาเท่านั้น ที่เหลือโปรแกรม Visual Basic จะจัดการให้เอง โดยช่องรับข้อความและปุ่มต่างๆนั้นเราจะทำการใช้ ActiveX Control จัดการ

นอกจากจะลดความซับซ้อนลงแล้วการใช้ ActiveX Control ช่วยในการเขียนโปรแกรมจะทำให้โปรแกรมที่เราเขียนกับโปรแกรมที่ผู้อื่นเขียนนั้น ตั้งอยู่บนมาตรฐานเดียวกัน ทำให้การปรับปรุงโปรแกรมทำได้ง่าย

2. สร้างแอปพลิเคชันที่ใช้งานกับฐานข้อมูล

เป็นแอปพลิเคชันที่มีการใช้งานกันมากที่สุดเพราะระบบร้านค้า, คลังสินค้า, ระบบบัญชี, ระบบบริหารงานบุคคลหรือแม้แต่ E-Commerce ทั้งหมดต่างๆ ต้องมีส่วนที่ติดต่อกับฐานข้อมูลที่แน่นอน

โปรแกรม Visual Basic ได้ช่วยในการสร้างแอปพลิเคชันกับฐานข้อมูลเป็นเรื่องที่ทำได้ง่ายดาย เพราะมีเครื่องมือต่างๆ ที่อำนวยความสะดวกในการเขียนโปรแกรมเพื่อใช้งานข้อมูลจากฐานข้อมูล และไม่จำกัดด้วยว่าเป็นฐานข้อมูลแบบใดทั้งฐานข้อมูลส่วนบุคคล, ฐานข้อมูลผ่านเครือข่าย, หรือฐานข้อมูลผ่านอินเทอร์เน็ต จากความสามารถที่หลากหลายนี้จึงทำให้ Visual Basic เหมาะกับการสร้างแอปพลิเคชันที่เกี่ยวกับฐานข้อมูลในโลกธุรกิจยุคปัจจุบันนี้

3. สร้างแอปพลิเคชันแบบใหม่กับอินเทอร์เน็ต

ปัจจุบันอินเทอร์เน็ตนับว่ามีความสำคัญกับชีวิตของคนที่ใช้ไอทีมากขึ้น ซึ่ง Visual Basic เปิดโอกาสให้เราสามารถสร้างแอปพลิเคชันเพื่อรองรับการทำงานร่วมกับอินเทอร์เน็ตได้ด้วย

จุดเด่นของ Visual Basic อีกข้อหนึ่งคือ เปิดโอกาสให้เรานำแอปพลิเคชันปกติที่เดิมทำงานกับคอมพิวเตอร์ส่วนบุคคล สามารถดัดแปลงให้ใช้กับงานอินเทอร์เน็ตได้ไม่ยากทำให้ไม่ต้องถึงงานเดิมที่มีอยู่

2.5 เริ่มสร้างแอปพลิเคชันด้วย Visual Basic

ขั้นที่ 1: ออกแบบแอปพลิเคชัน

ต้องทราบให้แน่ชัดก่อนว่าแอปพลิเคชันที่เราจะสร้างนั้นใช้ประโยชน์อะไร, ต้องมีความสามารถอะไรบ้าง, มีรูปร่างหน้าตาเป็นอย่างไร ซึ่งจำเป็นที่จะต้องคิดให้รอบคอบและเขียนออกมาให้ชัดเจน โดยอาจจะร่างคร่าวๆ ในกระดาษเป็นต้น

ขั้นที่ 2: ตกแต่งหน้าแอปพลิเคชัน

เป็นการตกแต่งรูปร่างของแอปพลิเคชันตามที่ได้ออกแบบไว้ พร้อมกับการกำหนดค่าพารามิเตอร์ต่างๆ ให้กับคอนโทรลแต่ละตัวในแอปพลิเคชันนั้น

ขั้นที่ 3: เขียนโค้ดกำกับแอปพลิเคชัน

ขั้นตอนการเขียนโค้ด หรือการเขียนโปรแกรมเพื่อทำการควบคุมการทำงานของแอปพลิเคชันนั้น เราจะใช้การเขียนโปรแกรมแบบ Event Driven Programming ซึ่งจะเป็นการเขียนโค้ดเพื่อรองรับเหตุการณ์ต่างๆ ที่จะเกิดขึ้นกับตัวคอนโทรลต่างๆ ในแอปพลิเคชันของเรา

ขั้นที่ 4: การทดสอบการทำงานของแอปพลิเคชัน

เมื่อเขียนโค้ดเสร็จเรียบร้อยแล้ว ก็มาถึงเวลาที่จะทำการทดสอบการทำงานของแอปพลิเคชันที่ได้ทำการสร้างขึ้น ซึ่งประกอบไปด้วยคอนโทรลต่างๆ ที่ปรับแต่งไว้และโค้ดที่เขียนเพื่อจัดการกับเหตุการณ์ต่างๆ

ขั้นที่ 5: บันทึกเก็บไว้ในคอมพิวเตอร์

หลังจากทดสอบจนแน่ใจว่าแอปพลิเคชันที่สร้างนั้นทำงานได้ถูกต้อง เราจึงจะบันทึกเก็บไว้ ซึ่งสามารถแก้ไขและเพิ่มเติมความสามารถอื่นๆ ได้ในภายหลัง

ขั้นที่ 6: สร้างไฟล์ .EXE (Make)

เมื่อเราสร้างแอปพลิเคชันเสร็จแล้ว เราอาจจะต้องการที่จะนำเอาแอปพลิเคชันที่ได้สร้างขึ้นมานั้น ให้สามารถเรียกใช้งานได้เองโดยไม่ต้องเรียกผ่านโปรแกรม Visual Basic หรือต้องการนำไปใช้งานในคอมพิวเตอร์เครื่องอื่นๆ ซึ่งเราจะทำได้โดยการสร้างไฟล์เอกซ์คิวต์ (ที่มีนามสกุล .EXE)

2.6 ฐานข้อมูล (Database)

ฐานข้อมูล (Data Base) คือ วิธีการจัดเก็บข้อมูลที่สัมพันธ์กันอย่างเป็นระเบียบ ซึ่งจะช่วยให้ง่ายต่อการงานและค้นหาข้อมูล ซึ่งฐานข้อมูลที่คนส่วนใหญ่คุ้นเคยคือ ฐานข้อมูลเชิงสัมพันธ์ เป็นรูปแบบการจัดเก็บข้อมูลที่มีความสัมพันธ์กัน โดยจะมองข้อมูลในลักษณะของตารางต่างๆ ที่มีความสัมพันธ์กัน

2.6.1 องค์ประกอบของฐานข้อมูล

การใช้งานฐานข้อมูลจะต้องประกอบด้วยองค์ประกอบดังต่อไปนี้

1. แอปพลิเคชันฐานข้อมูล (Database Application)

แอปพลิเคชันฐานข้อมูล เป็นแอปพลิเคชันที่สร้างไว้ให้ผู้ใช้ติดต่อกับฐานข้อมูลได้อย่างสะดวก ซึ่งมีรูปแบบของการติดต่อกับฐานข้อมูลแบบ เมนู หรือกราฟฟิก โดยไม่จำเป็นต้องมีความรู้มากก็สามารถเรียกใช้ฐานข้อมูลได้

2. ระบบจัดการฐานข้อมูล (DBMS)

ระบบจัดการฐานข้อมูลเป็นซอฟต์แวร์ที่ทำหน้าที่จัดการข้อมูลในฐานข้อมูล ทั้งการจัดเก็บ การแสดงผล การค้นหา การสำรองข้อมูล เป็นต้น โดยจะเป็นเครื่องมือในการทำงานของผู้บริหารข้อมูล และเป็นตัวกลางที่เชื่อมผ่านระหว่างแอปพลิเคชันฐานข้อมูลที่สร้างขึ้นกับตัวข้อมูลในฐานข้อมูล เช่น Microsoft Access, FoxPro เป็นต้น

3. ดาต้าเบสเซิร์ฟเวอร์ (Database Server)

ดาต้าเบสเซิร์ฟเวอร์ เป็นคอมพิวเตอร์ที่ให้บริการในการจัดการฐานข้อมูล ซึ่งก็คือคอมพิวเตอร์ที่มีระบบจัดการฐานข้อมูล

4. ข้อมูล (Data)

ข้อมูล คือ ตัวเนื้อหาของข้อมูลที่เรากำลังใช้งาน ซึ่งจะถูกรักษาไว้ในหน่วยความจำของดาต้าเบสเซิร์ฟเวอร์ โดยจะถูกเรียกมาใช้จากระบบจัดการฐานข้อมูล

5. ผู้บริหารฐานข้อมูล (Database Administrator)

ผู้บริหารฐานข้อมูลเป็นคนที่ทำหน้าที่ในการดูแลข้อมูลที่มีในฐานข้อมูลผ่านระบบจัดการฐานข้อมูล ซึ่งจะทำการควบคุมให้การทำงานนั้นเป็นไปอย่างราบรื่นและยังทำหน้าที่ในการกำหนดผู้ที่มีสิทธิ์ในการใช้งานฐานข้อมูล จะกำหนดในเรื่องความปลอดภัยของการใช้งาน พร้อมทั้งดูแล Database Server ให้ทำงานอย่างปกติด้วย

ฐานข้อมูลมีหลักการพื้นฐานมาจากคณิตศาสตร์ในเรื่องทฤษฎีเซต (Theory of Set) โดยที่การจัดเก็บหรือแสดงข้อมูลให้ผู้ใช้เห็นจะเป็นแบบตารางข้อมูลต่างๆจะแสดงในรูปของตาราง โดยในแต่ละตารางนั้นจะเป็นการจัดรวบรวมข้อมูลที่ประเภทเดียวกันไว้ด้วยกัน แต่ละแถวที่ประกอบขึ้นเป็นตารางนั้นก็คือ Record ซึ่งจะเป็นที่เก็บข้อมูลแต่ละชุดของตารางนั้น และในแต่ละแถวก็จะประกอบด้วยคอลัมน์ซึ่งเป็นหน่วยย่อยที่แสดงคุณสมบัติของข้อมูลแต่ละแถว ในแต่ละตารางมักจะมีบางคอลัมน์หรืออาจจะหลายๆ คอลัมน์ประกอบกัน ที่สามารถบอกถึงความแตกต่างของฐานข้อมูลในแต่ละแถวได้ คอลัมน์ดังกล่าวเรียกว่า Primary Key เช่น คอลัมน์เลขประจำตัวพนักงานที่แตกต่างกันทุกคน และในการเชื่อมโยงตารางฐานข้อมูลที่มีหลายตารางนั้น จะกระทำได้อีกต่อเมื่อแต่ละตารางมีคอลัมน์เกี่ยวข้องกัน ซึ่งคอลัมน์ที่เกี่ยวข้องนี้เรียกว่า foreign key

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 ชุดคำสั่งที่ใช้จัดการกับข้อมูล

แบ่งออกเป็น 2 ประเภทหลักๆคือ

1. จัดการกับโครงสร้างฐานข้อมูล เป็นการสร้าง ลบ หรือการแก้ไขฐานข้อมูล และตารางเช่น สร้างตารางฐานข้อมูลของพนักงานขึ้นมาชุดคำสั่งต่างๆ ที่เกี่ยวกับการสร้างหรือเปลี่ยนแปลงโครงสร้างของฐานข้อมูลมีศัพท์เรียกว่า Data Definition Language หรือ DDL
2. จัดการกับส่วนของฐานข้อมูลที่มีอยู่ในตารางต่างๆ ในฐานข้อมูล เป็นการทำงานในเรื่องของการเพิ่ม การลบ หรือการแก้ไขข้อมูลในตารางต่างๆ เช่น การเพิ่มข้อมูลพนักงานใหม่เข้าไปในตารางข้อมูลพนักงาน คำสั่งประเภทนี้เรียกว่า Data Manipulation Language หรือ DML

2.6.3 โครงสร้างของภาษา

Structure Query Language หรือ SQL เป็นภาษามาตรฐานที่ใช้ในการจัดการกับฐานข้อมูลโดยเฉพาะที่เป็นแบบแอปพลิเคชันที่ใช้งานกับฐานข้อมูลแบบของ SQL เป็นภาษาที่มีประสิทธิภาพสูงสามารถเรียนรู้และใช้งานอย่างง่ายตาย จึงทำให้เป็นที่นิยมอย่างรวดเร็ว นอกจากนี้ยังสามารถใช้กับการจัดการฐานข้อมูลโดยผ่านโปรแกรมก็ได้ เช่น หากเราใช้ Microsoft Access เราก็สามารถพิมพ์คำสั่ง SQL ได้โดยตรงในโปรแกรม Access หรือ จะใช้คำสั่ง SQL ผ่าน Visual Basic ก็ได้ เราจะต้องใช้ SQL เพื่อทำการจัดการข้อมูลในฐานข้อมูลได้หลายอย่าง เช่น การแสดงข้อมูลจากฐานข้อมูลแบบมีเงื่อนไข การเพิ่ม การลบ และการนำข้อมูลจากตารางหลายๆตารางมาแสดงร่วมกันได้ เป็นต้น

ภาษาของ SQL ประกอบไปด้วย 3 ส่วนใหญ่ๆคือ

1. Data Definition Language (DDL) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับโครงสร้างของฐานข้อมูล เช่น การสร้างฐานข้อมูล ปรับปรุงโครงสร้างของฐานข้อมูล เป็นต้น
2. Data Manipulation Language (DML) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการข้อมูลในฐานข้อมูล เช่น การแสดงข้อมูลแบบมีเงื่อนไข การลบข้อมูล การเพิ่มข้อมูล และการแสดงที่มาจากตารางหลายๆตาราง เป็นต้น
3. กลุ่มฟังก์ชัน Aggregate Function เป็นฟังก์ชันพิเศษของภาษา SQL ที่มีหน้าที่เฉพาะอย่าง เช่น หาผลรวมของเรคคอร์ด ค่าสูงสุด ค่าต่ำสุด เป็นต้น เป็นกลุ่มฟังก์ชันที่เป็นประโยชน์มาก เพราะจะช่วยลดภาระให้ไม่ต้องเขียนโค้ดจัดการเอง

สำหรับการใช้งานภาษา SQL ร่วมกับ Visual Basic เพื่อจัดการฐานข้อมูล จะใช้งานกลุ่มคำสั่ง DML ร่วมกับกลุ่มฟังก์ชัน Aggregate และกำหนดเงื่อนไขโดยใช้ตัวดำเนินการด้านต่างๆ

คำสั่งในกลุ่ม DML จะมีคำสั่งพื้นฐานอยู่ 4 คำสั่งคือ

1. DELETE เป็นคำสั่งที่ใช้สำหรับลบข้อมูลหรือลบเรคคอร์ดใดๆในตาราง
2. INSERT เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูลหรือเพิ่มเรคคอร์ดใดๆเข้าไปในตาราง
3. SELECT เป็นคำสั่งที่ใช้สำหรับเลือกข้อมูลหรือแสดงเรคคอร์ดที่ต้องการ
4. UPDATE ใช้สำหรับแก้ไขข้อมูลหรือแก้ไขเรคคอร์ดใดๆในตาราง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาและวิจัยเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีารนำไปใช้

2.6.3.1 ตัวดำเนินการ (Operator)

ตัวดำเนินการเปรียบเทียบที่น่าสนใจ ได้แก่

- = คือ เท่ากับ (Equality)
- <> คือ ไม่เท่ากับ (Inequality)
- < คือ น้อยกว่า (Less Than)
- > คือ มากกว่า (Greater Than)
- <= คือ น้อยกว่าหรือเท่ากับ (Less Than or Equal To)
- >= คือ มากกว่าหรือเท่ากับ (Greater Than or Equal To)

2.6.3.2 ตัวดำเนินการด้านตรรกะ (Logical Operator)

ตัวดำเนินการด้านตรรกะที่นิยมใช้จะมีอยู่ 3 คือ And, Or, Not

2.6.3.3 กลุ่มฟังก์ชัน Aggregate

กลุ่มฟังก์ชัน Aggregate เป็นฟังก์ชันที่ใช้คำนวณทางคณิตศาสตร์ โดยช่วยให้การนำเสนอผลการค้นหาเป็นไปอย่างมีประสิทธิภาพ ซึ่งมีฟังก์ชันพื้นฐานดังต่อไปนี้

AGV ()	หน้าที่ หาค่าเฉลี่ยของฟิลด์จากรีคคอร์ดทั้งหมด
COUNT ()	หน้าที่ นับจำนวนเรคคอร์ด
FIRST	หน้าที่ หาค่าแรกในฟิลด์
LAST	หน้าที่ หาค่าสุดท้ายในฟิลด์
MAX	หน้าที่ หาค่ามากที่สุด หรือค่าสูงสุด
MIN	หน้าที่ หาค่าน้อยที่สุด หรือค่าต่ำสุด
SUM	หน้าที่ หาผลรวมทั้งหมดของฟิลด์

2.6.4 ลักษณะการใช้งานของกลุ่มคำสั่ง DML

เป็นคำสั่งที่ใช้สำหรับ ลบ หรือ เร็คคอร์ดใดๆ ออกจากตารางมีรูปแบบการใช้งาน 2 ลักษณะใหญ่ๆ คือ

รูปแบบที่ 1	DELETE FORM	ชื่อตาราง WHERE เงื่อนไข
รูปแบบที่ 2	DELETE FORM	ชื่อตาราง

ชื่อตารางนี้ หมายถึงชื่อของตารางที่ต้องการลบส่วนเงื่อนไขจะหมายถึงเงื่อนไขที่ใช้ใน

การลบข้อมูล หรือเร็คคอร์ดในตารางนั้น

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4.1 คำสั่ง INSERT

เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูล หรือเพิ่มเรคคอร์ดเข้าไปในตารางในกรณีที่ฟิลด์เป็นข้อมูลชนิด TEXT ต้องใช้เครื่องหมาย “,” กำกับด้วยฟิลด์นั้นด้วยมีรูปแบบการใช้งาน 2 ลักษณะ

รูปแบบที่ 1 INSERT INTO table name (field1, field2,...) VALUES(value1, value2,...)
รูปแบบที่ 2 INSERT INTO table name 1

ตัวแปร Table name	หมายถึง ชื่อตารางที่ต้องการเพิ่มเรคคอร์ดเข้าไป
ตัวแปร Table channel	หมายถึง เลือกข้อมูลจากตารางที่ชื่อ tablename1 ตามเงื่อนไขที่ระบุไว้ในตัวแปร criteria แล้วนำมาเพิ่มที่ตาราง tablename2
ตัวแปร Field	หมายถึง ชื่อของฟิลด์ต่างๆที่อยู่ในตารางที่ต้องการเพิ่มข้อมูล
ตัวแปร Value	หมายถึงค่าของฟิลด์จะเพิ่มเข้าไปโดยที่ต้องระบุค่าให้ตรงกับฟิลด์

2.6.4.2 คำสั่ง SELECT

สำหรับเลือกดึงข้อมูล (Retrieve Data) ที่เราต้องการจากตารางที่ระบุไว้ในคำสั่งที่มีความยืดหยุ่นสูงมาก เพราะว่าเงื่อนไขในการนำข้อมูลออกมาจากตารางมีมากมายหลายลักษณะแต่มีรูปแบบการใช้งานพื้นฐานอยู่ 2 ลักษณะคือ

SELECT*FORM ชื่อตาราง

หรือ

Select ฟิลด์ที่1,ฟิลด์ที่2,...,ฟิลด์ที่n FORM ชื่อตาราง WHERE เงื่อนไข

ชื่อตาราง	หมายถึง ชื่อตารางที่ต้องการดึงข้อมูล
ตัวแปร ฟิลด์ที่1,ฟิลด์ที่2,...,ฟิลด์ที่n	หมายถึง ชื่อฟิลด์ที่ต้องการดึงข้อมูลถ้ามีมากกว่า 1 ฟิลด์จะใช้เครื่องหมาย,คั่นระหว่างฟิลด์
เงื่อนไข	หมายถึง เงื่อนไขในการดึงข้อมูลอาจเป็นคำสั่งทางคณิตศาสตร์ หรือเป็นคำสั่ง SELECT ซ้อนอยู่ข้างไหนก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ โดยสงวนลิขสิทธิ์ไว้เพื่อประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอยู่ข้างไหนก็ได้

2.6.4.3 คำสั่ง UPDATE

เป็นคำสั่งที่ใช้สำหรับแก้ไข หรือเปลี่ยนแปลงข้อมูลในเรคคอร์ดที่มีอยู่ในตาราง มีรูปแบบการใช้งานดังต่อไปนี้

```
UPDATE ชื่อตาราง SET ชื่อฟิลด์ = ค่าที่กำหนด WHERE เงื่อนไข
```

2.6.5 การเชื่อมโยงตาราง

การเชื่อมโยงตารางโดยใช้คำสั่ง SELECT วิธีนี้เป็นการเชื่อมโยงตาราง โดยระบุฟิลด์ที่มีความสัมพันธ์กัน เช่น

```
SELECT Student, FirstName, LastName, MajorDesc
FROM Student, Major
WHERE Student, MajorID = majored
```

2.6.5.1 การเชื่อมโยงตารางโดยใช้คำสั่ง INNER JOIN

เป็นการเชื่อมโยงของข้อมูลจาก 2 ตารางเข้าด้วยกันโดยระบุชื่อฟิลด์ที่มีความสำคัญกันหลังคำสั่ง ON เป็นเงื่อนไขที่ใช้การ JOIN ตาราง โดยที่ฟิลด์ดังกล่าว จะต้องเป็นข้อมูลชนิดเดียวกัน มีลักษณะเช่นเดียวกับการใช้คำสั่ง SELECT มีรูปแบบการใช้งานดังนี้

```
SELECT fieldname
FORM table name 1
INNER JOIN table name 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

ตัวแปรFieldname หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2

ตัวแปรTablename1 หมายถึงชื่อตารางที่1

ตัวแปรTablename2 หมายถึงชื่อตารางที่2

ตัวแปรFieldname1 หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 1

ตัวแปรFieldname2 หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 2

ตัวแปรOperation หมายถึงเงื่อนไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาคือเป็นมรดกของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่ข้อมูลและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5.2 การเชื่อมโยงโดยใช้คำสั่ง LEFT JOIN

การเชื่อมโยงของ 2 ตารางเข้าด้วยกัน โดยกำหนดให้ตารางแรก (ตารางทางซ้าย ของคำสั่ง LEFT JOIN) เป็นหลัก แล้วนำตารางที่ 2 เข้ามาทำการเชื่อมโยงตามเงื่อนไขที่ได้ระบุเอาไว้ มีรูปแบบการใช้งานได้ดังนี้

```
SELECT fieldname
FROM table name 1 LEFT JOIN table name 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

ตัวแปรFieldname	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2
ตัวแปรTablename1	หมายถึงชื่อตารางที่1
ตัวแปรTablename2	หมายถึงชื่อตารางที่2
ตัวแปรFieldname1	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่ 1
ตัวแปรFieldname2	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่ 2
ตัวแปรOperation	หมายถึงเงื่อนไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง

2.6.5.3 การเชื่อมโยงคำสั่งโดยให้คำสั่ง RIGHT JOIN

จะเป็นการเชื่อมโยง 2 ตารางเข้าด้วยกัน แต่กำหนดให้ตารางที่อยู่ทางด้านขวาของคำสั่ง RIGHT JOIN เป็นหลัก แล้วนำเร็คคอร์ดของตารางที่อยู่ทางด้านซ้ายของคำสั่ง RIGHT JOIN เข้ามาร่วมแสดงเฉพาะที่เร็คคอร์ดที่ตรงตามเงื่อนไขที่ระบุไว้ในรูปแบบการใช้งานดังนี้

```
SELECT fieldname
FROM table name 1 RIGHT JOIN table name 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

ตัวแปรFieldname	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2
ตัวแปรTablename1	หมายถึงชื่อตารางที่1
ตัวแปรTablename2	หมายถึงชื่อตารางที่2
ตัวแปรFieldname1	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่ 1
ตัวแปรFieldname2	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่ 2
ตัวแปรOperation	หมายถึงเงื่อนไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น กรุณาอย่านำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.6 เทคโนโลยี ActiveX Data Object

ADO เป็นเทคโนโลยีที่มีแนวความคิดมาจาก DAO บางส่วน กล่าวคือจะมองฐานข้อมูลเป็นออบเจกต์เช่นกัน แต่จะใช้ OLEDB Provider เป็นตัวจัดการ ข้อมูลและโครงสร้างของฐานข้อมูลแทน โดยที่จะไม่นิยามออบเจกต์ขึ้นมา เพื่อแทนโครงสร้างของฐานข้อมูลแต่ละส่วนเหมือนกับ DAO แต่จะใช้วิธีการสร้าง OLEDB Provider ให้กับ RDBMS แต่ละชนิดแทน เช่น เมื่อฐานข้อมูลเป็นชนิด JET ก็จะใช้ Microsoft Jet OLEDB Provider ถ้าเป็น Oracle ก็จะใช้ Microsoft OLEDB Provider for Oracle เป็นต้น ทำให้ออบเจกต์ในโมเดล ADO ไม่ยุ่งยากซับซ้อนเหมือนออบเจกต์ DAO ใน Visual Basic สามารถเรียกใช้ OLEDB Provider ได้ 2 วิธีคือ

1. อาศัยคอนโทรล ADO Data (ADO Data Control) ร่วมกับกลุ่มคอนโทรล Bound & ActiveX Bound Control ที่มีคำว่า OLEDB ต่อท้าย
2. โดยเรียกใช้งานกลุ่มออบเจกต์ ADO โดยตรง

ชุดออบเจกต์ใน ADO 2.5

สำหรับในโมเดลของ ADO 2.5 จะประกอบด้วยออบเจกต์ 9 ตัวคือ

ออบเจกต์ Command	เป็นออบเจกต์ตัวกลางที่ใช้สำหรับการส่งคำสั่ง (Execute) ไปยังฐานข้อมูลเป้าหมายที่ต้องการติดต่อโดยมีศัพท์เรียกฐานข้อมูล ดังกล่าวว่า Data Sources
ออบเจกต์ Start	ใช้สำหรับเริ่มต้นเปิดการเชื่อมต่อกับฐานข้อมูล
ออบเจกต์ Error	จะเป็นออบเจกต์ที่เก็บข้อมูลเกี่ยวข้องกับการผิดพลาดใน ส่วนของการเข้าถึงของข้อมูลของ OLEDB Provider
ออบเจกต์ Field	ใช้สำหรับเก็บฟิลด์ใดฟิลด์หนึ่งในฐานข้อมูล
ออบเจกต์ Parameter	ใช้สำหรับในการรับ-ส่ง ตัวแปร หรือ อากิวเมนต์ระหว่าง การทำคิวรีหรือ Stored Procedure ในออบเจกต์ Command
ออบเจกต์ Property	จะใช้สำหรับในการทำการกำหนดค่าของคุณสมบัติต่างๆ ของออบเจกต์ Record set
ออบเจกต์ Record	จะใช้สำหรับในการเก็บค่าของเรคคอร์ดใดเรคคอร์ดหนึ่ง ที่อยู่ในออบเจกต์ Record set
ออบเจกต์ Record set	จะเป็นผลลัพธ์ที่ได้จากฐานข้อมูลโดยจะประกอบด้วย กลุ่มของเรคคอร์ด และฟิลด์ที่ได้มาจากตารางที่ เชื่อมต่อ เป็นออบเจกต์ที่มีความสำคัญมากที่สุด

2.6.7 วิธีติดต่อกับฐานข้อมูลของ Visual Basics 10

โปรแกรม Visual Basics 10 มีความสามารถในการจัดการกับฐานข้อมูลได้โดยจะมี

วิธีการจัดการอยู่ 2 แบบ คือ

1. การใช้ Data Control เป็นวิธีการที่ง่ายและมีความสะดวกที่สุดในการที่จะทำ การติดต่อกับฐานข้อมูลเนื่องจาก Data Control จะทำการติดต่อกับฐานข้อมูลและจัดการกับ

ฐานข้อมูลในตารางโดยอัตโนมัติเช่น การเปิดฐานข้อมูล การแสดงและแก้ไขข้อมูลในตาราง อย่างไรก็ตาม การใช้ Data Control ยังมีข้อจำกัดอยู่พอสมควร เช่น ไม่มีฟังก์ชันในการลบข้อมูล

2. การใช้ Data Object วิธีนี้จะต้องเขียนโปรแกรมเพื่อทำการติดต่อกับฐานข้อมูล โดยการใช้ Object ต่างๆ ที่ Visual Basics 10 มีมาให้โดยการเพิ่ม ลบ หรือการแก้ไขข้อมูลจะต้องทำการเขียนโปรแกรมเอง แต่ข้อดีของวิธีนี้คือสามารถติดต่อกับข้อมูลจากหลายๆ ตารางพร้อมกัน สามารถสร้างคิวรีตอนรันโปรแกรมได้ และสามารถควบคุมความผิดพลาดต่างๆ ได้ดีกว่าใช้ Data Control รวมทั้งสามารถใช้ภาษา SQL เพื่อจัดการกับฐานข้อมูลได้อีกด้วย

2.6.7.1 การใช้งานดาต้าคอนโทรล (Data Control)

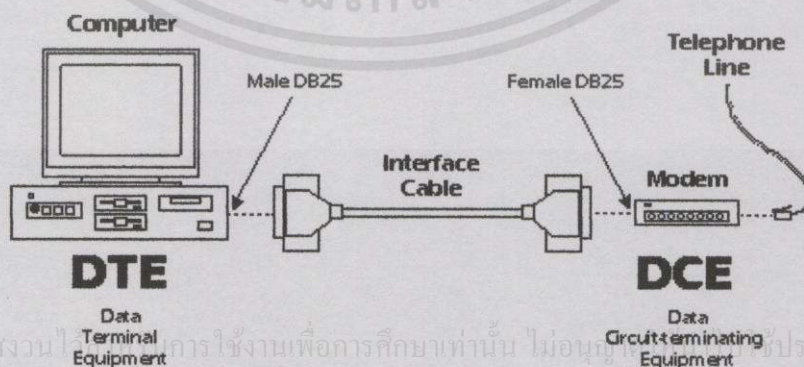
ดาต้าคอนโทรลเป็นเครื่องมือที่ช่วยให้สามารถจัดการกับข้อมูลในตาราง เช่นการเพิ่ม ลบ แก้ไข หรือการค้นหาข้อมูลที่ถูกต้องเป็นต้น

คุณสมบัติที่สำคัญสำหรับดาต้าคอนโทรล

Data Source	เป็นการกำหนดชื่อของดาต้าคอนโทรลที่ต้องการเชื่อมโยง
Delafield	เป็นการกำหนดว่าเราต้องการให้แสดงหรือแก้ไขข้อมูลใดในตาราง
Database Name	เป็นที่อยู่ของไฟล์ฐานข้อมูลที่ต้องการ โดยมีการระบุชื่อไฟล์คั่น
Connect	เป็นคุณสมบัติที่ใช้บอกว่าต้องการติดต่อกับฐานข้อมูลประเภทใดเช่น MS-Access
RecordSetType	เป็นชนิดของเร็คคอร์ดเซตที่ต้องการเข้าถึงฐานข้อมูล

2.7 การติดต่อสื่อสารและส่งผ่านข้อมูล

การติดต่อสื่อสารแบบ Host Link เป็นการเชื่อมต่อกับตัวควบคุมแบบ PLC กับคอมพิวเตอร์ทั่วไปผ่านทางพอร์ทของคอมพิวเตอร์ (COM 1, 2, 3, 4, 5 ฯลฯ) ส่วนมากจะนิยมใช้มาตรฐานในการส่งแบบอนุกรมเพื่อทำให้สามารถควบคุม PLC จากคอมพิวเตอร์ได้



รูปที่ 2.2 การติดต่อสื่อสารและส่งผ่านข้อมูล

สำหรับคอมพิวเตอร์หนึ่งเครื่องสามารถที่จะต่อเข้ากับตัวควบคุมแบบ PLC ได้เป็นจำนวนมากโดยใช้การเชื่อมต่อหลายๆ ตัวเข้าด้วยกัน เรียกว่า PC Link โดยจะใช้ในการติดต่อแบบ Host Link จะต้องผ่านอุปกรณ์ที่เรียกว่า Host Link Units ซึ่งจะต้องทำการตั้งค่าต่างๆ ที่จำเป็นที่จะทำการใช้ในการติดต่อแบบ Host Link Units ผ่านสวิตช์ของเครื่องควบคุมแบบ PLC

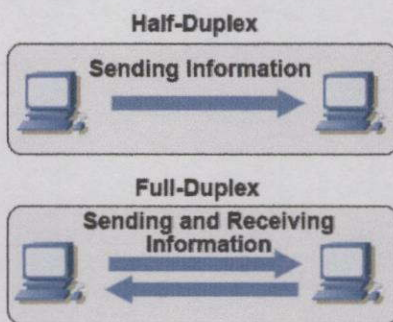
2.7.1 การสื่อสารข้อมูลแบบอนุกรม

ถ้าทำการกล่าวถึงเรื่องของการสื่อสารของข้อมูลแบบอนุกรมแล้วนั้น แสดงว่า จะต้องมีการสื่อสารข้อมูลแบบขนานด้วย ซึ่งการสื่อสารแบบขนานก็คือข้อมูลในทุกๆ บิตในแต่ละ เวิร์ดจะถูกส่งออกไปพร้อมๆ กัน ขึ้นอยู่กับว่าเวิร์ดดังกล่าวมีขนาดเท่าไร ทัวไปก็คือ 1 ไบต์ หรือ 8 บิตนั่นเอง การส่งข้อมูลแบบขนานนี้จะมีข้อกำหนดทางด้านระยะทาง ซึ่งโดยทั่วไปจะส่งในระยะไม่เกิน 3 - 5 ฟุต เท่านั้น ทั้งนี้ยังขึ้นอยู่กับอัตราความเร็วที่ใช้ในการส่งข้อมูลด้วย ยิ่งอัตราในการส่ง สูงก็จะได้ระยะทางที่มีความสั้นลง การส่งข้อมูลแบบขนานนั้นนิยมในระบบที่ต้องการความเร็วสูง มากๆ แต่อุปกรณ์ไม่ยุ่งยากมากนัก ส่วนการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะถูกทยอยส่งออกไป ทีละบิตจนหมดครบทั้งเวิร์ด โดยส่งผ่านสัญญาณเพียงเส้นเดียวและในการใช้งานจริงนั้นจะต้องมี สายสัญญาณอีกเส้นเป็นสายสัญญาณกราวด์ (Ground) ดังนั้น เมื่อเราส่งข้อมูลในแบบอนุกรมเราจะ สามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้น ในขณะที่ส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อย เท่ากับจำนวนบิตบวกกับสายสัญญาณระดับแรงดัน Ground อีก 1 เส้น ที่สำคัญการส่งข้อมูลแบบ อนุกรมนั้นจะสามารถส่งได้ไกลกว่า เช่น ถ้าส่งตามมาตรฐานของ RS-232 ที่จะกล่าวต่อไปใน ภายหลังจะสามารถส่งได้ไกลถึง 30 ถึง 40 ฟุต โดยไม่ต้องใช้อุปกรณ์ขับสัญญาณเพิ่มเติมแต่เดิม ใดๆ อย่างไรก็ตามในการส่งข้อมูลยังมีข้อกำหนดบางประการเพื่อให้ได้รับข้อมูลที่ถูกต้อง แม่นยำ และมีความน่าเชื่อถือสูง จะต้องมีวิธีตรวจสอบความถูกต้องของข้อมูลได้ในระหว่างที่มีการสื่อสารกัน อยู่ด้วย ซึ่งจะได้กล่าวต่อไป

2.7.1.1 การส่งข้อมูลแบบซิมเพล็กซ์ (Simplex) และ ดูเพล็กซ์ (Duplex)

ในการสื่อสารไม่ว่าจะเป็นการสื่อสารข้อมูลหรือเป็นการสื่อสารทั่วไปนั้น ย่อม จะต้องประกอบด้วยผู้รับและผู้ส่งแน่นอนตายตัวอยู่ตลอดเวลา เช่น การสื่อสารข้อมูลระหว่างเครื่อง คอมพิวเตอร์กับเครื่องพิมพ์ เป็นต้น การสื่อสารของอุปกรณ์ที่มีผู้รับและผู้ส่งตายตัวนั้น เราเรียกว่า การสื่อสารแบบซิมเพล็กซ์ กล่าวคือ การสื่อสารเป็นไปในลักษณะทิศทางเดียวตลอดเวลาซึ่งจะมีที่ใช้ ไม่มากนัก การสื่อสารโดยทั่วไปนั้นจะเป็นแบบดูเพล็กซ์ คือมีทิศทางการสื่อสารเป็นแบบสองทิศทาง มีทั้งไปและกลับ โดยการสื่อสารในลักษณะดูเพล็กซ์นั้นแบ่งออกได้เป็น 2 ชนิด คือ แบบฮาร์ฟดูเพล็กซ์ (Half Duplex) นิยมเขียนย่อกันว่า HEX ซึ่งจะมีทิศทางในการสื่อสารในลักษณะที่ผลัดกันเป็น ผู้ส่งและผู้รับพร้อมกัน และแบบฟูลดูเพล็กซ์ (Full Duplex) นิยมเขียนย่อว่า FDX จะมีทิศทางการ สื่อสารในลักษณะสัญญาณรับสัญญาณหนึ่ง และสัญญาณส่งอีกทิศทางหนึ่งหรือกล่าวได้อีกนัยหนึ่งว่า สัญญาณรับและส่งจะมีสายตัวนำสัญญาณแยกออกจากกันโดยเด็ดขาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

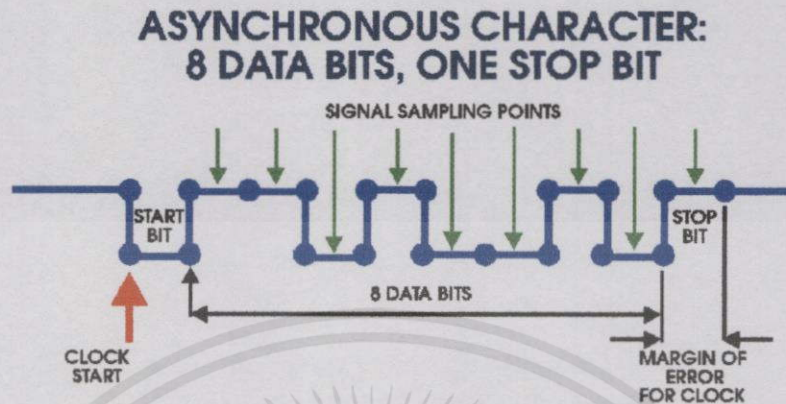


ภาพที่ 2.3 การส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์ (Half duplex) และ แบบฟูลดูเพล็กซ์ (Full duplex)

2.7.1.2 โพรโตคอลของการสื่อสารแบบอนุกรม

เมื่อพิจารณาการส่งข้อมูลแบบอนุกรมให้ดีจะพบว่ามีปัญหาหนึ่งที่เกิดขึ้นอยู่เสมอ ก็คือ การตัดสินใจว่าข้อมูลที่ได้รับนั้นมีจุดเริ่มต้นที่ใด ดังนั้น จึงมีการกำหนดข้อตกลงในการสื่อสารขึ้นเพื่อแก้ปัญหานี้ โดยข้อตกลงดังกล่าวจะเรียกว่า โพรโตคอล (Protocol) ของการสื่อสารข้อมูลแบบอนุกรม โดยสามารถที่จะแบ่งออกเป็น 2 ประเภทใหญ่ๆ ด้วยกันคือ โพรโตคอลสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous) โดยการสื่อสารข้อมูลแบบซิงโครนัสนี้ ข้อมูลจะถูกทำการส่งออกมาอย่างสม่ำเสมอและช่วงเวลาระหว่างบิตและเวิร์ด จะมีค่าที่เท่ากันเสมอ ส่วนการสื่อสารข้อมูลแบบ อะซิงโครนัสจะเป็นหัวใจของการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ ในปัจจุบันซึ่งการสื่อสารแบบนี้ช่วงเวลาระหว่างบิตจะมีค่าที่เท่ากันกับแบบซิงโครนัส แต่จะมีระยะห่างระหว่างเวิร์ดนั้นแตกต่างกันไปเป็นวินาที นาที ชั่วโมง หรือ วัน เป็นต้น ขึ้นอยู่กับฝ่ายรับสามารถรอได้หรือไม่เท่านั้น เมื่อไม่มีข้อกำหนดทางด้านระยะเวลาระหว่างเวิร์ดแล้วทางผู้ส่งและผู้รับจะเข้าใจกันตรงกันได้อย่างไรที่ใดคือจุดที่เริ่มต้นและจุดที่สิ้นสุดของแต่ละเวิร์ด เพื่อแก้ไขปัญหานี้จึงมีการกำหนดข้อตกลงเกี่ยวกับรูปแบบของข้อมูลที่ใช้ในการทำการส่งให้ทางผู้รับสามารถเข้าใจว่าจุดใดที่เป็นจุดเริ่มต้นของเวิร์ด ข้อกำหนดดังกล่าวกำหนดให้แต่ละเวิร์ดจะต้องขึ้นต้นด้วยบิตที่ เรียกว่า บิตเริ่มต้น (Start Bit) ซึ่งจะต้องมีข้อมูลเป็นลอจิก 0 เสมอ จากนั้นตามด้วยบิตข้อมูลที่ต้องการส่งมีความยาว 5 ถึง 8 บิต ถัดจากบิตข้อมูลก็จะเป็นบิตพาริตีบิต ซึ่งทำหน้าที่เป็นบิตสำหรับตรวจสอบความถูกต้องของข้อมูลที่ได้รับว่ามีความถูกต้องหรือไม่ บิตพาริตีนี้มี 2 ประเภท คือ อีเวนพาริตี (Even Parity) ซึ่งจะกำหนดจำนวนบิตที่เป็นลอจิก 1 ในบิตที่เป็นข้อมูลมีจำนวนเป็นคี่ ในการส่งข้อมูลบางครั้งอาจจะไม่มีการใช้บิตพาริตีก็ได้ ถ้าหากการสื่อสารในครั้งนั้นมีความน่าเชื่อถือสูง มีสัญญาณรบกวนต่ำเป็นการเพิ่มความเร็วในการสื่อสารได้ด้วยบิตสุดท้ายในรูปแบบก็คือ บิตสุดท้าย (Stop Bit) ทำหน้าที่บอกทางผู้รับว่าในขณะที่ข้อมูลทางผู้รับได้รับนั้นครบเวิร์ดแล้วขอให้เตรียมชุดรับเวิร์ดต่อไปได้ บิตสุดท้ายนี้ถูกกำหนดให้เป็นลอจิก 1 บิต หรือ 2 บิต ก็ได้ จากรูปแบบดังกล่าวจะเห็นว่าเรามีรูปแบบสำหรับการสื่อสารมากมาย เช่น 5E1 (5Databit,Even Parity,1 Stop Bit),7E1(7Databit,Even Parity,1 Stop Bit) และ 8E1 (8Databit,Even Parity,1 Stop Bit) เป็นต้น ในการใช้งานทั่วไปเรานิยมใช้กันอยู่เพียง 2 รูปแบบคือ 7E1 และ 8E1 จะเลือกรูปแบบใดขึ้นอยู่กับสภาพของสายส่งสัญญาณว่ามีสัญญาณรบกวนมากเพียงใด ถ้าหากสายส่งมีสัญญาณรบกวนมากก็ควรจะใช้ 7E1 แต่ถ้าสายส่งสัญญาณมีสภาพดีสัญญาณรบกวนต่ำการใช้ 8E1

จะเร็วกว่า เป็นต้น ทั้งนี้ต้องมีการตกลงกันล่วงหน้าระหว่างทางผู้รับและผู้ส่งว่าจะใช้รูปแบบใดในการสื่อสาร ลักษณะของข้อมูลที่ถูกส่งออกไปจะมีลักษณะดังรูป



ภาพที่ 2.4 แบบมาตรฐานการสื่อสารข้อมูลแบบอะซิงโครนัส

2.7.1.3 มาตรฐานสัญญาณอนุกรมแบบ RS232C

มาตรฐานการสื่อสารข้อมูลแบบอนุกรมกำหนดโดย EIA (Electronics Industries Association) มาตรฐาน RS232C ได้ถูกตีพิมพ์ในปี ค.ศ. 1969 เริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่างคอมพิวเตอร์กับโมเด็มในสมัยนั้น

ตัวอักษร RS แทน “Recommend Standard”

232 แทนหมายเลขของมาตรฐาน

ตัวอักษร C แทนให้รู้ว่ามาตรฐานได้รับการแก้ไขกี่ครั้ง



ภาพที่ 2.5 ลักษณะสมบัติทางไฟฟ้าของ RS-232

ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232

- ถูกออกแบบให้ใช้กับอุปกรณ์พวกลักษณะ Discrete
- ใช้การอินเทอร์เฟสแบบ Unbalanced

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในโครงการนำร่องเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาตจากทางผู้จัดทำ

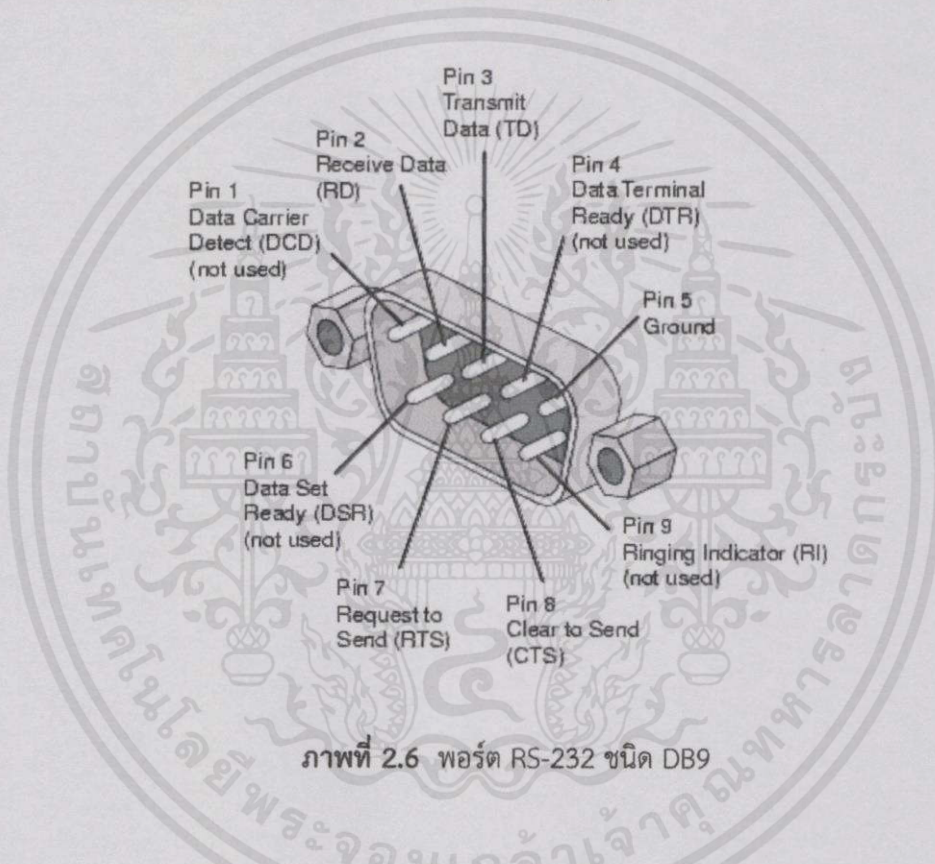
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้วงจรอื่นใดเชื่อมต่อกับสายสัญญาณนำร่อง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อัตราเร็วในการส่งข้อมูลมีค่า < 20 กิโลบิตต่อวินาที (Kbps)

- ระยะทางสูงสุดในการส่งข้อมูลมีค่า < 15 เมตร
- ทำให้เกิด Crosstalk ที่มีค่ามาก

2.7.1.4 พอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรม (Serial Communication Device) หรือเรียกว่า “Serial Port” เครื่องคอมพิวเตอร์โดยปกติจะมีพอร์ตชนิดนี้อยู่แล้ว 2 พอร์ต คือ พอร์ตขนาด 9 ขา (9-pins) มีรูปร่างเหมือนสี่เหลี่ยมคางหมู มีเข็มนูนออกมา 9 เข็ม เรียกหัวชนิดนี้ว่า “DB-9 Connector Male Type” อีกชนิดหนึ่งคือ พอร์ตขนาด 25 ขา (25-pins) มีรูปร่างเช่นเดียวกับกับแบบ 9 ขา แต่มีขนาดที่ยาวกว่า เรียกหัวชนิดนี้ว่า “DB-25 Connector Male Type”



ภาพที่ 2.6 พอร์ต RS-232 ชนิด DB9

V.24/RS-232 Interface

SOURCE	SIGNAL DESIGNATION	PIN NO.	PIN NO.	SIGNAL DESIGNATION	SOURCE
DTE	Secondary Transmitted Data	14	1	Shield	Common
DCE	Transmitter Signal Element Timing	15	2	Transmitted Data (TD)	DTE
DCE	Secondary Received Data	16	3	Received Data (RD)	DCE
DCE	Receiver Signal Element Timing	17	4	Request to Send (RTS)	DTE
DTE	Local Loopback (LL)	18	5	Clear to Send (CTS)	DCE
DTE	Secondary Request to Send	19	6	Data Set Ready (DSR)	DCE
DTE	Data Terminal Ready (DTR)	20	7	Signal Ground	Common
DTE	Remote Loopback (RL)	21	8	Received Line Signal Detector (DCD)	DCE
DCE	Ring Indicator (RI)	22	9	+ VOLTAGE	-
DTE/DCE	Data Signal Rate Selector	23	10	- VOLTAGE	-
DTE	Ext. Transmit Signal Element Timing	24	11	Unassigned	-
DCE	Test Mode	25	12	Secondary Received Line Signal Detector	DCE
			13	Secondary Clear to Send	DCE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในโครงการวิจัยของศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร สำนักงานปลัดกระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.7 พอร์ต RS-232 ชนิด DB25

ตารางที่ 2.7 ขาสัญญาณต่างๆในการส่งข้อมูลแบบอนุกรม RS-232 Standard

DB-9 pins	DB-25 pins	Function
1	8	Carrier Detect
2	3	Received Data
3	2	Transmitted Data
4	20	Data Terminal Ready
5	7	Signal Ground
6	6	Data Set Ready
7	4	Request to Send
8	5	Clear to Send
9	22	Ring Indicator

