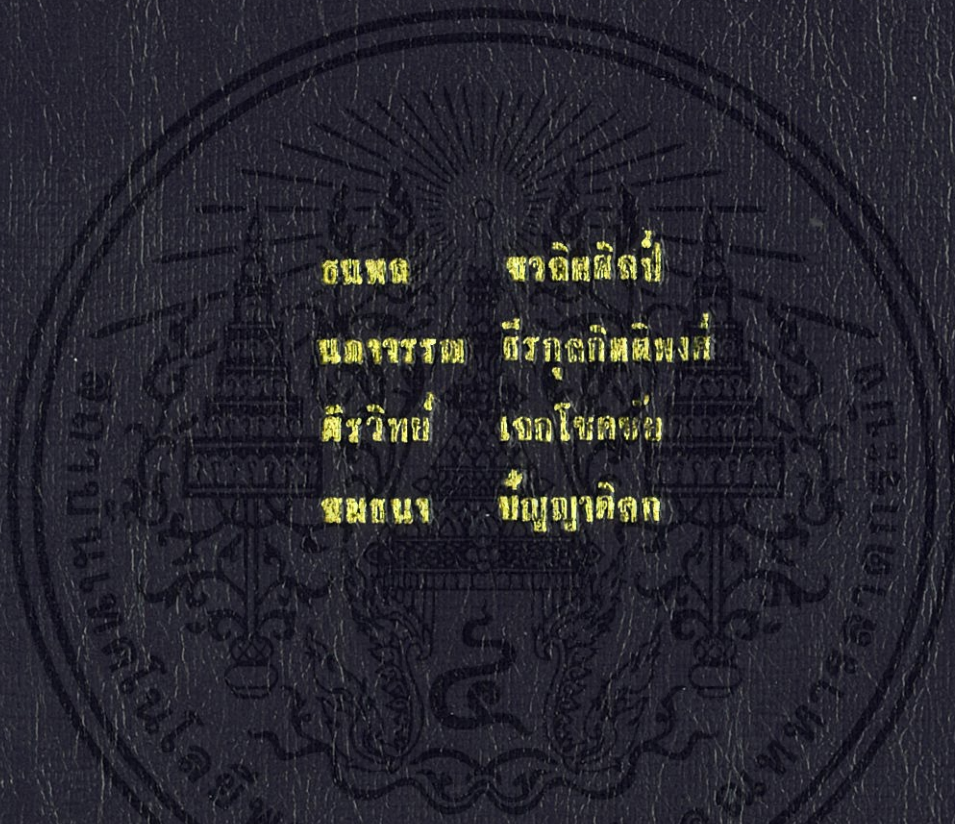


แบบจำลองระบบคลังสินค้า
WAREHOUSE SYSTEM MODEL



ปริญญาโท สาขาบริหารงานส่วนหน้าของงานด้านวิชาการ หลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาศึกษาศาสตร์ วิทยาลัยราชภัฏบุรีรัมย์

คณะศึกษาศาสตร์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2555

แบบจำลองระบบคลังสินค้า
WAREHOUSE SYSTEM MODEL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAREHOUSE SYSTEM MODEL



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2012

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่ข้อมูลใดๆของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองระบบคลังสินค้า

โดย

นายธนพล	ชวลิตศิลป์	52010449
นางสาวนภาพรรณ	ธีรกุลกิตติพงศ์	52010565
นายศิริวิทย์	เจอโชคชัย	52011190
นางสาวสมธนา	ปัญญาติลก	52011239

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร.วันชัย	รัฐรุจา
อาจารย์เทพจิตร	เชยโกคา
ผู้ช่วยศาสตราจารย์ ดร.วรรณดี	เพชรณีนีล้ำค่า

ปีการศึกษา 2555

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ นำเสนอแบบจำลองการทำงานของระบบคลังสินค้า โดยโครงสร้างของระบบประกอบด้วยส่วนควบคุมหลัก ส่วนสายพานลำเลียง ส่วนคัดแยกและส่วนโปรแกรมควบคุมระบบจุดมุ่งหมายของโครงการนี้คือการจำลองและออกแบบระบบคลังสินค้าอัตโนมัติโดยใช้ระบบคอมพิวเตอร์เข้ามาควบคุมให้แสดงผลและทำตามคำสั่งที่ได้ป้อนเข้าไป โดยเริ่มจากการออกแบบและประกอบโครงสร้างของระบบและเขียนโปรแกรม Arduino ควบคุมการทำงานของไมโครคอนโทรลเลอร์ ET-EASY168STAMP ร่วมกับวงจรถออิเล็กทรอนิกส์ เพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAREHOUSE SYSTEM MODEL

By

Mr.Tanapon	Chavalitsin	52010449
Ms.Napawan	Teerakulkittipong	52010565
Mr.Sirawit	Jerchokchai	52011190
Ms.Somthana	Panyadilok	52011239

Advisors

Assoc.Prof.Dr.Vanchai	Riewruja
Mr.Thepjit	Cheypoca
Asst.Prof.Dr.Wandee	Petchmaneelumka

Academic Year 2012

ABSTRACT

This thesis presents about warehouse system model. The structure consists of a main controller, conveyor section, sorting section and system programming section. The purpose of this project is to design and implement the automatic warehouse system controlled via computer and operate under the command. Beginning of operation is design the structure of system. Then C language, Arduino program, electronic circuits and ET-EASY168STAMP are used for control system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ที่ปรึกษา ทั้งรศ.ดร.วันชัย ธีร์รุจา อาจารย์เทพจิตร เขยโกคา และผศ.ดร.วรรณดี เพชรมณีล้ำค่า เป็นอย่างสูงที่ได้ให้ปรึกษาแนวทางในการแก้ปัญหา ความคิดริเริ่มให้ความเอาใจใส่ดูแลสอบถามถึงความก้าวหน้าอย่างสม่ำเสมอทำให้คณะผู้จัดทำสามารถทำงานได้อย่างเป็นระบบและใช้เวลาได้อย่างมีประสิทธิภาพ

ขอขอบคุณสาขาวิชาวิศวกรรมการวัดและควบคุม หลักสูตรวิศวกรรมระบบควบคุม ที่ได้เอื้อเฟื้ออุปกรณ์และสถานที่ในการทำวิจัยจนผู้จัดทำสามารถทำปริญญานิพนธ์ฉบับนี้เสร็จสิ้น

ขอขอบคุณเพื่อนๆ ทุกคนสำหรับกำลังใจสนับสนุนอุปกรณ์ที่ขาดเหลือและคอยถามไถ่ความคืบหน้าของโครงการอยู่เสมอ จนสามารถสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ทางคณะผู้จัดทำขอกราบขอบพระคุณบิดามารดาและครอบครัว ที่คอยเป็นกำลังใจที่ดีที่สุดมา ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้

คณะผู้จัดทำ

ธนพล

ชวลิตศิลป์

นภาพรรณ

ธีรกุลกิตติพงศ์

ศิริวิทย์

เจจโจชชัย

สมธนา

ปัญญาติลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ในการศึกษา	1
1.3 ขอบเขตการศึกษา	1
1.4 ขั้นตอนในการดำเนินงาน	2
1.4.1 ศึกษาทฤษฎีที่เกี่ยวข้อง	2
1.4.2 ออกแบบและทดลอง	2
1.5 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 ไมโครคอนโทรลเลอร์ AVR รุ่น ET-EASY168 STAMP	3
2.1.1 คุณสมบัติของบอร์ด ET-EASY168 STAMP	4
2.1.2 หน้าที่ของขาสัญญาณในการทำงานแบบ AVR Microcontroller	5
2.2 โปรแกรมของไมโครคอนโทรลเลอร์ AVR	5
2.3 มอเตอร์ไฟฟ้ากระแสตรง	7
2.3.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง	7
2.3.1.1 สเตเตอร์ (Stator)	7
2.3.1.2 ตัวหมุน (Rotor)	8
2.3.2 หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action)	10
2.3.3 การควบคุมมอเตอร์ไฟฟ้ากระแสตรงด้วยสัญญาณพัลส์วidthมอดูเลชัน	10
2.4 การขับมอเตอร์ด้วย L298	11
2.5 การควบคุมตำแหน่งของมอเตอร์กับเอ็นโค้ดเดอร์	13
2.6 ทฤษฎีและหลักการการทำงานของเซนเซอร์	19
2.7 การควบคุมการทำงานเซนเซอร์ด้วยไอซี LM339	21
2.8 โปรแกรม visual c#	21
2.8.1 ทฤษฎีและหลักการเขียนโปรแกรมควบคุมการทำงานของเครื่อง	22
2.8.2 การใช้งาน visual c#	23
2.8.2.1 การใช้งาน Pen และ Brush	24
2.8.2.2 การใส่สีพื้นหลัง	24
2.8.2.3 การใช้งาน Brush	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกไปเผยแพร่ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 3 การปรับปรุงและพัฒนา	27
3.1 โครงสร้างโดยรวมของระบบลำเลียง	27
3.1.1 ระบบลำเลียง (Part A)	27
3.1.2 ระบบลำเลียง (Part B, D, E)	27
3.1.3 ระบบลำเลียง (Part C)	28
3.2 วงจรไฟฟ้าโดยรวมของระบบลำเลียง	29
3.2.1 วงจรขับมอเตอร์ไฟฟ้ากระแสตรง	29
3.2.2 วงจรควบคุมการทำงานของเซนเซอร์และวงจรเซนเซอร์	29
3.2.3 การต่อวงจรใช้งานจริงทั้งระบบ	31
3.3 การปรับปรุงฐานยึดเซนเซอร์	31
3.4 การพัฒนาการจ่ายไฟ	32
3.5 การปรับปรุงเซนเซอร์	32
3.6 การออกแบบในส่วนของโปรแกรม	34
3.6.1 โปรแกรม arduino	34
3.6.1.1 หลักการทำงานของระบบ	35
3.6.1.2 หลักการทำงานของโปรแกรมส่วนลำเลียงสินค้าขึ้น(Part A)	36
3.6.1.3 หลักการทำงานของโปรแกรมส่วนลำเลียงสินค้า(Part B)	37
3.6.1.4 หลักการทำงานของโปรแกรมส่วนลำเลียงสินค้า(Part D)	37
3.6.1.5 หลักการทำงานของโปรแกรมส่วนลำเลียงสินค้า(Part E)	38
3.6.1.6 หลักการทำงานของโปรแกรมส่วนคัดแยก(Part C)	38
3.6.2 โปรแกรม Visual C#	41
3.6.2.1 ภาพรวมของโปรแกรม Visual C#	41
3.6.2.2 การกำหนดคุณสมบัติของฟอร์มและคอนโทรลต่างๆ ในการออกแบบ Visual C#	42
บทที่ 4 การทดลองและผลการทดลอง	44
4.1 ลำดับขั้นตอนการทำงาน	44
4.2 ผลการทดสอบการทำงานระบบลำเลียง	44
4.2.1 เมื่อกดปุ่ม Start ที่หน้าจอติดต่อกันระหว่างผู้ใช้งานกับโปรแกรม	44
4.2.2 เมื่อกดปุ่ม Stop ที่หน้าจอติดต่อกันระหว่างผู้ใช้งานกับโปรแกรม	45
บทที่ 5 บทวิจารณ์และสรุป	46
5.1 บทวิจารณ์	46
5.2 สรุปผล	46
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	48

เอกสารนี้เป็นเอกสารตัวอย่างสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น

สารบัญ(ต่อ)