มาตรฐานของการรับส่งข้อมูลแบบอนุกรม (RS-232) นี้ได้กำหนดขึ้นเพื่อให้คอมพิวเตอร์ต่างยี่ห้อกัน หรืออุปกรณ์ต่อพ่วงแต่ละชนิดรับส่งข้อมูลกันได้เมื่อทำตามมาตรฐาน ไม่สนใจว่าอุปกรณ์หรือคอมพิวเตอร์นั้นจะผลิตมาจากที่ใด

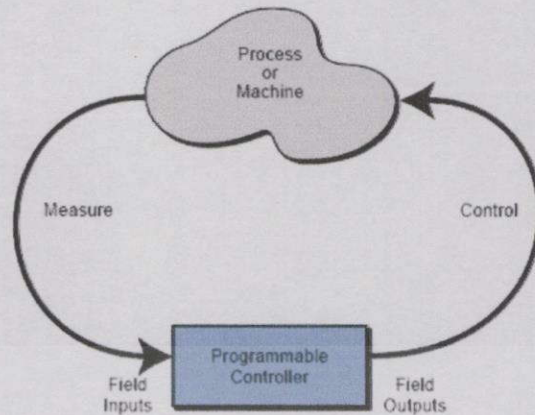
2.8 ตัวควบคุมเชิงตรรกะแบบโปรแกรมได้ (Programmable logic control :PLC)

2.8.1 นิยามของตัวควบคุมเชิงตรรกะแบบโปรแกรมได้

โปรแกรมเมเบิลลอจิกคอนโทรล มีอีกชื่อหนึ่งว่า โปรแกรมเมเบิลคอนโทรเลอร์ หรือ PLC ก็คืออุปกรณ์ประเภทโซลิตสเตท ซึ่งเป็นตระกูลหนึ่งของคอมพิวเตอร์ โดยการนำวงจรรวม (Integrated Circuit :IC) มาประยุกต์ใช้เป็นอุปกรณ์ควบคุมแทนที่อุปกรณ์จำพวกรีเลย์ หรือพวกแมกเนติกคอนแทรกเตอร์

PLC มีชุดคำสั่งต่างๆเช่น คำสั่งเกี่ยวกับระบบซีควেনซ์ คำสั่งการหน่วงเวลา คำสั่งการนับ คำสั่งทางคณิตศาสตร์ คำสั่งการจัดการข้อมูล รวมถึงคำสั่งที่ใช้ในระบบสื่อสารต่างๆ เพื่อใช้ในการควบคุมทางอุตสาหกรรมและกระบวนการผลิต

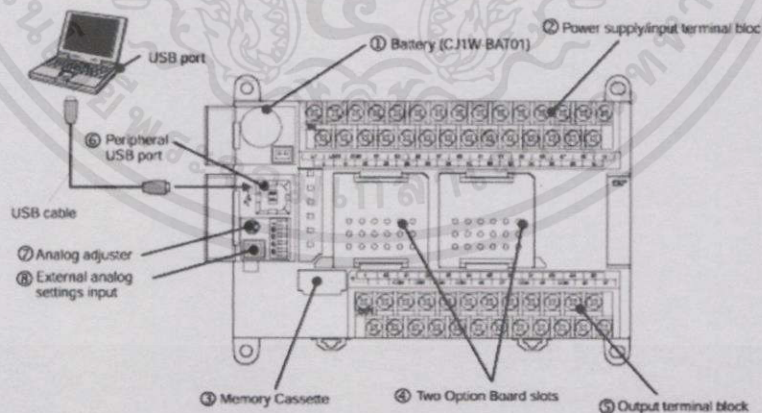
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.8 ไดอะแกรมแสดงการประยุกต์ใช้งานของระบบ PLC

ถึงแม้ว่าในปัจจุบัน PLC นั้นจะมีหลายนิยาม แต่จากคำนิยามของ “โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์” ตามมาตรฐานของ IEC 1131, PART1 “ระบบปฏิบัติการทางด้านดิจิทัล ออกแบบมาให้ใช้งานในอุตสาหกรรม ซึ่งใช้หน่วยความจำที่สามารถโปรแกรมได้ในการเก็บคำสั่งที่ผู้ใช้กำหนดขึ้น (User Program) เพื่อเป็นเครื่องมือในการกำหนดฟังก์ชันหรือเงื่อนไขในการทำงานเช่น การทำงานแบบลอจิก, การทำงานแบบซีควเอนซ์, การใช้งานไทม์เมอร์, การใช้งานเคาน์เตอร์ และ ฟังก์ชันทางคณิตศาสตร์ เพื่อควบคุมอุปกรณ์ดิจิทัลอินพุตและเอาต์พุต หรือแอนะล็อก อินพุต และเอาต์พุต ของเครื่องจักรหรือกระบวนการผลิตต่างๆ นอกจากนี้ ทั้งระบบ PLC และอุปกรณ์ภายนอกที่ใช้งาน จะต้องสามารถเชื่อมต่อหรือสื่อสารกับระบบควบคุมทางอุตสาหกรรม, เครื่องมือหรืออุปกรณ์ต่าง ๆ และใช้งานร่วมกันได้ง่าย”

2.8.2 ส่วนประกอบของตัวควบคุมเชิงตรรกะแบบโปรแกรมได้ รุ่น CP1L



รูปที่ 2.9 โครงสร้างภายนอกของตัวควบคุมเชิงตรรกะแบบโปรแกรมได้ รุ่น CP1L

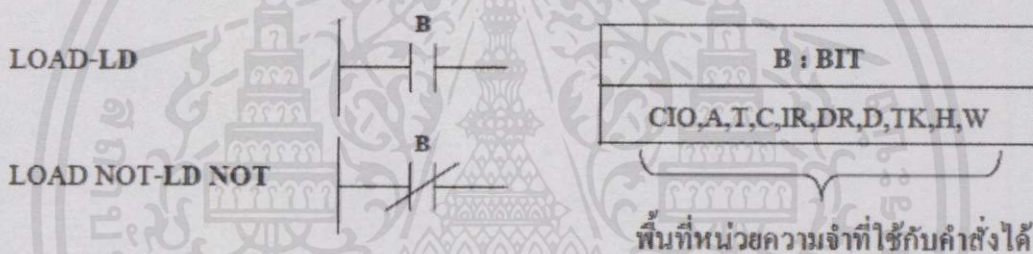
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.8 สามารถอธิบายความหมายของแต่ละส่วนได้ดังนี้

- 1 คือ แบตเตอรี่ (Battery)
- 2 คือ ขั้วต่อไฟและอินพุท (Power supply/Input terminal)
- 3 คือ ช่องเสียบหน่วยความจำ (Memory cassette)
- 4 คือช่องเสียบเพื่อเพิ่มพอร์ตติดต่อสื่อสาร (Option board slots)
- 5 คือ ขั้วต่อเอาต์พุต (Output terminal)
- 6 คือ พอร์ตเชื่อมต่อกับอุปกรณ์ป้อนโปรแกรม (USB Port)
- 7 คือ ปุ่มปรับอนาล็อก (Analog adjuster)
- 8 คือ ขั้วต่ออินพุตสำหรับอนาล็อก setting (External analog setting input)

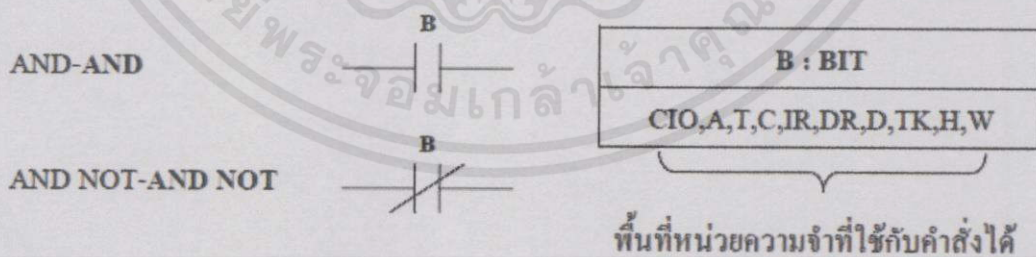
2.8.3 กลุ่มคำสั่งพื้นฐาน

2.8.3.1 การใช้คำสั่ง LOAD(LD), LOAD NOT(LD NOT)



รูปที่ 2.10 ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง LD และ LD NOT

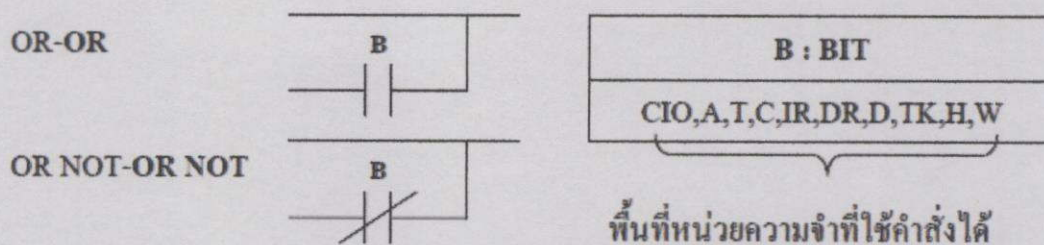
2.8.3.2 การใช้คำสั่ง AND , AND NOT



รูปที่ 2.11 ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง AND และ AND NOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.3.3 การใช้คำสั่ง OR , OR NOT



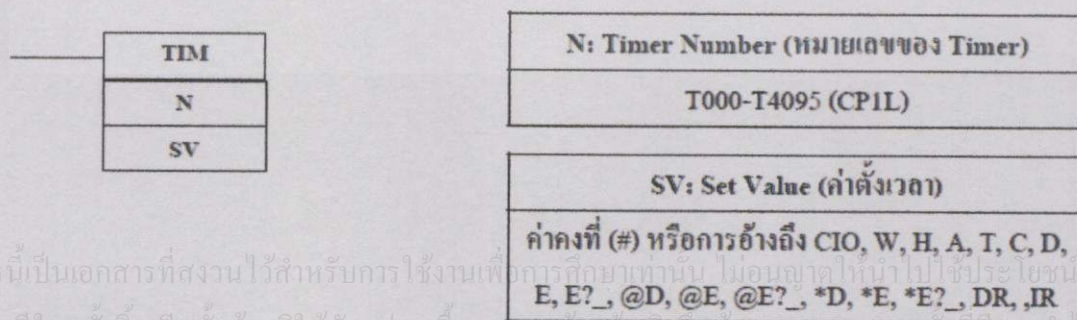
รูปที่ 2.12 ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง OR และ OR NOT

2.8.3.4 การ OUT , OUT NOT



รูปที่ 2.13 ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง OUT และ OUT NOT

2.8.2.5 การใช้คำสั่ง TIMER : TIM

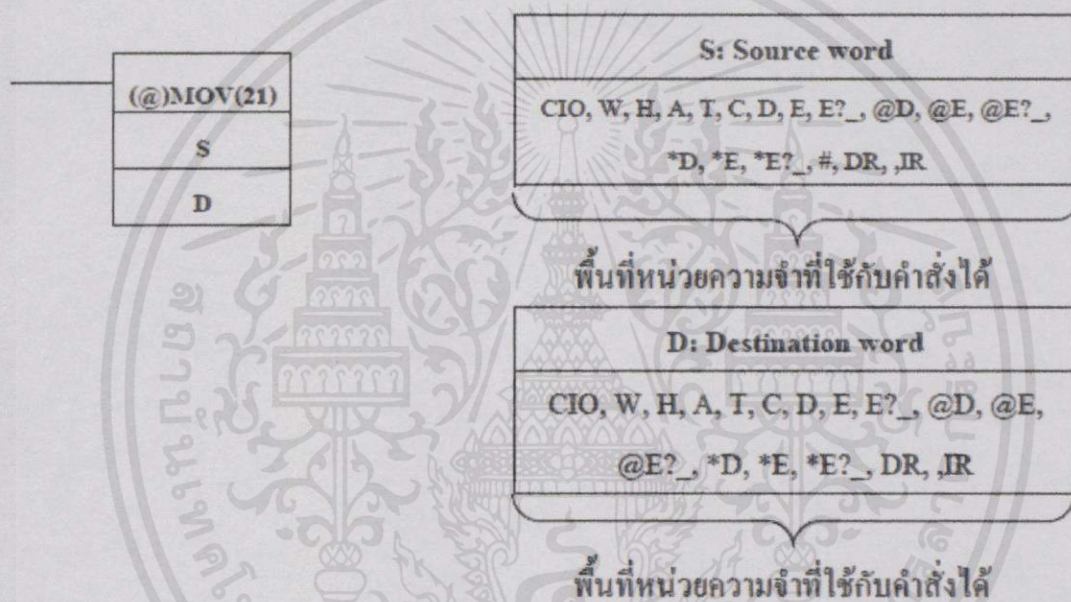


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.14 ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง TIMER

- N = Timer Number (เบอร์ 0000 - 4095) เลือกว่าจะใช้ Timer ตัวที่เท่าใด
- SV = Set Value ค่าตั้งเวลา ใช้กำหนดว่าจะให้ Timer ตั้งเวลานานเท่าใด ซึ่ง SV ที่ตั้งนั้น จะถูกคูณด้วย 0.1 เพื่อแปลงเป็นระยะเวลาจริง ซึ่งสามารถ
- กำหนด SV เป็นค่าคงที่ #0000-9999 (000.0-999.9 วินาที คูณด้วย 0.1 วินาที)
 - กำหนด SV เป็น แอดเดรส CIO,A,T,C,IR,DR,D,TK,H,W โดยใส่ค่าตั้งเวลาที่ เป็นค่าคงที่ 0000-9999 ไว้ใน แอดเดรส ที่อ้างถึงอีกทีหนึ่ง (ค่าที่กำหนดจะคูณด้วย 0.1 วินาทีเช่นเดียวกับการกำหนดแบบค่าคงที่)

2.8.2.6 การใช้คำสั่ง MOVE : MOV(21)



รูปที่ 2.15 ชุดคำสั่งและการเขียน Ladder Diagram คำสั่ง MOVE

2.8.2.5 การใช้คำสั่ง END

เมื่อสิ้นสุดการเขียนโปรแกรมแล้วจะต้องจบด้วยคำสั่ง END(01) เสมอ ถ้าไม่มีคำสั่งนี้ เมื่อผู้ใช้สั่ง RUN โปรแกรมที่เขียนขึ้น PLC จะเกิด ERROR โดยสังเกตที่ PLC ไฟ ERROR/ALARM สีแดงจะติดค้าง นั่นแสดงว่าไม่มีคำสั่ง END(01)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 การเชื่อมต่อแบบ Host Link

เป็นคำสั่งที่ใช้ในการควบคุมต่างๆเช่นอ่านและเขียนหน่วยความจำอินพุทหรือเอาต์พุต และใช้ในการเปลี่ยนโหมดของเครื่องควบคุม รวมทั้งยังสามารถใช้ในการล๊อคบิตข้อมูลโดยการส่งคำสั่งโฮสลิงค์ระหว่างคอมพิวเตอร์กับเครื่องควบคุม

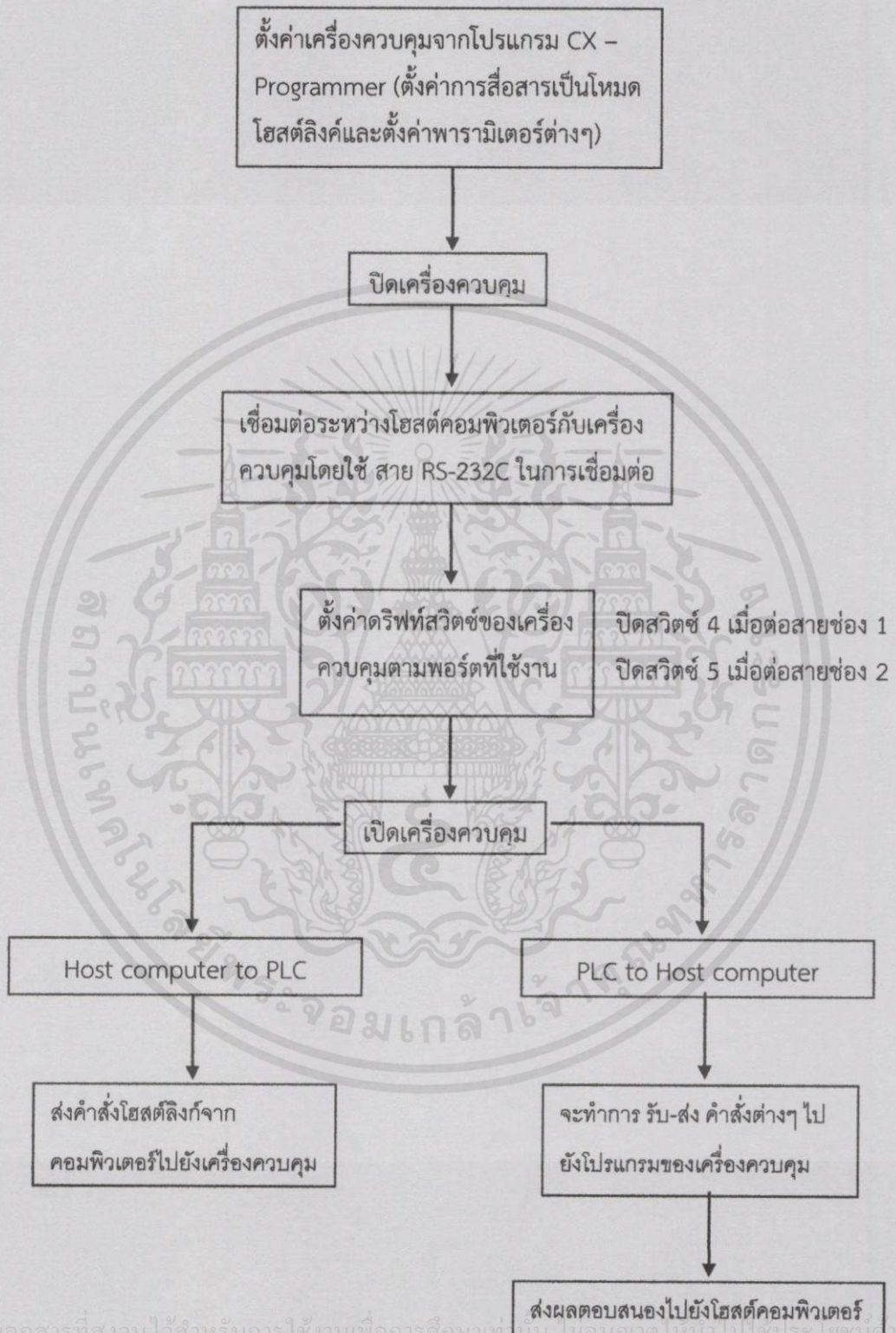
ใช้การเชื่อมโยงแบบโฮสลิงค์ในการตรวจสอบข้อมูลต่างๆ เช่นสถานการณ์ดำเนินงานของข้อมูลที่เกิดพลาด และข้อมูลคุณภาพของเครื่องควบคุม หรือใช้ในการส่งข้อมูล

ตารางที่ 2.8 ฟังก์ชันการสื่อสารที่เชื่อมโยงพื้นที่ที่มีอยู่ในเครื่องควบคุม

การเชื่อมต่อ	ชนิดของคำสั่ง	วิธีการสื่อสาร	องค์ประกอบ
โฮสต์ คอมพิวเตอร์ 	คำสั่งโฮสลิงค์ (C Mode) <div style="border: 1px solid black; padding: 5px; display: inline-block;">Host link command</div>	สร้างกรอบคำสั่ง ที่ใช้ในการสื่อสาร ส่งไปยังเครื่อง ควบคุมซึ่งเครื่อง ควบคุมก็จะมี ผลตอบสนอง มายังโฮสต์ คอมพิวเตอร์	โดยการเชื่อมต่อคอมพิวเตอร์ที่เป็นโฮสต์ แบบ 1:1 หรือ 1:N  OR 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 กระบวนการในการรับ - ส่งคำสั่ง Host Link



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

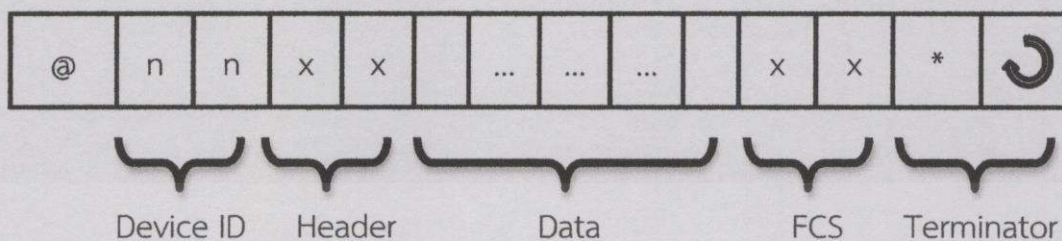
รูปที่ 2.16 กระบวนการรับ - ส่ง คำสั่ง Host Link

ตารางที่ 2.9 คำสั่ง Host Link ในส่วนของ Header

ชนิด	โค้ดส่วนหัว	ชื่อ	ฟังก์ชัน
คำสั่งของการอ่านหน่วยความจำของอินพุตและ เอาต์พุต	RR	CIO AREA READ	อ่านค่าของข้อมูลที่ระบุจำนวนไว้ของพื้นที่ CIO โดยเริ่มจากค่าเริ่มต้นไปยังค่าที่ระบุไว้
	RL	LINK AREA READ	อ่านค่าของข้อมูลที่ระบุจำนวนไว้ของพื้นที่ LINK โดยเริ่มจากค่าเริ่มต้นไปยังค่าที่ระบุไว้
	RH	HR AREA READ	อ่านค่าของข้อมูลที่ระบุจำนวนไว้ของพื้นที่ Holding โดยเริ่มจากค่าเริ่มต้นไปยังค่าที่ระบุไว้
	RC	PV AREA READ	อ่านค่าของข้อมูลที่ระบุจำนวนไว้ของตัวนับและตัวจับเวลา โดยเริ่มจากค่าเริ่มต้นไปยังค่าที่ระบุไว้
	RG	T/C STATUS READ	อ่านค่าสถานะของข้อมูลที่ระบุจำนวนไว้ของตัวนับและตัวจับเวลา โดยเริ่มจากค่าเริ่มต้นไปยังค่าที่ระบุไว้
	RD	DM AREA READ	อ่านค่าของข้อมูลที่ระบุจำนวนไว้ของพื้นที่ DM โดยเริ่มจากค่าเริ่มต้นไปยังค่าที่ระบุไว้
	RJ	AR AREA READ	อ่านค่าของข้อมูลที่ระบุจำนวนไว้ของพื้นที่ AR โดยเริ่มจากค่าเริ่มต้นไปยังค่าที่ระบุไว้
คำสั่งของการเขียนหน่วยความจำของอินพุตและ เอาต์พุต	WR	CIO AREA WRITE	เขียนค่าของข้อมูลไปยังพื้นที่ CIO โดยเริ่มจากค่าที่ระบุไว้
	WL	LINK AREA WRITE	เขียนค่าของข้อมูลไปยังพื้นที่ LINK โดยเริ่มจากค่าที่ระบุไว้
	WH	HR AREA WRITE	เขียนค่าของข้อมูลไปยังพื้นที่ HR โดยเริ่มจากค่าที่ระบุไว้
	WC	PV AREA WRITE	เขียนตัวแปรไปยังตัวนับและตัวจับเวลาของค่าที่ระบุไว้
	WD	DM AREA WRITE	เขียนค่าของข้อมูลไปยังพื้นที่ DM โดยเริ่มจากค่าที่ระบุไว้
	WJ	AR AREA WRITE	เขียนค่าของข้อมูลไปยังพื้นที่ AR โดยเริ่มจากค่าที่ระบุไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 รูปแบบในการส่งคำสั่ง Host Link



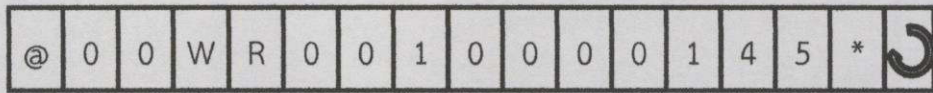
รูปที่ 2.17 รูปแบบของคำสั่ง Host Link

การรับ-ส่งข้อมูลนั้นจะมีรูปแบบการในการรับ-ส่งข้อมูลดังรูป โดยจะประกอบด้วยส่วนต่างๆ ซึ่งรูปแบบการส่งหรือการรับนั้นจะเริ่มด้วย “@” เสมอ ซึ่งรูปแบบต่างๆมีส่วนประกอบดังนี้

Device ID	คือ เลขของเครื่องควบคุมซึ่งประกอบด้วย 2 ไบต์ โดยมีเลขตั้งแต่ 00 – FF
Header	คือ คำสั่งที่ใช้ในการจัดการกับเครื่องควบคุม ประกอบด้วย 2 ไบต์ ตัวอย่างเช่น ในการอ่านค่าจาก เครื่องควบคุม คำสั่งที่ใช้ในไบต์แรกจะเป็น “R” ส่วนใน ไบต์ ถัดไปจะไปคำสั่งว่าต้องการอ่านค่าส่วนไหนของเครื่องควบคุม
Data	คือ ข้อมูลที่แสดงตำแหน่งของเครื่องควบคุม ซึ่งข้อมูลนี้จะประกอบด้วย 8 ไบต์ โดย 4 ไบต์แรกจะเป็นข้อมูลของเซแวลเครื่องควบคุมและอีก 4 ไบต์หลังจะเป็น ตำแหน่งของเซแวลนั้นๆ ตัวอย่างเช่น ต้องการสั่งให้ตำแหน่ง 10.01 ทำงาน ใน 4 ไบต์แรกจะเป็น “0010” และอีก 4 ไบต์หลังจะเป็น “0002” ซึ่งรูปแบบข้อมูล จะต้องเป็น HEX
FCS	คือ มีไว้เพื่อการตรวจสอบข้อผิดพลาดในการสื่อสารในบิตสตรีมแบบอนุกรม ซึ่งในการ คำนวณหาค่า FCS นั้นทำได้โดยนำค่าตั้งแต่ “@” จนถึงข้อมูลตัวสุดท้าย ซึ่งข้อมูลนั้น จะต้องเป็น 8 – bit ซึ่งหาได้จากตาราง ASCII จากนั้นทำการ XOR กันทั้งหมด โดยคำตอบ ที่ได้จะแปลงเป็นเลขฐาน 16 ก็จะได้ค่า FCS
Terminator	คือ ส่วนที่ปิดท้ายของคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.3 ตัวอย่างการคำนวณ หาค่า FCS



@ 0100 0000

XOR

0 0011 0000

XOR

0 0011 0000

XOR

W 0101 0111

XOR

R 0101 0010

XOR

0 0011 0000

XOR

0 0011 0000

XOR

1 0011 0001

XOR

0 0011 0000

XOR

0 0011 0000

XOR

0 0011 0000

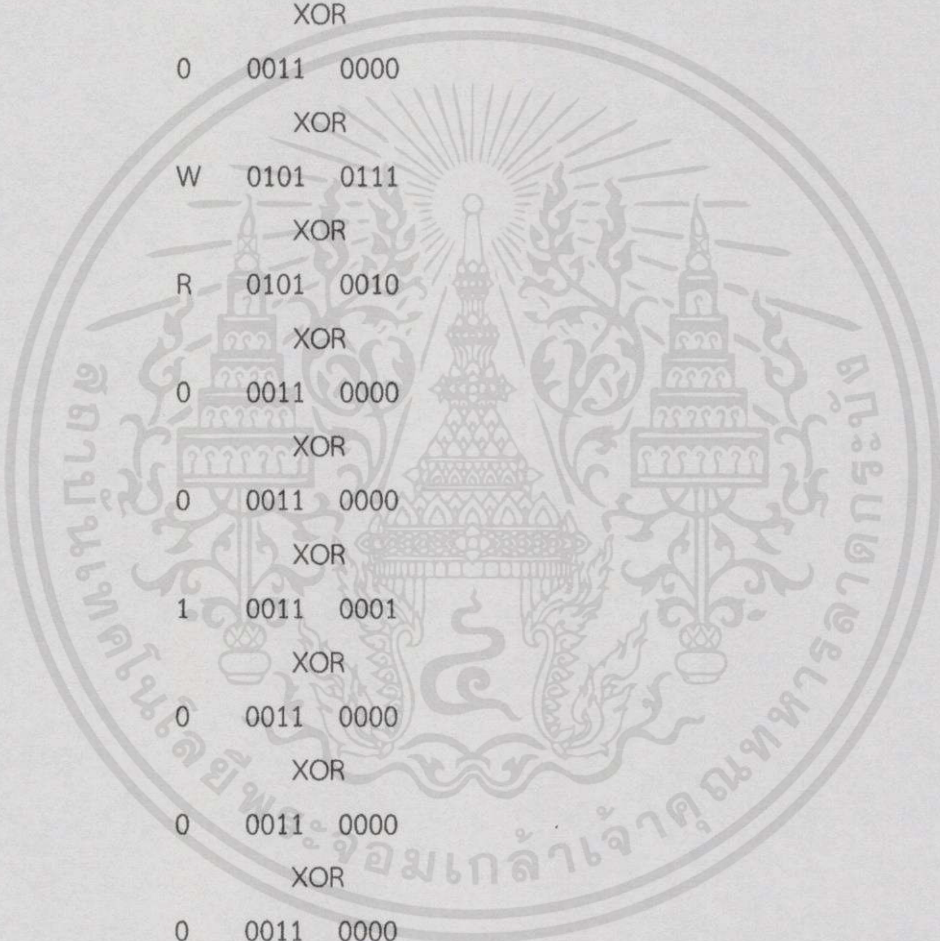
XOR

0 0011 0000

XOR

1 0011 0001

FCS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 0100 0101 = 45₁₆
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 =====

บทที่ 3

การออกแบบโปรแกรมควบคุม

3.1 กล่าวนำ

การออกแบบโปรแกรมควบคุมนั้นเป็นการออกแบบโปรแกรมเพื่อนำไปใช้งานจริง โดยโปรแกรมนั้นจะใช้ควบคุมสิ่งต่างๆซึ่งโปรแกรมควบคุมสามารถตอบสนองความต้องการของผู้ที่ใช้งานได้โดยตรง ในการออกแบบโปรแกรมควบคุมนั้นจะมีองค์ประกอบหลายอย่างที่ใช้ออกแบบ คือ การทำงานของอุปกรณ์ที่ใช้ควบคุมจะต้องสัมพันธ์กันกับโปรแกรมที่ใช้ควบคุม ซึ่งโปรแกรมที่จะใช้สร้างหน้าควบคุมจะใช้โปรแกรม Visual Basic 2010 ในการสร้างโดยสิ่งที่ต้องคำนึงหลักๆก็คือการเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ควบคุม หรือ PLC ซึ่งจะต้องศึกษาการเชื่อมต่อและนำมาประยุกต์ใช้กับการเขียนโปรแกรม

สำหรับในส่วนของโปรแกรมควบคุมนั้นจะเป็นโปรแกรมที่ใช้จัดการบริหารกระเช้าจอตลอด ซึ่งในการบริหารการจัดการกระเช้าจอตลอดนั้นได้แบ่งโปรแกรมออกเป็นสองส่วนหลักคือส่วนที่ควบคุมกระเช้าในส่วนนี้จะเป็นส่วนของพนักงานที่ใช้ควบคุมกระเช้าทั้งหมดและอีกส่วนคือส่วนจัดการข้อมูลทั้งหมดจะเป็นส่วนของผู้ที่สามารถดูข้อมูลในการจอตลอดได้ทั้งหมด

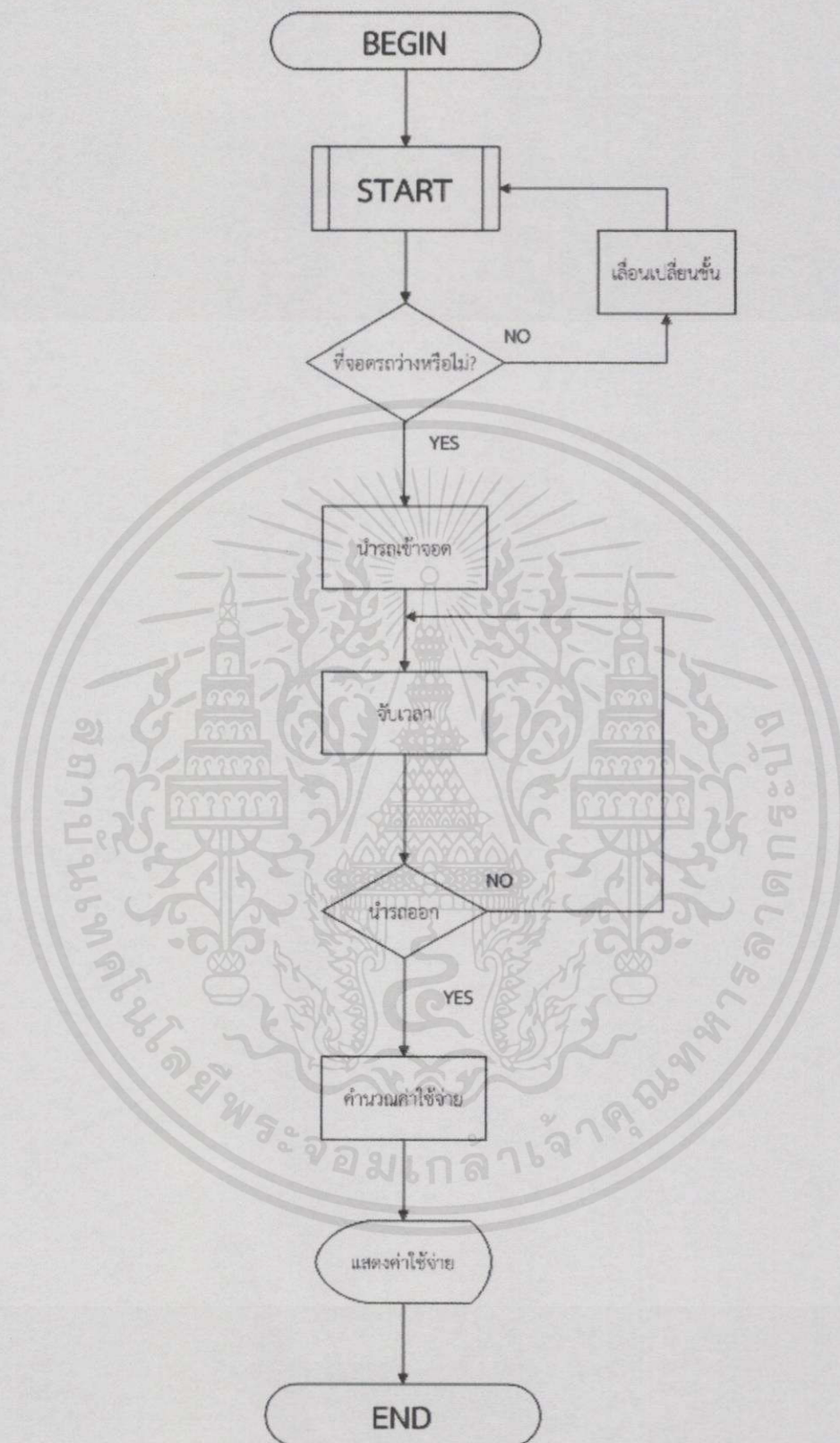
3.2 การออกแบบโปรแกรมควบคุม

การออกแบบโปรแกรมควบคุมนั้นอย่างแรกจะต้องกำหนดเงื่อนไขต่างๆของการทำงานทั้งหมดและทำการเขียนแลตเตอร์โคดแกรมให้กับ PLC ที่ควบคุมต่อมาทำการออกแบบหน้าควบคุมโดยใช้โปรแกรม Visual Basic 2010 ในการออกแบบหน้าควบคุมทั้งหมด

3.2.1 โปรแกรมการทำงานของระบบ

ส่วนของตัวโปรแกรมที่ใช้ในการดำเนินการระบบของกระเช้าจอตลอดนั้น ตัวโปรแกรมการตรวจสอบว่าชั้นที่อยู่ปัจจุบันว่ามีสถานะเป็นเช่นไร ถ้าชั้นปัจจุบันนั้นว่างอยู่ก็จะทำการนำรถเข้าจอด แต่ถ้าชั้นปัจจุบันไม่ว่าง ระบบจะทำการเลื่อนไปชั้นถัดไปและทำวนจนกว่าจะไปถึงชั้นที่มีสถานะว่าง โดยเมื่อนำรถเข้าจอดแล้วระบบจะเริ่มทำการจับเวลาเพื่อนำไปใช้ในการคำนวณค่าใช้จ่าย โดยจะหยุดนับเวลาที่ต่อเมื่อมีการนำรถออก ระบบจะทำการนำเวลาที่ค้างอยู่ในระบบมาคำนวณค่าใช้จ่ายแล้วแสดงผลออกมา

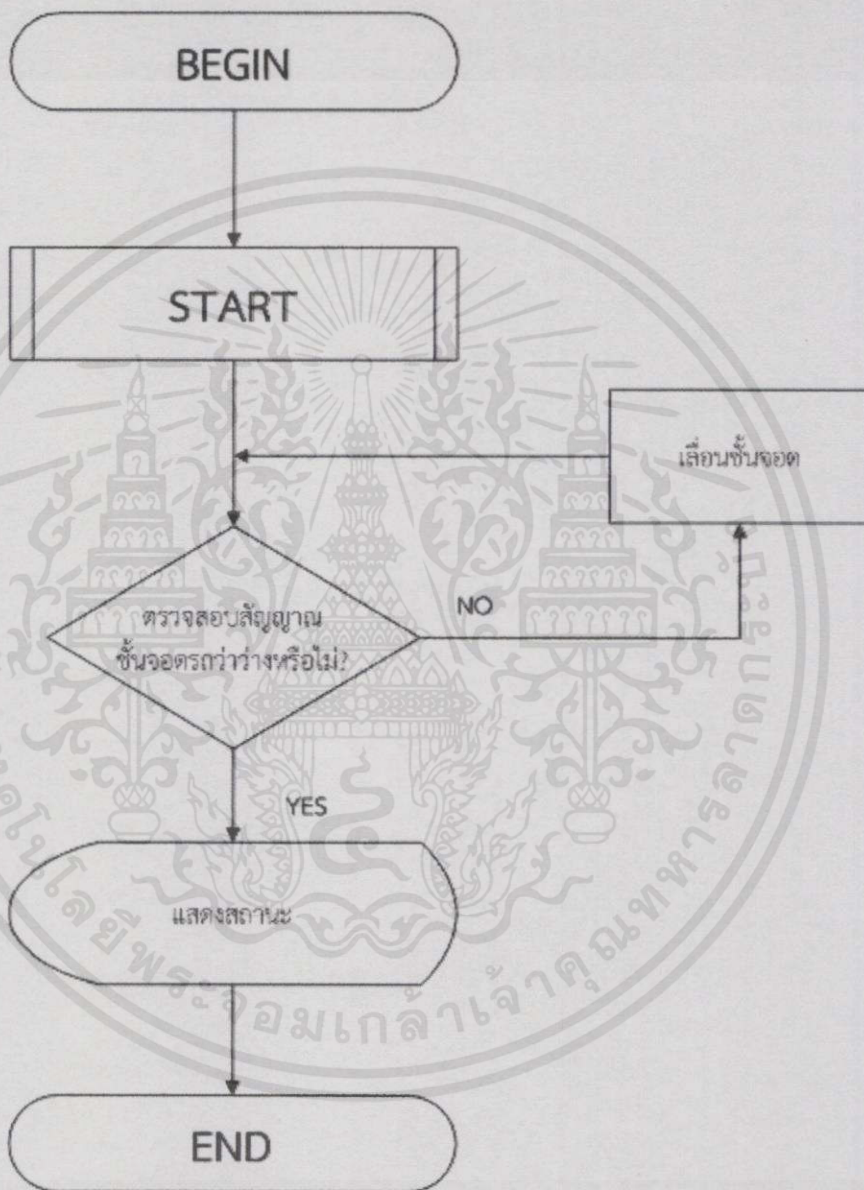
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ภาพที่ 3.1 Flow chart แสดงขั้นตอนการทำงานของระบบกระเช้าจอดรถ
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โปรแกรมตรวจสอบสถานะชั้นจอตรด (ว่าง/ไม่ว่าง)

ส่วนของโปรแกรมนี้เป็นโปรแกรมแสดงผลโดยจะเป็นส่วนการรับค่า จากตัวโปรแกรมว่ามีสถานะเป็นอย่างไร โดยตรวจสอบตำแหน่งของชั้นจากเซ็นเซอร์และเคาเตอร์ จากตัวโมเดล

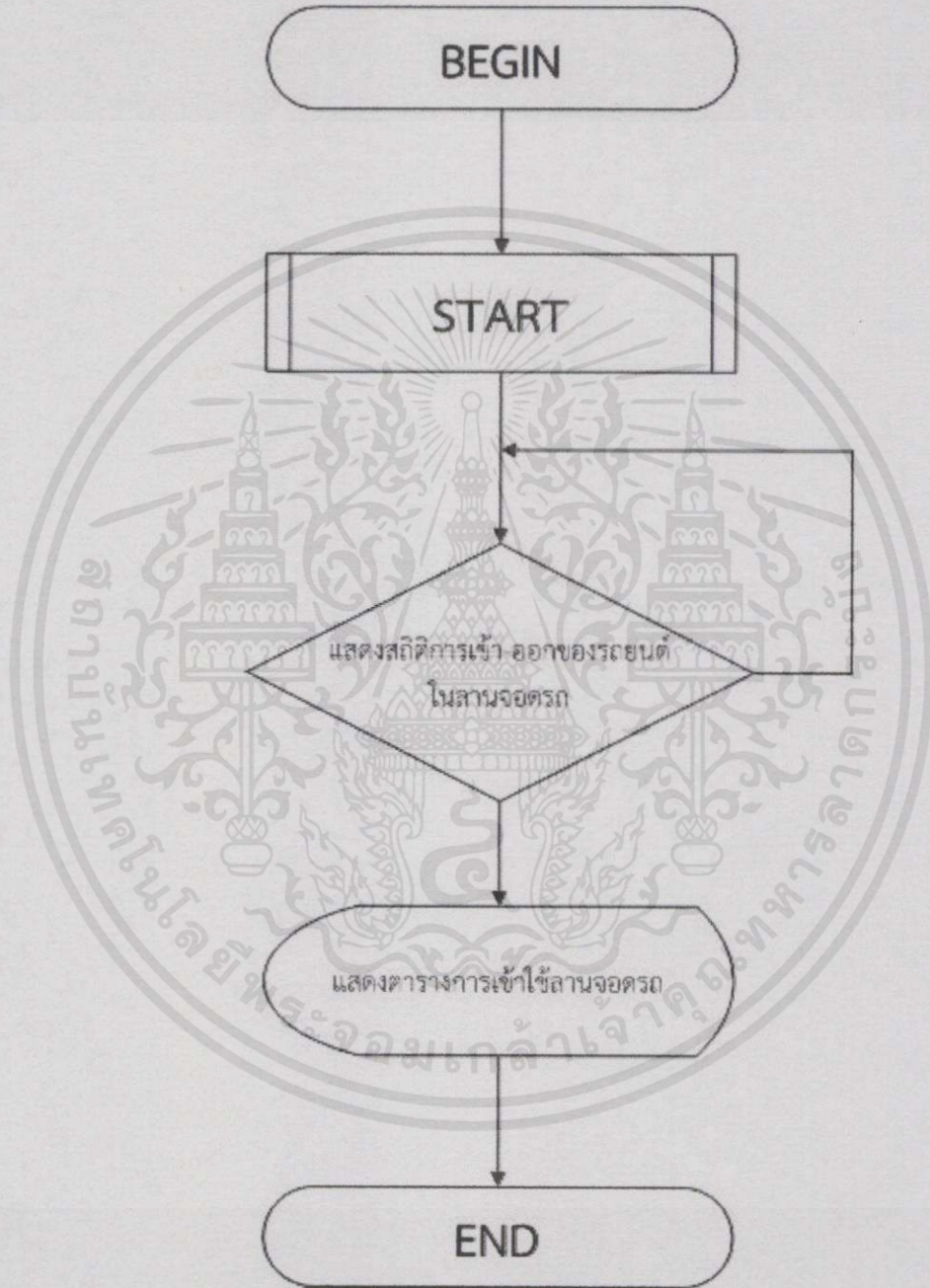


รูปที่ 3.2 Flow chart แสดงการตรวจสอบสถานะชั้นจอตรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 โปรแกรมแสดงสถิติการเข้าจอดในกระเช้าจอตrolleyยนต์

ส่วนของโปรแกรมแสดงสถิติการเข้าจอดในกระเช้าจอตrolleyยนต์นั้นเป็นโปรแกรมที่จะนำเข้าข้อมูลจากฐานข้อมูลใน MS Access ทั้งหมดมาแสดง โดยจะระบุรายละเอียดต่างๆ ไม่ว่าจะเป็น ทะเบียนรถ เวลาเข้า-ออก เวลารวม ค่าใช้จ่าย ชั้นที่จอด



ภาพที่ 3.3 Flow chart โปรแกรมแสดงสถิติการเข้าจอดในกระเช้าจอตrolleyยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบแลคเตอร์ไคอะแกรม

3.3.1 กำหนดเงื่อนไขของการทำงาน

3.3.1.1 กำหนดอินพุต และ เอาท์พุต ของ PLC

ในการกำหนดนั้นจะต้องกำหนดตามเงื่อนไขที่กำหนดไว้ โดยจะเป็นการกำหนดแอดเดรสให้กับอุปกรณ์ที่ใช้งาน ซึ่งจะใช้ในการเขียน Ladder

I/O address	Instruction	Remark
0.00	Proximity Sensor (Right)	Input
0.10	Proximity Sensor (Left)	Input
10.01	1 floor (Internal relay)	Input
10.02	2 floor (Internal relay)	Input
10.03	3 floor (Internal relay)	Input
10.04	4 floor (Internal relay)	Input
10.05	5 floor (Internal relay)	Input
10.06	6 floor (Internal relay)	Input
10.07	7 floor (Internal relay)	Input
100.02	Relay (Motor)	Output
100.03	Relay (Buzzer)	Output
100.04	7 - Segment	Output
100.05	7 - Segment	Output
100.06	7 - Segment	Output

ตารางที่ 3.1 ตารางแสดงอินพุตและเอาต์พุต

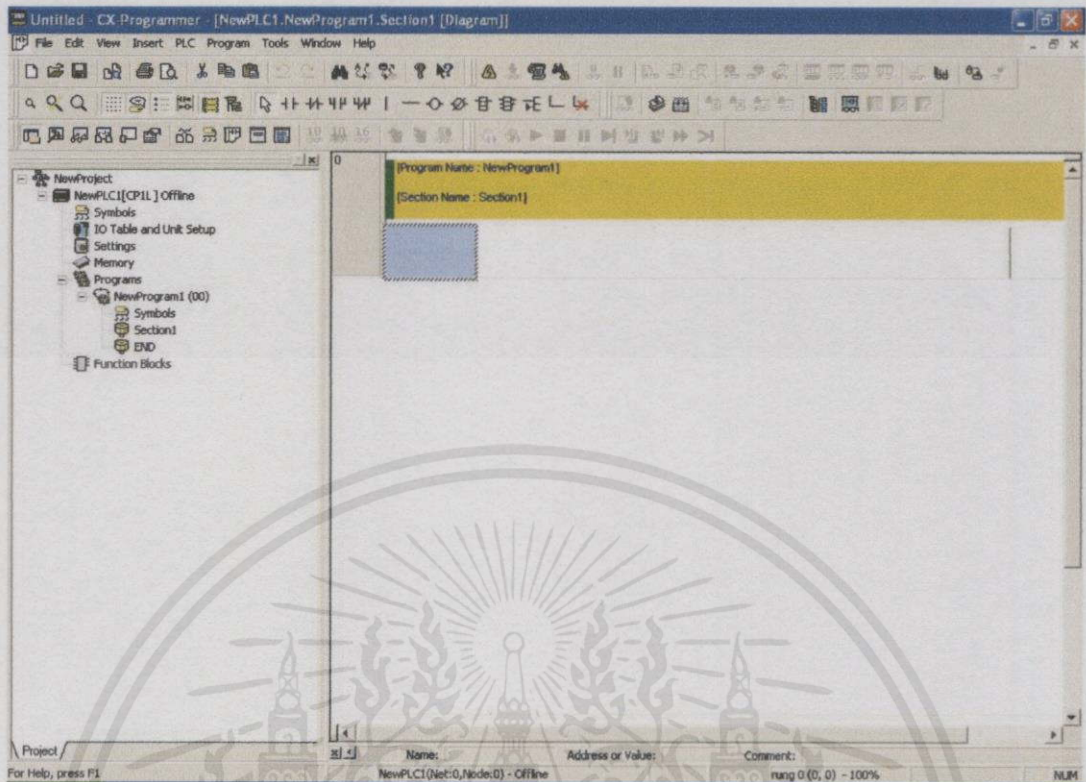
3.4 กำหนดการหมุนของกระเช้าจอตลอดทั้ง 7 กระเช้า

สำหรับการกำหนดการหมุนของกระเช้านั้นจะกำหนดว่าให้กระเช้าที่ 1 เป็นกระเช้าเริ่มต้น โดยในการหมุนกระเช้านั้นจะมี Proximity Sensor ในการนับชั้น โดยถ้าต้องการไปชั้นไหนกระเช้าจะต้องหมุนไปตามชั้นที่ต้องการ

3.4.1 เขียนแลคเตอร์ไคอะแกรม ตามเงื่อนไขที่กำหนด

ในการเขียนโปรแกรมนั้นจะใช้โปรแกรม CX – Programmer ในการเขียน Ladder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.4 หน้าจอโปรแกรม CX-Programmer

CX-Programmer Information						
C	W	O	I	N	SPACE	
Ctrl+1	Ctrl+4	Ctrl+K	Next Addr.	Find bit	Information Show/Hide	
W	M	O	B	L	Ctrl+Shift+I	
Ctrl+W	Ctrl+3	Ctrl+J	Ctrl+L	Prev. Jump	Comment	

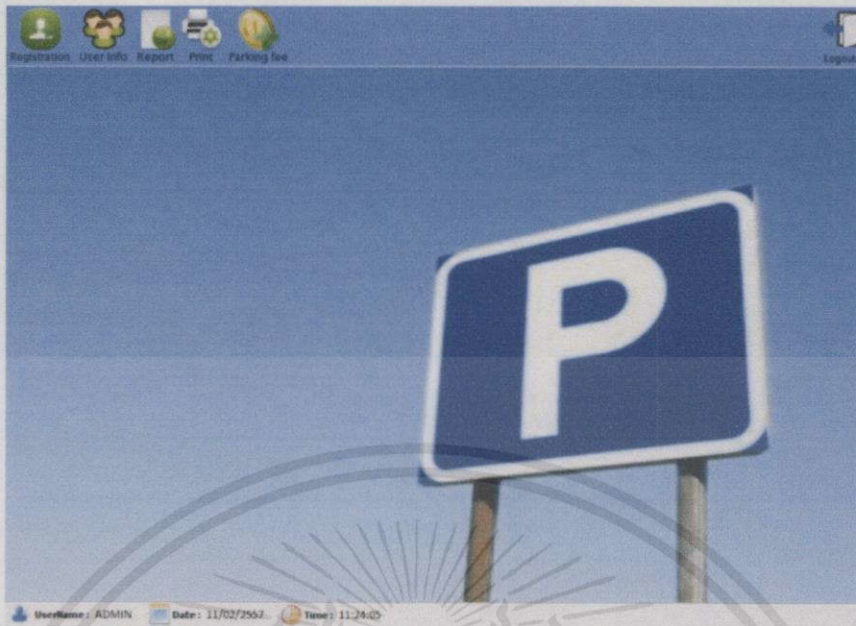
ภาพที่ 3.5 Shortcut Key ไว้สำหรับเขียนแลตเตอร์ตามเงื่อนไขต่างๆ

3.5 การออกแบบ Visual Basic

การออกแบบนั้นจะกำหนดการใช้งานเป็นสองส่วนคือส่วนที่เป็นแอดมินและส่วนที่เป็นยูสเซอร์ดังนี้

3.5.1 หน้าแอดมิน

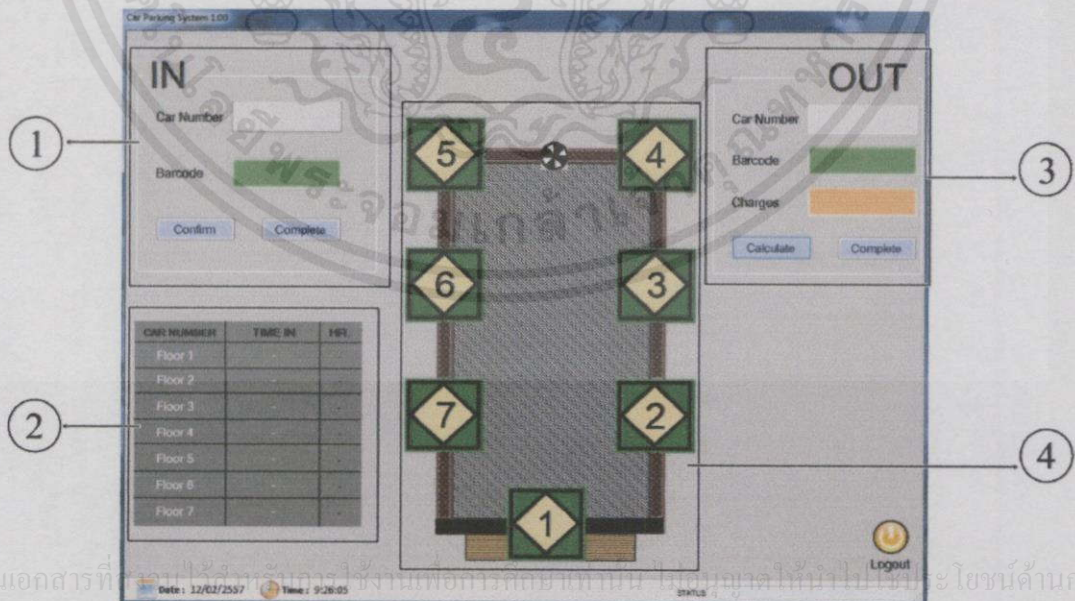
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6 หน้าจอที่เป็นส่วนของ ADMIN

โดยจะประกอบด้วยส่วนที่เป็นปุ่มกดเพื่อเชื่อมต่อไปยังหน้าต่างๆที่ใช้งานและส่วนที่แสดงสถานะต่างๆโดยหน้านี้เป็นหน้าหลักของแอดมินที่จะใช้เพื่อกำหนดและดูข้อมูลต่างๆในการจอดรถ และยังรวมถึงการกำหนดคูดิสเคอร์เนมและพาสเวิร์ดของผู้ที่ใช้งานอีกด้วย

3.5.2 หน้ายูสเซอร์



ภาพที่ 3.7 หน้าจอที่เป็นส่วนของ USER

สำหรับหน้ายูสเซอร์นั้นเป็นหน้าที่ให้ผู้ที่ควบคุมนั้นได้ควบคุมรถที่เข้าจอดกระเช้า และรถที่ออกจากกระเช้าจอดรวมถึงเวลาในการจอดรถโดยหลักนั้นจะแบ่งเป็น 4 ส่วนหลักๆได้แก่

- | | |
|-----------|--|
| ส่วนที่ 1 | เป็นส่วนที่กรอกข้อมูลนั้นก็คือทะเบียนรถและบาร์โค้ดของบัตรจอดรถโดยข้อมูลนั้นจะจัดเก็บอยู่ในฐานข้อมูลหลัก |
| ส่วนที่ 2 | เป็นตารางที่ใช้แสดงรถที่จอดอยู่ในกระเช้าจอดรถโดยตารางนี้จะประกอบด้วยทะเบียนรถ เวลาที่รถเข้าจอด และจำนวนชั่วโมงที่จอดรถ |
| ส่วนที่ 3 | เป็นส่วนที่ใช้สำหรับรถที่จะออกจากกระเช้าจอดรถโดยผู้ควบคุมจะยิงบาร์โค้ดของบัตรจอดรถและจะแสดงค่าใช้จ่ายในการจอดรถ |
| ส่วนที่ 4 | เป็นส่วนของกราฟิกที่ใช้แสดงสถานะของรถที่เข้าจอดในกระเช้าจอดรถโดยกราฟิกนั้นจะเปลี่ยนตามการทำงานของกระเช้าจอดรถโดยถ้ากระเช้าว่างจะโชว์สถานะสีเขียวส่วนกระเช้าไหนที่มีรถจอดอยู่จะแสดงสถานะเป็นสีแดง |

3.5 รายละเอียดของโปรแกรมทั้งหมด

โปรแกรมที่ใช้จะถูกแบ่งเป็นสองส่วนคือส่วนยูสเซอร์จะเป็นส่วนที่ใช้จัดการรถเข้าออก และดูสถานะในการจอดรถและส่วนของแอดมินจะเป็นส่วนที่ใช้ในการดูข้อมูลต่างๆในการจอดรถและใช้การกำหนดอัตราค่าใช้บริการรวมทั้งยังสามารถพิมพ์รายงานการจอดรถได้และยังเป็นส่วนที่ใช้กำหนดยูสเซอร์เนมและรหัสผ่านของผู้ควบคุมกระเช้า โดยมีรายละเอียดดังนี้

1. เมื่อเปิดโปรแกรมขึ้นมาจะปรากฏหน้าล็อกอินโดยหน้าล็อกอินนั้นจะแบ่งเป็นสองส่วนคือ แอดมินและยูสเซอร์โดยหน้านี้จะเป็นส่วนเริ่มต้นของโปรแกรมที่จะเลือกว่าจะเข้าไปใช้ในส่วนไหน
2. เมื่อเลือกเข้าส่วนของแอดมินจะเปิดหน้าหลักของแอดมินขึ้นมาซึ่งในหน้าหลักนี้เป็นหน้าที่ใช้ดูข้อมูลต่างๆในการจอดรถและใช้กำหนดยูสเซอร์และรหัสผ่านของยูสเซอร์ เมื่อเลือกเข้าที่ Registration จะเข้าไปยังหน้าที่ใช้บันทึกข้อมูลสำหรับผู้ที่ควบคุมกระเช้าจอดรถโดยมีการบันทึกข้อมูลต่างๆรวมทั้งกำหนดรหัสผ่านในการเข้าใช้งานด้วยซึ่งข้อมูลนี้ก็จะแสดงในตารางเก็บข้อมูล
3. เมื่อเลือกเข้าที่ Report ก็แสดงข้อมูลต่างๆในการจอดรถ เช่น ทะเบียนรถ เวลาเข้า เวลาออก วันที่จอด เวลาทั้งหมดที่จอดรถ ชั้นที่จอด ค่าใช้จ่ายทั้งหมด เป็นต้น และเมื่อเข้าหน้า Print เป็นหน้าที่ใช้สำหรับพิมพ์ข้อมูลจากรายงานที่บันทึกไว้ข้างต้นนำมาพิมพ์เป็นรายงานรวมถึงยังแสดงกราฟสถิติของรถที่จอดในแต่ละชั้น และหน้าสุดท้ายคือหน้ากำหนดค่าใช้จ่ายในการจอดรถทั้งหมด ซึ่งส่วนที่กล่าวมาทั้งหมดนั้นจะเป็นส่วนของแอดมินที่เข้าใช้งานได้

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาและวิจัยเท่านั้น ไม่สามารถนำไปใช้เพื่อการพาณิชย์ได้

เมื่อเลือกเข้าส่วนของยูสเซอร์จะเปิดหน้าที่ใช้ควบคุมกระเช้าทั้งหมดโดยยูสเซอร์จะเป็นคนจัดการในส่วนของรถที่เข้าจอดและรถที่ออกโดยจะมีกราฟฟิกจำลองจากกระเช้าจอดจริงไปใช้แสดงผลได้ด้วยและยังมีตารางการเข้าจอดรถไว้สำหรับดูข้อมูลการจอด

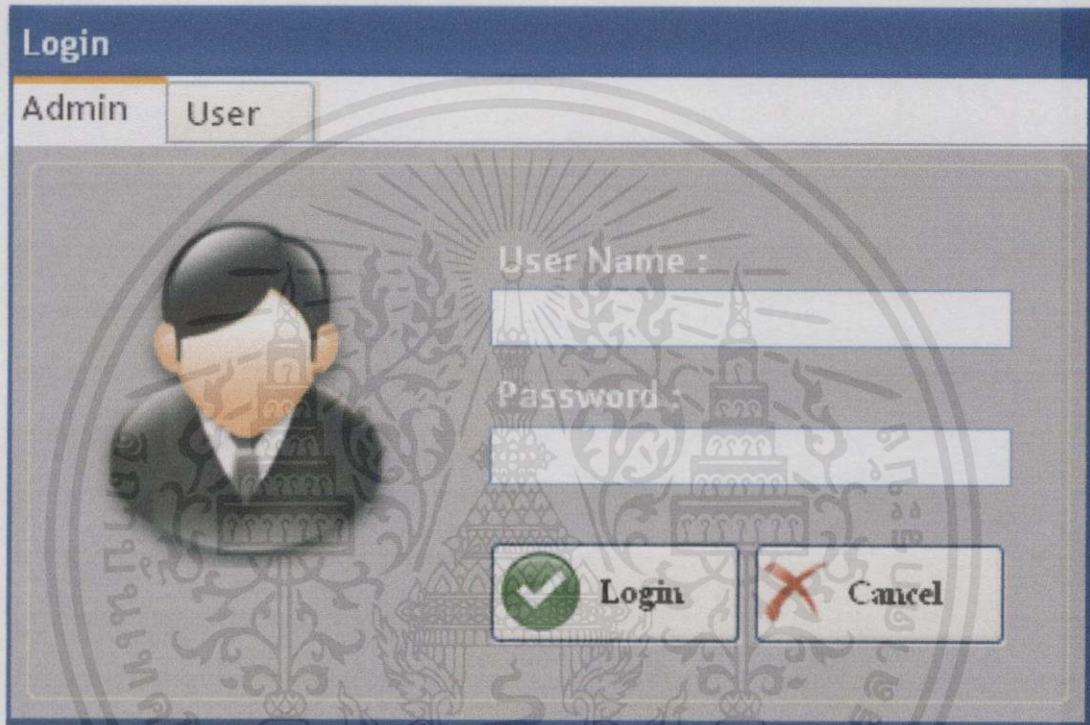
บทที่ 4

ผลการทดลอง



4.1 ผลการทดลอง RUN โปรแกรมการควบคุมกระเช้าจอดรถด้วยคอมพิวเตอร์

โปรแกรมที่ใช้ในการควบคุมมีอยู่ด้วยกัน 3 ส่วนใหญ่ๆ ดังนี้

4.1.1 หน้าโปรแกรมเข้าสู่ระบบ (Login)



ภาพที่ 4.1 ส่วนของหน้าจอ Login

1. Admin ใช้ในการเข้าสู่ระบบในฐานะ Admin
2. User ใช้ในการเข้าสู่ระบบในฐานะ User
3.  Login เมื่อใส่รหัสเสร็จแล้วก็จะนำไปสู่หน้าต่างใช้งานต่อไป
4.  Cancel ใช้เพื่อการออกจากโปรแกรม
5. Forget Password เมื่อมีการใส่รหัสผิดพลาดจะแสดงปุ่ม Forget Password ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

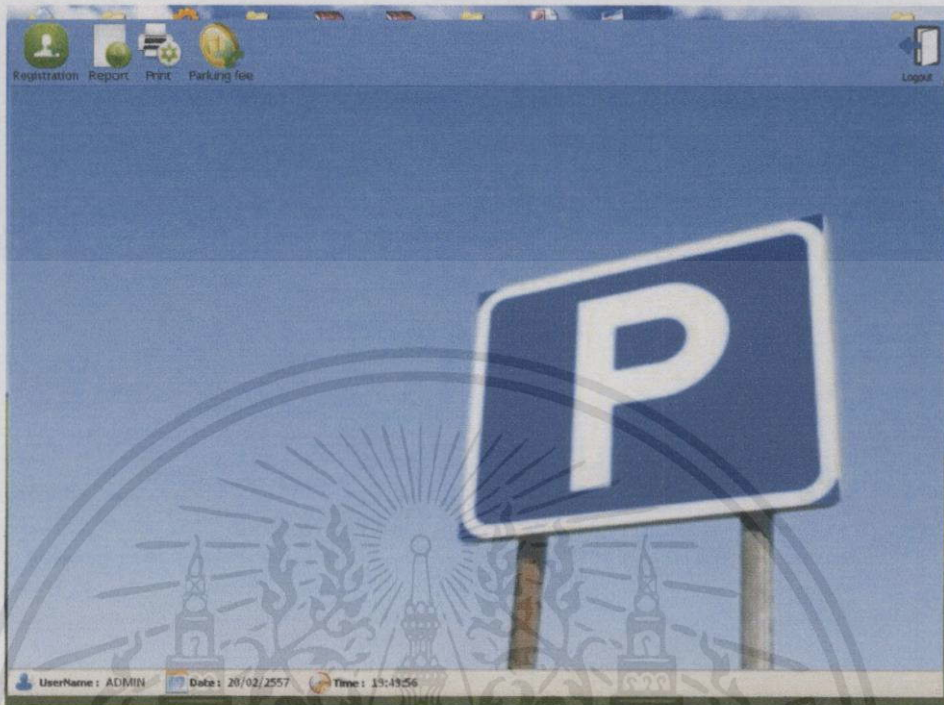
ภาพที่ 4.2 การแสดง Forget Password

เมื่อกด Forget Password จะแสดงหน้าต่าง ดังรูป 4.3 จากนั้นกรอกข้อมูลให้ครบถ้วนจากนั้นกด Ok รหัสผ่านก็จะแสดงออกมา


ภาพที่ 4.3 หน้าต่างตรวจสอบรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 หน้า ADMIN สำหรับผู้ดูแลระบบ




ภาพที่ 4.4 หน้าหลักของ ADMIN

1.  เพิ่มข้อมูลและตั้งรหัสผ่านสำหรับพนักงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาพที่ 4.5 ช่องสำหรับกรอกข้อมูลพนักงาน ให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งเมื่อกรอกข้อมูลแล้วกด Create ข้อมูลจะทำการบันทึกลงไปในระบบฐานข้อมูล
 หากกด Get Data จะทำการแสดงข้อมูลของพนักงานในฐานข้อมูลทั้งหมด


Registration Details					
Remove					
User Name	Date Of Birth	Place Of Birth	Email ID	User ID	Password
GAME	4/12/2556	BANGKOK	gamepot89@...	GAME	1234

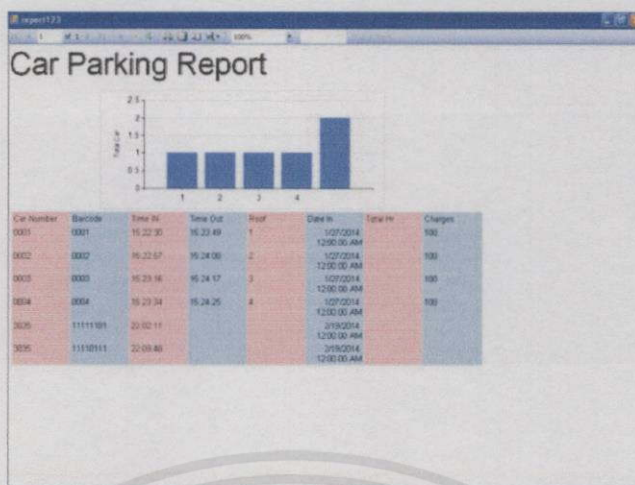
ภาพที่ 4.6 แสดงข้อมูลพนักงานในระบบฐานข้อมูล

2.  แสดงการให้บริการรถเช่าจอร์ด เวลาเข้า เวลาออกและค่าใช้จ่าย


Report							
Remove							
Call Number	Barcode	Time In	Time Out	Floor	Date In	Total Hr	Charges
0001	0001	15:22:56	15:23:49	1	27/1/2557		100
0002	0002	15:22:57	15:24:08	2	27/1/2557		100
0003	0003	15:23:16	15:24:17	3	27/1/2557		100
0004	0004	15:23:34	15:24:25	4	27/1/2557		100

ภาพที่ 4.7 ข้อมูลในฐานข้อมูลที่แสดงการให้บริการลูกค้า

- เอกสารนี้เป็นเอกสาร
3.  พิมพ์เอกสารรายงานและแสดงแผนภูมิแท่งเพื่อนำไปใช้งานต่อไป
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสาร ทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.8 หน้าจอแสดงผลประกอบการ

4.  ใช้สำหรับกำหนดอัตราค่าบริการ

Parking Fee

Daily


1 Hr. Baht / Hr.

2 - 3 Hr. Baht / Hr.

4 - 6 Hr. Baht / Hr.

Flat Rate Baht/Day

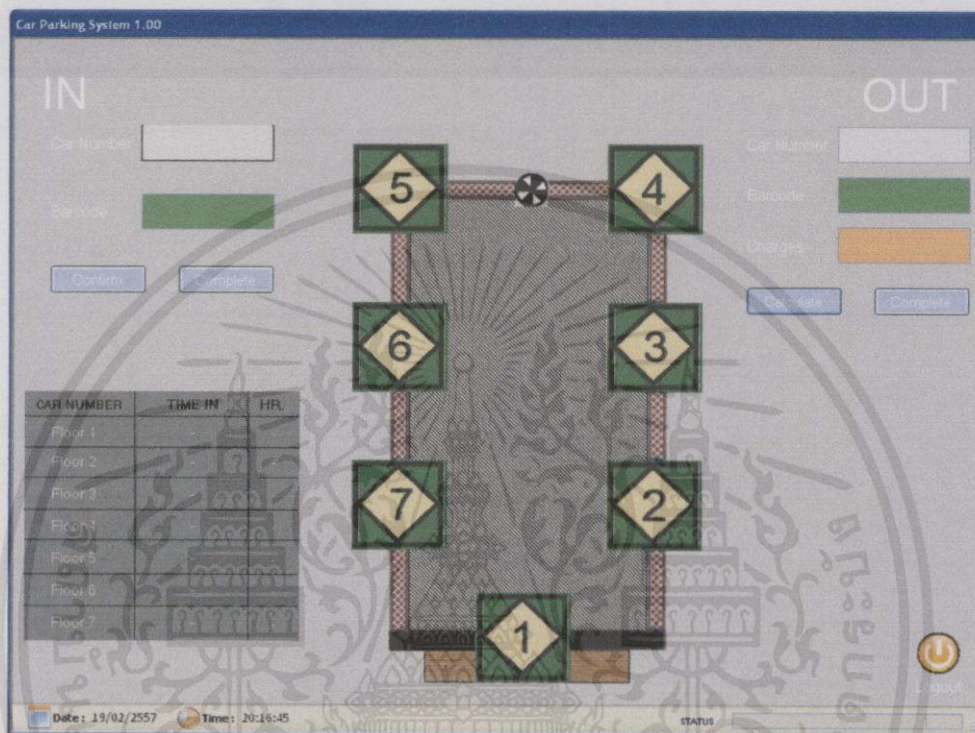
ภาพที่ 4.9 กำหนดอัตราค่าบริการ

5.  ออกจากโปรแกรม

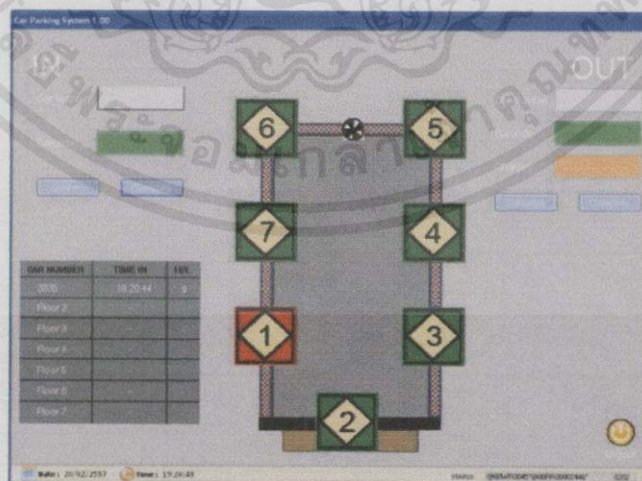
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 หน้า USER

สำหรับพนักงานเพื่อยิงบาร์โค้ดและกรอกข้อมูลทะเบียนรถเพื่อบันทึกข้อมูลลงในระบบฐานข้อมูลแสดงกราฟฟิคของกระเช้าจอดรถว่ามีชั้นใดที่ว่างและในขณะที่กระเช้าแต่ละชั้นอยู่ในตำแหน่งใดเมื่อมีรถที่ต้องการออกจากกระเช้าจะมีการยิงบาร์โค้ดเพื่อนำบาร์โค้ดที่ตรงกันจากฐานข้อมูลมาคำนวณค่าใช้จ่าย

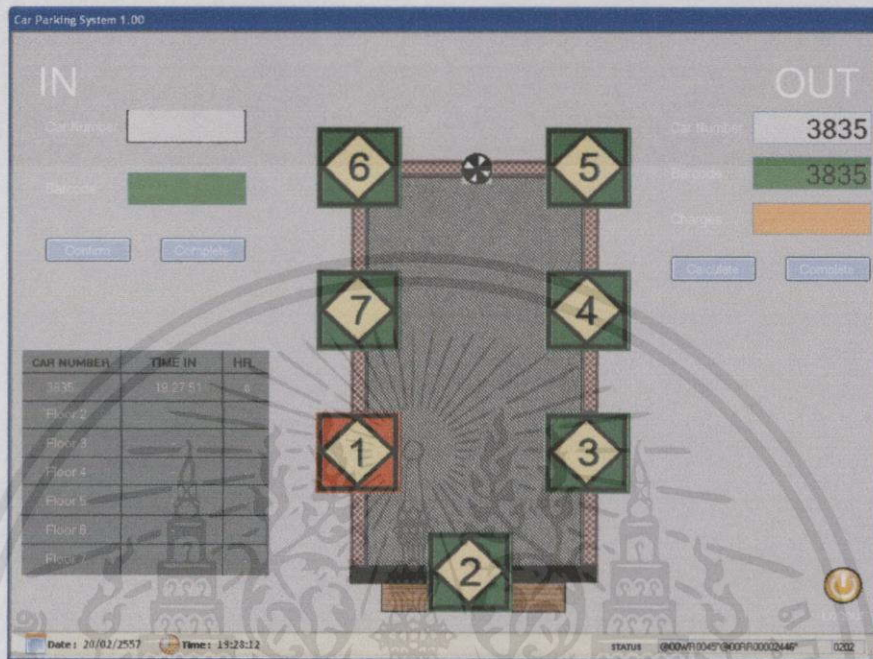


ภาพที่ 4.10 ส่วนของ USER



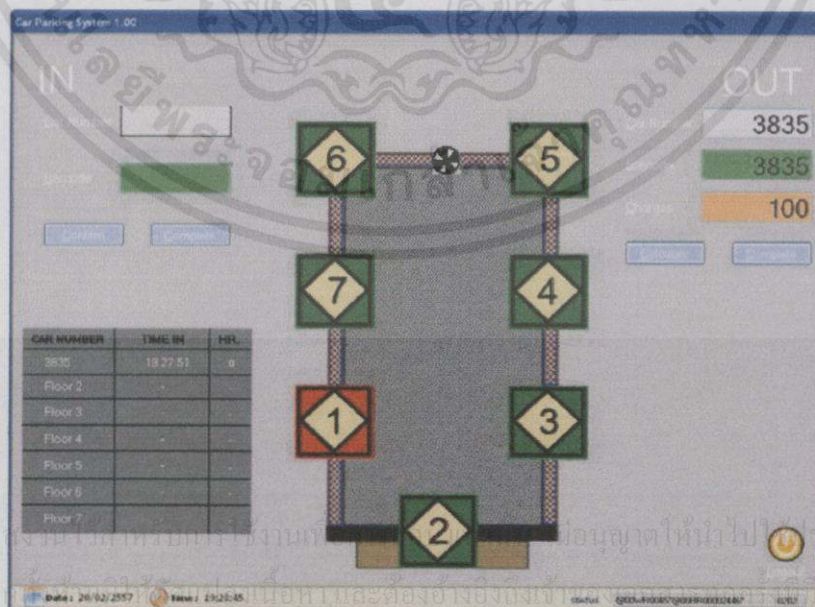
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ภาพที่ 4.11 เมื่อมีรถเข้ามาจอดในชั้นที่ 1 หลังจากนั้นก็จะแสดงสถานะเป็นสีแดงนำไปใช้

เมื่อรถที่จอดในชั้นที่ 1 ต้องการออกจะทำการใส่ค่าบาร์โค้ดจากนั้นระบบฐานข้อมูลจะนำข้อมูลเลขทะเบียนมาแสดงในช่องเลขทะเบียน



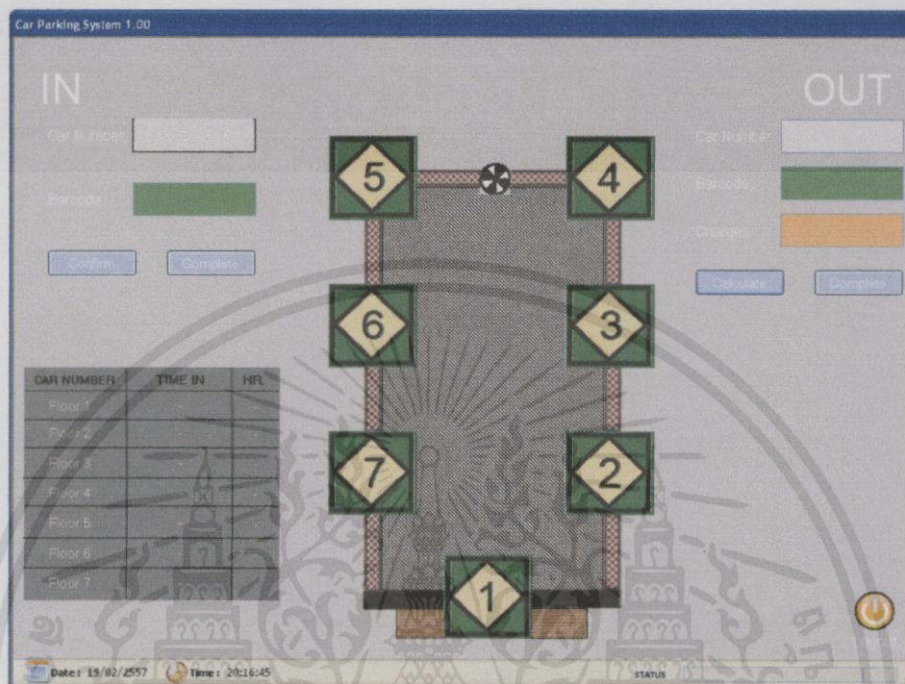
ภาพที่ 4.12 การดึงฐานข้อมูลเลขทะเบียนมาแสดงผล

เมื่อระบบดึงเลขทะเบียนขึ้นมาแล้วทางพนักงานจะต้องกด Calculate เพื่อทำการคำนวณค่าใช้จ่ายของลูกค้า



ภาพที่ 4.13 แสดงการคำนวณค่าใช้จ่าย

เมื่อลูกค้าชำระค่าบริการแล้วพนักงานจะต้องกดปุ่ม Complete กระเช้าจะทำการเลื่อนชั้นมายังชั้นที่รถต้องการจะออก ดังแสดงในภาพ 4.14 ซึ่งจะเห็นว่าในชั้นที่จะกลับมาเป็นสีเขียว นั้น แสดงว่าในชั้นที่ 1 สามารถนำรถเข้าใช้บริการได้ตามปกติ



ภาพที่ 4.14 ลูกค้าชำระค่าบริการและนำรถออก

4.2 สรุปผลการทดลอง

การสรุปผลจะแบ่งออกเป็นสองส่วนคือ ส่วนของพนักงาน และ ส่วนของผู้ดูแล ซึ่งจะสามารถอธิบายได้ดังนี้

ส่วนของพนักงาน จะเป็นการบอกสถานะของกระเช้าแต่ละชั้นว่าว่างหรือไม่ ถ้าว่างสถานะจะแสดงเป็นสีเขียว แต่ถ้ากระเช้านั้นไม่ว่างสถานะจะแสดงเป็นสีแดง นอกจากบอกสถานะแล้วพนักงานยังจะต้องกรอกข้อมูล ยิงบาร์โค้ด เพื่อให้ระบบบันทึกข้อมูลลงในฐานข้อมูล เมื่อรถของลูกค้ามีความต้องการออกจากกระเช้าพนักงานจะต้อนำบาร์โค้ดมายิงอีกรอบและระบบจะนำข้อมูลที่อยู่ในฐานข้อมูลมาคิดค่าบริการตามอัตราค่าบริการที่ผู้ดูแลตั้งอัตราค่าบริการไว้

ส่วนของผู้ดูแล ผู้ดูแลจะทำการกรอกข้อมูลลงในระบบฐานข้อมูลเมื่อมีพนักงานมาสมัครเพิ่ม และตรวจสอบความเรียบร้อยของการให้บริการ รวมถึงตั้งอัตราค่าบริการตามความเหมาะสม ซึ่งในส่วนของผู้ดูแลจะใช้งานระบบฐานข้อมูลเป็นหลัก ซึ่งต่างจากส่วนของพนักงานที่จะใช้ในส่วนของการฝึกเป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์โครงการ

5.1 สรุปโครงการ

5.1.1 เนื่องจากปัจจุบันในเมืองหลวงมีพื้นที่จำกัดในการขยายพื้นที่ด้านกว้าง ซึ่งส่งผลกระทบต่อการใช้สอยต่างๆ แต่ด้วยการแก้ไขปัญหาลแล้วถึงแม้ว่าจะขยายพื้นที่ด้านกว้างไม่ได้ก็ยังสามารถขยายความสูงได้เพื่อเพิ่มประสิทธิภาพในการใช้สอยพื้นที่ให้มากขึ้น ด้วยจำนวนรถที่มีมากขึ้นทุกวันซึ่งสวนทางกับพื้นที่ที่มีอย่างจำกัดของเมืองหลวง โครงการนี้จึงสามารถใช้ได้ดีกับโลกปัจจุบันและเป็นทางออกที่ดีที่จะช่วยแก้ปัญหาพื้นที่ที่มีอยู่อย่างจำกัดของโลกปัจจุบันนี้ โดยกระเช้าจอร์รถที่ควบคุมด้วยระบบคอมพิวเตอร์

5.1.2 การควบคุมกระเช้าจอร์รถนั้นจะควบคุมด้วยคอมพิวเตอร์โดยผ่านโปรแกรม Visual Basic 2010 ซึ่งจะเชื่อมต่อกับ PLC โดยใช้ Host link

5.1.3 เมื่อเปิดโปรแกรมขึ้นมา โปรแกรมจะประกอบด้วยสองส่วนหลักๆ คือ ส่วนผู้ดูแล และ ส่วนของพนักงาน ส่วนของผู้ดูแลนั้นจะประกอบด้วยข้อมูลต่างๆ เช่น ข้อมูลพนักงาน ข้อมูลการใช้บริการ โดยจะเก็บไว้ในระบบฐานข้อมูล ส่วนของพนักงานเมื่อเริ่มโปรแกรม โปรแกรมจะส่งค่า Host link เพื่อให้กระเช้าหมุนมาที่ชั้น 1 ซึ่งเมื่อมีรถเข้ามาจอดแล้วนั้น โปรแกรมจะสั่งให้กระเช้าหมุนไปที่ว่างทันที นอกจากนั้นส่วนของพนักงานก็จะทำหน้าที่คำนวณค่าบริการอีกด้วย

5.2 สรุปโครงการโดยรวม

โครงการนี้ใช้ในการควบคุมกระเช้าจอร์รถ ซึ่งชุดทดลองกระเช้าจอร์รถนี้มีทั้งหมด 7 ชั้น การหมุนของกระเช้าใช้มอเตอร์เป็นตัวขับเคลื่อนโดยผู้ใช้โปรแกรมจะสั่งกระเช้าทำงานจากคอมพิวเตอร์ โดยควบคุมผ่าน PLC ซึ่งระบบจะทำงานอัตโนมัติ โดยกระเช้าจะเคลื่อนที่ไปยังชั้นที่ว่างตลอดเวลา เมื่อรถต้องการออกจากกระเช้าจะใช้บาร์โค้ดเป็นตัวสั่งการ เมื่อยิงบาร์โค้ดแล้วระบบจะหาว่าบาร์โค้ดนี้อยู่ที่ชั้นใด เมื่อพบแล้วกระเช้าจะเคลื่อนที่มาโดยอัตโนมัติเช่นกัน ในส่วนของระบบฐานข้อมูลจะใช้ Microsoft Access 2010 เป็นตัวเก็บฐานข้อมูลต่างๆ ซึ่งผู้ใช้สามารถเพิ่มข้อมูลลงในฐานข้อมูลผ่าน Visual Basic ได้เลย

5.3 ปัญหาที่พบและอุปสรรคในการทำงาน

1. กระเช้าเคลื่อนที่ไม่ตรงกับค่าที่สั่ง เช่น กระเช้าอยู่ชั้นที่ 1 เมื่อสั่งให้เคลื่อนที่ไปยังชั้นที่ 5 กระเช้าจะเคลื่อนที่มาถึงชั้นที่ 4 เหตุมาจากกระเช้ามีการแกว่งจึงทำให้เซ็นเซอร์จับค่าได้ 2 ครั้ง ทำให้เกิดการนับซ้ำ ซึ่งจะแก้ไขได้โดยนำ Dip Up มาใช้เพื่อให้ทำงานเฉพาะขอบขาขึ้นเท่านั้น และได้เพิ่ม Timer เข้ามาหน่วงเวลาเพื่อรอให้กระเช้าหยุดนิ่งเสียก่อน

2. การเชื่อม PLC กับ คอมพิวเตอร์ เกิดจากปัญหาการสั่งค่า Host link จากคอมพิวเตอร์ไปยัง PLC แล้วไม่มีการตอบกลับจาก PLC ซึ่งอาจเกิดได้หลายสาเหตุ เช่น พอร์ตที่ใช้เชื่อมต่อมีการใช้งานซ้อนกัน ค่าบอรรถระหว่าง PLC กับคอมพิวเตอร์มีค่าที่ไม่ตรงกันซึ่งแก้ไขได้โดยการตั้งค่าระหว่าง PLC กับคอมพิวเตอร์ให้ตรงกัน

บรรณานุกรม

[1] Omron electronics . 2007 . Beginning of PLC Lv.1 . Omron electronics

[2] PLC easy . CP1 Series from <http://www.plceasy.com/>

[3] Professional Database Programming with VB2010 & VC# 2010 . IDC Premier Ltd.,

[4] Application develop by Visual basics



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

///// USER /////
Imports System.Data.OleDb
Imports System.IO
Imports System.Drawing.Drawing2D
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.IO.Ports

Public Class Form1
    Dim cn As New
    OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
    Dim T1 As Double
    Dim T2 As Double
    Dim T3 As Double
    Dim T4 As Double
    Dim time As String
    Dim txt As String
    Protected BoxImages(16) As Image
    Dim myPort As Array
    Delegate Sub SetTextCallback(ByVal
[text] As String)

    Private Sub Timer2_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
        TimeStatusLabel.Text =
Now.ToLongTimeString.ToString()

    End Sub

    Private Sub form1_FormClosing(ByVal
sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs
) Handles Me.FormClosing
        Me.Dispose()
        SerialPort1.Close()
        FrmLogin.Show()

    End Sub
    Sub clear()
        txtCarNumber.Clear()
        TimeStatusLabel.Text = Now
        DateStatusLabel.Text = Today
    End Sub
    Private Function RequiredEntry() As
Boolean
        If txtCarNumber.Text = "" Then
            MsgBox("Please enter Car
Number...", MsgBoxStyle.Critical,
"Attention")
            Return True
        End If
    End Function

    End Function
    Private Sub Form1_Load(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ToolStripStatusLabel3.Text =
        "Date :"
        DateStatusLabel.Text =
        Format(Now, "dd/MM/yyyy")
        ToolStripStatusLabel6.Text =
        "Time :"

        TextBox1.Text =
        Mid(rtbReceived.Text, 20, 2)

        BoxImages(0) = My.Resources.num1
        BoxImages(1) = My.Resources.num2
        BoxImages(2) = My.Resources.num3
        BoxImages(3) = My.Resources.num4
        BoxImages(4) = My.Resources.num5
        BoxImages(5) = My.Resources.num6
        BoxImages(6) = My.Resources.num7
        BoxImages(7) = My.Resources.num1r
        BoxImages(8) = My.Resources.num2r
        BoxImages(9) = My.Resources.num3r
        BoxImages(10) =
        My.Resources.num4r
        BoxImages(11) =
        My.Resources.num5r
        BoxImages(12) =
        My.Resources.num6r
        BoxImages(13) =
        My.Resources.num7r
        BoxImages(14) = My.Resources.C1
        BoxImages(15) = My.Resources.C2

        Timer1.Start()

        'RS(232)
        myPort =
        IO.Ports.SerialPort.GetPortNames()
        SerialPort1.PortName = "COM1"
        'Set SerialPort1 to the selected COM port
        at startup
        SerialPort1.BaudRate = 9600
        'Set Baud rate to the selected value on
        SerialPort1.Parity =
        IO.Ports.Parity.Even
        SerialPort1.StopBits =
        IO.Ports.StopBits.Two
        SerialPort1.DataBits = 7
        SerialPort1.Open()
        SerialPort1.Write("@00WR00100002"
& FcsCal("@00WR00100002") & vbCr)
        SerialPort1.Write("@00RR01000001"
& FcsCal("@00RR01000001") & vbCr)
    Dim cn As New
    OleDbConnection("Provider=Microsoft.ACE.0

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำใช้