	หน้า
เอกสารอ้างอิง	49
ภาคผนวก ก โปรแกรมควบคุม	51
ก.1 โปรแกรมส่วน Main Controller (Master)	52
ก.2 โปรแกรมส่วนสายพานลำเลียงขาขึ้น (Part A)	54
ก.3 โปรแกรมส่วนสายพานลำเลียง (Part B)	56
ก.4 โปรแกรมส่วนสายพานลำเลียง (Part D)	57
ก.5 โปรแกรมส่วนสายพานลำเลียง (Part E)	59
ก.6 โปรแกรมส่วนคัดแยก (Part C)	60
ก.7 โปรแกรมส่วนการแสดงผล (Visual C#)	66
ภาคผนวก ข เอกสารคู่มือการใช้งานอุปกรณ์อิเล็กทรอนิกส์	70
ข.1 คู่มือการเลือกใช้ L298D	70
ข.2 คู่มือการใช้ TCRT5000	81
ภาคผนวก ค การเริ่มต้นการใช้โปรแกรม Arduino	89
ค.1 การติดตั้ง Driver ของ USB Bridge ของบอร์ด ET-EASY168 STAMP	89
ค.2 การพัฒนาโปรแกรม ET-EASY168 STAMP แบบ Arduino Project	94
ค.3 การติดตั้งโปรแกรม Arduino	95
ภาคผนวก ง โปสเตอร์แบบจำลองระบบคลังสินค้า	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างการใช้งานโปรแกรมArduino	6
2.2 เฟรม	7
2.3 ขดลวดพันอยู่รอบขั้วแม่เหล็ก	8
2.4 ขั้วแม่เหล็ก	8
2.5 โรเตอร์	9
2.6 แปรรงถ่าน	9
2.7 ช่องแปรรงถ่าน	10
2.8 สัญญาณพัลส์วิทมอดูเลชันมีความกว้างพัลส์เท่ากับ 10 เปอร์เซ็นต์	11
2.9 สัญญาณพัลส์วิทมอดูเลชันมีความกว้างพัลส์เท่ากับ 50 เปอร์เซ็นต์	11
2.10 สัญญาณพัลส์วิทมอดูเลชันมีความกว้างพัลส์เท่ากับ 90 เปอร์เซ็นต์	11
2.11 ขาสัญญาณของไอซี L298	12
2.12 วงจรเพื่อทำแผ่น PCB	12
2.13 วงจรการขับมอเตอร์ด้วยไอซี L298	12
2.14 Rotary encoder	13
2.15 การสร้าง Pulse ของ Rotary encoder	13
2.16 Pulse ของ Rotary แบบสองเฟส	14
2.17 การตรวจเช็คสัญญาณ Pulse ของ Rotary encoder	15
2.18 ลักษณะ Pulse ของ Rotary encoder ทั้งสองเฟส	16
2.19 ความละเอียดของ Pulse	16
2.20 Block Diagram ของ Rotary encoder	17
2.21 การตรวจจับขอบขาขึ้นและขอบขาลงของทั้ง 2 เฟส	18
2.22 วิธีการตรวจจับทิศทางการหมุนของ Rotary encoder	18
2.23 การเช็คความกว้างของ Pulse	18
2.24 เซนเซอร์ตรวจจับแสงอินฟราเรดรุ่น TCRT5000	19
2.25 วงจรเซนเซอร์ตรวจจับแสงอินฟราเรดรุ่น TCRT5000	20
2.26 ไอซี LM339	21
2.27 ตำแหน่งขาสัญญาณของไอซี LM339	21
3.1 ลักษณะของระบบลำเลียง (Part A)	27
3.2 ลักษณะของระบบลำเลียง (Part B, D และ E)	28
3.3 ลักษณะของส่วนคัดแยก (Part C)	28
3.4 วงจรขับมอเตอร์ไฟฟ้ากระแสตรง	29

สารบัญญรูป(ต่อ)

รูปที่	หน้า
3.5 วงจรเซนเซอร์ TCRT5000 ที่ใช้งานจริง	30
3.6 วงจรควบคุมการทำงานของเซนเซอร์	30
3.7 การต่อวงจรรวมของระบบ	31
3.8 การติดตั้งเซนเซอร์ของ Part B และ Part D, Eแบบใหม่	31
3.9 โครงสร้างบอร์ดจ่ายไฟ	32
3.10 วงจรเซนเซอร์ที่ใช้เซนเซอร์อินฟาเรด	33
3.11 วงจรเซนเซอร์ที่ใช้TCRT 5000	33
3.12 ขั้นตอนการทำงานของทั้งระบบ	35
3.13 ขั้นตอนการทำงานของสายพานลำเลียงขาขึ้น (Part A)	36
3.14 ขั้นตอนการทำงานของสายพานลำเลียง (Part B)	37
3.15 ขั้นตอนการทำงานของสายพานลำเลียง (Part D)	37
3.16 ขั้นตอนการทำงานของสายพานลำเลียง (Part E)	38
3.17 ขั้นตอนการทำงานของส่วนคัดแยก (Part C)	40
3.18 หน้าจอควบคุมการทำงานโปรแกรม Visual C#	41
4.1 หน้าจอแสดงผลการทำงานเมื่อกดปุ่ม Start	44
4.2 หน้าจอแสดงผลการทำงานเมื่อกดปุ่ม Stop	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 การจัดสรรขาสัญญาณของชุดบอร์ด ET-EASY168 STAMP	4
3.1 การกำหนดคุณสมบัติของฟอร์มและคอนโทรลต่างๆในการออกแบบ Visual C#	42
5.1 ปัญหาที่พบและแนวทางแก้ไขพัฒนา	47



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในระบบอุตสาหกรรมการผลิตทั่วไปนั้นมักมีระบบหลายๆ ระบบที่เกี่ยวข้องกันทำงานต่อกันอย่างเป็นขั้นเป็นตอน และกระบวนการที่เกิดขึ้นหลังจากการผลิตสินค้าเริ่มต้นจากกระบวนการลำเลียงสินค้า ถัดมานั้นเป็นกระบวนการในการคัดแยกสินค้า และขั้นตอนของกระบวนการจัดเก็บเข้าคลังสินค้า ตลอดจนถึงข้อมูลของสินค้าหรือผลิตภัณฑ์ที่จัดเก็บจากการผลิตเพื่อให้ได้ตรงตามมาตรฐานประเภทคุณภาพและตรงตามลักษณะที่เราต้องการ หากในอุตสาหกรรมใช้ทรัพยากรมนุษย์ในการปฏิบัติงานในกระบวนการข้างต้นอาจทำให้เกิดความล่าช้าสูญเสียพลังงานสิ้นเปลืองค่าใช้จ่ายด้านแรงงาน ในระบบอุตสาหกรรมจึงมีการนำระบบอัตโนมัติมาช่วยในกระบวนการผลิตซึ่งสามารถก่อให้เกิดประโยชน์ทางด้านต่างๆ เช่น ด้านทรัพยากรมนุษย์ เป็นการประหยัดแรงงาน ด้านต้นทุนการผลิต ลดค่าใช้จ่าย กระบวนการผลิตไม่หยุดชะงักจึงทำให้เรามีความสนใจที่จะศึกษาการควบคุมแบบอัตโนมัติโดยใช้คอมพิวเตอร์เป็นตัวช่วยในการควบคุม

1.2 วัตถุประสงค์ในการศึกษา

1. เพื่อนำความรู้ทางวิศวกรรมระบบควบคุมมาใช้ในการออกแบบพัฒนาและปรับปรุงระบบการลำเลียงคัดแยกและจัดเก็บสินค้าให้เกิดประสิทธิภาพสูงสุด
2. เพื่อสร้างโปรแกรมสำเร็จรูปที่สามารถควบคุมและจำลองการทำงานของกระบวนการต่างๆ ในระบบคลังสินค้าผ่านทางหน้าจอคอมพิวเตอร์
3. เพื่อศึกษาการเชื่อมต่อข้อมูลระหว่างคอมพิวเตอร์กับแบบจำลองระบบคลังสินค้าและนำไปประยุกต์ใช้งาน
4. เพื่อเปลี่ยนจากการใช้การควบคุมโดย PLC มาเป็นการควบคุมโดยโปรแกรมสำเร็จรูปที่สร้างขึ้น

1.3 ขอบเขตการศึกษา

ในการทำโครงการนี้ จำเป็นต้องเรียนรู้และเข้าใจในตัวระบบการลำเลียงและคัดแยกสินค้า ก่อนที่จะเริ่มพัฒนาโครงการ หลังจากศึกษาข้อมูลพบว่า โครงการนี้เป็นการออกแบบและควบคุมโดยนำระบบอัตโนมัติมาใช้จัดการระบบการลำเลียงสินค้าให้ทำงานเป็นลำดับและระบบลำเลียงในโรงงานอุตสาหกรรม จะแบ่งออกเป็นสองส่วน ส่วนแรกฮาร์ดแวร์(Hardware) ส่วนที่สองซอฟต์แวร์(Software) ส่วนที่สามโปรเซส(Process) คือการเขียนโปรแกรมควบคุมระบบการลำเลียงสินค้าให้ทำงานเป็นลำดับขั้นตอน และส่วนที่สี่ มอนิเตอร์คือส่วนแสดงสถานะภาพการทำงานของระบบการลำเลียงสินค้าทั้งหมดและสามารถทำการควบคุมระบบจากส่วนนี้ได้ ซึ่งเราจะทำการพัฒนาระบบนี้ให้มี

เอกสารนี้เป็นประโยชน์อย่างยิ่งขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนในการดำเนินงาน

1.4.1 ศึกษาทฤษฎีที่เกี่ยวข้อง

- การทำงานและโครงสร้างของระบบคลังสินค้า
- โปรแกรม Visual C#
- คุณสมบัติของเซนเซอร์ทฤษฎีการควบคุมมอเตอร์กระแสตรงโดยใช้ PWM (Pulse Width Modulation) ในการส่งสัญญาณควบคุมมอเตอร์ทฤษฎีทางกลศาสตร์
- สถาปัตยกรรมและโครงสร้างของตัวควบคุมโดยใช้ไมโครคอนโทรลเลอร์ AVR รุ่น EASY 168 STAMP พร้อมหลักการเขียนโปรแกรมควบคุมโดยใช้โปรแกรม Arduino

1.4.2 ออกแบบและทดลอง

- ทดลองตรวจสอบการทำงานของระบบคลังสินค้าในทุกส่วน คือ กระบวนการลำเลียงสินค้า กระบวนการในการคัดแยกสินค้า และกระบวนการจัดเก็บเข้าคลังสินค้า ตลอดจนเก็บข้อมูลของสินค้าหรือผลิตภัณฑ์ที่จัดเก็บจากการผลิต
- ปรับปรุงและพัฒนาทุกๆส่วนในระบบคลังสินค้า เช่น ระบบเซนเซอร์ โปรแกรมที่ใช้ควบคุม ระบบมอเตอร์
- ทดลองการทำงานของระบบคลังสินค้า เมื่อได้รับการปรับปรุงและพัฒนาแล้ว
- สรุปผลที่ได้จากการทดลองและอภิปรายผลการทดลอง

1.5 รายละเอียดของปริญญานิพนธ์

วิทยานิพนธ์ฉบับนี้มีเนื้อหาประกอบด้วย 5 บทดังนี้

บทที่ 1 ได้กล่าวถึงความเป็นมาของปริญญานิพนธ์เล่มนี้ วัตถุประสงค์ในการศึกษา ขอบเขตการศึกษา และขั้นตอนในการดำเนินงานรวมถึงรายละเอียดของปริญญานิพนธ์ในแต่ละบท

บทที่ 2 ทฤษฎีที่เกี่ยวข้องกล่าวถึงทฤษฎีที่เกี่ยวข้องในระบบคลังสินค้าในกระบวนการลำเลียงสินค้า กระบวนการในการคัดแยกสินค้า และกระบวนการจัดเก็บเข้าคลังสินค้า ตลอดจนเก็บข้อมูลของสินค้า

บทที่ 3 หลักการออกแบบนำเสนอการประกอบโครงสร้างของระบบรวมถึงแนวคิดในการออกแบบระบบจำลองและโปรแกรมควบคุมอัตโนมัติ

บทที่ 4 กล่าวถึงการทดลองการรับค่าข้อมูลจากเซนเซอร์และการแสดงผลข้อมูลบนหน้าจอบทคอมพิวเตอร์การทำงานของระบบเป็นผลจากการโปรแกรมควบคุมระบบอัตโนมัติ

บทที่ 5 บทวิจารณ์และสรุปกล่าวถึงการสรุปผลการดำเนินงานปัญหาที่เกิดขึ้นแนวทางในการแก้ไขข้อเสนอแนะและแนวทางในการศึกษาค้นคว้าพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎี

ระบบคลังลำเลียงสินค้า มีการทำงานโดยเริ่มจากการลำเลียงในส่วนแรก (Part A) แล้วส่งต่อไปยัง ส่วนสายพานลำเลียง (Part B) ไปยังส่วนคัดแยก (Part C) จะมีการคัดแยกสี ให้ไปทางซ้ายและขวา จากนั้นสินค้าไหลผ่านส่วนสายพานลำเลียงของ Part D และ Part E โดยองค์ประกอบหลักที่สำคัญ เป็นอย่างมากในการออกแบบโครงสร้างและพัฒนาส่วนของโปรแกรมที่ควรศึกษาเพิ่มเติมเพื่อที่จะ นำมาใช้ในการทำโครงสร้างส่วนคลังเก็บสินค้าซึ่งมีรายละเอียดดังนี้

- ไมโครคอนโทรลเลอร์ตระกูล AVR รุ่น ET-EASY168 STAMP
- โปรแกรมไมโครคอนโทรลเลอร์ AVR
- มอเตอร์ไฟฟ้ากระแสตรงและการควบคุม
- การขับมอเตอร์ด้วย L298
- การควบคุมตำแหน่งของมอเตอร์กับเอ็นโค้ดเดอร์
- เซนเซอร์ TCRT5000
- การควบคุมเซนเซอร์ด้วย LM339
- โปรแกรม Visual c#

2.1 ไมโครคอนโทรลเลอร์ AVR รุ่น ET-EASY168 STAMP

ไมโครคอนโทรลเลอร์ AVR เป็นไอซีไมโครคอนโทรลเลอร์ของบริษัท ATMEL มีสถาปัตยกรรม ภายในเป็นแบบ RISC (reduced instruction set computer) โดยใช้สัญญาณนาฬิกาเพียง 1 ลูก ในการปฏิบัติงานใน 1 คำสั่ง โดยจะประกอบด้วยหน่วยความจำโปรแกรมภายในที่เป็นแบบแฟลช โปรแกรมข้อมูลได้แบบ In-System programmable และในบางเบอร์ยังสามารถมีการกำหนด ตำแหน่งของหน่วยความจำที่สร้างเป็นบูตโหลดเดอร์ (เขียนโปรแกรมเพื่อติดต่อกับ PC หรือไอซีตัว อื่นๆ และยังสามารถโปรแกรมให้กับตัวเองได้) มีขนาดของหน่วยความจำตามเบอร์ของไอซีแต่ละตัว ในโครงการในส่วนของการจัดเก็บเข้าคลังสินค้านี้ได้เลือกไมโครคอนโทรลเลอร์ AVR ATmega168 รุ่น ET-EASY168 STAMP ใช้ในการควบคุมระบบ ซึ่งมีรายละเอียดดังนี้

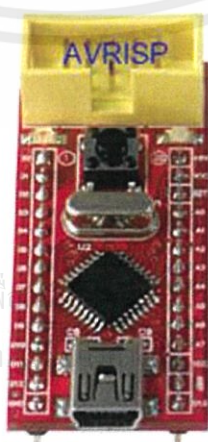
AVR ATmega168 รุ่น ET-EASY168 STAMP เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR8 ขนาดเล็กจิ๋วโดยมีขนาดของบอร์ดเพียง 2cm x 5cm เท่านั้นซึ่งขนาดบอร์ดประมาณเท่ากับ ตัวถังของไอซี 28 DIP 300 โดยเลือกใช้ไมโครคอนโทรลเลอร์ตระกูล AVR8 เบอร์ ATmega168 ของ ATMEL เป็น MCU ประจำบอร์ดโดยเลือกใช้ MCU ที่มีรูปร่างตัวถังแบบ 32 TQFP พร้อมวงจรรอบ นอกที่จำเป็นอย่าง Oscillator และ Reset รวมไว้ด้วยภายในบอร์ดนอกจากนี้แล้วภายในตัวบอร์ดยัง ได้รวมเอาไอซี USB Bridge ของ FTDI เบอร์ FT232R เพื่อใช้ติดต่อสื่อสารแบบอนุกรมด้วย RS232 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB ได้โดยตรงทำให้บอร์ด ET-EASY168 STAMP เป็นบอร์ด ทดลองขนาดเล็กที่เพียบพร้อมไปด้วยวงจรพื้นฐานที่จำเป็นต่อการใช้งานไมโครคอนโทรลเลอร์ตระกูล

AVR8 อย่างแท้จริงเพียงแต่เสียบสาย USB จากพอร์ต USB ของเครื่องคอมพิวเตอร์ PC เข้ากับหัว USB ของบอร์ด ET-EASY168 STAMP ก็สามารถทำการเขียนโปรแกรมและ Download Code ให้กับ MCU เพื่อทำการทดลองได้ทันที

2.1.1 คุณสมบัติของบอร์ด ET-EASY168 STAMP

- เลือกใช้ MCU ตระกูล AVR8 เบอร์ ATmega168 ของ ATMEL Run ความถี่ 16.00 MHz
- มีหน่วยความจำ Flash สำหรับเขียนโปรแกรม 16KByte ถ้าใช้การพัฒนาโปรแกรมผ่านระบบ AVRISP หรือ 14Kbyte เมื่อใช้การพัฒนาโปรแกรมผ่านระบบ Boot Loader RS232
- มี SRAM ใช้งานขนาด 1KByte และ EEPROM ใช้งานขนาด 512 Byte
- มี GPIO ใช้งานจำนวน 22 บิต
- Digital GPIO จำนวน 14 บิต
- Analog Input (ADC) ขนาดความละเอียด 10 บิตจำนวน 8 ช่อง
- ใช้งานกับแรงดันไฟตรงขนาด +5VDC โดยใช้ได้ทั้งกับแหล่งจ่าย +5VDC/500mA จากพอร์ต USB และจากแหล่งจ่าย +5VDC จากภายนอกได้ด้วยพร้อม LED Power แสดงสถานะของแหล่งจ่าย
- มีวงจร External Reset แบบ RC Reset และ Switch Reset พร้อมภายในบอร์ด
- หัวต่อใช้งานวางตัวบน Pin Header ระยะห่าง 2.54mm(100mil) ขนาด 28 Pin (ด้านละ 14Pin) ระยะห่าง 600mil(1.5cm) ง่ายต่อการนำไปต่อประยุกต์ใช้งานและขยายวงจร I/O สามารถใช้กับ Project Board และ PCB เอนกประสงค์ได้โดยง่าย
- มีหัวต่อ USB สำหรับเชื่อมต่อสื่อสารกับคอมพิวเตอร์ PC ผ่าน USB Bridge ของ FTDI ในรูปแบบของการสื่อสารอนุกรม RS232 สำหรับใช้งานสื่อสารและ Download Code ให้กับ MCU ในบอร์ด
- มีหัว AVRISP แบบ IDE 10PIN สำหรับใช้ Download โปรแกรมให้กับ MCU ภายในบอร์ดในกรณี ไม่ต้องการใช้การพัฒนาโปรแกรมผ่านทาง Boot Loader

ตารางที่ 2.1 การจัดสรรขาสัญญาณของชุดบอร์ด ET-EASY168 STAMP

AVR	Arduino	Pin	ET-EASY168 STAMP	Pin	Arduino	AVR
PD0	Digital-0	1		28	+5V(+Vin)	+5V(+Vin)
PD1	Digital-1	2		27	+VCC(+5V)	+VCC(+5V)
PD2	Digital-2	3		26	RESET#	RESET(PC6)
PD3	Digital-3	4		25	Analog-0	PC0/ADC0
PD4	Digital-4	5		24	Analog-1	PC1/ADC1
PD5	Digital-5	6		23	Analog-2	PC2/ADC2
PD6	Digital-6	7		22	Analog-3	PC3/ADC3
PD7	Digital-7	8		21	Analog-4	PC4/ADC4
PB0	Digital-8	9		20	Analog-5	PC5/ADC5
PB1	Digital-9	10		19	Analog-6	ADC6
PB2	Digital-10	11		18	Analog-7	ADCT
PB3	Digital-11	12		17	+VCC(+5V)	+VCC(+5V)
PB4	Digital-12	13		16	+AREF	+AREF
GND	GND	14		15	Digital-13	PB5

ETT

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่สามารถนำออกนอกองค์กรไปใช้โดยไม่ขออนุญาตจากฝ่ายไอทีขององค์กรได้

2.1.2 หน้าทีของขาสัญญาณในการใช้งานแบบ AVR Microcontroller

- +5V(+Vin) เป็นขาสำหรับใช้เป็นจุดรับแรงดันขนาด +5VDC จากภายนอกเพื่อใช้เป็นแหล่งจ่ายไฟเลี้ยงให้กับบอร์ด
- +VCC(+5V) เป็นขาแหล่งจ่ายไฟจุดเดียวกันกับที่ป้อนให้กับ +VCC ของ MCU ซึ่งจุดนี้จะรับแรงดันมาจาก 2 แหล่งด้วยกันคือขารับแรงดัน+5V(+Vin) จากขา 28 ของบอร์ดและจากขา
- VUSB(+5V)จากขั้ว USB ของบอร์ดโดยมี Diode ป้องกันการย้อนกลับของแรงดันไว้แล้ว
- +AREF เป็นขาสำหรับรับสัญญาณแรงดันอ้างอิง (Analog Reference) ให้กับวงจร Analog Input ในกรณีต้องการใช้แรงดันอ้างอิงจากภายนอก
- RESET# เป็นขาสัญญาณ RESET ของ CPU ทำงานที่ Logic “0”
- PB[0-5] เป็นขา I/O แบบ Digital สามารถใช้งานเชื่อมต่อกับสัญญาณ Logic TTL (5V) ต่างๆ
- PD[0-7] เป็นขา I/O แบบ Digital สามารถใช้งานเชื่อมต่อกับสัญญาณ Logic TTL (5V) ต่างๆ
- PC[0-5] เป็นขา I/O ซึ่งสามารถกำหนดให้เป็นได้ทั้ง Digital และ Analog Input
- ADC6,ADC7 เป็นขา Input แบบ Analog สามารถรับ Input แบบ Analog 0..+5V

2.2 โปรแกรมของไมโครคอนโทรลเลอร์ AVR

ในการพัฒนาโปรแกรมซอฟต์แวร์ของบอร์ด ET-EASY168 STAMP นั้นผู้ใช้สามารถเลือกใช้วิธีการพัฒนาโปรแกรมได้ 2 รูปแบบด้วยกันคือ

1. AVR Micro Controller เป็นการพัฒนาโปรแกรมตามรูปแบบของ AVR Micro Controller ปรกติซึ่งสามารถเลือกใช้โปรแกรมภาษาใดๆก็ได้ที่รองรับการใช้งานร่วมกับ AVR เบอร์ ATmega168 ซึ่งผู้ใช้สามารถเลือกใช้โปรแกรมในการพัฒนาได้ตามความถนัดเช่นภาษาเบสิก BASCOM-AVR หรือภาษาซีเช่น Code Vision และWinAVR เป็นต้น