```

LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
    If cn.State =
ConnectionState.Open Then
        cn.Close()
    End If
    cn.Open()
    Dim dr1 As OleDbDataReader
    Dim com1 As New OleDbCommand

```

```

    If TextBox3.Enabled = True Then
        com1.CommandText = "select
[ชั่วโมงรถ],[ชั่วโมงที่2ถึง3],[ชั่วโมงที่4ขึ้นไป],[รายละเอียด] from
FeeInfo where number = 1"
        com1.Connection = cn
        dr1 = com1.ExecuteReader
        If dr1.Read Then

```

```

            T1 = "" &
UCase(dr1(0)).ToString() & ""
            T2 = "" &
UCase(dr1(1)).ToString() & ""
            T3 = "" &
UCase(dr1(2)).ToString() & ""
            T4 = "" &
UCase(dr1(3)).ToString() & ""

```

```

        End If
        cn.Close()
        dr1.Close()
        Return
    End If

```

```
End Sub
```

```

Private Sub PictureBox14_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
PictureBox14.Click
    Me.Close()
End Sub

```

```

Private Sub btnConfirm_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
btnConfirm.Click
    If RequiredEntry() = True Then
        Return
    End If
    Try

```

```

        Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O

```

```

LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")

```

```

    If cn.State =
ConnectionState.Open Then
        cn.Close()
    End If
    cn.Open()

```

```

    If txtCarNumber.Enabled =
True Then

```

```

        Dim sSQL As String =
"insert into
PayInfo(CarNumber,Barcode,TimeIn,DateIn)
values(@d1,@d2,@d3,@d4)"
        Dim cmd As OleDbCommand =
New OleDbCommand(sSQL, cn)

```

```

        ' CarNumber
        Dim CarNumber As
OleDbParameter = New
OleDbParameter("@d1", OleDbType.VarWChar,
25)
        CarNumber.Value =
txtCarNumber.Text.ToString()

```

```
cmd.Parameters.Add(CarNumber)
```

```

        ' Barcode
        Dim Barcode As
OleDbParameter = New
OleDbParameter("@d2", OleDbType.VarWChar,
25)
        Barcode.Value =
txtBarcode.Text.ToString()

```

```
cmd.Parameters.Add(Barcode)
```

```

        ' TimeIn
        Dim TimeIn As
OleDbParameter = New
OleDbParameter("@d3", OleDbType.VarWChar,
15)

```

```

        TimeIn.Value =
TimeStatusLabel.Text.ToString()

```

```
cmd.Parameters.Add(TimeIn)
```

```

        ' DateIn
        Dim DateIn As
OleDbParameter = New
OleDbParameter("@d4", OleDbType.VarWChar,
15)

```

```

        DateIn.Value =
DateStatusLabel.Text.ToString()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นกรณีที่ได้รับอนุญาตเป็นอย่างอื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cmd.Parameters.Add(DateIn)

Then
    If cmd.ExecuteNonQuery()
        cn.Close()
        MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
    Else
        MsgBox("failed... ",
MsgBoxStyle.Critical, "Registration
failed")
        Return
    End If
Else
    MsgBox.Show("Plese
Confirm Car Number", "Attention",
MsgBoxButtons.OK,
MsgBoxIcon.Warning)
    End If
Catch ex As Exception
    MsgBox.Show(ex.Message.ToString(),
    "Data Error")
    Exit Sub
End Try
End Sub

Private Sub Button2_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

    Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
    If cn.State =
ConnectionState.Open Then
        cn.Close()
    End If
    cn.Open()
    Dim dr1 As OleDbDataReader
    Dim com As New OleDbCommand
    com.CommandText = "select
[Barcode] from PayInfo where Barcode =
@Barcode"

    ' Barcode
    Dim Barcode As OleDbParameter =
New OleDbParameter("@Barcode",
OleDbType.VarWChar, 15)

Barcode.Value =
UCase(txtBarcode.Text.ToString())
com.Parameters.Add(Barcode)
com.Connection = cn

dr1 = com.ExecuteReader
If dr1.Read Then

    If UCase(dr1("Barcode")) =
UCase(txtBarcode.Text) Then

        Dim sSQL As String =
"Update PayInfo Set Roof = @d5 where
Barcode =' & txtBarcode.Text & """"
        Dim cmd As OleDbCommand =
New OleDbCommand(sSQL, cn)

        If TextBox1.Text = "0101"
Or TextBox1.Text = "01" Then

            'Roof
            Dim Roof As
OleDbParameter = New
OleDbParameter("@d5", OleDbType.VarWChar,
15)
            Roof.Value = "1"

            cmd.Parameters.Add(Roof)

            If
cmd.ExecuteNonQuery() Then
                cn.Close()
                MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
                Dim car As String
                = txtCarNumber.Text
                number1.Text =
txtCarNumber.Text
                txtCarNumber.Text
                = ""
                txtBarcode.Text =
""
                Label8.Text =
TimeStatusLabel.Text.ToString()
                Hr1.Start()
                rtbReceived.Text
                = ""
                TextBox1.Text =
""
                If number2.Text =
"Floor 2" Then
                    SerialPort1.Write("@00WR00100004" &
FcsCal("@00WR00100004") & vbCr)
                ElseIf
number3.Text = "Floor 3" Then

```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SerialPort1.Write("@00WR00100008" &
FcsCal("@00WR00100008") & vbCr)
    ElseIf
number4.Text = "Floor 4" Then

SerialPort1.Write("@00WR00100010" &
FcsCal("@00WR00100010") & vbCr)
    ElseIf
number5.Text = "Floor 5" Then

SerialPort1.Write("@00WR00100020" &
FcsCal("@00WR00100020") & vbCr)
    ElseIf
number6.Text = "Floor 6" Then

SerialPort1.Write("@00WR00100040" &
FcsCal("@00WR00100040") & vbCr)
    ElseIf
number7.Text = "Floor 7" Then

SerialPort1.Write("@00WR00100080" &
FcsCal("@00WR00100080") & vbCr)
    End If

    Timer1.Stop()
    Timer3.Start()
    Timer4.Stop()
    Timer5.Stop()
    Timer6.Stop()
    Timer7.Stop()
    Timer8.Stop()
    Timer9.Stop()

Else
    MsgBox("failed...
", MsgBoxStyle.Critical, "Registration
failed")

    Return
End If

ElseIf TextBox1.Text =
"0202" Or TextBox1.Text = "02" Then

    'Roof
    Dim Roof As
OleDbParameter = New
OleDbParameter("@d5", OleDbType.VarWChar,
15)

    Roof.Value = "2"

cmd.Parameters.Add(Roof)

    If
cmd.ExecuteNonQuery() Then
        cn.Close()

MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
Dim car1 As
String = txtCarNumber.Text
number2.Text =
txtCarNumber.Text
txtCarNumber.Text
= ""
txtBarcode.Text =
""
Label9.Text =
TimeStatusLabel.Text.ToString()
Hr2.Start()
rtbReceived.Text
= ""
""
TextBox1.Text =
If number3.Text =
"Floor 3" Then
SerialPort1.Write("@00WR00100008" &
FcsCal("@00WR00100008") & vbCr)
    ElseIf
number4.Text = "Floor 4" Then
SerialPort1.Write("@00WR00100010" &
FcsCal("@00WR00100010") & vbCr)
    ElseIf
number5.Text = "Floor 5" Then
SerialPort1.Write("@00WR00100020" &
FcsCal("@00WR00100020") & vbCr)
    ElseIf
number6.Text = "Floor 6" Then
SerialPort1.Write("@00WR00100040" &
FcsCal("@00WR00100040") & vbCr)
    ElseIf
number7.Text = "Floor 7" Then
SerialPort1.Write("@00WR00100080" &
FcsCal("@00WR00100080") & vbCr)
    ElseIf
number1.Text = "Floor 1" Then
SerialPort1.Write("@00WR00100002" &
FcsCal("@00WR00100002") & vbCr)
    End If

Timer1.Stop()
Timer3.Stop()
Timer4.Start()
Timer5.Stop()
Timer6.Stop()
Timer7.Stop()
Timer8.Stop()
Timer9.Stop()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    MsgBox("failed...
", MsgBoxStyle.Critical, "Registration
failed")
    Return
End If

ElseIf TextBox1.Text =
"0303" Or TextBox1.Text = "03" Then

    'Roof
    Dim Roof As
OleDbParameter = New
OleDbParameter("@d5", OleDbType.VarWChar,
15)

    Roof.Value = "3"

cmd.Parameters.Add(Roof)

If
cmd.ExecuteNonQuery() Then
    cn.Close()
    MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
    Dim car As String
= txtCarNumber.Text
txtCarNumber.Text
= ""
""
TimeStatusLabel.Text.ToString()
Hr3.Start()
rtbReceived.Text
= ""
""
    TextBox1.Text =
    If number4.Text =
"Floor 4" Then
        SerialPort1.Write("@00WR00100010" &
FcsCal("@00WR00100010") & vbCr)
        ElseIf
number5.Text = "Floor 5" Then
            SerialPort1.Write("@00WR00100020" &
FcsCal("@00WR00100020") & vbCr)
            ElseIf
number6.Text = "Floor 6" Then
                SerialPort1.Write("@00WR00100040" &
FcsCal("@00WR00100040") & vbCr)
                ElseIf
number7.Text = "Floor 7" Then
                    SerialPort1.Write("@00WR00100080" &
FcsCal("@00WR00100080") & vbCr)
                    ElseIf
number1.Text = "Floor 1" Then
                        SerialPort1.Write("@00WR00100002" &
FcsCal("@00WR00100002") & vbCr)
                        ElseIf
number2.Text = "Floor 2" Then
                            SerialPort1.Write("@00WR00100004" &
FcsCal("@00WR00100004") & vbCr)
                            End If
                                Timer1.Stop()
                                Timer3.Stop()
                                Timer4.Stop()
                                Timer5.Start()
                                Timer6.Stop()
                                Timer7.Stop()
                                Timer8.Stop()
                                Timer9.Stop()
                                    Else
                                        MsgBox("failed...
", MsgBoxStyle.Critical, "Registration
failed")
                                        Return
                                        End If
                                            ElseIf TextBox1.Text =
"0404" Or TextBox1.Text = "04" Then
                                                'Roof
                                                Dim Roof As
OleDbParameter = New
OleDbParameter("@d5", OleDbType.VarWChar,
15)
                                                Roof.Value = "4"
                                                cmd.Parameters.Add(Roof)
                                                If
cmd.ExecuteNonQuery() Then
                                                    cn.Close()
                                                    MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
                                                    Dim car As String
= txtCarNumber.Text
number4.Text =
txtCarNumber.Text
= ""
txtBarcode.Text =

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label11.Text =
TimeStatusLabel.Text.ToString()
Hr4.Start()
rtbReceived.Text
= ""
""
TextBox1.Text =
""
If number5.Text =
"Floor 5" Then
SerialPort1.Write("@00WR00100020" &
FcsCal("@00WR00100020") & vbCr)
ElseIf
number6.Text = "Floor 6" Then
SerialPort1.Write("@00WR00100040" &
FcsCal("@00WR00100040") & vbCr)
ElseIf
number7.Text = "Floor 7" Then
SerialPort1.Write("@00WR00100080" &
FcsCal("@00WR00100080") & vbCr)
ElseIf
number1.Text = "Floor 1" Then
SerialPort1.Write("@00WR00100002" &
FcsCal("@00WR00100002") & vbCr)
ElseIf
number2.Text = "Floor 2" Then
SerialPort1.Write("@00WR00100004" &
FcsCal("@00WR00100004") & vbCr)
ElseIf
number3.Text = "Floor 3" Then
SerialPort1.Write("@00WR00100008" &
FcsCal("@00WR00100008") & vbCr)
End If
Timer1.Stop()
Timer3.Stop()
Timer4.Stop()
Timer5.Stop()
Timer6.Start()
Timer7.Stop()
Timer8.Stop()
Timer9.Stop()
Else
MsgBox("failed...
", MsgBoxStyle.Critical, "Registration
failed")
Return
End If
ElseIf TextBox1.Text =
"0505" Or TextBox1.Text = "05" Then
'Roof
Dim Roof As
OleDbParameter = New
OleDbParameter("@d5", OleDbType.VarWChar,
15)
Roof.Value = "5"
cmd.Parameters.Add(Roof)
If
cmd.ExecuteNonQuery() Then
cn.Close()
MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
Dim car As String
= txtCarNumber.Text
number5.Text =
txtCarNumber.Text
txtCarNumber.Text
= ""
""
txtBarcode.Text =
Label12.Text =
TimeStatusLabel.Text.ToString()
Hr5.Start()
rtbReceived.Text
= ""
""
TextBox1.Text =
If number6.Text =
"Floor 6" Then
SerialPort1.Write("@00WR00100040" &
FcsCal("@00WR00100040") & vbCr)
ElseIf
number7.Text = "Floor 7" Then
SerialPort1.Write("@00WR00100080" &
FcsCal("@00WR00100080") & vbCr)
ElseIf
number1.Text = "Floor 1" Then
SerialPort1.Write("@00WR00100002" &
FcsCal("@00WR00100002") & vbCr)
ElseIf
number2.Text = "Floor 2" Then
SerialPort1.Write("@00WR00100004" &
FcsCal("@00WR00100004") & vbCr)
ElseIf
number3.Text = "Floor 3" Then
SerialPort1.Write("@00WR00100008" &
FcsCal("@00WR00100008") & vbCr)
ElseIf
number4.Text = "Floor 4" Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าวิจัยเท่านั้น มิใช่เพื่อการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คิดแบบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SerialPort1.Write("@00WR00100010" &
FcsCal("@00WR00100010") & vbCrLf)
End If

Timer1.Stop()
Timer3.Stop()
Timer4.Stop()
Timer5.Stop()
Timer6.Stop()
Timer7.Start()
Timer8.Stop()
Timer9.Stop()

Else
    MsgBox("failed...
", MsgBoxStyle.Critical, "Registration
failed")

    Return
End If

ElseIf TextBox1.Text =
"0606" Or TextBox1.Text = "06" Then

    'Roof
    Dim Roof As
OleDbParameter = New
OleDbParameter("@d5", OleDbType.VarWChar,
15)

    Roof.Value = "6"

cmd.Parameters.Add(Roof)

If
cmd.ExecuteNonQuery() Then
    cn.Close()
    MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
    Dim car As String
    number6.Text =
txtCarNumber.Text

txtCarNumber.Text
= ""

""

TimeStatusLabel.Text.ToString()
Hr6.Start()
rtbReceived.Text
= ""

""

    TextBox1.Text =

If number7.Text =
"Floor 7" Then

SerialPort1.Write("@00WR00100080" &
FcsCal("@00WR00100080") & vbCrLf)
ElseIf
number1.Text = "Floor 1" Then

SerialPort1.Write("@00WR00100002" &
FcsCal("@00WR00100002") & vbCrLf)
ElseIf
number2.Text = "Floor 2" Then

SerialPort1.Write("@00WR00100004" &
FcsCal("@00WR00100004") & vbCrLf)
ElseIf
number3.Text = "Floor 3" Then

SerialPort1.Write("@00WR00100008" &
FcsCal("@00WR00100008") & vbCrLf)
ElseIf
number4.Text = "Floor 4" Then

SerialPort1.Write("@00WR00100010" &
FcsCal("@00WR00100010") & vbCrLf)
ElseIf
number5.Text = "Floor 5" Then

SerialPort1.Write("@00WR00100020" &
FcsCal("@00WR00100020") & vbCrLf)
End If
Timer1.Stop()
Timer3.Stop()
Timer4.Stop()
Timer5.Stop()
Timer6.Stop()
Timer7.Stop()
Timer8.Start()
Timer9.Stop()

Else
    MsgBox("failed...
", MsgBoxStyle.Critical, "Registration
failed")

    Return
End If

ElseIf TextBox1.Text =
"0707" Or TextBox1.Text = "07" Then

    'Roof
    Dim Roof As
OleDbParameter = New
OleDbParameter("@d5", OleDbType.VarWChar,
15)

    Roof.Value = "7"

cmd.Parameters.Add(Roof)

If
cmd.ExecuteNonQuery() Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cn.Close()
        MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
Dim car As String
= txtCarNumber.Text
txtCarNumber.Text
= ""
""
txtBarcode.Text =
Label14.Text =
TimeStatusLabel.Text.ToString()
Hr7.Start()
rtbReceived.Text
= ""
""
TextBox1.Text =
If number1.Text =
"Floor 1" Then
SerialPort1.Write("@00WR00100002" &
FcsCal("@00WR00100002") & vbCr)
number2.Text = "Floor 2" Then
SerialPort1.Write("@00WR00100004" &
FcsCal("@00WR00100004") & vbCr)
number3.Text = "Floor 3" Then
SerialPort1.Write("@00WR00100008" &
FcsCal("@00WR00100008") & vbCr)
number4.Text = "Floor 4" Then
SerialPort1.Write("@00WR00100010" &
FcsCal("@00WR00100010") & vbCr)
number5.Text = "Floor 5" Then
SerialPort1.Write("@00WR00100020" &
FcsCal("@00WR00100020") & vbCr)
number6.Text = "Floor 6" Then
SerialPort1.Write("@00WR00100040" &
FcsCal("@00WR00100040") & vbCr)
End If
Timer1.Stop()
Timer3.Stop()
Timer4.Stop()
Timer5.Stop()
Timer6.Stop()
Timer7.Stop()
Timer8.Stop()
Timer9.Start()
Else
MsgBox("failed...
", MsgBoxStyle.Critical, "Registration
failed")
Return
End If
End If
End If
End If
SerialPort1.Write("@00RR01000001"
& FcsCal("@00RR01000001") & vbCr)
Exit Sub
End Sub
Private Sub btnCal_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles btnCal.Click
If
Len(Trim(txtCarNumberOut.Text)) = 0 Then
MsgBox.Show("Please enter
car number", "Input Error",
MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
txtCarNumberOut.Focus()
Exit Sub
End If
Try
Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
If cn.State =
ConnectionState.Open Then
cn.Close()
End If
cn.Open()
Dim dr1 As OleDbDataReader
Dim com As New OleDbCommand
com.CommandText = "select
[CarNumber],[TimeIn] from PayInfo where
CarNumber = @CNumber"
' CarNumber
Dim CNumber As OleDbParameter
= New OleDbParameter("@CNumber",
OleDbType.VarWChar, 20)
CNumber.Value =
UCase(txtCarNumberOut.Text.ToString())
com.Parameters.Add(CNumber)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุผลเบี่ยงเบนเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

com.Connection = cn
dr1 = com.ExecuteReader

If dr1.Read Then
    Dim diff As TimeSpan
    Dim timeSS As String

    time = "" &
    UCase(dr1(1)).ToString() & ""

    Dim time1 As Date
    Dim time2 As Date
    time1 =
    Convert.ToDateTime(time)
    time2 =
    Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff =
    time2.Subtract(time1)
    timeSS =
    FormatNumber(diff.TotalMinutes, 0)

    If
    UCase(dr1("CarNumber")) =
    UCase(txtCarNumberOut.Text) Then
        Dim sSQL As String =
        "Update PayInfo Set
        Timeout=@d4,Charges=@d5,TotalHr=@d6 where
        CarNumber ='" & txtCarNumberOut.Text &
        ""

        Dim cmd As
        OleDbCommand = New OleDbCommand(sSQL, cn)

        'Timeout
        Dim Timeout As
        OleDbParameter = New
        OleDbParameter("@d4", OleDbType.VarWChar,
        15)
        Timeout.Value =
        TimeStatusLabel.Text.ToString()

        cmd.Parameters.Add(Timeout)

        'Charge
        Dim charge As
        OleDbParameter = New
        OleDbParameter("@d5", OleDbType.VarWChar,
        15)

        If 60 >= timeSS Then
            charge.Value = T1

            cmd.Parameters.Add(charge)
            Then
                ElseIf 120 >= timeSS
                Then
                    charge.Value = T1
                    + T2

                    cmd.Parameters.Add(charge)
                    ElseIf 180 >= timeSS
                    Then
                        charge.Value = T1
                        + T2 + T2

                        cmd.Parameters.Add(charge)
                        ElseIf 240 >= timeSS
                        Then
                            charge.Value = T1
                            + T2 + T2 + T3

                            cmd.Parameters.Add(charge)
                            ElseIf 300 >= timeSS
                            Then
                                charge.Value = T1
                                + T2 + T2 + T2 + T3

                                cmd.Parameters.Add(charge)
                                ElseIf 360 >= timeSS
                                Then
                                    charge.Value = T1
                                    + T2 + T2 + T2 + T3 + T3

                                    cmd.Parameters.Add(charge)
                                    Else
                                        charge.Value = T4

                                        cmd.Parameters.Add(charge)
                                        End If

                                        'TotalHr
                                        Dim TThr As
                                        OleDbParameter = New
                                        OleDbParameter("@d4", OleDbType.VarWChar,
                                        15)

                                        If
                                        txtCarNumberOut.Text = number1.Text Then
                                            TThr.Value =
                                            Label15.Text

                                            cmd.Parameters.Add(TThr)

                                            ElseIf
                                            txtCarNumberOut.Text = number2.Text Then
                                                TThr.Value =
                                                Label16.Text

                                                cmd.Parameters.Add(TThr)

                                                ElseIf
                                                txtCarNumberOut.Text = number3.Text Then
                                                    TThr.Value =
                                                    Label17.Text

                                                    cmd.Parameters.Add(TThr)
                    Then
                        cmd.Parameters.Add(TThr)
                    End If
                End If
            End If
        End If
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

Catch ex As Exception

MessageBox.Show(ex.Message.ToString(),
"Data Error")
Exit Sub
End Try
End Sub

Private Sub Button4_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

If txtCarNumberOut.Text =
number1.Text Then
number1.Text = "Floor 1"
txtCarNumberOut.Text = ""
txtBarcode1.Text = ""
TextBox3.Text = ""
rtbReceived.Text = ""
TextBox1.Text = ""
Hr1.Stop()
Label18.Text = " - "
Label15.Text = "-"

SerialPort1.Write("@00WR00100002" &
FcsCal("@00WR00100002") & vbCr)
Timer1.Stop()
Timer3.Stop()
Timer4.Stop()
Timer5.Stop()
Timer6.Stop()
Timer7.Stop()
Timer8.Stop()
Timer9.Start()

ElseIf txtCarNumberOut.Text =
number2.Text Then
number2.Text = "Floor 2"
txtCarNumberOut.Text = ""
txtBarcode1.Text = ""
TextBox3.Text = ""
rtbReceived.Text = ""
TextBox1.Text = ""
Hr2.Stop()
Label19.Text = " - "
Label16.Text = "-"

SerialPort1.Write("@00WR00100004" &
FcsCal("@00WR00100004") & vbCr)
Timer1.Stop()
Timer3.Start()
Timer4.Stop()
Timer5.Stop()
Timer6.Stop()
Timer7.Stop()

ElseIf txtCarNumberOut.Text =
number3.Text Then
number3.Text = "Floor 3"
txtCarNumberOut.Text = ""
txtBarcode1.Text = ""
TextBox3.Text = ""
rtbReceived.Text = ""
TextBox1.Text = ""
Hr3.Stop()
Label10.Text = " - "
Label17.Text = "-"

SerialPort1.Write("@00WR00100008" &
FcsCal("@00WR00100008") & vbCr)
Timer1.Stop()
Timer3.Stop()
Timer4.Start()
Timer5.Stop()
Timer6.Stop()
Timer7.Stop()
Timer8.Stop()
Timer9.Stop()

ElseIf txtCarNumberOut.Text =
number4.Text Then
number4.Text = "Floor 4"
txtCarNumberOut.Text = ""
txtBarcode1.Text = ""
TextBox3.Text = ""
rtbReceived.Text = ""
TextBox1.Text = ""
Hr4.Stop()
Label11.Text = " - "
Label18.Text = "-"

SerialPort1.Write("@00WR00100010" &
FcsCal("@00WR00100010") & vbCr)
Timer1.Stop()
Timer3.Stop()
Timer4.Stop()
Timer5.Start()
Timer6.Stop()
Timer7.Stop()
Timer8.Stop()
Timer9.Stop()

ElseIf txtCarNumberOut.Text =
number5.Text Then
number5.Text = "Floor 5"
txtCarNumberOut.Text = ""
txtBarcode1.Text = ""
TextBox3.Text = ""
Timer8.Stop()
Timer9.Stop()

```

```

rtbReceived.Text = ""
TextBox1.Text = ""
Hr5.Stop()
Label12.Text = " - "
"
Label19.Text = "- "
Timer5.Stop()
Timer6.Stop()
Timer7.Stop()
Timer8.Start()
Timer9.Stop()
End If
SerialPort1.Write("@00WR00100020" &
FcsCal("@00WR00100020") & vbCr)
Private Sub Timer1_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
Dim g As Graphics =
CreateGraphics()
g.DrawImage(BoxImages(0), 486,
584, 100, 95)
g.DrawImage(BoxImages(1), 623,
444, 100, 95)
g.DrawImage(BoxImages(2), 623,
277, 100, 95)
g.DrawImage(BoxImages(3), 623,
110, 100, 95)
g.DrawImage(BoxImages(4), 355,
110, 100, 95)
g.DrawImage(BoxImages(5), 355,
277, 100, 95)
g.DrawImage(BoxImages(6), 355,
444, 100, 95)
g.DrawImage(BoxImages(15), 526,
141, 35, 35)
g.Dispose()
End Sub
Private Sub Timer3_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick
Graphics
Dim g As Graphics =
CreateGraphics()
g.DrawImage(BoxImages(14), 526,
141, 35, 35)
'.....Floor1
If number1.Text = "Floor 1" Then
g.DrawImage(BoxImages(0),
355, 444, 100, 95)
Else
g.DrawImage(BoxImages(7),
355, 444, 100, 95)
End If
'.....Floor2
If number2.Text = "Floor 2" Then
SerialPort1.Write("@00WR00100040" &
FcsCal("@00WR00100040") & vbCr)
Timer1.Stop()
Timer3.Stop()
Timer4.Stop()
Timer5.Stop()
Timer6.Stop()
Timer7.Start()
Timer8.Stop()
Timer9.Stop()
ElseIf txtCarNumberOut.Text =
number7.Text Then
number7.Text = "Floor 7"
txtCarNumberOut.Text = ""
txtBarcode1.Text = ""
TextBox3.Text = ""
rtbReceived.Text = ""
TextBox1.Text = ""
Hr7.Stop()
Label14.Text = " - "
"
Label22.Text = "- "
Timer1.Stop()
Timer3.Stop()
Timer4.Stop()

```

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ห้ามแจกจ่ายให้ผู้อื่นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามนำให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        g.DrawImage(BoxImages(1),
486, 584, 100, 95)
    Else
        g.DrawImage(BoxImages(8),
486, 584, 100, 95)

    End If
    .....Floor3
    If number3.Text = "Floor 3" Then
        g.DrawImage(BoxImages(2),
623, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(9),
623, 444, 100, 95)

    End If
    .....Floor4
    If number4.Text = "Floor 4" Then
        g.DrawImage(BoxImages(3),
623, 277, 100, 95)
    Else
        g.DrawImage(BoxImages(10),
623, 277, 100, 95)

    End If
    .....Floor5
    If number5.Text = "Floor 5" Then
        g.DrawImage(BoxImages(4),
623, 110, 100, 95)
    Else
        g.DrawImage(BoxImages(11),
623, 110, 100, 95)

    End If
    .....Floor6
    If number6.Text = "Floor 6" Then
        g.DrawImage(BoxImages(5),
355, 110, 100, 95)
    Else
        g.DrawImage(BoxImages(12),
355, 110, 100, 95)

    End If
    .....Floor7
    If number7.Text = "Floor 7" Then
        g.DrawImage(BoxImages(6),
355, 277, 100, 95)
    Else
        g.DrawImage(BoxImages(13),
355, 277, 100, 95)

    End If
    g.Dispose()
    .....

```

Graphics

```

    End Sub

    Private Sub Timer4_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer4.Tick
    .....
    Graphics
    .....
    Dim g As Graphics =
CreateGraphics()
    g.DrawImage(BoxImages(15), 526,
141, 35, 35)
    .....Floor1
    If number1.Text = "Floor 1" Then
        g.DrawImage(BoxImages(0),
355, 277, 100, 95)
    Else
        g.DrawImage(BoxImages(7),
355, 277, 100, 95)

    End If
    .....Floor2
    If number2.Text = "Floor 2" Then
        g.DrawImage(BoxImages(1),
355, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(8),
355, 444, 100, 95)

    End If
    .....Floor3
    If number3.Text = "Floor 3" Then
        g.DrawImage(BoxImages(2),
486, 584, 100, 95)
    Else
        g.DrawImage(BoxImages(9),
486, 584, 100, 95)

    End If
    .....Floor4
    If number4.Text = "Floor 4" Then
        g.DrawImage(BoxImages(3),
623, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(10),
623, 444, 100, 95)

    End If
    .....Floor5
    If number5.Text = "Floor 5" Then
        g.DrawImage(BoxImages(4),
623, 277, 100, 95)
    Else
        g.DrawImage(BoxImages(11),
623, 277, 100, 95)

    End If
    .....Floor6

```

End If

.....Floor6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        If number6.Text = "Floor 6" Then
            g.DrawImage(BoxImages(5),
623, 110, 100, 95)
        Else
            g.DrawImage(BoxImages(12),
623, 110, 100, 95)

        End If
        .....Floor7
        If number7.Text = "Floor 7" Then
            g.DrawImage(BoxImages(6),
355, 110, 100, 95)
        Else
            g.DrawImage(BoxImages(13),
355, 110, 100, 95)

        End If
        g.Dispose()
        .....
Graphics
.....

    End Sub

    Private Sub Timer5_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer5.Tick
        .....
Graphics
.....
        Dim g As Graphics =
CreateGraphics()
        g.DrawImage(BoxImages(14), 526,
141, 35, 35)
        .....Floor1
        If number1.Text = "Floor 1" Then
            g.DrawImage(BoxImages(0),
355, 110, 100, 95)
        Else
            g.DrawImage(BoxImages(7),
355, 110, 100, 95)

        End If
        .....Floor2
        If number2.Text = "Floor 2" Then
            g.DrawImage(BoxImages(1),
355, 277, 100, 95)
        Else
            g.DrawImage(BoxImages(8),
355, 277, 100, 95)

        End If
        .....Floor3
        If number3.Text = "Floor 3" Then
            g.DrawImage(BoxImages(2),
355, 444, 100, 95)
        Else
            g.DrawImage(BoxImages(9),
355, 444, 100, 95)

        End If
        .....Floor4
        If number4.Text = "Floor 4" Then
            g.DrawImage(BoxImages(3),
486, 584, 100, 95)
        Else
            g.DrawImage(BoxImages(10),
486, 584, 100, 95)

        End If
        .....Floor5
        If number5.Text = "Floor 5" Then
            g.DrawImage(BoxImages(4),
623, 444, 100, 95)
        Else
            g.DrawImage(BoxImages(11),
623, 444, 100, 95)

        End If
        .....Floor6
        If number6.Text = "Floor 6" Then
            g.DrawImage(BoxImages(5),
623, 277, 100, 95)
        Else
            g.DrawImage(BoxImages(12),
623, 277, 100, 95)

        End If
        .....Floor7
        If number7.Text = "Floor 7" Then
            g.DrawImage(BoxImages(6),
623, 110, 100, 95)
        Else
            g.DrawImage(BoxImages(13),
623, 110, 100, 95)

        End If
        g.Dispose()
        .....
Graphics
.....

    End Sub

    Private Sub Timer6_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer6.Tick
        .....
Graphics
.....
        Dim g As Graphics =
CreateGraphics()