2. Arduino Project นั้นเป็นการพัฒนาโปรแกรมโดยใช้โปรแกรมและชุดคำสั่งในการเขียนโปรแกรมโดยใช้ภาษาซี (C++) ซึ่งเป็นส่วนของ “Arduino Project” โดยที่เป็นโครงการที่ได้พัฒนาไมโครคอนโทรลเลอร์ AVR แบบ Open Source ซึ่งเป็นสิ่งที่กำลังได้รับความนิยมอย่างแพร่หลายเนื่องจากเป็นโครงการที่เปิดเผยทั้ง Source Code ในการพัฒนาให้ทั้งหมดและยังมีตัวอย่างโครงงานพร้อมตัวอย่างโปรแกรมการทดลองต่างๆแจกจ่ายให้ผู้สนใจนำมาใช้ศึกษาเรียนรู้และทดลองได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆและผู้สนใจสามารถเข้าไปค้นหารายละเอียดต่างๆของArduino Project นี้ได้<http://www.arduino.cc/>

ซึ่งในการทำโครงงานนี้ได้เลือกใช้โปรแกรมArduinoในการเขียนโปรแกรมควบคุมซึ่งเป็นโปรแกรมภาษาซีที่ใช้เขียนเพื่อควบคุมการทำงานของ AVR ซึ่งมีความสะดวกและใช้งานง่าย สามารถรองรับการทำงานได้ทั้ง Windows, Linux, Macintosh OSX มีชุดคำสั่งที่ง่ายต่อการใช้งาน แต่สามารถทำงานที่มีความซับซ้อนมากๆได้ มี Library ให้ใช้งานมากมาย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งการสื่อสารอนุกรม

คำสั่งในกลุ่มนี้จะเป็นการติดต่อสื่อสารระหว่าง AVR กับ PC โดยเมื่อใช้คำสั่งในกลุ่มนี้แล้วจะไม่สามารถใช้คำสั่งสัญญาณ PDO และ PD1 ได้

- **Serial.begin(int speed)** เป็นคำสั่งเปิดการทำงานของพอร์ตอนุกรมระหว่าง AVR กับ PC โดยต้องกำหนดความเร็วในการสื่อสารให้ตรงกันในที่คือ 19200speed คือความเร็วในการสื่อสาร (300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 และ 115200)

2.3 มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง เป็นต้นกำลังขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรม เพราะมีคุณสมบัติที่เด่นในด้าน การปรับความเร็วได้ตั้งแต่ความเร็วต่ำสุดจนถึงสูงสุด นิยมใช้กันมาก ในโรงงานอุตสาหกรรม เช่น โรงงานทอผ้า โรงงานเส้นใยโพลีเอสเตอร์ โรงงานถลุงโลหะหรือให้ เป็นต้นกำลังในการขับเคลื่อนรถไฟฟ้า เป็นต้นในการศึกษาเกี่ยวกับมอเตอร์ไฟฟ้ากระแสตรงจึงควรรู้จัก อุปกรณ์ต่าง ๆ ของมอเตอร์ไฟฟ้ากระแสตรงและเข้าใจถึงหลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบต่าง ๆ

2.3.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงที่ส่วนประกอบที่สำคัญ 2 ส่วนดังนี้

2.3.1.1 สเตเตอร์ (Stator)

ประกอบด้วยเฟรม (Frame) เป็นโครงภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจาก ขั้วเหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่นๆให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนา ม้วนเป็นรูปทรงกระบอก

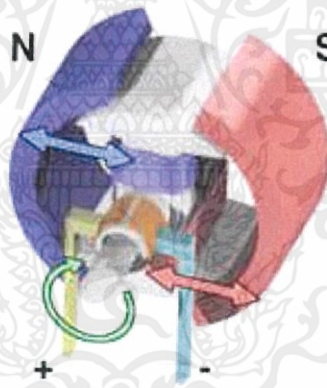


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น กรุณาอย่าให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และนำข้อมูลข้างต้นไปใช้
รูปที่ 2.2 เฟรม



รูปที่ 2.3 ขดลวดพันอยู่รอบขั้วแม่เหล็ก

ขั้วแม่เหล็ก (Pole) ประกอบด้วย 2 ส่วนคือแกนขั้วแม่เหล็กและขดลวดส่วนแรกแกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กบางๆ กั้นด้วยฉนวนประกอบกันเป็นแท่งยึดติดกับเฟรม ส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์เรียกว่าขั้วแม่เหล็ก (Pole Shoes) มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุดเพื่อให้เกิดช่องอากาศน้อยที่สุด เพื่อให้เกิดช่องอากาศน้อยที่สุดจะส่งผลให้เส้นแรงแม่เหล็กจากขั้วแม่เหล็กผ่านไปยังโรเตอร์มากที่สุดแล้วทำให้เกิดแรงบิดหรือกำลังบิดของโรเตอร์มากเป็นการทำให้มอเตอร์มีกำลังหมุน (Torque)

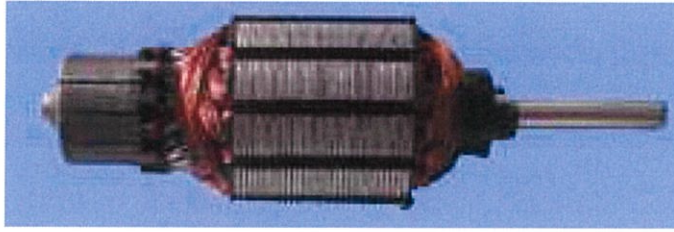


รูปที่ 2.4 ขั้วแม่เหล็ก

ส่วนที่สอง ขดลวดสนามแม่เหล็ก (Field Coil) จะพันอยู่รอบๆแกนขั้วแม่เหล็กขดลวดนี้ทำหน้าที่รับกระแสจากภายนอกเพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้น และเส้นแรงแม่เหล็กนี้จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาร์เมเจอร์ทำให้เกิดแรงบิดขึ้น

2.3.1.2 ตัวหมุน (Rotor)

ตัวหมุนหรือเรียกว่าโรเตอร์ตัวหมุนนี้ทำให้เกิดกำลังงานมีแกนวางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์ เพื่อให้หน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 โรเตอร์

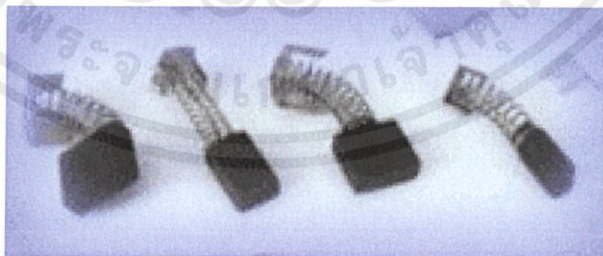
ตัวโรเตอร์ประกอบด้วย 4 ส่วนด้วยกัน คือ

1. แกนเพลลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลลานี้จะวางอยู่บนแบร์ริง เพื่อบังคับให้หมุนอยู่ในแนวตั้งไม่มีการสั่นสะเทือนได้

2. แกนเหล็กอาร์มาเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์มาเจอร์ซึ่งสร้างแรงบิด (Torque)

3. คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่แต่ละซี่มีฉนวนไมก้า (mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์ จะมีร่องสำหรับใส่ปลายสายของขดลวดอาร์มาเจอร์ ตัวคอมมิวเตเตอร์นี้อัดแน่นติดกับแกนเพลลา เป็นรูปกลมทรงกระบอก มีหน้าที่สัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยัง ขดลวดอาร์มาเจอร์เพื่อสร้างเส้นแรงแม่เหล็กอีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับเส้นแรงแม่เหล็กอีกส่วน ซึ่งเกิดจากขดลวดขั้วแม่เหล็ก ดังกล่าวมาแล้วเรียกว่าปฏิกิริยามอเตอร์ (Motor action)

4. ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่องสลอต (Slot) ของแกนอาร์มาเจอร์ ขนาดของลวดจะเล็กหรือใหญ่และจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับการออกแบบของตัวโรเตอร์ชนิดนั้นๆ เพื่อที่จะให้เหมาะสมกับงานต่างๆ ที่ต้องการ ควรศึกษาต่อไปในเรื่องการพันอาร์มาเจอร์ (Armature Winding) ในโอกาสต่อไป



รูปที่ 2.6 แปรงถ่าน (Brushes)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ช่องแปรงถ่าน

ทำด้วยคาร์บอนมีรูปร่างเป็นแท่งสี่เหลี่ยมผืนผ้าในช่องแปรงมีสปริงกดอยู่ด้านบนเพื่อให้ถ่านนี้สัมผัสกับซี่คอมมิวเตเตอร์ตลอดเวลาเพื่อรับกระแส และส่งกระแสไฟฟ้าระหว่างขดลวดอาร์มาเจอร์กับวงจรไฟฟ้าจากภายนอก คือถ้าเป็นมอเตอร์กระแสไฟฟ้าตรงจะทำหน้าที่รับกระแสจากภายนอกเข้าไปยังคอมมิวเตเตอร์ให้ลวดอาร์มาเจอร์เกิดแรงบิดทำให้มอเตอร์หมุนได้

2.3.2 หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action)

หลักการของมอเตอร์ไฟฟ้ากระแสตรง เมื่อเป็นแรงดันไฟฟ้ากระแสตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะ แปรงถ่านผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ใต้ขึ้น จะเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน ตามคุณสมบัติของเส้นแรงแม่เหล็ก จะไม่ตัดกันทิศทางตรงข้ามจะหักล้างกัน และทิศทางเดียวจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวของอาร์มาเจอร์ ซึ่งวางแกนเพลลาและแกนเพลลานี้ สวมอยู่กับตลับลูกปืนของมอเตอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ ขณะที่ตัวอาร์มาเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความว่าตัวหมุน การที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิกริยาต่อกัน ทำให้ขดลวดอาร์มาเจอร์ หรือโรเตอร์ หมุนไปนั้นเป็นไปตามกฎซ้ายของเฟลมมิ่ง (Fleming's left hand rule)

2.3.3 การควบคุมมอเตอร์ไฟฟ้ากระแสตรงด้วยสัญญาณพัลส์วิมมอดูเลชั่น

สัญญาณพัลส์วิมมอดูเลชั่น (Pulse Width Modulation: PWM) คือ การปรับความกว้างของพัลส์โดยการนำเอาสองสัญญาณมาเปรียบเทียบกันคือสัญญาณสามเหลี่ยมกับสัญญาณที่ใช้ปรับความกว้างของพัลส์ โดยอาศัยวงจรเปรียบเทียบแรงดันของไมโครคอนโทรลเลอร์เป็นตัวกำหนดสัญญาณพัลส์วิมมอดูเลชั่นในการควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง

ลักษณะของสัญญาณพัลส์วิมมอดูเลชั่นที่ได้จากไมโครคอนโทรลเลอร์ที่ใช้ในการควบคุม

ความเร็วของมอเตอร์ไฟฟ้ากระแสตรงนั้นสามารถแสดงได้ดังนี้ ไม่นานมานี้เห็นว่าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 สัญญาณพัลส์วิธมอดูเลชันมีความกว้างพัลส์เท่ากับ 10 เปอร์เซ็นต์

เช่น จากรูปที่ 2.8 พบว่าสัญญาณมี duty cycle เป็น 10 เปอร์เซ็นต์ ของคาบสัญญาณทั้งหมด คือ การขับมอเตอร์ด้วยระดับแรงดันไฟฟ้ากระแสตรง 10 เปอร์เซ็นต์ ของระดับแรงดันไฟฟ้าสูงสุดในวงจรขับ