```

```

g.DrawImage(BoxImages(15), 526,
141, 35, 35)
.....Floor1
If number1.Text = "Floor 1" Then
g.DrawImage(BoxImages(0),
623, 110, 100, 95)
Else
g.DrawImage(BoxImages(7),
623, 110, 100, 95)

End If
.....Floor2
If number2.Text = "Floor 2" Then
g.DrawImage(BoxImages(1),
355, 110, 100, 95)
Else
g.DrawImage(BoxImages(8),
355, 110, 100, 95)

End If
.....Floor3
If number3.Text = "Floor 3" Then
g.DrawImage(BoxImages(2),
355, 277, 100, 95)
Else
g.DrawImage(BoxImages(9),
355, 277, 100, 95)

End If
.....Floor4
If number4.Text = "Floor 4" Then
g.DrawImage(BoxImages(3),
355, 444, 100, 95)
Else
g.DrawImage(BoxImages(10),
355, 444, 100, 95)

End If
.....Floor5
If number5.Text = "Floor 5" Then
g.DrawImage(BoxImages(4),
486, 584, 100, 95)
Else
g.DrawImage(BoxImages(11),
486, 584, 100, 95)

End If
.....Floor6
If number6.Text = "Floor 6" Then
g.DrawImage(BoxImages(5),
623, 444, 100, 95)
Else
g.DrawImage(BoxImages(12),
623, 444, 100, 95)

End If
.....Floor7
If number7.Text = "Floor 7" Then

```

```

g.DrawImage(BoxImages(6),
623, 277, 100, 95)
Else
g.DrawImage(BoxImages(13),
623, 277, 100, 95)

End If
g.Dispose()

.....
Graphics
.....

End Sub

Private Sub Timer7_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer7.Tick
.....
Graphics
.....
Dim g As Graphics =
CreateGraphics()
g.DrawImage(BoxImages(14), 526,
141, 35, 35)
.....Floor1
If number1.Text = "Floor 1" Then
g.DrawImage(BoxImages(0),
623, 277, 100, 95)
Else
g.DrawImage(BoxImages(7),
623, 277, 100, 95)

End If
.....Floor2
If number2.Text = "Floor 2" Then
g.DrawImage(BoxImages(1),
623, 110, 100, 95)
Else
g.DrawImage(BoxImages(8),
623, 110, 100, 95)

End If
.....Floor3
If number3.Text = "Floor 3" Then
g.DrawImage(BoxImages(2),
355, 110, 100, 95)
Else
g.DrawImage(BoxImages(9),
355, 110, 100, 95)

End If
.....Floor4
If number4.Text = "Floor 4" Then
g.DrawImage(BoxImages(3),
355, 277, 100, 95)
Else

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยราชภัฏวไลยอลงกรณ์
 ไม่ว่ากรณิใดๆทั้งสิ้น ยกเว้นมีให้เหตุผลบ่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        g.DrawImage(BoxImages(10),
355, 277, 100, 95)

    End If
    .....Floor5
    If number5.Text = "Floor 5" Then
        g.DrawImage(BoxImages(4),
355, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(11),
355, 444, 100, 95)

    End If
    .....Floor6
    If number6.Text = "Floor 6" Then
        g.DrawImage(BoxImages(5),
486, 584, 100, 95)
    Else
        g.DrawImage(BoxImages(12),
486, 584, 100, 95)

    End If
    .....Floor7
    If number7.Text = "Floor 7" Then
        g.DrawImage(BoxImages(6),
623, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(13),
623, 444, 100, 95)

    End If
    g.Dispose()
    .....
Graphics
.....

End Sub

Private Sub Timer8_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer8.Tick
    .....
Graphics
.....

    Dim g As Graphics =
CreateGraphics()
    g.DrawImage(BoxImages(15), 526,
141, 35, 35)
    .....Floor1
    If number1.Text = "Floor 1" Then
        g.DrawImage(BoxImages(0),
623, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(7),
623, 444, 100, 95)

    End If
    .....Floor2
    If number2.Text = "Floor 2" Then
        g.DrawImage(BoxImages(1),
623, 277, 100, 95)
    Else
        g.DrawImage(BoxImages(8),
623, 277, 100, 95)

    End If
    .....Floor3
    If number3.Text = "Floor 3" Then
        g.DrawImage(BoxImages(2),
623, 110, 100, 95)
    Else
        g.DrawImage(BoxImages(9),
623, 110, 100, 95)

    End If
    .....Floor4
    If number4.Text = "Floor 4" Then
        g.DrawImage(BoxImages(3),
355, 110, 100, 95)
    Else
        g.DrawImage(BoxImages(10),
355, 110, 100, 95)

    End If
    .....Floor5
    If number5.Text = "Floor 5" Then
        g.DrawImage(BoxImages(4),
355, 277, 100, 95)
    Else
        g.DrawImage(BoxImages(11),
355, 277, 100, 95)

    End If
    .....Floor6
    If number6.Text = "Floor 6" Then
        g.DrawImage(BoxImages(5),
355, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(12),
355, 444, 100, 95)

    End If
    .....Floor7
    If number7.Text = "Floor 7" Then
        g.DrawImage(BoxImages(6),
486, 584, 100, 95)
    Else
        g.DrawImage(BoxImages(13),
486, 584, 100, 95)

    End If
    g.Dispose()

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.....
Graphics
.....
    End Sub

    Private Sub Timer9_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer9.Tick
.....
Graphics
.....
    Dim g As Graphics =
CreateGraphics()
    g.DrawImage(BoxImages(14), 526,
141, 35, 35)
.....'Floor1
    If number1.Text = "Floor 1" Then
        g.DrawImage(BoxImages(0),
486, 584, 100, 95)
    Else
        g.DrawImage(BoxImages(7),
486, 584, 100, 95)

    End If
.....'Floor2
    If number2.Text = "Floor 2" Then
        g.DrawImage(BoxImages(1),
623, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(8),
623, 444, 100, 95)

    End If
.....'Floor3
    If number3.Text = "Floor 3" Then
        g.DrawImage(BoxImages(2),
623, 277, 100, 95)
    Else
        g.DrawImage(BoxImages(9),
623, 277, 100, 95)

    End If
.....'Floor4
    If number4.Text = "Floor 4" Then
        g.DrawImage(BoxImages(3),
623, 110, 100, 95)
    Else
        g.DrawImage(BoxImages(10),
623, 110, 100, 95)

    End If
.....'Floor5
    If number5.Text = "Floor 5" Then
        g.DrawImage(BoxImages(4),
355, 110, 100, 95)
    Else
        g.DrawImage(BoxImages(11),
355, 110, 100, 95)
    End If
.....'Floor6
    If number6.Text = "Floor 6" Then
        g.DrawImage(BoxImages(5),
355, 277, 100, 95)
    Else
        g.DrawImage(BoxImages(12),
355, 277, 100, 95)

    End If
.....'Floor7
    If number7.Text = "Floor 7" Then
        g.DrawImage(BoxImages(6),
355, 444, 100, 95)
    Else
        g.DrawImage(BoxImages(13),
355, 444, 100, 95)

    End If
    g.Dispose()
.....
Graphics
.....
    End Sub
    Public Function FcsCal(ByVal strText
As String) As String
        Dim fcstemp As String
        Dim fcs As Byte = 0
        Dim i As Byte
        For i = 0 To
(CByte(strText.Length) - 1)
            fcs = fcs Xor
Val(Asc(strText(i)))
        Next
        fcstemp = CStr(Hex(fcs)) + "*"
        Return fcstemp
    End Function

    Private Sub
SerialPort1_DataReceived(ByVal sender As
Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventAr
gs) Handles SerialPort1.DataReceived
        ReceivedText(SerialPort1.ReadExisting())
    End Sub
    Private Sub ReceivedText(ByVal [text]
As String)
        If Me.rtbReceived.InvokeRequired
Then
            Dim x As New
SetTextCallback(AddressOf ReceivedText)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารได้
 ไม่มีการแก้ไขทั้งส่วนอื่นทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Me.Invoke(x, New Object()
    {(text)})
    Else
        Me.rtbReceived.Text &= [text]
        Me.TextBox1.Text &=
Mid(rtbReceived.Text, 20, 2)
    End If
End Sub

```

```

Private Sub Hr1_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Hr1.Tick
    Dim diff As TimeSpan
    Dim timeS As String
    Dim time1 As Date
    Dim time2 As Date
    time1 =
Convert.ToDateTime(Label8.Text)
    time2 =
Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff = time2.Subtract(time1)
    timeS =
FormatNumber(diff.TotalHours, 0)
    Label15.Text = timeS
End Sub

```

```

Private Sub Hr2_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Hr2.Tick
    Dim diff As TimeSpan
    Dim timeS As String
    Dim time1 As Date
    Dim time2 As Date
    time1 =
Convert.ToDateTime(Label9.Text)
    time2 =
Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff = time2.Subtract(time1)
    timeS =
FormatNumber(diff.TotalHours, 0)
    Label16.Text = timeS
End Sub

```

```

Private Sub Hr3_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Hr3.Tick
    Dim diff As TimeSpan
    Dim timeS As String
    Dim time1 As Date
    Dim time2 As Date
    time1 =
Convert.ToDateTime(Label10.Text)
    time2 =
Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff = time2.Subtract(time1)

```

```

    timeS =
FormatNumber(diff.TotalHours, 0)
    Label17.Text = timeS
End Sub

```

```

Private Sub Hr4_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Hr4.Tick
    Dim diff As TimeSpan
    Dim timeS As String
    Dim time1 As Date
    Dim time2 As Date
    time1 =
Convert.ToDateTime(Label11.Text)
    time2 =
Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff = time2.Subtract(time1)
    timeS =
FormatNumber(diff.TotalHours, 0)
    Label18.Text = timeS
End Sub

```

```

Private Sub Hr5_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Hr5.Tick
    Dim diff As TimeSpan
    Dim timeS As String
    Dim time1 As Date
    Dim time2 As Date
    time1 =
Convert.ToDateTime(Label12.Text)
    time2 =
Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff = time2.Subtract(time1)
    timeS =
FormatNumber(diff.TotalHours, 0)
    Label19.Text = timeS
End Sub

```

```

Private Sub Hr6_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Hr6.Tick
    Dim diff As TimeSpan
    Dim timeS As String
    Dim time1 As Date
    Dim time2 As Date
    time1 =
Convert.ToDateTime(Label13.Text)
    time2 =
Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff = time2.Subtract(time1)
    timeS =
FormatNumber(diff.TotalHours, 0)
    Label20.Text = timeS
End Sub

```

```

Private Sub Hr7_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Hr7.Tick
    Dim diff As TimeSpan
    Dim timeS As String
    Dim time1 As Date
    Dim time2 As Date
    time1 =
Convert.ToDateTime(Label14.Text)
    time2 =
Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff = time2.Subtract(time1)
    timeS =
FormatNumber(diff.TotalHours, 0)
    Label22.Text = timeS
End Sub

Private Sub
TextBox2_TextChanged(ByVal sender As
System.Object, ByVal e As
System.EventArgs)

End Sub
End Class

////////// Admin Main Page //////////

Imports System.Data.OleDb

Public Class FrmMain

Private Sub Timer1_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    TimeStatusLabel.Text =
Now.ToLongTimeString.ToString()
End Sub

Private Sub FrmMain_FormClosing(ByVal
sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs
) Handles Me.FormClosing
    If FrmLogin.CBformState.Text = ""
Then
        Me.Dispose()
        FrmLogin.Show()

FrmLogin.TxtAdminUserName.Text = ""

FrmLogin.TxtAdminPassword.Text = ""
FrmLogin.txtUserName.Text = ""

Private Sub Hr7_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Hr7.Tick
    Dim diff As TimeSpan
    Dim timeS As String
    Dim time1 As Date
    Dim time2 As Date
    time1 =
Convert.ToDateTime(Label14.Text)
    time2 =
Convert.ToDateTime(Now.ToLongTimeString.T
oString())
    diff = time2.Subtract(time1)
    timeS =
FormatNumber(diff.TotalHours, 0)
    Label22.Text = timeS
End Sub

Private Sub
TextBox2_TextChanged(ByVal sender As
System.Object, ByVal e As
System.EventArgs)

End Sub
End Class

////////// Admin Main Page //////////

Imports System.Data.OleDb

Public Class FrmMain

Private Sub Timer1_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    TimeStatusLabel.Text =
Now.ToLongTimeString.ToString()
End Sub

Private Sub FrmMain_FormClosing(ByVal
sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs
) Handles Me.FormClosing
    If FrmLogin.CBformState.Text = ""
Then
        Me.Dispose()
        FrmLogin.Show()

FrmLogin.TxtAdminUserName.Text = ""

FrmLogin.TxtAdminPassword.Text = ""
FrmLogin.txtUserName.Text = ""

FrmLogin.txtPassword.Text = ""

Return
End If

Try
    Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
    If cn.State =
ConnectionState.Open Then
        cn.Close()
    End If
    cn.Open()

Dim dr1 As OleDbDataReader
Dim com As New OleDbCommand

If FrmLogin.CBformState.Text
= "User" Then
    com.CommandText = "select
[UserID],[UserName] from userinfo where
UserID = '" & FrmLogin.txtUserName.Text &
""

    com.Connection = cn

dr1 = com.ExecuteReader
If dr1.Read Then
    Dim intResponse As
Integer

intResponse =
MessageBox.Show("Do you really want to
logout?", "Response",
MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
    If intResponse =
MsgBoxResult.No Then
        e.Cancel = True
' Exit Sub
    Else
        MsgBox("Thank
you!! '" &
UCase(dr1("UserName")).ToString() & "' &
vbNewLine & "' & vbNewLine & " God Bless
U...", MsgBoxStyle.Information, "God
Bless...")

cn.Close()
Me.Dispose()
FrmLogin.Show()

FrmLogin.TxtAdminUserName.Text = ""

FrmLogin.TxtAdminPassword.Text = ""
FrmLogin.txtUserName.Text = ""
FrmLogin.txtUserNa

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น มิใช่ผู้เผยแพร่โดยไม่ขออนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FrmLogin.txtPassword.Text = ""
    End If
End If
Else
    If
FrmLogin.CBformState.Text = "Admin" Then
    Dim intResponse As
Integer
        intResponse =
    MessageBox.Show("Do you really want to
logout? ", "Response",
    MessageBoxButtons.YesNo,
    MessageBoxIcon.Question)
    If intResponse =
MsgBoxResult.No Then
        e.Cancel = True
        ' Exit Sub
    Else
        MsgBox("Thank
you, God Bye...",
    MsgBoxStyle.Information, "God Bless
U...")
        cn.Close()
        Me.Dispose()
        FrmLogin.Show()
FrmLogin.TxtAdminUserName.Text = ""
FrmLogin.TxtAdminPassword.Text = ""
FrmLogin.txtUserName.Text = ""
FrmLogin.txtPassword.Text = ""
    End If
    End If
    End If
    Catch ex As Exception
        MsgBox(ex.Message(),
    MsgBoxStyle.Critical, "Error...")
    End Try
End Sub

Private Sub FrmMain_Load(ByVal sender
As Object, ByVal e As System.EventArgs)
Handles Me.Load
    ToolStripStatusLabel1.Text =
"UserName :"
    ToolStripStatusLabel3.Text =
"Date :"
    DateStatusLabel.Text =
Format(Now, "dd/MM/yyyy")
    ToolStripStatusLabel5.Text =
"Time :"

    Try
        If FrmLogin.CBformState.Text
= "User" Then
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()

            Dim dr1 As
OleDbDataReader
            Dim com As New
OleDbCommand
                com.CommandText = "select
[UserID],[UserName] from UserInfo where
UserID = @UName"
                ' UserName
                Dim UName As
OleDbParameter = New
OleDbParameter("@UName",
OleDbType.VarWChar, 30)
                UName.Value =
UCase(FrmLogin.txtUserName.Text.ToString(
))
                com.Parameters.Add(UName)
                com.Connection = cn
                dr1 = com.ExecuteReader
                If dr1.Read Then
                    UserNameStatusLabel.Text =
UCase(dr1("UserName").ToString())
                    ToolStripRegistartion.Visible = False
                End If
                cn.Close()
                Return
            End If
        Else
            UserNameStatusLabel.Text
= "ADMIN"
            ToolStripRegistartion.Visible = True
        End If
    Catch ex As Exception
        MsgBox(ex.Message(),
    MsgBoxStyle.Critical, "Error...")
    End Try
End Sub

```

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันพระจอมเกล้าเจ้าพระยา ใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Me.Close()
End Sub

```

```

Private Sub ToolStripRegistartion_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripRegistartion.Click
    Me.Dispose()
    Registration.Show()
End Sub

```

```

Private Sub ToolStripUserInfo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripUserInfo.Click
    Me.Dispose()
    UserInfo.Show()
End Sub

```

```

Private Sub ToolStripLogout_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripLogout.Click
    Me.Dispose()
    FrmLogin.Show()
End Sub

```

```

Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click
    Me.Dispose()
    Form2.Show()
End Sub

```

```

Private Sub ToolStripButton3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton3.Click
    Me.Dispose()
    FrmFee.Show()
End Sub

```

```

Private Sub DailyToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Me.Dispose()
    Preprint.Show()

```

```
End Sub
```

```

Private Sub MonthlyToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Me.Dispose()
    Preprint.Show()
End Sub

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Me.Dispose()
    test.Show()
End Sub

```

```

Private Sub ToolStripSplitButton1_ButtonClick(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Form3.ShowDialog()
End Sub

```

```

Private Sub ToolStripSplitButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripSplitButton1.Click
    Form3.ShowDialog()
End Sub

```

```
End Class
```

```

///////// Registration Page //////////
Imports System.Data.OleDb
Imports System.IO

```

```
Public Class Registration
```

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Me.Close()
End Sub

```

```

Private Sub Registration_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    Me.Dispose()
    FrmMain.Show()
End Sub

```

```
Sub clear()
```

```

txtUserName.Clear()
dtpDOB.Value = Today
txtPOB.Clear()

```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ไม่ว่ากรณีใดๆ ห้ามเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย

```

txtEmailID.Clear()
txtUserID.Clear()
txtpassword.Clear()
txtConformPass.Clear()
End Sub
Private Function RequiredEntry() As Boolean
    If txtUserName.Text = "" Or
txtPOB.Text = "" Or txtEmailID.Text = ""
Or txtUserID.Text = "" Or
txtpassword.Text = "" Or
txtConformPass.Text = "" Then
        MsgBox("Please enter all
information....", MsgBoxStyle.Critical,
"Attention")
        Return True
    Exit Function
    End If
End Function

Private Sub BBrowse_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles BBrowse.Click
    Dim OpenFileDialog As New
OpenFileDialog()
    Try
        With OpenFileDialog
            .FileName = ""
            .Title = "Photo:"
            .Filter = "Image files:
(*.jpg)|*.jpg|(*.jpeg)|*.jpeg|(*.png)|*.p
ng|(*.Gif)|*.Gif|(*.bmp)|*.bmp| All Files
(*.*)|*.*"
            If .ShowDialog =
Windows.Forms.DialogResult.OK Then
                Me.EmpPic.Image =
System.Drawing.Image.FromFile(.FileName)
            End If
        End With
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error...")
        End Try
    End Sub

Private Sub BRemove_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles BRemove.Click
    Me.EmpPic.Image =
System.Drawing.Image.FromFile(My.Applicat
ion.Info.DirectoryPath &
"\Image\personal.PNG")
End Sub

Private Sub Button2_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Call clear()
End Sub

Private Sub Registration_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    dtpDOB.Value = Today
End Sub

Private Sub button1_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles button1.Click
    If RequiredEntry() = True Then
        Return
    End If
    Try
        Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
        If cn.State =
ConnectionState.Open Then
            cn.Close()
        End If
        cn.Open()
        If txtpassword.Text =
txtConformPass.Text Then
            Dim sSQL As String =
"insert into
UserInfo(UserID,Pass,UserName,DateOfBirth
,PlaceOfBirth,EmailID,UserPic)
values(@d1,@d2,@d3,@d4,@d5,@d6,@d7)"
            Dim cmd As OleDbCommand =
New OleDbCommand(sSQL, cn)
            ' UserID
            Dim UserID As
OleDbParameter = New
OleDbParameter("@d1", OleDbType.VarWChar,
8)
            UserID.Value =
txtUserID.Text.ToString()
            cmd.Parameters.Add(UserID)
            ' Password
            Dim Password As
OleDbParameter = New
OleDbParameter("@d2", OleDbType.VarWChar,
8)
            Password.Value =
txtpassword.Text.ToString()

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เพื่อการศึกษาเท่านั้น ไม่อาจ
 หมายความว่าให้ทั้งสามอีกทั้งสามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

```

cmd.Parameters.Add(Password)
    ' UserName
    Dim UserName As
OleDbParameter = New
OleDbParameter("@d3", OleDbType.VarWChar,
15)
    UserName.Value =
txtUserName.Text.ToString()
cmd.Parameters.Add(UserName)
    ' DateOfBirth
    Dim DOB As OleDbParameter
= New OleDbParameter("@d4",
OleDbType.Date, 15)
    DOB.Value =
dtpDOB.Text.ToString()
    cmd.Parameters.Add(DOB)
    ' PlaceOfBirth
    Dim POB As OleDbParameter
= New OleDbParameter("@d5",
OleDbType.VarWChar, 25)
    POB.Value =
txtPOB.Text.ToString()
    cmd.Parameters.Add(POB)
    ' EmailID
    Dim EmailID As
OleDbParameter = New
OleDbParameter("@d6", OleDbType.VarWChar,
25)
    EmailID.Value =
txtEmailID.Text.ToString()
    cmd.Parameters.Add(EmailID)

Dim MemStream As New
MemoryStream
Dim DataPic_Update As
Byte()

Me.EMP.Pic.Image.Save(MemStream,
Imaging.ImageFormat.Png)
    DataPic_Update =
MemStream.GetBuffer()

MemStream.Read(DataPic_Update, 0,
MemStream.Length)

Dim photo As
OleDbParameter = New
OleDbParameter("@d7", SqlDbType.Image)
photo.Value =
DataPic_Update
cmd.Parameters.Add(photo)

If cmd.ExecuteNonQuery()
Then
    cn.Close()
    MsgBox("New user
registered successfully... ",
MsgBoxStyle.Information, "Record Saved")
Else
    MsgBox("New user
registration failed... ",
MsgBoxStyle.Critical, "Registration
failed")
Return
End If
Else
    MsgBox.Show("Password
& conform password does not match",
"Attention", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
End If

Catch ex As Exception
    MsgBox.Show(ex.Message.ToString(),
"Data Error")
Exit Sub
End Try
End Sub

Private Sub
txtUserName_KeyPress(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs)
Handles txtUserName.KeyPress
    Dim NotAllowed As String =
"1234567890~@#%&*()-_
+={[}]';',!.><?/\"

    'Allowed Letters only
    If e.KeyChar <> ControlChars.Back
= True Then
        If
NotAllowed.IndexOf(e.KeyChar) = -1 =
False Then
            e.Handled = True
        End If
    End If
End Sub
End Sub
    ' image content

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุผลเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub txtPOB_KeyPress(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs)
Handles txtPOB.KeyPress
    Dim NotAllowed As String =
"1234567890~`@#%&*( )_-
+={[ ]};:;,!.><?/\\"

```

```

'Allowed Letters only
If e.KeyChar <> ControlChars.Back
= True Then
    If
NotAllowed.IndexOf(e.KeyChar) = -1 =
False Then
        e.Handled = True
    End If
End If
End Sub

```

```

Private Sub txtUserID_KeyPress(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs)
Handles txtUserID.KeyPress
    Dim NotAllowed As String =
"~`@%&+={[ ]}(!:;, '><?/\-.#+"
'Allowed letters numbers and ( _
$ *)

```

```

If e.KeyChar <> ControlChars.Back
= True Then
    If
NotAllowed.IndexOf(e.KeyChar) = -1 =
False Then
        e.Handled = True
    End If
End If
End Sub

```

```

Private Sub
txtpassword_KeyPress(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs)
Handles txtpassword.KeyPress
    Dim NotAllowed As String =
"~`@%&+={[ ]}(!:;, '><?/\-.#+"
'Allowed letters numbers and ( _
$ *)

```

```

If e.KeyChar <> ControlChars.Back
= True Then
    If
NotAllowed.IndexOf(e.KeyChar) = -1 =
False Then
        e.Handled = True
    End If
End If
End Sub

```

```

Private Sub
txtConformPass_KeyPress(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs)
Handles txtConformPass.KeyPress
    Dim NotAllowed As String =
"~`@%&+={[ ]}(!:;, '><?/\-.#+"
'Allowed letters numbers and ( _
$ *)

```

```

If e.KeyChar <> ControlChars.Back
= True Then
    If
NotAllowed.IndexOf(e.KeyChar) = -1 =
False Then
        e.Handled = True
    End If
End If
End Sub

```

```

Private Sub Button4_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
Me.Dispose()
FrmRegistrationDetails.Show()
End Sub

```

```

End Class
////////// Report Page //////////

```

```

Imports System.Data.OleDb

Public Class Form2
    Dim cn As New
System.Data.OleDb.OleDbConnection
    Dim cmd As
System.Data.OleDb.OleDbCommand
    Dim myDA As
System.Data.OleDb.OleDbDataAdapter
    Dim myDataSet As System.Data.DataSet

```

```

Sub DataSetFill()
    Try
        cn.ConnectionString =
"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;"
        If cn.State =
System.Data.ConnectionState.Closed Then
            cn.Open()
        End If
        cmd = New
System.Data.OleDb.OleDbCommand("SELECT
(CarNumber)as [Car Number]; (Barcode)as

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีกรณีการใช้งาน

```

[Barcode],[TimeIn]as [Time
In],[TimeOut]as[Time Out],[Roof]as
[Floor],[DateIn] as [Date
In],[TotalHr]as [Total Hr],[Charges] as
[Charges] FROM PayInfo ", cn)
myDA = New
System.Data.OleDb.OleDbDataAdapter(cmd)
myDataSet = New
System.Data.DataSet
myDA.Fill(myDataSet,
"PayInfo")
DataGridView1.DataSource =
myDataSet.Tables("PayInfo").DefaultView
cn.Close()
Catch ex As Exception
MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
End Try
End Sub

Private Sub Form2_FormClosing(ByVal
sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs
) Handles Me.FormClosing
Me.Dispose()
FrmMain.Show()
End Sub

Private Sub Form2_Load(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Call DataSetFill()
End Sub

Private Sub Delete1_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles Delete1.Click
Try
If MsgBox.Show("Do you
really want to delete the record?",
"Response", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) =
Windows.Forms.DialogResult.Yes Then
Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
If cn.State =
ConnectionState.Open Then
cn.Close()
End If
cn.Open()
Dim temp As Integer = 0

```

```

Dim cq As String =
"delete from Payinfo where
CarNumber=@DELETE;"

cmd = New
OleDbCommand(cq)
cmd.Connection = cn

cmd.Parameters.Add(New
OleDbParameter("@DELETE",
System.Data.OleDb.OleDbType.VarChar, 6,
"CarNumber"))

cmd.Parameters("@DELETE").Value =
DataGridView1.SelectedCells(0).Value.ToSt
ring()

temp =
cmd.ExecuteNonQuery()
If temp > 0 Then
cn.Close()
Call DataSetFill()
MessageBox.Show("Successfully deleted",
"Record", MessageBoxButtons.OK,
MessageBoxIcon.Information)
Else
Call DataSetFill()
MessageBox.Show("No
record found", "Sorry",
MessageBoxButtons.OK,
MessageBoxIcon.Error)
End If

End If
Catch ex As Exception
MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
End Try
End Sub

End Class
////////// Print Page //////////
Public Class Form3

Private Sub Form3_Load(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
'TODO: This line of code loads
data into the 'TestingDataSet.PayInfo'
table. You can move, or remove it, as
needed.

Me.PayInfoTableAdapter.Fill(Me.TestingDat
aSet.PayInfo)

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้ภายในเท่านั้น ไม่สามารถเผยแพร่หรือจำหน่ายโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Me.ReportViewer1.RefreshReport()
    End Sub
End Class

```

```

////////// Fee Page //////////

```

```

Imports System.Data.OleDb
Imports System.IO

```

```

Public Class FrmFee

```

```

    Dim cn As New
    System.Data.OleDb.OleDbConnection
    Dim cmd As
    System.Data.OleDb.OleDbCommand
    Dim myDA As
    System.Data.OleDb.OleDbDataAdapter
    Dim myDataSet As System.Data.DataSet

```

```

    Private Function RequiredEntry() As Boolean

```

```

        If TextBox01.Text = "" Or
        TextBox02.Text = "" Or TextBox03.Text =
        "" Or TextBox04.Text = "" Then

```

```

            MsgBox("Please enter all
            charge", MsgBoxStyle.Critical,
            "Attention")

```

```

            Return True
            Exit Function

```

```

        End If

```

```

    End Function

```

```

    Private Sub FrmFee_FormClosing(ByVal
    sender As Object, ByVal e As
    System.Windows.Forms.FormClosingEventArgs
    ) Handles Me.FormClosing

```

```

        Me.Dispose()
        FrmMain.Show()

```

```

    End Sub

```

```

    Private Sub FrmFee_Load(ByVal sender
    As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load

```

```

        TextBox01.Enabled = False
        TextBox02.Enabled = False
        TextBox03.Enabled = False
        TextBox04.Enabled = False

```

```

    Try

```

```

        Dim cn As New

```

```

        OleDbConnection("Provider=Microsoft.ACE.O
        LEDB.12.0;Data Source=" &
        My.Application.Info.DirectoryPath.ToStrin
        g() & "\Backup\testing.Accdb;Persist
        Security Info=False;")

```

```

        If cn.State =
        ConnectionState.Open Then

```

```

            cn.Close()

```

```

    End If
    cn.Open()

```

```

    Dim dr1 As OleDbDataReader
    Dim com1 As New OleDbCommand

```

```

    If TextBox01.Enabled = False

```

```

    Then

```

```

        com1.CommandText =

```

```

        "select [ชั่วโมงแรก],[ชั่วโมงที่2ถึง3],[ชั่วโมงที่4ขึ้นไป],[วอ
        เคียน] from FeeInfo where number = 1"

```

```

        com1.Connection = cn

```

```

        dr1 = com1.ExecuteReader

```

```

        If dr1.Read Then

```

```

            TextBox01.Text = "" &

```

```

            UCase(dr1(0)).ToString() & ""

```

```

            TextBox02.Text = "" &

```

```

            UCase(dr1(1)).ToString() & ""

```

```

            TextBox03.Text = "" &

```

```

            UCase(dr1(2)).ToString() & ""

```

```

            TextBox04.Text = "" &

```

```

            UCase(dr1(3)).ToString() & ""

```

```

        End If

```

```

        cn.Close()

```

```

        dr1.Close()

```

```

        Return

```

```

    End If

```

```

    Catch ex As Exception

```

```

        MsgBox(ex.Message(),

```

```

        MsgBoxStyle.Critical, "Error...")