รูปที่ 2.9 สัญญาณพัลส์วิธมอดูเลชันมีความกว้างพัลส์เท่ากับ 50 เปอร์เซ็นต์

หรือ จากรูปที่ 2.9 พบว่าสัญญาณมี duty cycle เป็น 50 เปอร์เซ็นต์ ของคาบสัญญาณทั้งหมด คือ การขับมอเตอร์ด้วยระดับแรงดันไฟฟ้ากระแสตรง 50 เปอร์เซ็นต์ ของระดับแรงดันไฟฟ้าสูงสุดในวงจรขับ



รูปที่ 2.10 สัญญาณพัลส์วิธมอดูเลชันมีความกว้างพัลส์เท่ากับ 90 เปอร์เซ็นต์

หรือ จากรูปที่ 2.10 พบว่าสัญญาณมี duty cycle เป็น 90 เปอร์เซ็นต์ ของคาบสัญญาณทั้งหมด คือ การขับมอเตอร์ด้วยระดับแรงดันไฟฟ้ากระแสตรง 90 เปอร์เซ็นต์ ของระดับแรงดันไฟฟ้าสูงสุดในวงจรขับ

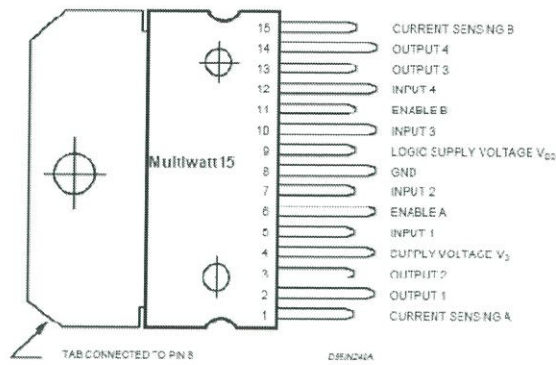
2.4 การขับมอเตอร์ด้วย L298

เนื่องจากในการที่จะควบคุมทิศทาง และความเร็วของมอเตอร์ จำเป็นต้องใช้ IC DRIVER MOTOR ในการขับมอเตอร์ และเลือกใช้ L298 เนื่องจากคุณสมบัติดังต่อไปนี้

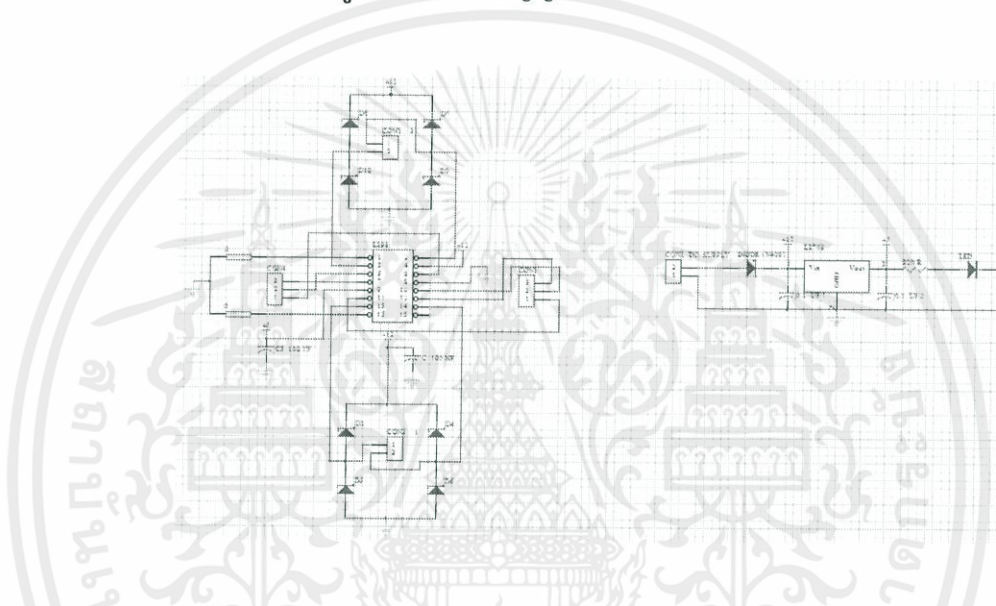
- สามารถรองรับแหล่งจ่ายแรงดันได้ถึง 46 โวลต์
- กระแสไฟฟ้าตรงรวมได้ 4 แอมป์

- ทนต่ออุณหภูมิที่สูงได้

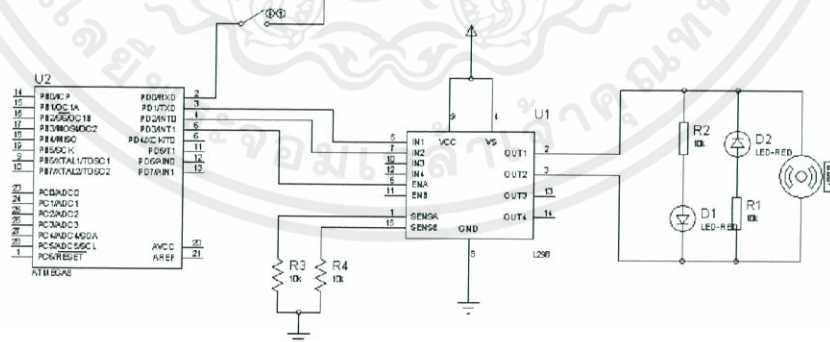
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ขาสัญญาณของไอซี L298



รูปที่ 2.12 วงจรเพื่อทำแผ่น PCB

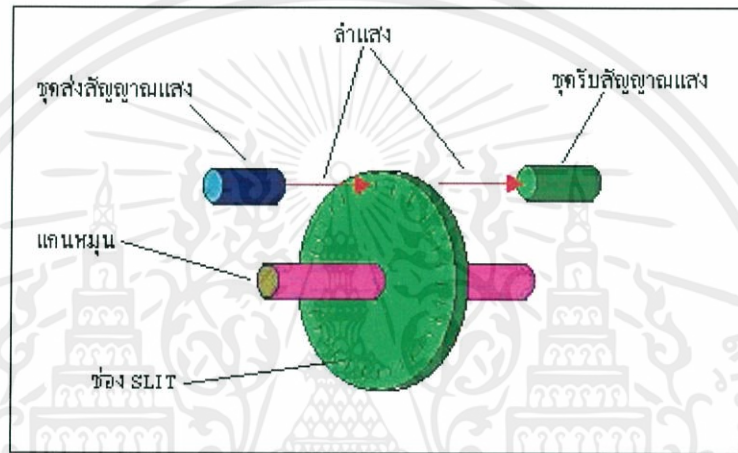


รูปที่ 2.13 วงจรการขับมอเตอร์ด้วยไอซี L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

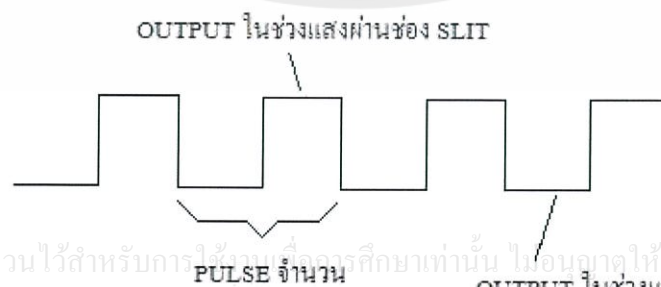
2.5 การควบคุมตำแหน่งของมอเตอร์กับเอ็นโค้ดเดอร์

ในโครงการนี้ได้นำเอา Rotary encoder เข้ามาใช้ และ อ่านค่าออกมาเพื่อตรวจสอบทิศทาง และ ตำแหน่งของการเคลื่อนที่เพื่อนำมาทำการเปรียบเทียบตำแหน่งการเคลื่อนที่กับค่า Input โดย หลักการทำงานของ Encoder มีดังนี้ Rotary encoder มีลักษณะเป็นแผ่นกลมมีแกนอยู่ตรงกลาง และ ที่แผ่นกลมจะมีช่องเล็กๆที่แสงสามารถ ส่องผ่านได้เป็นจำนวนมากเราเรียกช่องนี้ว่า ช่อง slit ซึ่งที่ ด้านหนึ่งของ แผ่นกลมนี้ จะมีตัวส่งแสง Infrared ไปยังตัวรับสัญญาณแสง Infrared ซึ่งจะอยู่ในด้าน ตรงกันข้าม ดังรูปที่ 2.14



รูปที่ 2.14 Rotary encoder

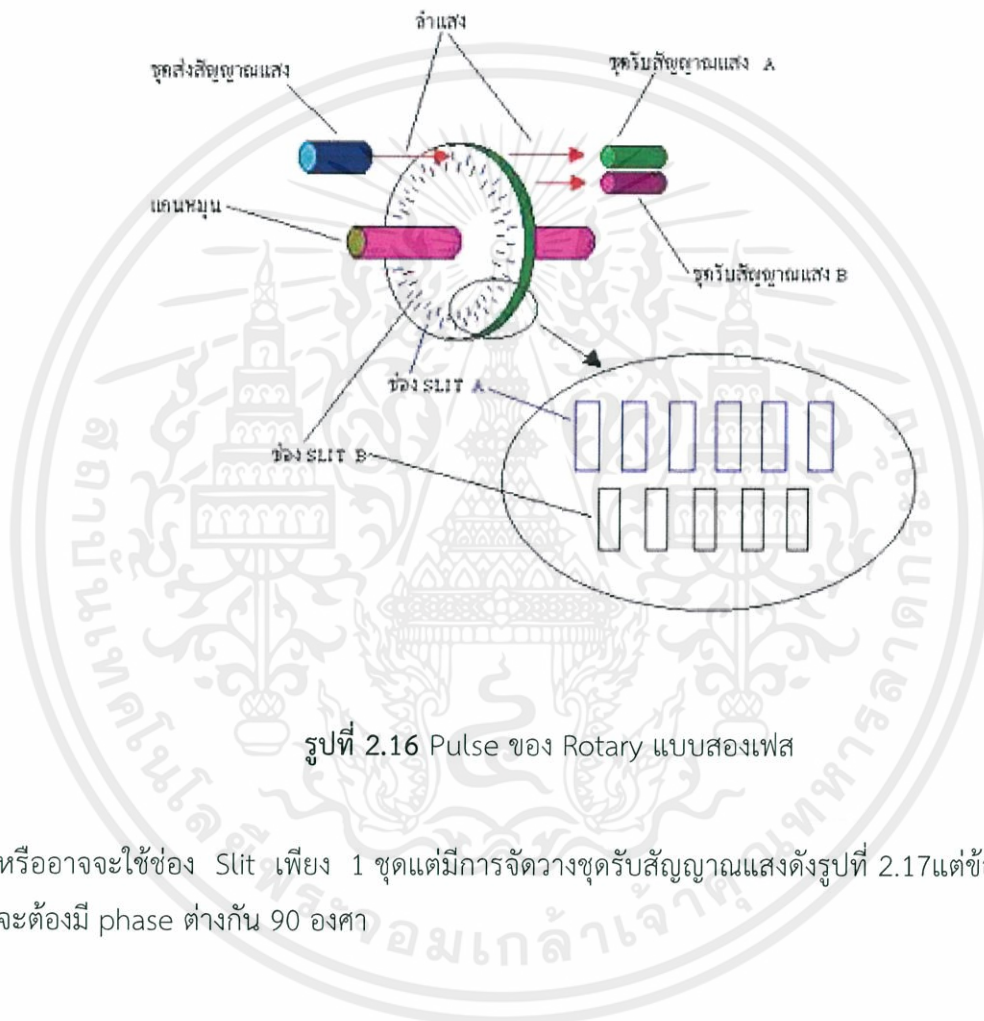
เมื่อหมุนแกนหมุนทำให้แผ่นกลมหมุนไปตัดลำแสง Infrared ดังนั้นชุดรับแสง infrared จึงมี แสงมากระทบเป็นช่วงๆ (pulse) ตามจังหวะที่แสงผ่านช่อง slit จึงทำให้ สัญญาณ output ของ ชุดรับแสง infrared มีลักษณะเป็น pulse