```

```

    End Try

```

```

    End Sub

```

```

    Private Sub Button01_Click(ByVal
    sender As System.Object, ByVal e As
    System.EventArgs) Handles Button01.Click

```

```

        TextBox01.Enabled = True

```

```

        TextBox02.Enabled = True

```

```

        TextBox03.Enabled = True

```

```

        TextBox04.Enabled = True

```

```

    End Sub

```

```

    Private Sub Button02_Click(ByVal
    sender As System.Object, ByVal e As
    System.EventArgs) Handles Button02.Click

```

```

    Try

```

```

        Dim cn As New

```

```

        OleDbConnection("Provider=Microsoft.ACE.O
        LEDB.12.0;Data Source=" &

```

```

        My.Application.Info.DirectoryPath.ToStrin

```

```

        g() & "\Backup\testing.Accdb;Persist

```

```

        Security Info=False;")

```

```

        If cn.State =

```

```

        ConnectionState.Open Then

```

```

            cn.Close()

```

```

        End If

```

```

    End Sub

```

```

End Class

```

```


```

```


```

```


```

```

g() & "\\BackUp\\testing.Accdb;Persist
Security Info=False;")
    If cn.State =
ConnectionState.Open Then
        cn.Close()
    End If
    cn.Open()
    Dim dr1 As OleDbDataReader
    Dim com As New OleDbCommand
    com.CommandText = "select
[number] from FeeInfo where number =
@Number"

    ' Number
    Dim Number As OleDbParameter
= New OleDbParameter("@Number",
OleDbType.VarWChar, 20)
    Number.Value = 1
    com.Parameters.Add(Number)
    com.Connection = cn
    dr1 = com.ExecuteReader

    If dr1.Read Then
        If UCase(dr1("number")) =
1 Then
            Dim sSQL As String =
"Update FeeInfo Set ชั่วโมงแรก=@d1, ชั่วโมงที่2ถึง
3=@d2, ชั่วโมงที่4ขึ้นไป=@d3, รายเดือน=@d4 where number
= 1 "
            Dim cmd As
OleDbCommand = New OleDbCommand(sSQL, cn)

            ' ชั่วโมงแรก
            Dim Hrone As
OleDbParameter = New
OleDbParameter("@d1", OleDbType.VarWChar,
15)
            Hrone.Value =
TextBox01.Text.ToString()

            cmd.Parameters.Add(Hrone)

            ' ชั่วโมงที่2ถึง3
            Dim Hrtwo As
OleDbParameter = New
OleDbParameter("@d2", OleDbType.VarWChar,
15)
            Hrtwo.Value =
TextBox02.Text.ToString()

            cmd.Parameters.Add(Hrtwo)

            ' ชั่วโมงที่4ขึ้นไป
            Dim Hrfour As
OleDbParameter = New
OleDbParameter("@d3", OleDbType.VarWChar,
15)
            Hrfour.Value =
TextBox03.Text.ToString()

            cmd.Parameters.Add(Hrfour)

            ' รายเดือน
            Dim Monthly As
OleDbParameter = New
OleDbParameter("@d4", OleDbType.VarWChar,
15)
            Monthly.Value =
TextBox04.Text.ToString()

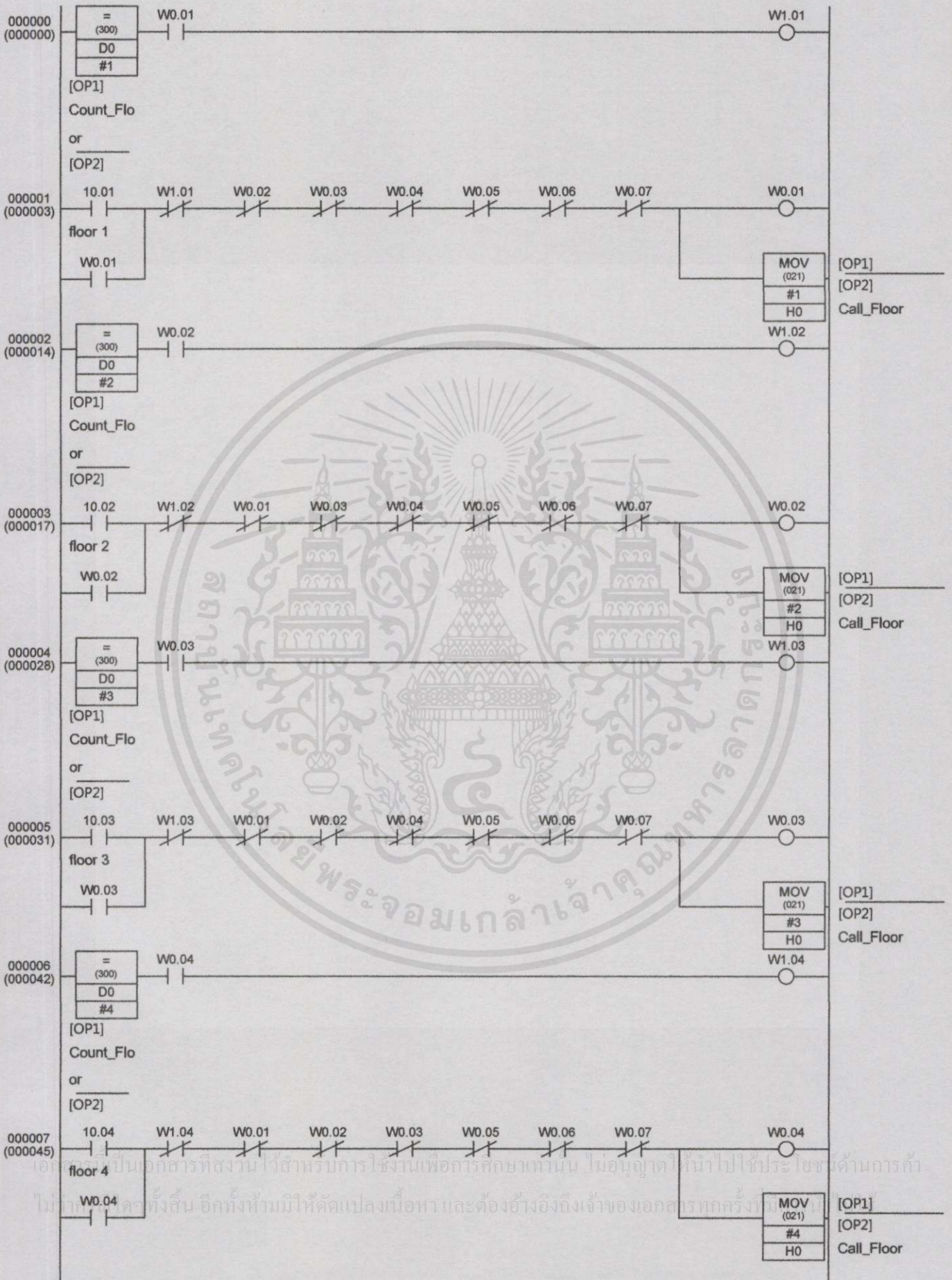
            cmd.Parameters.Add(Monthly)

            If
cmd.ExecuteNonQuery() Then
                cn.Close()
                MsgBox("
successfully... ",
MsgBoxStyle.Information, "Record Saved")
                TextBox01.Enabled
                TextBox02.Enabled
                TextBox03.Enabled
                TextBox04.Enabled
            Else
                MsgBox("failed...
", MsgBoxStyle.Critical, "Registration
failed")
                Return
            End If
        Else
            MsgBox("failed... ",
MsgBoxStyle.Critical, "Registration
failed")
        End If
    End If
    Catch ex As Exception
        MsgBox.Show(ex.Message.ToString(),
"Data Error")
    Exit Sub
    End Try
End Sub

Private Sub GroupBox2_Enter(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
GroupBox2.Enter, GroupBox3.Enter
End Sub
End Clas

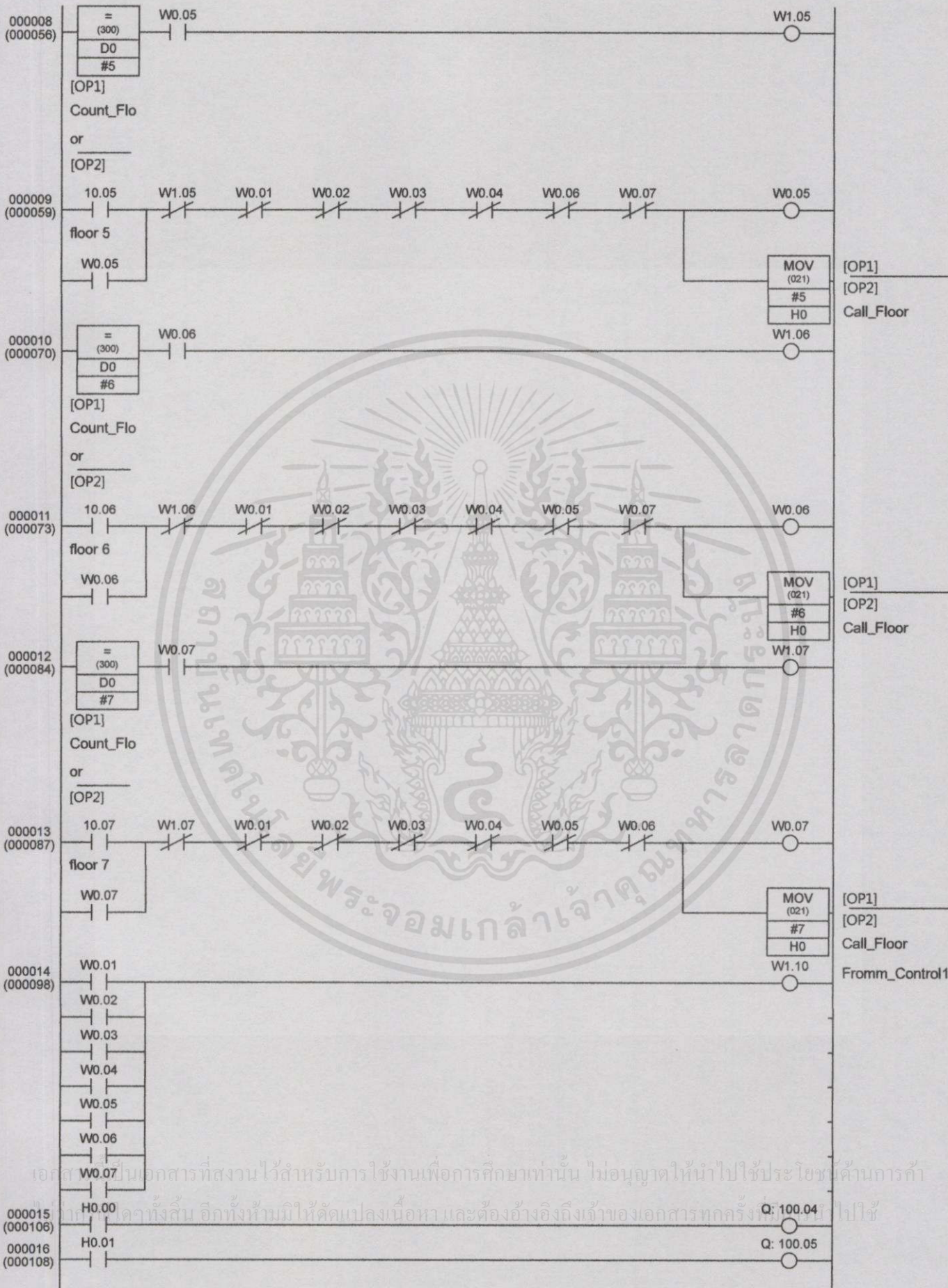
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำออกจำหน่ายหรือเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัยได้ หากมีผู้ใดฝ่าฝืน กรุณาแจ้งให้ทางมหาวิทยาลัยทราบ เพื่อดำเนินการตามกฎหมายต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ทำกำไรจากหนังสือพิมพ์ อื่นๆ ห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

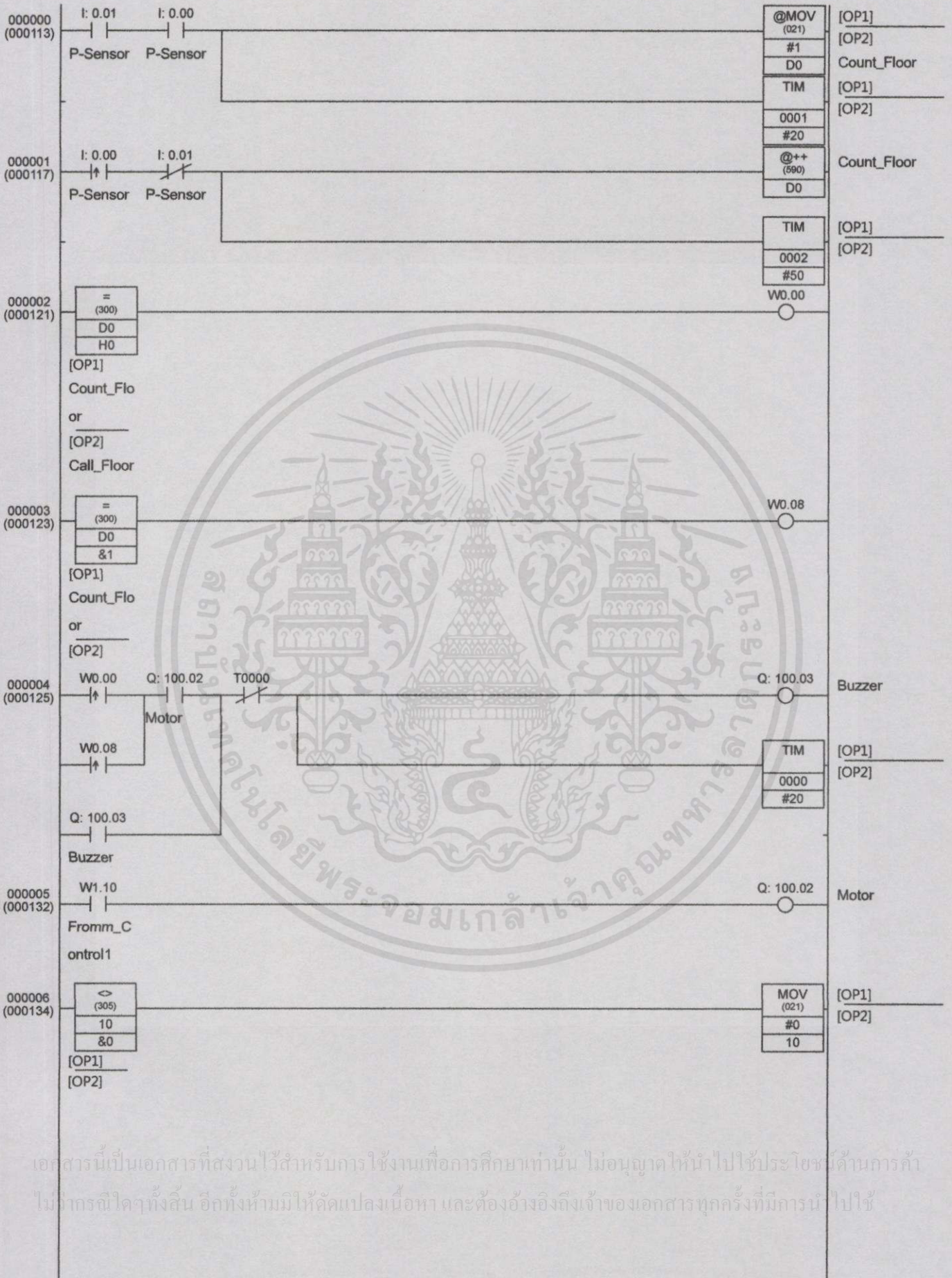


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

หากมีข้อผิดพลาดใดๆ กรุณาแจ้งให้ทราบทันที และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ทำกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ทำการตีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP1L

CPU Units and Expansion Units

When it comes to controllers for compact machines, Omron's new CP1L series offers the compactness of a micro-PLC with the capability of a modular PLC.

But this new and exciting range is not only compact, it is scaleable, has a faster processing speed than other controllers and is in a class of its own when it comes to price/performance. Naturally, it is compatible with all other devices in the Omron PLC line up.

- 4 high-speed encoder inputs and 2 high-speed pulse outputs
- CPUs with AC or DC supply and 14, 20, 30 or 40 I/O built-in
- Instruction set compatible with CP1H-, CJ1-, and CS1 series PLC
- Optional RS232C and RS-422A/485 serial ports
- USB programming port
- Scaleable with a wide range of I/O units (maximum up to 160 I/O points)
- Motion functionality
- One and the same software as other Omron controllers



CPU Unit Specification

Item	Type	AC power supply models	DC power supply models
	Model	CP1L-□□□-A	CP1L-□□□-D
Power supply		100 to 240 VAC 50/60 Hz	24 VDC
Operating voltage range		85 to 264 VAC	20.4 to 26.4 VDC
Power consumption		50 VA max. (CP1L-M40/M30DR-A) (See next page.) 30 VA max. (CP1L-L20/L14DR-A)	20 W max. (CP1L-M40/M30□□-D) (See next page.) 13 W max. (CP1L-L20/L14□□-D)
Inrush current (See note.)		100 to 120 VAC inputs: 20 A max. (for cold start at room temperature) 8 ms max. 200 to 240 VAC inputs: 40 A max. (for cold start at room temperature), 8 ms max.	30 A max. (for cold start at room temperature) 20 ms max.
External power supply		300 mA at 24 VDC (CP1L-M30/M40) 200 mA at 24 VDC (CP1L-L14/L20)	None
Insulation resistance		20 MΩ min. (at 500 VDC) between the external AC terminals and GR terminals	No insulation between primary and secondary for DC power supply
Dielectric strength		2,300 VAC at 50/60 Hz for 1 min between the external AC and GR terminals, leakage current: 5 mA max.	No insulation between primary and secondary for DC power supply
Noise immunity		Conforms to IEC 61000-4-4. 2 kV (power supply line)	
Vibration resistance		Conforms to JIS C0040. 10 to 57 Hz, 0.075-mm amplitude, 57 to 150 Hz, acceleration: 9.8 m/s ² in X, Y, and Z directions for 80 minutes each. Sweep time: 8 minutes x 10 sweeps = total time of 80 minutes)	
Shock resistance		Conforms to JIS C0041. 147 m/s ² three times each in X, Y, and Z directions	
Ambient operating temperature		0 to 55°C	
Ambient humidity		10% to 90% (with no condensation)	
Ambient operating environment		No corrosive gas	
Ambient storage temperature		-20 to 75°C (Excluding battery.)	
Power holding time		10 ms min.	2 ms min.

Note: The above values are for a cold start at room temperature for an AC power supply, and for a cold start for a DC power supply.

- A thermistor (with low-temperature current suppression characteristics) is used in the inrush current control circuitry for the AC power supply. The thermistor will not be sufficiently cooled if the ambient temperature is high or if a hot start is performed when the power supply has been OFF for only a short time. In those cases the inrush current values may be higher (as much as two times higher) than those shown above. Always allow for this when selecting fuses and breakers for external circuits.
- A capacitor charge-type delay circuit is used in the inrush current control circuitry for the DC power supply. The capacitor will not be charged if a hot start is performed when the power supply has been OFF for only a short time, so in those cases the inrush current values may be higher (as much as two times higher) than those shown above.

Current Consumption

The power consumption shown on page 1 is the maximum power consumption. To obtain the correct power consumption for the system configuration, calculate the power consumption for the external power supply from the current consumption given below for the CPU Unit, Expansion Units, and Expansion I/O Units.

CPU Units

Model	Current consumption		External power supply
	5 VDC	24 VDC	24 VDC
CP1L-M40DR-A	0.22 A	0.08 A	0.3 A max.
CP1L-M40DR-D	0.22 A	0.08 A	---
CP1L-M40DT-D	0.31 A	0.03 A	---
CP1L-M40DT1-D	0.31 A	0.03 A	---
CP1L-M30DR-A	0.21 A	0.07 A	0.3 A max.
CP1L-M30DR-D	0.21 A	0.07 A	---
CP1L-M30DT-D	0.28A	0.03 A	---
CP1L-M30DT1-D	0.28 A	0.03 A	---
CP1L-L20DR-A	0.20 A	0.05 A	0.2 A max.
CP1L-L20DR-D	0.20A	0.05 A	---
CP1L-L20DT-D	0.24 A	0.03 A	---
CP1L-L20DT1-D	0.24 A	0.03 A	---
CP1L-L14DR-A	0.18 A	0.04 A	0.2 A max.
CP1L-L14DR-D	0.18 A	0.04 A	---
CP1L-L14DT-D	0.21 A	0.03 A	---
CP1L-L14DT1-D	0.21 A	0.03A	---

- Note 1.** The current consumption of the CP1W-ME05M Memory Cassette and the CP1W-CIF01/CIF11 Option Boards are included in the current consumption of the CPU Unit.
- 2.** CPU Units with DC power do not provide an external power supply.
- 3.** The current consumptions given in the following table must be added to the current consumption of the CPU Unit if an Expansion Unit or Expansion I/O Unit is connected.
- 4.** The external power supply cannot be used if an Expansion Unit or Expansion I/O Unit is connected to a CPU Unit with 14 or 20 I/O points.

Expansion Units and Expansion I/O Units

Unit name	Model	Current consumption		
		5 VDC	24 VDC	
Expansion I/O Units	40 I/O points	CP1W-40EDR	0.080 A	0.090 A
	24 inputs	CP1W-40EDT	0.160 A	---
	16 outputs	CP1W-40EDT1	---	---
	20 I/O points	CP1W-20EDR1	0.103 A	0.044 A
	12 inputs	CP1W-20EDT	0.130 A	---
	8 outputs	CP1W-20EDT1	---	---
	16 outputs	CP1W-16ER	0.042 A	0.090 A
	8 inputs	CP1W-8ED	0.018 A	---
	8 outputs	CP1W-8ER	0.026 A	0.044 A
		CP1W-8ET	0.075 A	---
	CP1W-8ET1	---	---	
Analog Input Unit	4 inputs	CP1W-AD041	0.080 A	0.120 A
Analog Output Unit	4 outputs	CP1W-DA041	0.080 A	0.120 A
	2 outputs	CP1W-DA021	0.095 A	0.040 A
Analog I/O Unit	2 inputs and 1 output	CP1W-MAD11	0.083 A	0.110 A
Temperature Sensor Units	K or J thermocouple inputs	CP1W-TS001	0.040 A	0.059 A
		CP1W-TS002	---	---
	Pt or JPt platinum resistance thermometer inputs	CP1W-TS101	0.054 A	0.073 A
CompoBus/S I/O Link Unit	8 inputs and 8 outputs	CP1W-SRT21	0.029 A	---

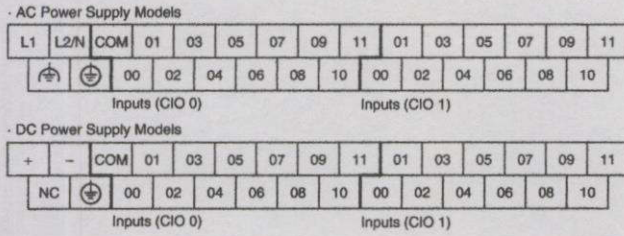
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU Units

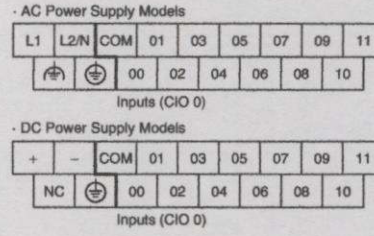
Type	CP1L-M40 (40 points)	CP1L-M30 (30 points)	CP1L-L20 (20 points)	CP1L-L14 (14 points)	
Item	Models	CP1L-M40□□□□	CP1L-M30□□□□	CP1L-L20□□□□	CP1L-L14□□□□
Control method	Stored program method				
I/O control method	Cyclic scan with immediate refreshing				
Program language	Ladder diagram				
Function blocks	Maximum number of function block definitions: 128 Maximum number of instances: 256 Languages usable in function block definitions: Ladder diagrams, structured text (ST)				
Instruction length	1 to 7 steps per instruction				
Instructions	Approx. 500 (function codes: 3 digits)				
Instruction execution time	Basic instructions: 0.55 μs min. Special instructions: 4.1 μs min.				
Common processing time	0.4 ms				
Program capacity	10K steps		5K steps		
Number of tasks	288 (32 cyclic tasks and 256 interrupt tasks)				
Scheduled interrupt tasks	1 (interrupt task No. 2, fixed)				
	Input interrupt tasks		4 (interrupt task No. 140 to 143, fixed)		
(Interrupt tasks can also be specified and executed for high-speed counter interrupts and executed.)					
Maximum subroutine number	256				
Maximum jump number	256				
I/O areas	Input bits	24: CIO 0.00 to CIO 0.11 and CIO 1.00 to CIO 1.11	18: CIO 0.00 to CIO 0.11 and CIO 1.00 to CIO 1.05	12: CIO 0.00 to CIO 0.11	8: CIO 0.00 to CIO 0.07
	Output bits	16: CIO 100.00 to CIO 100.07 and CIO 101.00 to CIO 101.07	12: CIO 100.00 to CIO 100.07 and CIO 101.00 to CIO 101.03	8: CIO 100.00 to CIO 100.07	6: CIO 100.00 to CIO 100.05
	1:1 Link Area	1,024 bits (64 words): CIO 3000.00 to CIO 3063.15 (CIO 3000 to CIO 3063)			
	Serial PLC Link Area	1,440 bits (90 words): CIO 3100.00 to CIO 3189.15 (CIO 3100 to CIO 3189)			
Work bits	8,192 bits (512 words): W000.00 to W511.15 (W0 to W511) CIO Area: 37,504 bits (2,344 words): CIO 3800.00 to CIO 6143.15 (CIO 3800 to CIO 6143)				
TR Area	16 bits: TR0 to TR15				
Holding Area	8,192 bits (512 words): H0.00 to H511.15 (H0 to H511)				
AR Area	Read-only (Write-prohibited): 7168 bits (448 words): A0.00 to A447.15 (A0 to A447) Read/Write: 8192 bits (512 words): A448.00 to A959.15 (A448 to A959)				
Timers	4,096 bits: T0 to T4095				
Counters	4,096 bits: C0 to C4095				
DM Area	32 Kwords: D0 to D32767		10 Kwords: D0 to D9999, D32000 to D32767		
Data Register Area	16 registers (16 bits): DR0 to DR15				
Index Register Area	16 registers (32 bits): IR0 to IR15				
Task Flag Area	32 flags (32 bits): TK0000 to TK0031				
Trace Memory	4,000 words (500 samples for the trace data maximum of 31 bits and 6 words.)				
Memory Cassette	A special Memory Cassette (CP1W-ME05M) can be mounted. Note: Can be used for program backups and auto-booting.				
Clock function	Supported. Accuracy (monthly deviation): -4.5 min to -0.5 min (ambient temperature: 55°C), -2.0 min to +2.0 min (ambient temperature: 25°C), -2.5 min to +1.5 min (ambient temperature: 0°C)				
Communications functions	One built-in peripheral port (USB 1.1): For connecting Support Software only.				
	A maximum of two Serial Communications Option Boards can be mounted.		A maximum of one Serial Communications Option Board can be mounted.		
Memory backup	Flash memory: User programs, parameters (such as the PLC Setup), comment data, and the entire DM Area can be saved to flash memory as initial values. Battery backup: The Holding Area, DM Area, and counter values (flags, PV) are backed up by a battery.				
Battery service life	5 years at 25°C. (Use the replacement battery within two years of manufacture.)				
Built-in input terminals	40 (24 inputs, 16 outputs)	30 (18 inputs, 12 outputs)	20 (12 inputs, 8 outputs)	14 (8 inputs, 6 outputs)	
Number of connectable Expansion Units and Expansion I/O Units	CP-series Expansion Unit and Expansion I/O Units: 3 max.		CP-series Expansion Units and Expansion I/O Units: 1 max.		
Max. number of I/O points	160 (40 built in + 40 per Expansion (I/O) Unit × 3 Units)	150 (30 built in + 40 per Expansion (I/O) Unit × 3 Units)	60 (20 built in + 40 per Expansion (I/O) Unit × 1 Unit)	54 (14 built in + 40 per Expansion (I/O) Unit × 1 Unit)	
Interrupt inputs	6 inputs (Response time: 0.3 ms)			4 inputs (Response time: 0.3 ms)	
Interrupt inputs counter mode	6 inputs (Response frequency: 5 kHz max. for all interrupt inputs), 16 bits Up or down counters			4 inputs (Response frequency: 5 kHz max. for all interrupt inputs), 16 bits Up or down counters	
Quick-response inputs	6 points (Min. input pulse width: 50 μs max.)			4 points (Min. input pulse width: 50 μs max.)	
Scheduled interrupts	1				
High-speed counters	4 counters, 2 axes (24-VDC input) 4 inputs: Differential phases (4x), 50 kHz or Single-phase (pulse plus direction, up/down, increment), 100 kHz Value range: 32 bits, Linear mode or ring mode Interrupts: Target value comparison or range comparison				
Pulse outputs (models with transistor outputs only)	Pulse outputs	Trapezoidal or S-curve acceleration and deceleration (Duty ratio: 50% fixed) 2 outputs, 1 Hz to 100 kHz (CCW/CW or pulse plus direction)			
	PWM outputs	Duty ratio: 0.0% to 100.0% (specified in increments of 0.1% or 1%) 2 outputs, 0.1 to 6553.5 Hz or 1 to 32,800 Hz (Accuracy: ±5% at 1 kHz)			
Analog control	1 (Setting range: 0 to 255)				
External analog input	1 input (Resolution: 1/256, Input range: 0 to 10 V). Not isolated.				

Input Terminal Block Arrangement (Top Block)

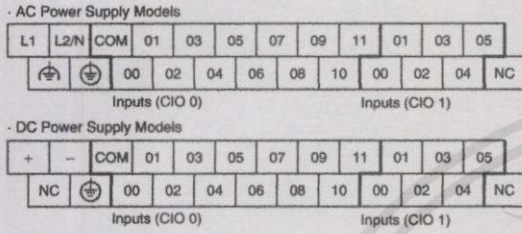
CP1L-M40



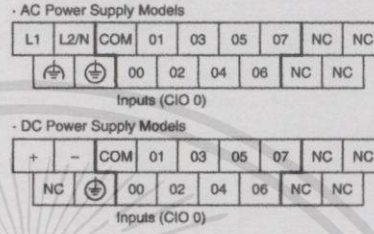
CP1L-L20



CP1L-M30



CP1L-L14



Built-in Input Area

CPU Units

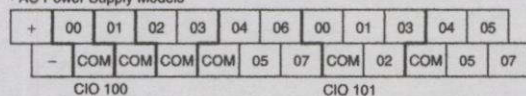
Number of inputs	Input terminal block		Input operation			High-speed counter operation		Origin search
	Word	Bit	Normal Inputs	Interrupt inputs	Quick-response inputs	Operation settings • High-speed counters enabled • Phase-Z signal reset		Origin searches enabled for pulse outputs 0 and 1
14	CIO 0	00	Normal input 0	---	---	High-speed counter 0 (increment)	High-speed counter 0 (phase-A, increment, or count input)	---
		01	Normal input 1	---	---	High-speed counter 1 (increment)	High-speed counter 0 (phase-B, decrement, or count input)	---
		02	Normal input 2	---	---	High-speed counter 2 (increment)	High-speed counter 1 (phase-A, increment, or count input)	Pulse output 0: Origin proximity input signal (See note 1.)
		03	Normal input 3	---	---	High-speed counter 3 (increment)	High-speed counter 1 (phase-B, decrement, or count input)	Pulse output 01 Origin proximity input signal (See note 1.)
		04	Normal input 4	Interrupt input 0	Quick-response input 0	Counter 0, phase-Z/reset input	High-speed counter 0 (phase-Z/reset)	---
		05	Normal input 5	Interrupt input 1	Quick-response input 1	Counter 1, phase-Z/reset input	High-speed counter 1 (phase-Z/reset)	---
		06	Normal input 6	Interrupt input 2	Quick-response input 2	Counter 2, phase-Z/reset input		Pulse output 0: Origin input signal
		07	Normal input 7	Interrupt input 3	Quick-response input 3	Counter 3, phase-Z/reset input		Pulse output 1: Origin input signal
		08	Normal input 8	Interrupt input 4	Quick-response input 4	---		---
		09	Normal input 9	Interrupt input 5	Quick-response input 5	---		---
		10	Normal input 10	---	---	---		Pulse output 0: Origin proximity input signal (See note 2.)
20	CIO 0	11	Normal input 11	---	---	---		Pulse output 1: Origin proximity input signal (See note 2.)
30	CIO 1	00	Normal input 12	---	---	---		---
		01	Normal input 13	---	---	---		---
		02	Normal input 14	---	---	---		---
		03	Normal input 15	---	---	---		---
		04	Normal input 16	---	---	---		---
		05	Normal input 17	---	---	---		---
40	CIO 1	06	Normal input 18	---	---	---		---
		07	Normal input 19	---	---	---		---
		08	Normal input 20	---	---	---		---
		09	Normal input 21	---	---	---		---
		10	Normal input 22	---	---	---		---
		11	Normal input 23	---	---	---		---

Note 1. The origin proximity input signals for CPU Units with 14 points are bits 02 and 03 of CIO 0.
 2. The origin proximity input signals for CPU Units with 20 points are bits 10 and 11 of CIO 0.

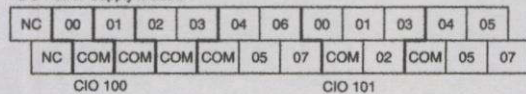
Output Terminal Block Arrangement (Bottom Block)

CP1L-M40

· AC Power Supply Models

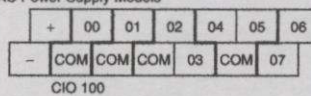


· DC Power Supply Models

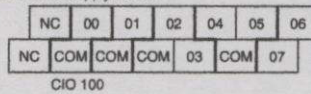


CP1L-L20

· AC Power Supply Models

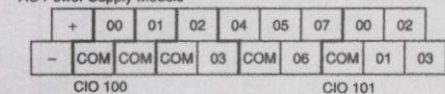


· DC Power Supply Models

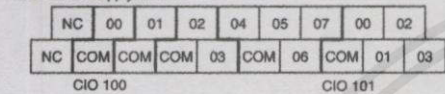


CP1L-M30

· AC Power Supply Models

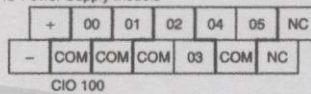


· DC Power Supply Models

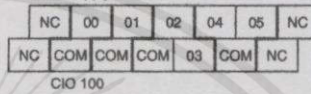


CP1L-L14

· AC Power Supply Models



· DC Power Supply Models



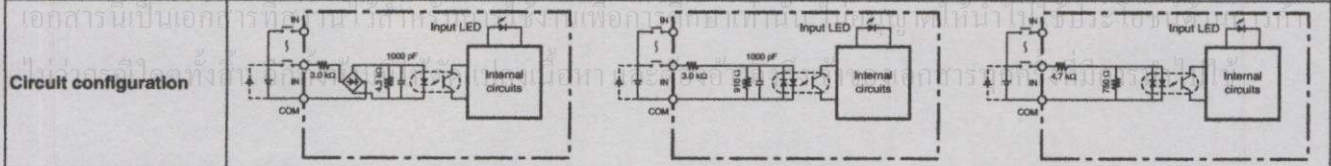
Built-in Output Area

CPU Units

Number of outputs	Output Terminal Block	When the instructions to the right are not executed	When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed		When the origin search function is set to be used in the PLC Setup, and an origin search is executed by the ORG instruction	When the PWM instruction is executed		
			Word	Bit	Normal output	Fixed duty ratio pulse output		Variable duty ratio pulse output
						CW/CCW	Pulse plus direction	
14	C100	00	Normal output 0	Pulse output 0 (CW)	Pulse output 0 (pulse)	---		
		01	Normal output 1	Pulse output 0 (CCW)	Pulse output 0 (direction)	---		
		02	Normal output 2	Pulse output 1 (CW)	Pulse output 1 (pulse)	---		
		03	Normal output 3	Pulse output 1 (CCW)	Pulse output 1 (direction)	---		
		04	Normal output 4	---	---	Origin search 0 (Error counter reset output)	---	
		05	Normal output 5	---	---	Origin search 1 (Error counter reset output)	---	
		06	Normal output 6	---	---	---	---	
20	C100	07	Normal output 7	---	---	---		
		08	Normal output 8	---	---	---		
30	C101	01	Normal output 9	---	---	---		
		02	Normal output 10	---	---	---		
		03	Normal output 11	---	---	---		
		04	Normal output 12	---	---	---		
		05	Normal output 13	---	---	---		
		06	Normal output 14	---	---	---		
		07	Normal output 15	---	---	---		

Input Specifications

ITEM	Specifications		
	High-speed counter inputs (phases A and B)	Interrupt inputs and quick-response inputs	Normal inputs
CP1L	CIO 0.00 to CIO 0.03	CIO 0.04 to CIO 0.09	CIO 0.10, CIO 0.11 and CIO 1.00 to CIO 1.11
Input voltage	24 VDC +10%/−15%		
Applicable sensors	2-wire sensors		
Input impedance	3.0 kΩ		4.7 kΩ
Input current	7.5 mA typical		5 mA typical
ON voltage	17.0 VDC min.		14.4 VDC min.
OFF voltage/current	1 mA max. at 5.0 VDC		
ON delay	2.5 μs max.	50 μs max.	1 ms max.
OFF delay	2.5 μs max.	50 μs max.	1 ms max.



High-speed Counter Function Input Specifications

Input bits: CIO 0.00 to CIO 0.03

Item	Specifications
ON/OFF delay	<ul style="list-style-type: none"> Pulse plus direction input mode Increment mode Up/down input mode <ul style="list-style-type: none"> Differential phase input mode

Interrupt Input Counter Mode

Input bits: CIO 0.04 to CIO 0.09

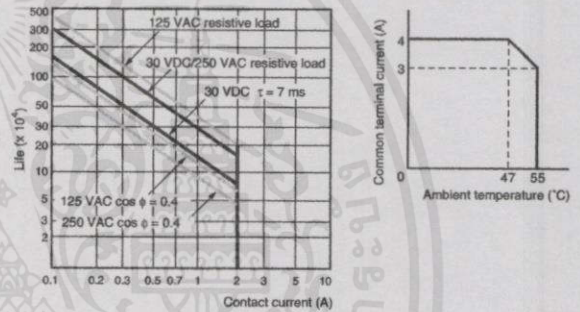
Item	Specifications
ON/OFF delay	

Output Specifications

CPU Units with Relay Outputs

Item	Specifications	
Max. switching capacity	2 A, 250 VAC ($\cos\phi = 1$), 2 A, 24 VDC 4 A/common	
Min. switching capacity	5 VDC, 10 mA	
Service life of relay	Resistive load	100,000 operations (24 VDC)
	Inductive load	48,000 operations (250 VAC, $\cos\phi = 0.4$)
Mechanical	20,000,000 operations	
ON delay	15 ms max.	
OFF delay	15 ms max.	
Circuit configuration		

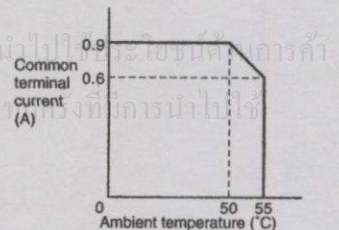
Note: Under the worst conditions, the service life of output contacts is as shown on the left. The service life of relays is as shown in the following diagram as a guideline.



CPU Units with Transistor Outputs (Sinking/Sourcing)

Item	Specifications
CP1L CPU Units	CIO 100.00 to CIO 100.03
Max. switching capacity	4.5 to 30 VDC; 300 mA/point, 0.9 A/common, 3.6 A/Unit (See notes 3 and 4.)