รูปที่ 2.15 การสร้าง Pulse ของ Rotary encoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

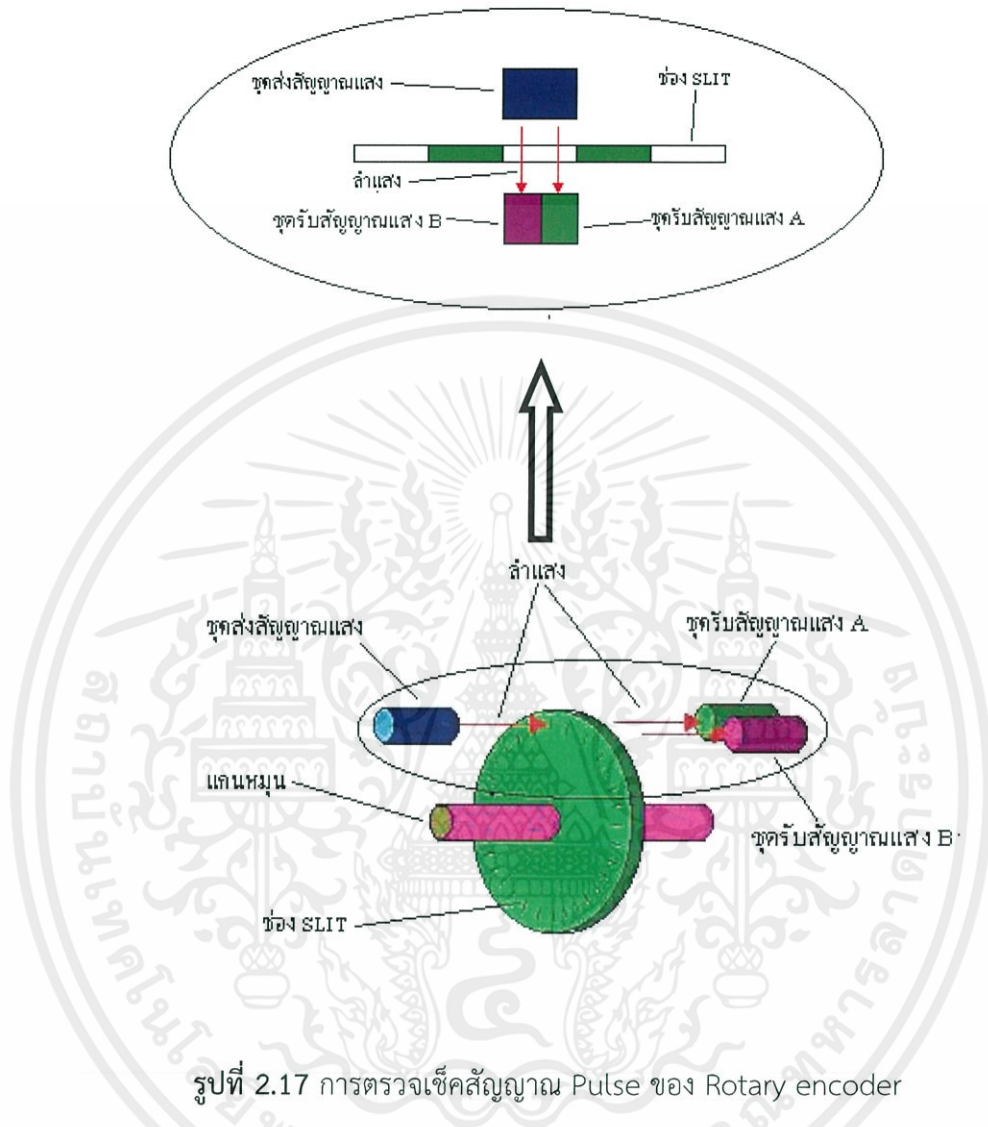
จำนวน pulse ที่ได้ออกมาจะเป็นตัวที่ชี้บ่งว่า motor หมุนไปกี่องศา หรือกี่รอบ ซึ่งเราสามารถคำนวณได้จากสูตร จำนวนรอบที่ motor หมุนไป = จำนวน pulse ค่าความละเอียดของ rotary encoder โดยที่ ค่าความละเอียดของ rotary encoder นั้นนิยมใช้หน่วยเป็น pulse / round หรือppr เช่น 1000 ppr ก็หมายความว่าเมื่อ motor หมุนไป 1 รอบจะมี pulse ออกมา 1000 pulse เป็นต้น ส่วนในเรื่องที่ว่าเราจะทราบได้อย่างไรว่า motor หมุนไปทิศทางใดนั้นส่วนใหญ่แล้วจะใช้วิธีสร้าง ช่อง slit เป็น 2 ชุดเหลื่อมกัน 90 องศา ดังรูปที่ 2.16



รูปที่ 2.16 Pulse ของ Rotary แบบสองเฟส

หรืออาจจะใช้ช่อง Slit เพียง 1 ชุดแต่มีการจัดวางชุดรับสัญญาณแสงดังรูปที่ 2.17 แต่ข้อสำคัญคือจะต้องมี phase ต่างกัน 90 องศา

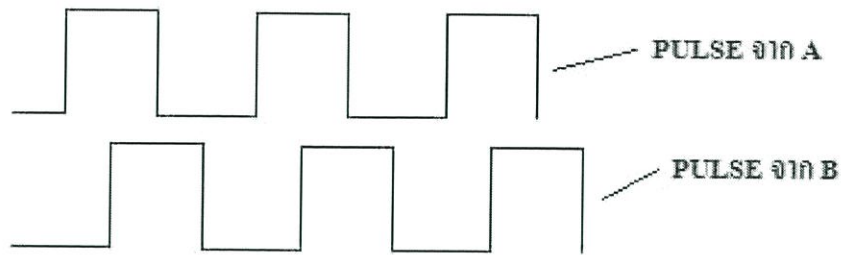
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การตรวจเช็คสัญญาณ Pulse ของ Rotary encoder

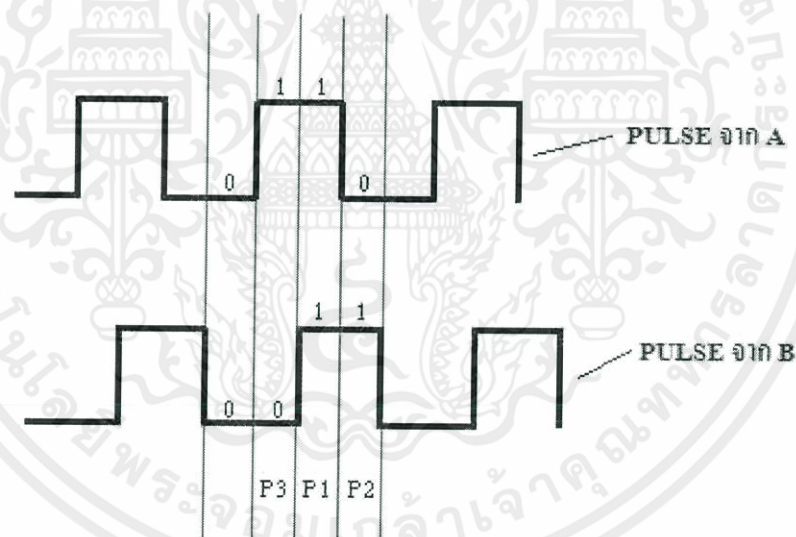
ดังนั้นสัญญาณ output จาก motor จึงมี 2 ชุด คือ A และ B โดยที่สัญญาณ pulse จาก A และ B ก็จะเลื่อนกัน 90 องศาด้วย อาจกล่าว ได้ว่าสัญญาณ output จาก motor มีค่าเท่ากับ 2 bit คือหนึ่ง bit มาจาก A และอีกหนึ่ง bit มาจาก B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 ลักษณะ Pulse ของ Rotary encoder ทั้งสองเฟส

ถ้าเราให้ Pulse ในช่วง high มีค่าเป็น “1” และ pulse ในช่วง low มีค่าเป็น “0” เราสามารถใช้ค่าดังกล่าวมาคำนวณหาทิศทางที่ motor หมุนได้ด้วยใช้วิธีการทาง digital คือการนำค่าที่อ่านได้มาทำการ exclusive OR (XOR) กันโดยการนำ bit ทางขวาของค่าเก่ามา XOR กับ bit ทางซ้ายของค่าใหม่ที่อ่านได้ตามตัวอย่างข้างล่าง



รูปที่ 2.19 ความละเอียดของ Pulse

สมมุติในตอนแรก motor อยู่ที่ตำแหน่ง P1 เพราะฉะนั้นค่าของตัวเลข binary ที่อ่านได้จาก encoder จะมีค่าเป็น 11 ถ้า Servo motor หมุนไปที่ตำแหน่ง P3 จะทำให้ค่าของตัวเลข binary ที่เราอ่านได้จาก encoder จะมีค่าเป็น 10 ดังนั้นถ้าเรานำเอาค่าทางขวา จากค่าที่อ่านได้ในครั้งแรก (11) ซึ่งก็คือ 1 มา XOR กับค่าทางซ้ายที่อ่านได้ในครั้งต่อมา (10) ซึ่งก็คือ 1 จะได้ผลลัพธ์เท่ากับ 0

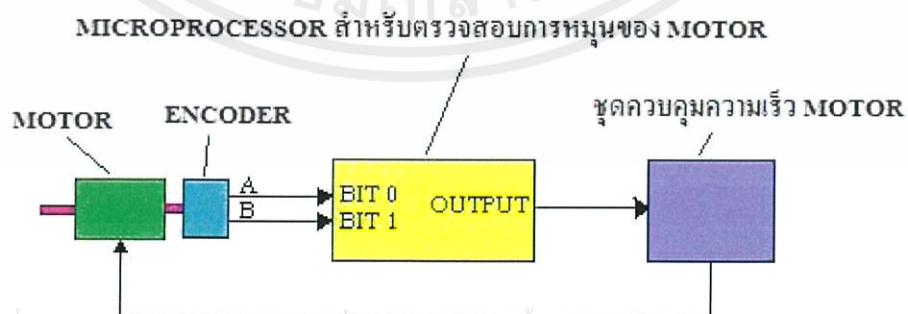
เพราะฉะนั้นค่า 0 จึงเป็นค่าที่บอกว่า motor หมุนไปทางขวา ในทางกลับกันถ้าในตอนแรก motor อยู่ที่ตำแหน่ง P1 และต่อมา motor หมุนไปที่ตำแหน่ง P2 ถ้าเรานำเอา ค่าทางขวา จากค่าที่อ่านได้ในครั้งแรก (11) ซึ่งก็คือ 1 มา XOR กับค่าทางซ้ายที่อ่านได้ในครั้งต่อมา (01) ซึ่งก็คือ 0 จะได้ผลลัพธ์เท่ากับ 1 เพราะฉะนั้นค่า 1 จึงเป็นค่าที่บอกว่า motor หมุนไปทางซ้ายและการ XOR ค่าต่างๆ

Pulse ที่ได้จาก rotary encoder จะถูกนำไปเชื่อมต่อเข้ากับวงจรนับ pulse หรือวงจรคำนวณเพื่อที่จะตรวจสอบว่า motor หมุนไปทิศทางใดและอยู่ที่ตำแหน่งใด ตามวิธีที่ได้กล่าวไว้ข้างต้นโดยส่วนมากแล้ว motor ที่ใช้ rotary encoder แบบนี้ จะมีจำนวนรอบ ของการหมุนที่ไม่จำกัด โดยจะหมุนไปกี่รอบ ก็ได้ ขึ้นอยู่กับ ความสามารถของวงจรนับ plus และวงจรคำนวณ

ดังนั้นเราอาจจะกล่าวได้ว่า encoder ที่ใช้กันอยู่โดยทั่วไปนั้นมีอยู่ 2 แบบใหญ่ๆคือเป็นแบบมีค่า output ที่ไม่จำกัด (incremental encoder) และแบบที่มีค่า output ที่จำกัด (absolute encoder) Incremental encoder มักจะเป็น encoder แบบ rotary encoder ดังที่ได้กล่าวไว้ในตอนต้นแล้วว่าขนาดของ output ของ encoder แบบนี้ขึ้นอยู่กับารออกแบบวงจรนับและวงจรคำนวณ

Absolute encoder ที่พบเห็นส่วนใหญ่จะเป็นแบบ potentiometer แต่ในบางครั้งก็อาจเป็นแบบ rotary encoder ก็ได้แต่จะมีการออกแบบลักษณะของช่อง slit ที่ต่างไปจาก rotary encoder ที่ได้ไว้ในตอนต้นโดยอาจจะมีค่าของ output ตั้งแต่ 4 bit ถึง 16 bit

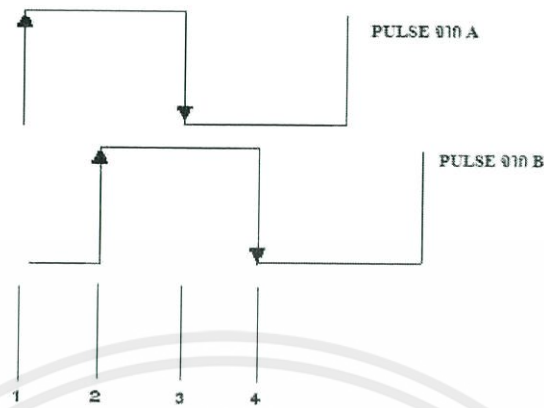
ในแง่ของการใช้งานแล้ว motor ที่มี encoder แบบ incremental encoder นั้นจะใช้งานยุ่งยากกว่า motor ที่มี encoder แบบ absolute encoder เพราะ output ของ increment encoder นั้นไม่สามารถเชื่อมต่อกับชุดควบคุม motor (controller) ได้ทันทีโดยต้องต่อผ่านวงจรนับ pulse และวงจรคำนวณที่ยุ่งยากซับซ้อนก่อนถึงจะต่อเข้ากับกับ controller ได้ แต่อย่างไรก็ตาม ในปัจจุบันเรามี microprocessor เช่น microcontroller ตระกูล PIC เป็นต้น ดังนั้นเราจึงสามารถนำมาใช้แทนวงจรนับ pulse และวงจรคำนวณได้จึงทำให้การใช้งานมีความง่ายขึ้นดังรูปที่ 2.25 แสดงไว้ข้างล่าง



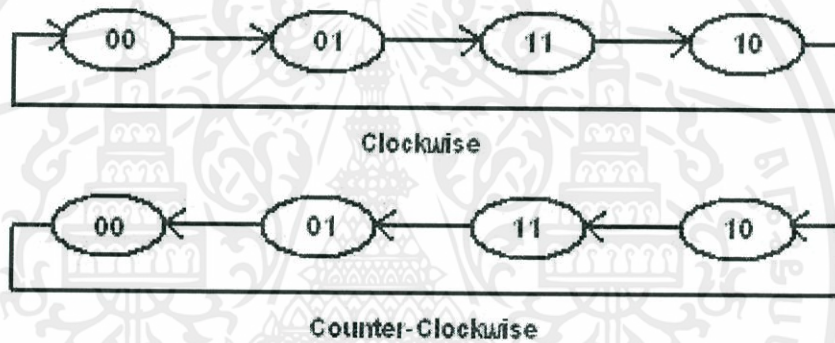
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.20 Block Diagram ของ Rotary encoder

ในโครงการอาศัยการตรวจนับขอบขาขึ้นและขาลงของทั้งเฟส A และเฟส B ทำให้สามารถแบ่ง 1 Pulse เป็น 4 ส่วน



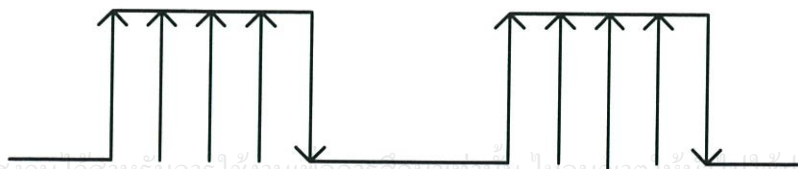
รูปที่ 2.21 การตรวจนับขอบขาขึ้นและขอบขาลงของทั้ง 2 เฟส



รูปที่ 2.22 วิธีการตรวจนับทิศทางการหมุนของ Rotary encoder

โดยอ่านค่าเข้ามาทาง Port1

- ในการนับจำนวน Pulse จะทำการนับขอบขาขึ้นของ Pulse A ว่าเท่าไร
- ในการหาความกว้างของ Pulse จะทำการเช็คขอบขาขึ้นของ A กับขอบขาลงของ A ว่ามีระยะห่างระหว่างกันเท่าไร โดยใช้ Timer เป็นตัวจับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.23 การเช็คความกว้างของ Pulse

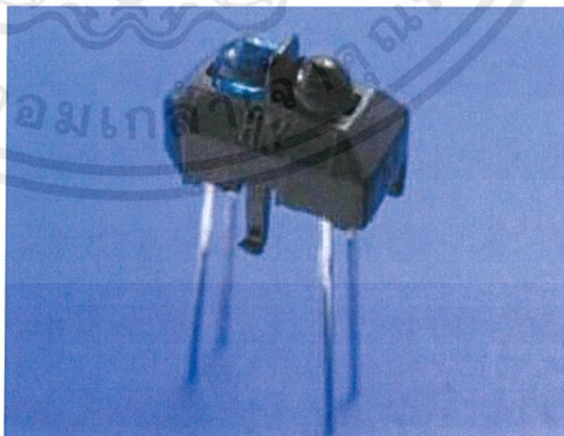
2.6 ทฤษฎีและหลักการการทำงานของเซนเซอร์

เซนเซอร์ (Sensor) โดยทั่วไปคืออุปกรณ์ที่ใช้ตรวจจับสัญญาณหรือปริมาณทางฟิสิกส์ต่างๆ เช่น อุณหภูมิ เสียง แสง แรงทางกล (Force) ความดันบรรยากาศ (Pressure) ระยะกระจัด (displacement) ความเร็ว (Speed) อัตราเร่ง (Acceleration) ระดับของเหลว (Liquid level) และ อัตราการไหล (Flow rate) จากนั้นจะทำหน้าที่เปลี่ยนเป็นสัญญาณออกหรือปริมาณเอาต์พุตที่ได้จากการวัดในอีกรูปแบบหนึ่งที่สามารถนำไปประมวลผลต่อได้

ซึ่งปัจจัยในการเลือกเซนเซอร์ใช้งานขึ้นอยู่กับปริมาณธรรมชาติของปริมาณทางฟิสิกส์ที่จะทำการวัดและควบคุมค่าเป็นสำคัญ รวมไปถึงราคาความน่าเชื่อถือตลอดจนคุณภาพของข้อมูลที่ทำกรวัด นอกจากนี้ยังมีปัจจัยสำคัญอื่นที่ควรพิจารณาอีกเช่นความเหมาะสมของเซนเซอร์ที่จะนำไปใช้ในสภาพแวดล้อมนั้นๆ ยกตัวอย่างเช่นเซนเซอร์ตรวจจับอุณหภูมิที่ถูกออกแบบให้ใช้งานในบ้านพักอาศัยทั่วไปจะมีความแตกต่างและไม่สามารถนำไปใช้แทนเซนเซอร์วัดอุณหภูมิในโรงงานผลิตรถยนต์ได้ ทั้งนี้เนื่องจากเซนเซอร์ตรวจจับอุณหภูมิซึ่งใช้ในกระบวนการผลิตต่างๆ ในโรงงานนั้นจำเป็นต้องมีอัตราความสามารถในการทนต่อสภาวะที่อุณหภูมิสูง ความดันสูงหรือสามารถทนต่อการกัดกร่อนได้สูงกว่าเซนเซอร์ที่ถูกออกแบบให้ใช้งานทั่วไป

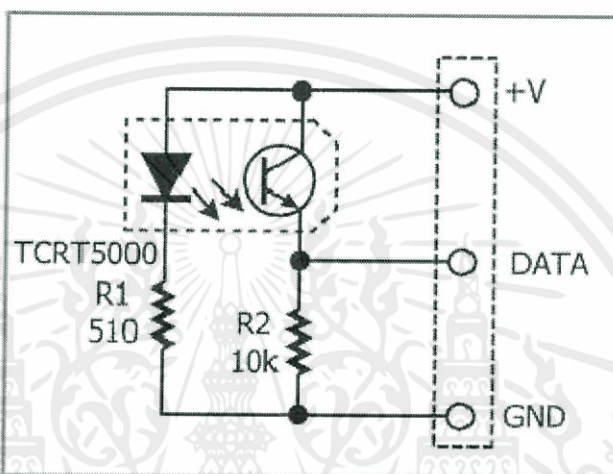
โดยทั่วไปเทคโนโลยีของเซนเซอร์ได้ถูกนำไปใช้เป็นองค์ประกอบหลักที่สำคัญในลักษณะงาน 2 ประเภทคือ

1. ใช้ตรวจจับปริมาณทางฟิสิกส์เพื่อนำไปแสดงผลการตรวจวัดหรือจัดเก็บบันทึกเป็นข้อมูลในระบบการวัด
2. ใช้ตรวจสอบสภาพกระบวนการในระบบการควบคุมซึ่งเซนเซอร์สำหรับการตรวจวัดข้อมูลที่เป็นตัวแปรทางฟิสิกส์โดยส่วนมากจะถูกนำไปใช้เป็นข้อมูลเพื่อแสดงสถานะสภาพของระบบในขณะนั้นเช่น เซนเซอร์วัดความเร็วในรถยนต์และมิเตอร์วัดความเร็วเป็นต้น แต่ในบางครั้งเซนเซอร์อาจจะใช้สำหรับการบันทึกข้อมูลเพื่อใช้สำหรับการบันทึกข้อมูลแสดงสมรรถนะของระบบได้เช่นกัน เช่น ทาโคกราฟ (Tachograph) ที่บันทึกข้อมูลแสดงเป็นกราฟของความเร็วเทียบกับเวลาในรถยนต์หรือรถบรรทุกเป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้รูปที่ 2.24 เซนเซอร์ตรวจจับแสงอินฟราเรดรุ่น TCRT5000 การนำไปใช้

เซนเซอร์ที่ใช้เป็นเซนเซอร์ตรวจจับแสงอินฟราเรด (Infrared) รุ่น TCRT5000 ดังรูปที่ 2.24 โดยแสงอินฟราเรดคือแสงที่มีความยาวคลื่นต่ำกว่าแสงสีแดงลงไปดังนั้นจึงไม่สามารถมองเห็นได้ด้วยสายตาของมนุษย์ซึ่งคุณสมบัตินี้เองจึงทำให้เซนเซอร์ชนิดที่ใช้แสงอินฟราเรดเป็นที่นิยมนำมาใช้กันมากโดยจะอาศัยหลักการของการสะท้อนของแสงกล่าวคือใช้อุปกรณ์ส่งแสงเป็นแหล่งกำเนิดปล่อยแสงออกไปและเมื่อแสงกระทบกับวัตถุด้านหน้ามันก็จะสะท้อนแสงกลับมาเข้าที่ตัวรับแสงส่วนอัตราของการสะท้อนกลับนั้นขึ้นอยู่กับสีและสภาพความมันของวัตถุที่สะท้อนเช่นสีดำจะมีอัตราการสะท้อนกลับน้อยกว่าสีขาวหรือสภาพพื้นผิวที่มีความราบเรียบเป็นมันวาวจะสามารถสะท้อนแสงได้ดีกว่าพื้นผิวที่มีลักษณะด้านและขรุขระ เป็นต้น