Min. switching capacity	4.5 to 30 VDC; 1 mA
Leakage current	0.1 mA max.
Residual voltage	0.6 V max. / 1.5 V max.
ON delay	0.1 ms max.
OFF delay	0.1 ms max. / 1 ms max.
Fuse	1/common (See note 2.)
Circuit configuration	

- Note 1. Do not apply a voltage or connect a load to an output terminal exceeding the maximum switching capacity.
 2. Fuses cannot be replaced by the user.
 3. Do not use more than 0.9 A total for CIO 100.00 to CIO 100.03.
 4. A maximum of 0.9 A per common can be switched at an ambient temperature of 50°C.



Pulse outputs

Output bits CIO 100.00 to CIO 100.03

Item	Specifications
Max. switching capacity	30 mA at 4.75 to 26.4 VDC
Min. switching capacity	7 mA at 4.75 to 26.4 VDC
Max. output frequency	100 kHz
Output waveform	

Pulse outputs

Output bits CIO 100.01, CIO 100.03

Item	Specifications
Max. switching capacity	30 mA at 4.75 to 26.4 VDC
Max. output frequency	CP1L: 32.8 kHz
PWM output precision	ON duty +5%, -0% at output frequency of 1 kHz
Output waveform	

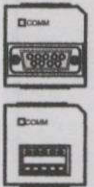
- Note 1.** The above values assume a resistive load and do not consider the impedance of the cable connecting the load.
2. The pulse widths during actual use may be smaller than the ones shown above due to pulse distortion caused by connecting cable impedance.

- Note 1.** The above values assume a resistive load and do not consider the impedance of the cable connecting the load.
2. The pulse widths during actual use may be smaller than the ones shown above due to pulse distortion caused by connecting cable impedance.

Serial Communications Specifications

Item	Function	Interface
Peripheral USB port	For connecting Peripheral Device.	Conforms to USB 1.1, B-type connector
Serial port 1	Host Link, No-protocol, NT Link (1: N), Serial PLC Link (See note.), Serial Gateway (CompoWay/F master, Modbus-RTU master), Modbus-RTU easy master function	The following can be used for either port. CP1W-CIF01 RS-232C Option Board
Serial port 2 (CP1L-M30/M40 only)		CP1W-CIF11 RS-422A/485 Option Board Can be used with either port.

Note: Serial PLC Link can be used with either serial port 1 or serial port 2.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connecting Expansion Units and Expansion I/O Units

CP-series and CPM1A-series Expansion Units and Expansion I/O Units can be connected to the CP1L. Up to three Expansion Units or Expansion I/O Units can be connected to a CPU Unit with 30 or 40 I/O points and one Expansion Unit or Expansion I/O Unit can be connected to a CPU Unit with 20 or 14 I/O points.

The functionality and performance of CP-series Expansion units and Expansion I/O Units is the same as the functionality and performance of CPM1A-series Expansion Units and Expansion I/O Units. CP-series Units are black, and CPM1A-series units are ivory.

Unit name		Output Method	Inputs	Outputs	Model	
					CP1W	CPM1A
Expansion I/O Units	8-point Input Unit		8	-	CP1W-8ED	CPM1A-8ED
	8-point Output Unit	Relay	-	8	CP1W-8ER	CPM1A-8ER
		Transistor (sinking)			CP1W-8ET	CPM1A-8ET
		Transistor (sourcing)			CP1W-8ET1	CPM1A-8ET1
	16-point Output Unit	Relay	-	16	CP1W-16ER	-
	20-point I/O Unit	Relay	12	8	CP1W-20EDR1	CPM1A-20EDR1
		Transistor (sinking)			CP1W-20EDT	CPM1A-20EDT
		Transistor (sourcing)			CP1W-20EDT1	CPM1A-20EDT1
	40-point I/O Unit	Relay	24	16	CP1W-40EDR	CPM1A-40EDR
		Transistor (sinking)			CP1W-40EDT	CPM1A-40EDT
Transistor (sourcing)				CP1W-40EDT1	CPM1A-40EDT1	
Expansion Units	Analog I/O Unit	Analog (resolution 1/256)	2	1	-	CPM1A-MAD01
		Analog (resolution 1/6000)			CP1W-MAD11	CPM1A-MAD11
	Analog Input Unit	Analog (resolution 1/6000)	4	-	CP1W-AD041	CPM1A-AD041
	Analog Output Unit	Analog (resolution 1/6000)	-	4	CP1W-DA041	CPM1A-DA041
			-	2	CP1W-DA021	-
	Temperature Sensor Unit	Thermocouple input	2	-	CP1W-TS001	CPM1A-TS001
			4	-	CP1W-TS002	CPM1A-TS002
		Platinum resistance input	2	-	CP1W-TS101	CPM1A-TS101
			4	-	CP1W-TS102	CPM1A-TS102
	Platinum resistance input and voltage/current output	2	1	-	CPM1A-TS101-DA	
	DeviceNet I/O Link Unit	-	I/O link of 32 input bits and 32 output bits	-	-	CPM1A-DRT21
	Profibus-DP I/O Link Unit	-	I/O link of 16 input bits and 16 output bits	-	-	CPM1A-PRT21
	CompoBus I/O Link Unit	-	I/O link of 8 input bits and 8 output bits	CP1W-SRT21	-	CPM1A-SRT21

CP1W-40EDR/40EDT/40EDT1/20EDR1/20EDT/20EDT1/16ER/8ED/8ER/8ET/8ET1 Expansion I/O Units

Expansion I/O Units can be connected to the CPU Unit to configure the required number of I/O points.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input Specifications of Expansion I/O Units

DC Inputs
(CP1W-40EDR/40EDT/40EDT1/20EDR1/20EDT/20EDT1/8ED)

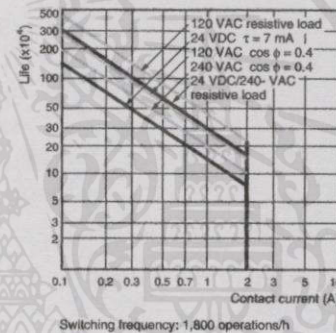
Item	Specifications
Input voltage	24 VDC +10%/-15%
Input impedance	4.7 kΩ
Input current	5 mA typical
ON voltage	14.4 VDC min.
OFF voltage	5.0 VDC max.
ON delay	0 to 32 ms max. (Default: 8 ms) (See note 1.)
OFF delay	0 to 32 ms max. (Default: 8 ms) (See note 1.)
Circuit configuration	

Note 1. Do not apply a voltage exceeding the rated voltage to an input terminal.
2. Can be set in the PLC Setup to 0, 0.5, 1, 2, 4, 8, 16 or 32 ms.
 The CP1W-40EDR/EDT/EDT1 are fixed at 16 ms.

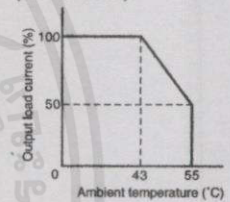
Relay Outputs
(CP1W-40EDR/20EDR1/16ER/8ER)

Item	Specifications	
Max. switching capacity	2 A, 250 VAC (cosφ = 1), 24 VDC 4 A/common	
Min. switching capacity	5 VDC, 10 mA	
Service life of relay	Resistive load	150,000 operations (24 VDC)
	Inductive load	100,000 operations (24 VAC cos = 0.4)
	Mechanical	20,000,000 operations
ON delay	15 ms max.	
OFF delay	15 ms max.	
Circuit configuration		

Note: Under the worst conditions, the service life of output contacts is as shown on the left. The service life of relays is as shown in the following diagram as a guideline.



Relationship between Output Load Current and Ambient Temperature (CP1W-16ER)



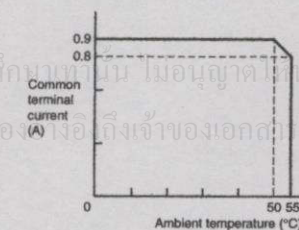
Transistor Outputs (Sinking/Sourcing)

Item	Specifications		
	CP1W-40EDT CP1W-40EDT1	CP1W-20EDT CP1W-20EDT1	CP1W-8ET CP1W-8ET1
Max. switching capacity (See note 3.)	4.5 to 30 VDC; 0.3 A/point 0.9 A/common 3.6 A/common	24 VAC +10%/-5%; 0.3 A/point 0.9 A/common 1.8 A/common	OUT00/OUT01: 0.2 A/point at 4.5 to 30 VDC OUT02 to OUT07: 0.3 A/point at 4.5 to 30 VDC
Leakage current	0.1 mA max.	0.1 mA max.	0.1 mA max.
Residual voltage	1.5 V max.	1.5 V max.	1.5 V max.
ON delay	0.1ms max.	0.1 ms max.	0.1 ms max.
OFF delay	1 ms max. at 24 VDC +10%/-5%, 5 to 300 mA	1 ms max. at 24 VDC +10%/-5%, 5 to 300 mA	1 ms max. at 24 VDC +10%/-5%, 5 to 300 mA
Fuse (See note 2.)	None	1/common	
Circuit configuration			

Note 1. Do not apply a voltage or connect a load to an output terminal exceeding the maximum switching capacity.
2. The fuses cannot be replaced by the user.

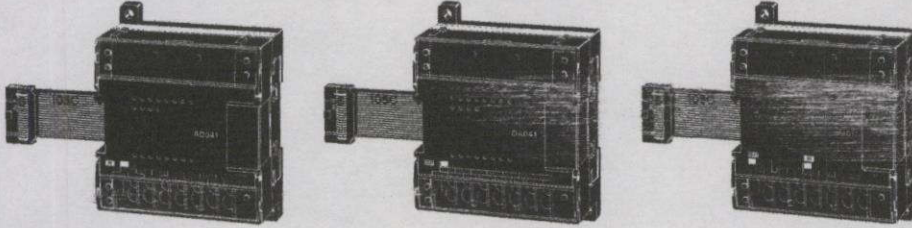
3. A maximum of 0.9 A per common can be switched at an ambient temperature of 50°C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับว่าได้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CP1W-AD041/DA041/DA021/MAD11 Analog Units

Analog values that are input are converted to binary data and stored in the input area, or binary data is output as analog values.



Analog Input Unit: CP1W-AD041

Model		CP1W-AD041	
Item		Input voltage	Input current
Number of inputs		4	
Input signal range		0 to 5 V, 1 to 5 V, 0 to 10 V, -10 to 10 V	0 to 20 mA 4 to 20 mA
Max. rated input		±15 V	±30 mA
External input impedance		1 MΩ min.	Approx. 250 Ω
Resolution		6000	
Overall accuracy	25°C	±0.3% of full scale	±0.4% of full scale
	0 to 55°C	±0.6% of full scale	±0.8% of full scale
Conversion time		2.0 ms/point	
A/D conversion data		Binary data with resolution of 6,000 Full scale for -10 to 10 V: F448 to 0BB8 hex Full scale for other ranges: 0000 to 1770 hex	
Averaging		Supported.	
Open-circuit detection		Supported.	
Insulation resistance		20 MΩ min. (at 250 VDC, between isolated circuits)	
Dielectric strength		500 VAC for 1 min (between isolated circuits)	
Isolation method		Photocoupler isolation (between analog inputs and secondary internal circuits). No isolation between input signals.	

Analog Output Unit: CP1W-DA041/DA021

Model		CP1W-DA041/DA021	
Item		Output voltage	Output current
Number of outputs		DA041: 4, DA021: 2	
Output signal range		0 to 5 V, 0 to 10 V, or -10 to 10 V	0 to 20 mA or 4 to 20 mA
Allowable external output load resistance		2 kΩ min.	350 Ω max.
External output impedance		0.5 Ω max.	---
Resolution		6000	
Overall accuracy	25°C	±0.4% of full scale	
	0 to 55°C	±0.8% of full scale	
Conversion time		2.0 ms/point	
D/A conversion data		Binary data with resolution of 6,000 Full scale for -10 to 10 V: F448 to 0BB8 hex Full scale for other ranges: 0000 to 1770 hex	
Insulation resistance		20 MΩ min. (at 250 VDC between isolated circuits)	
Dielectric strength		500 VAC for 1 min between isolated circuits	
Isolation method		Photocoupler isolation between analog inputs and secondary internal circuits. No isolation between analog input signals.	

Analog I/O Unit: CP1W-MAD11

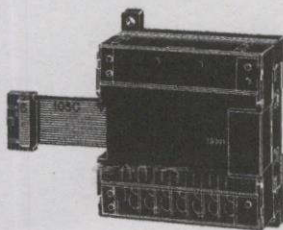
Model		CP1W-MAD11		
Item		Voltage I/O	Current I/O	
Analog Input Section	Number of inputs	2 inputs		
	Input signal range	0 to 5 V, 1 to 5V, 0 to 10 V, or -10 to 10V		
	Max. rated input	±15 V		
	External input impedance	1 MΩ min.		
	Resolution	1/6000 (full scale)		
	Overall accuracy	25°C	±0.3% of full scale	±0.4% of full scale
		0 to 55°C	±0.6% of full scale	±0.8% of full scale
A/D conversion data	Binary data (hexadecimal, 4 digits) -10 to 10 V: F448 to 0BB8 hex Full scale for other ranges: 0000 to 1770 hex			
Analog Output Section (See note 1.)	Averaging	Supported (Set for each input using a DIP switch.)		
	Disconnection detection	Supported		
	Number of outputs	1 output		
	Output signal range	1 to 5 V, 0 to 10 V, -10 to 10 V	0 to 20 mA, 4 to 20 mA	
	External output max. current	---		
	Allowable external output load resistance	1 kΩ min.	600 Ω max.	
	External input impedance	0.5 Ω max.		
	Resolution	1/6000 (full scale)		
	Overall accuracy	25°C	±0.4% of full scale	---
		0 to 55°C	±0.8% of full scale	---
Data setting	---			
D/A conversion data	Binary data (hexadecimal, 4 digits) -10 to 10 V: F448 to 0BB8 hex Full scale for other ranges: 0000 to 1770 hex			
Conversion time (See note 2.)	2 ms/point (6 ms for all points)			
Isolation method	Photocoupler isolation between analog I/O and internal circuits (There is no isolation between the analog I/O signals.)			

Note 1. The voltage output and current output can be used at the same time for analog outputs, but the total output current must not exceed 21 mA.

2. The conversion time is the total time for 2 analog inputs and 1 analog output.

Temperature Sensor Units: CP1W-TS001/TS002/TS101/TS102

By mounting a Temperature Sensor Unit to the PLC, inputs can be obtained from thermocouples or platinum resistance thermometers, and temperature measurements can be converted to binary data (4-digit hexadecimal) and stored in the input area of the CPU Unit.



Specifications

Item	Model	CP1W-TS001/002	CP1W-TS101/102
Number of inputs		2 (TS001), 4 (TS002)	2 (TS101), 4 (TS102)
Input types		K, J switchable (Note: Same for all inputs.)	Pt100, JPt100 switchable (Note: Same for all inputs.)
Indication accuracy		(The larger of the indicated value: $\pm 0.5\%$ and $\pm 2^\circ\text{C}$ (See note.)) ± 1 digit max.	(The larger of the indicated value: $\pm 0.5\%$ and $\pm 1^\circ\text{C}$) ± 1 digit max.
Conversion time		250 ms/2 points (TS001, TS101); 250 ms/4 points (TS002, TS102)	
Converted temperature data		Binary (4-digit hexadecimal)	
Isolation method		Photocoupler isolation between the temperature input signals.	

Note: The indication accuracy when using a K-type thermocouple for temperature less than -100°C is $\pm 4^\circ\text{C} \pm 1$ digit max.

Input Temperature Ranges for CP1W-TS001/002
(The rotary switch can be used to make the following range and input type settings.)

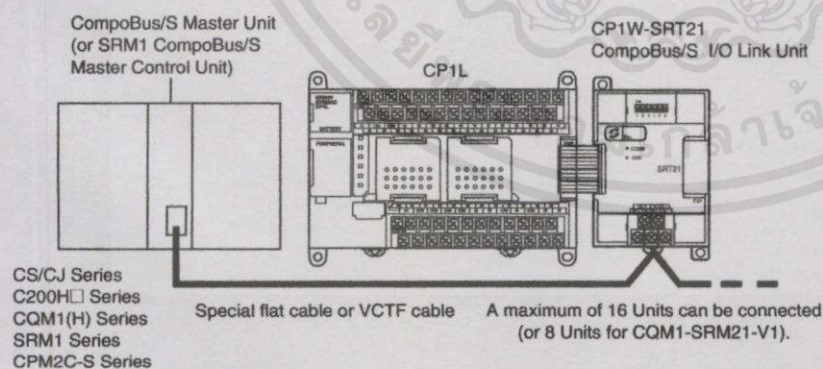
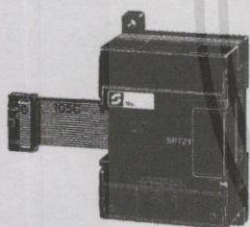
Input type	Range ($^\circ\text{C}$)	Range ($^\circ\text{F}$)
K	-200 to 1300	-300 to 2300
	0.0 to 500.0	0.0 to 900.0
J	-100 to 850	-100 to 1500
	0.0 to 400.0	0.0 to 750.0

Input Temperature Ranges for CP1W-TS101/102
(The rotary switch can be used to make the following range and input type settings.)

Input type	Range ($^\circ\text{C}$)	Range ($^\circ\text{F}$)
Pt100	-200.0 to 650.0	-300 to 1200.0
JPt100	-200.0 to 650.0	-300 to 1200.0

CP1W-SRT21 CompoBus/S I/O Link Unit

The CompoBus/S I/O Link Unit functions as a slave for a CompoBus/S Master Unit (or an SRM1 CompoBus/S Master Control Unit) to form an I/O Link with 8 inputs and 8 outputs between the CompoBus/S I/O Link Unit and the Master Unit.



Specifications

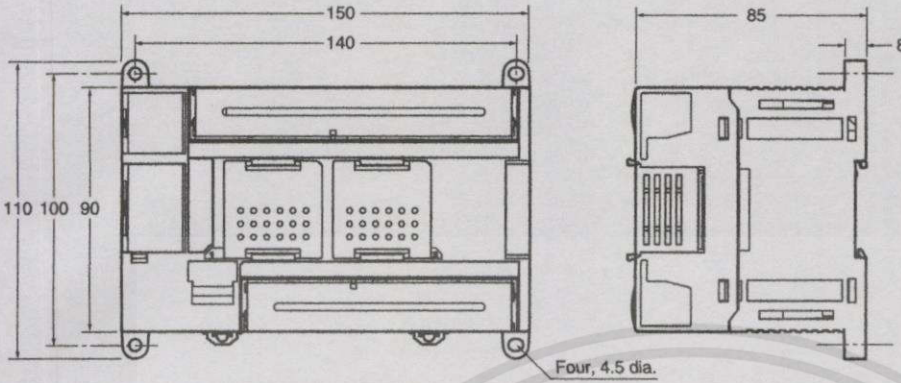
Item	Model	CP1W-SRT21
Master/Slave		CompoBus/S Slave
Number of I/O bits		8 input bits, 8 output bits
Number of words occupied in CP1L I/O memory		1 input word, 1 output word (Allocated in the same way as for other Expansion Units)
Node number setting		Set using the DIP switch (before the CPU Unit is turned ON.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensions

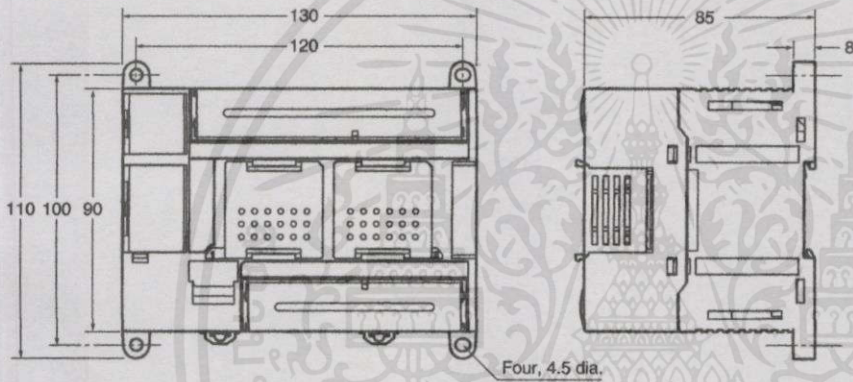
(Unit: mm)

CP1L CPU Units with 40 I/O Points



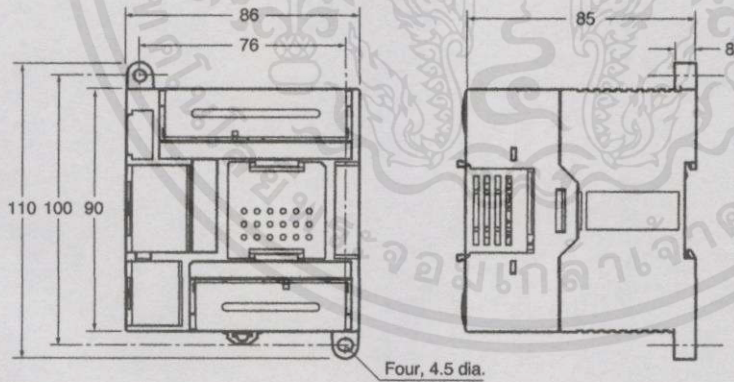
Weight:
675 g max.

CP1L CPU Units with 30 I/O Points



Weight:
610 g max.

CP1L CPU Units with 14 or 20 I/O Points



Weight:
380 g max.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Expansion Units and Expansion I/O Units

CP1W-20ED□

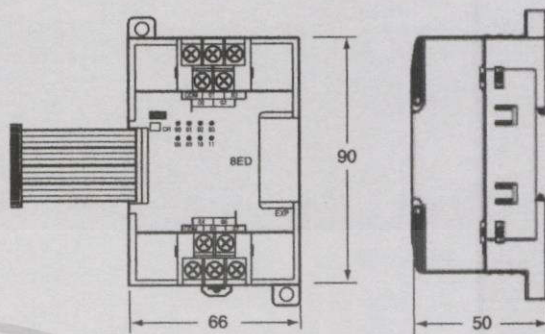
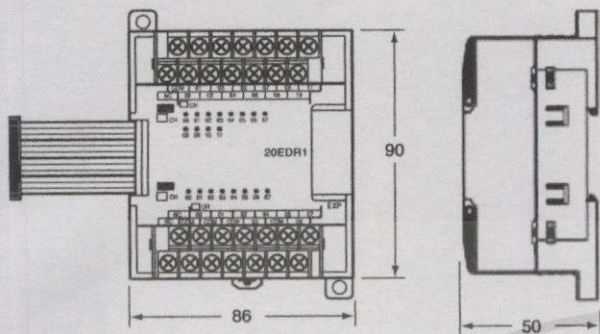
CP1W-16ER

CP1W-AD041/CP1W-DA041/CP1W-DA021

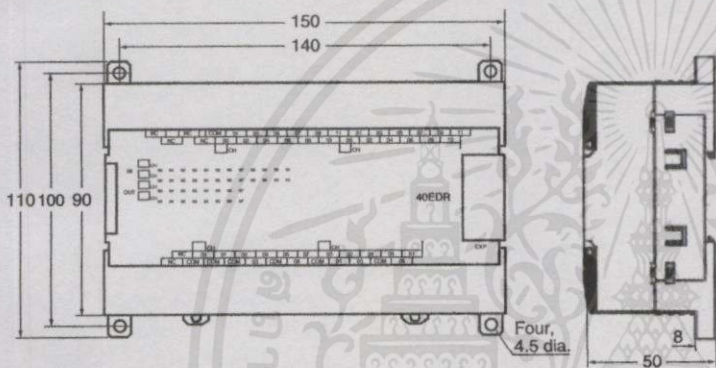
CP1W-MAD11/CP1W-TS□□□

CP1W-8E□□

CP1W-SRT21



CP1W-40ED□



Unit name	Model number	Weight
Expansion I/O Units	CP1W-40EDR	380 g
	CP1W-40EDT/-40EDT1	320 g
	CP1W-20EDR1/-20EDT/-20EDT1	300 g
	CP1W-16ER	280 g
	CP1W-8ED	200 g
Analog Units	CP1W-8ER/-8ET/-8ET1	250 g
	CP1W-AD041/-DA041/-DA021	200 g
Temperature Sensor Units	CP1W-MAD11	150 g
	CP1W-TS001/-TS002/-TS101/-TS102	250 g
CompoBus/S I/O Link Unit	CP1W-SRT21	200 g

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information





CPU Units

International Standards

The standards indicated in the "Standards" column are those current for UL, CSA, cULus, NK, and Lloyd standards and EC Directives as of the end of April 2007. The standards are abbreviated as follows: U: UL, U1: UL (Class 1 Division 2 Products for Hazardous Locations), C: CSA, UC: cULus, UC1: cULus (Class 1 Division 2 Products for Hazardous Locations), CU: cUL, N: NK, L: Lloyd, and CE: EC Directives

Ask your OMRON representative for the conditions under which the standards were met.

CP1L CPU Units

CPU Unit	Specifications				Model	Standards
	Power supply	Output method	Inputs	Outputs		
CP1L-M CPU Units with 40 Points 	AC power supply	Relay output	24	16	CP1L-M40DR-A	UC1, N, L, CE
	DC power supply	Transistor output (sinking)			CP1L-M40DR-D	
		Transistor output (sourcing)			CP1L-M40DT-D	
CP1L-M CPU Units with 30 Points 	AC power supply	Relay output	18	12	CP1L-M30DR-A	UC1, N, L, CE
	DC power supply	Transistor output (sinking)			CP1L-M30DR-D	
		Transistor output (sourcing)			CP1L-M30DT-D	
CP1L-L CPU Units with 20 Points 	AC power supply	Relay output	12	8	CP1L-L20DR-A	UC1, N, L, CE
	DC power supply	Transistor output (sinking)			CP1L-L20DR-D	
		Transistor output (sourcing)			CP1L-L20DT-D	
CP1L-L CPU Units with 14 Points 	AC power supply	Relay output	8	6	CP1L-L14DR-A	UC1, N, L, CE
	DC power supply	Transistor output (sinking)			CP1L-L14DR-D	
		Transistor output (sourcing)			CP1L-L14DT-D	

Options for CPU Units

Name	Specifications	Model	Standards
RS-232C Option Board	For CPU Unit option port.	CP1W-CIF01	UC1, N, L, CE
RS-422A/485 Option Board	For CPU Unit option port.	CP1W-CIF11	CE
Memory Cassette	Can be used for backing up programs or auto-booting.	CP1W-ME05M	

Programming Devices

Name	Specifications	Model	Standards	
CX-One FA Integrated Tool Package Ver. 2.0	CX-One is a package that integrates the Support Software for OMRON PLCs and components. CX-One runs on the following OS. OS: Windows 98SE, Me, NT 4.0 (Service Pack 6a), 2000 (Service Pack 3 or higher), or XP *CX-Thermo runs only on Windows 2000 (Service Pack 3 or higher) or XP. CX-One Ver. 2.0 includes CX-Programmer Ver. 7.□. For details, refer to the CX-One catalog (Cat. No. R134). *The software is provided on CDs for the CXONE-AL□□C-□EV2 and on DVD for the CXONE-AL□□D-□EV2. *Site licenses are available for users who must run the CX-One on many computers. Ask your OMRON representative for details.	1 license	CXONE-AL01C-EV2 CXONE-AL01D-EV2	---
		3 licenses	CXONE-AL03C-EV2 CXONE-AL03D-EV2	
		10 licenses	CXONE-AL10C-EV2 CXONE-AL10D-EV2	
		50 licenses	CXONE-AL50C-EV2 CXONE-AL50D-EV2	
USB Programming cable	A-type male to B-type male (Length: 1.8 m)	CP1W-CN221	---	
Programming Device Connecting Cable for CP1W-CIF01 RS-232C Option Board	Connects DOS computers, D-Sub 9-pin (Length: 2.0 m)	For anti-static connectors	XW2Z-200S-CV	---
	Connects DOS computers, D-Sub 9-pin (Length: 5.0 m)		XW2Z-500S-CV	
	Connects DOS computers, D-Sub 9-pin (Length: 2.0 m)		XW2Z-200S-V	
	Connects DOS computers, D-Sub 9-pin (Length: 5.0 m)		XW2Z-500S-V	
USB-Serial Conversion Cable (See note)	USB-RS-232C Conversion Cable (Length: 0.5 m) and PC driver (on a CD-ROM disc) are included. Complies with USB Specification 1.1 On personal computer side: USB (A plug connector, male) On PLC side: RS-232C (D-sub 9-pin, male) Driver: Supported by Windows 98, Me, 2000, and XP	CS1W-CIF31		

- Note: 1. Cannot be used with a peripheral USB port.
 2. CP1L PLCs are supported by CX-Programmer version 7.1 or higher.

Expansion Units

Name	Output method	Inputs	Outputs	Model	Standards
Expansion I/O Units	Relay	24	16	CP1W-40EDR	N, L, CE
	Transistor (sinking)			CP1W-40EDT	
	Transistor output (sourcing)			CP1W-40EDT1	
	Relay	12	8	CP1W-20EDR1	U, C, L, CE
	Transistor (sinking)			CP1W-20EDT	U, C, N, L, CE
	Transistor output (sourcing)			CP1W-20EDT1	
	Relay	---	16	CP1W-16ER	CE
	---	8		CP1W-8ED	U, C, N, L, CE
	Relay	---	8	CP1W-8ER	
	Transistor (sinking)	---		CP1W-8ET	
Transistor output (sourcing)	---	CP1W-8ET1			
Analog Input Unit	Analog (resolution: 1/6000)	4	---	CP1W-AD041	UC1, CE
Analog Output Unit	Analog (resolution: 1/6000)	---	4	CP1W-DA041	UC1, CE
		---	2	CP1W-DA021	UC1, CE
Analog I/O Unit	Analog (resolution: 1/6000)	2	1	CP1W-MAD11	U, C, N, CE
CompoBus/S I/O Link Unit	---	8 (I/O link input bits)	8 (I/O link input bits)	CP1W-SRT21	U, C, N, L, CE
Temperature Sensor Unit	2 thermocouple inputs			CP1W-TS001	U, C, N, L, CE
	4 thermocouple inputs			CP1W-TS002	
	2 platinum resistance thermometer inputs			CP1W-TS101	
	4 platinum resistance thermometer inputs			CP1W-TS102	

Optional Products, Maintenance Products and DIN Track Accessories

Name	Specifications	Model	Standards
Battery Set	For CP1L CPU Units (Use batteries within two years of manufacture.)	CJ1W-BAT01	CE
DIN Track	Length: 0.5 m; Height: 7.3 mm	PFP-50N	---
	Length: 1 m; Height: 7.3 mm	PFP-100N	
	Length: 1 m; Height: 16 mm	PFP-100N2	
End Plate	There are 2 stoppers provided with CPU Units and I/O Interface Units as standard accessories to secure the Units on the DIN Track.	PFP-M	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read and Understand this Catalog

Please read and understand this catalog before purchasing the product. Please consult your OMRON representative if you have any questions or comments.

WARRANTY

OMRON's exclusive warranty is that the products are free from defects in materials and workmanship for a period of one year (or other period if specified) from date of sale by OMRON.

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, REGARDING NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR PARTICULAR PURPOSE OF THE PRODUCTS. ANY BUYER OR USER ACKNOWLEDGES THAT THE BUYER OR USER ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE. OMRON DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED.

LIMITATIONS OF LIABILITY

OMRON SHALL NOT BE RESPONSIBLE FOR SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED ON CONTRACT, WARRANTY, NEGLIGENCE, OR STRICT LIABILITY.

In no event shall the responsibility of OMRON for any act exceed the individual price of the product on which liability is asserted.

IN NO EVENT SHALL OMRON BE RESPONSIBLE FOR WARRANTY, REPAIR, OR OTHER CLAIMS REGARDING THE PRODUCTS UNLESS OMRON'S ANALYSIS CONFIRMS THAT THE PRODUCTS WERE PROPERLY HANDLED, STORED, INSTALLED, AND MAINTAINED AND NOT SUBJECT TO CONTAMINATION, ABUSE, MISUSE, OR INAPPROPRIATE MODIFICATION OR REPAIR.

SUITABILITY FOR USE

OMRON shall not be responsible for conformity with any standards, codes, or regulations that apply to the combination of the product in the customer's application or use of the product.

Take all necessary steps to determine the suitability of the product for the systems, machines, and equipment with which it will be used.

Know and observe all prohibitions of use applicable to this product.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

PROGRAMMABLE PRODUCTS

OMRON shall not be responsible for the user's programming of a programmable product, or any consequence thereof.

CHANGE IN SPECIFICATIONS

Product specifications and accessories may be changed at any time based on improvements and other reasons. Consult with your OMRON representative at any time to confirm actual specifications of purchased product.

DIMENSIONS AND WEIGHTS

Dimensions and weights are nominal and are not to be used for manufacturing purposes, even when tolerances are shown.

PERFORMANCE DATA

Performance data given in this catalog is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of OMRON's test conditions, and the users must correlate it to actual application requirements. Actual performance is subject to the OMRON Warranty and Limitations of Liability.

Cat. No. P20E-EN-02

In the interest of product improvement, specifications are subject to change without notice.

OMRON EUROPE B.V.

Wegalaan 67-69,
NL-2132 JD, Hoofddorp,
The Netherlands

Phone: +31 23 568 13 00

Fax: +31 23 568 13 88

www.industrial.omron.eu