รูปที่ 2.25 วงจรเซนเซอร์ตรวจจับแสงอินฟราเรดรุ่น TCRT5000

หลักการของเซนเซอร์ตรวจจับแสงอินฟราเรดรุ่น TCRT5000

จากวงจรเซนเซอร์ตรวจจับแสงอินฟราเรดรุ่น TCRT5000 ดังรูปที่ 2.25 จะประกอบไปด้วยขาจำนวน 3 ขาซึ่งได้แก่ขาไฟเลี้ยง (+5V) ขากราวด์และขาสัญญาณเซนเซอร์ชนิดนี้จะมีหลักการของมันก็มีแค่ส่งแสงอินฟราเรดไปยังวัตถุที่ต้องการตรวจจับถ้าพบวัตถุนั้นก็จะสะท้อนแสงกลับมายังตัวรับสีที่สะท้อนได้ดีที่สุดก็คือสีขาวถ้าเป็นสีดำจะดูดกลืนได้มากกว่าเซนเซอร์อะไรก็แล้วแต่มีหลักการเหมือนกันคือต้องมีตัวส่งสัญญาณแล้วก็ตัวรับสัญญาณแล้วก็เอาสัญญาณที่ได้จากตัวรับมาประมวลผลคราวนี้ก็ขึ้นอยู่กับว่าจะเอาเซนเซอร์นั้นไปใช้งานทำอะไรโดยเซนเซอร์แสงสามารถทำได้หลายแบบแต่ละแบบทำงานในลักษณะเดียวกันคือมีตัวส่งแสงอินฟราเรดไปกระทบวัตถุถ้าวัตถุนั้นสามารถสะท้อนแสงได้ดีแสงดังกล่าวก็จะมาเป็นอินพุตให้อุปกรณ์รับแสงซึ่งอาจจะเป็นโฟโตไดโอดโฟโตทรานซิสเตอร์หรือแอลดีอาร์ก็ได้หรืออาจจะใช้อุปกรณ์เซ็นเซอร์แสงแบบสำเร็จก็มีขายคือมีทั้งตัวส่งและตัวรับในตัวถึงเดียวกันแต่ถ้าวัตถุไม่สะท้อนแสงหรือสะท้อนแสงได้น้อยเช่นวัตถุสีดำตัวรับก็จะไม่นำกระแสหรือนำกระแสเพียงเล็กน้อยผลจากการเปลี่ยนแปลงของเซ็นเซอร์สามารถนำไปเข้าวงจรไม่ว่ากรณีเปรียบเทียบแรงดันให้เป็นค่าลอจิกซึ่งจะง่ายกับการนำไปควบคุมงานที่ได้ออกแบบไว้การใช้

อินฟราเรดเซนเซอร์สิ่งที่ควรคำนึงมากอีกสิ่งหนึ่งคือแสงอินฟราเรดจากภายนอกซึ่งมีอยู่ทั่วไปและมีมากในตอนกลางวันและวันที่ฟ้าโปร่ง

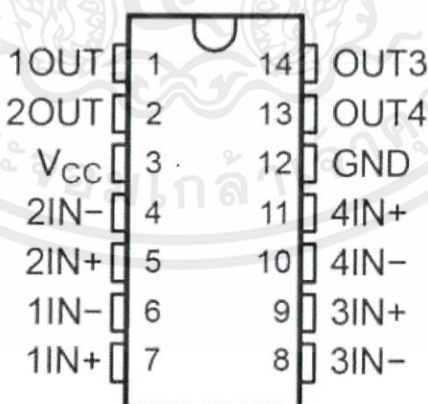
2.7 การควบคุมการทำงานเซนเซอร์ด้วยไอซี LM339

เนื่องจากในการควบคุมการทำงานเซนเซอร์ เลือกใช้ไอซี LM339 ซึ่งมีคุณสมบัติดังนี้

- รับแรงดันได้ 36 โวลต์
- ภายในมีออปแอมป์ 4 ตัว
- สามารถควบคุมเซนเซอร์ได้ 4 ตัว
- สามารถใช้งานร่วมกับ ตัวต้านทานปรับค่าได้ เพื่อปรับระยะเซนเซอร์



รูปที่ 2.26 ไอซี LM339



รูปที่ 2.27 ตำแหน่งขาสัญญาณของไอซี LM339

2.8 โปรแกรม Visual C#

โปรแกรม Visual C# เป็นโปรแกรมที่เขียนด้วยภาษาซีชาร์ปซึ่งมีต้นกำเนิดมาจากภาษาซีแต่ภาษาซีชาร์ปจะก้าวหน้ากว่ามากและยังได้รวบรวมข้อดีจากภาษาต่างๆเช่น Java, Delphi, C++ เข้า

ไว้ด้วยกันอีกทั้งยังมีความเรียบง่ายกว่าและมีเครื่องมือช่วยเหลือมากมายจึงสะดวกต่อการสร้างและพัฒนาโปรแกรม

2.8.1 ทฤษฎีและหลักการการเขียนโปรแกรมควบคุมการทำงานของเครื่อง

ในส่วนนี้ใช้โปรแกรม visual C# ในการเขียนโปรแกรมควบคุมและเชื่อมการทำงานของส่วนต่างๆ เนื่องจากการเขียนโปรแกรมส่วนใหญ่จะอาศัยเงื่อนไขหรือข้อกำหนดบางอย่างในการทำงาน ดังนั้นสิ่งที่เราจำเป็นต้องรู้จักคือจะกำหนดเงื่อนไขอย่างไรในการควบคุมโดยใช้คำสั่งต่อไปนี้เช่น

คำสั่ง If เป็นการเปรียบเทียบว่าตรงกับเงื่อนไขที่ต้องการหรือไม่หากตรงจะกระทำตามคำสั่ง หากไม่ตรงจะกระทำตามคำสั่งหากไม่ตรงจะข้ามไปยังคำสั่งถัดไป

คำสั่ง If...else เป็นการกำหนดทางเลือกอื่นในการทำงานนอกเหนือจากที่กำหนดใน if โดยที่หากเงื่อนไขไม่ตรงกับที่ระบุไว้ใน if โปรแกรมจะทำตามคำสั่งที่ระบุไว้ในส่วน else

คำสั่ง else if ในกรณีที่มีการตรวจสอบหลายๆเงื่อนไขทำให้ไม่สะดวกต่อการใช้ if หลายๆ ครั้งเราสามารถใช้อ else if เข้ามาช่วยตรวจสอบในแต่ละเงื่อนไขได้

การใช้คำสั่ง if ซ้อนกันในการทำงานของโปรแกรมบางครั้งการที่จะทำคำสั่งใดหรือไม่นั้น อาจขึ้นอยู่กับเงื่อนไขมากกว่า 1 อย่างหรือกล่าวอีกอย่างก็คือเงื่อนไขบางอย่างอาจมีเงื่อนไขแยกย่อยอีกมากมายเราอาจจำเป็นต้องใช้คำสั่ง if, else if และ else ซ้อนกันหลายๆตัวเพื่อการตรวจสอบเงื่อนไขเหล่านั้น

คำสั่ง switch...case ในกรณีที่เรามีเงื่อนไขที่ต้องตรวจสอบอยู่หลายเงื่อนไขบางครั้งหากเราใช้ else if อาจไม่สะดวกจึงมีการตรวจสอบเงื่อนไขในอีกลักษณะหนึ่งเพิ่มขึ้นมานั้นคือ switch case

คำสั่ง for เป็นการทำงานแบบซ้ำๆตามเงื่อนไขที่กำหนดโดยรูปแบบ for จะต้องมีส่วนที่ซ้ำๆ ทำหน้าที่เป็นตัวนับสำหรับกำหนดค่าเริ่มต้นและค่าสุดท้ายของการทำซ้ำ

คำสั่ง while ในการใช้รูปแบบ for เราจำเป็นต้องทราบค่าเริ่มต้นและค่าสุดท้ายของรูปแบบ แต่ในบางกรณีเราอาจไม่ทราบค่าที่แน่นอนของค่าเริ่มต้นและค่าสุดท้ายของรูปแบบแต่อาจทราบเพียงเงื่อนไขบางอย่างเท่านั้นด้วยเหตุนี้ภาษาคอมพิวเตอร์โดยทั่วไปจึงมีทางเลือกในการวนลูปโดยใช้วิธีกำหนดเงื่อนไขนั้นก็คือรูปแบบ while

คำสั่ง do...while รูปแบบ do...while นั้นจะมีการตรวจสอบเงื่อนไขที่ท้ายลูปแสดงว่า จะต้องมีการทำงานตามคำสั่งอย่างน้อย 1 ครั้ง ดังนั้นรูปแบบนี้จึงเหมาะกับงานที่ต้องการกระทำ บางอย่างไปก่อนในครั้งแรกแล้วค่อยตรวจสอบเงื่อนไขภายหลัง

คำสั่ง break เป็นคำสั่งในการหยุดทำคำสั่งส่วนที่เหลือต่อหรืออาจเป็นการออกจากลูปแม้ว่า จะยังวนลูปไม่ครบตามเงื่อนไขที่กำหนดก็ตามคำสั่งนี้อาจใช้ร่วมกับคำสั่ง switch, for, while, do...while โดยระบุคำสั่งนี้ณจุดที่ต้องการออกจากลูป

คำสั่ง continue จะใช้ในทางตรงกันข้ามกับ break คือให้วนลูป for, while หรือ do...while ถัดไปโดยไม่มีคำสั่งส่วนที่อยู่ต่อจาก continue

บล็อกและขอบเขตของตัวแปรการประกาศตัวแปรใน C# นั้นไม่จำเป็นต้องประกาศตัวแปร ทุกๆตัวที่ต้องใช้ไว้ล่วงหน้าเสมอไปแต่สามารถประกาศไว้ ณ จุดที่ต้องการใช้งานก็ได้เพราะบางครั้งการ ไม่ว่ากรณีที่จะต้องใช้ตัวแปรตัวใดตัวหนึ่งหรือไม่อาจขึ้นกับเงื่อนไขแต่สิ่งที่ต้องคำนึงก็คือหากเราประกาศตัวแปร

ไว้ในบล็อกของเงื่อนไขหรือภายในบล็อกของลูปแล้วเราจะนำตัวแปรไปใช้นอกบล็อกที่มันถูกสร้างขึ้น จะเกิดการผิดพลาดได้

2.8.2 การใช้งาน visual c#

1. เปิด visual studio ขึ้นมาแล้วเลือก File->New->Project->Windows Form Application

2. จากนั้นก็จะเข้าสู่หน้า form ของ window ให้คลิกเลือก form แล้วในแถบ properties (ถ้าไม่เจอให้ไปเลือก View->Properties window)

3. จากนั้นในแถบ properties ให้เลือก Events (รูปสายฟ้า) แล้วเลือก event ที่ชื่อว่า Paint ให้ดับเบิลคลิกที่ event นี้แล้วโปรแกรมจะแสดงหน้า Form1.cs โดยมี method ถูกสร้างขึ้นที่ชื่อว่า Form1_Paint

4. ใน method นี้สามารถเขียนโปรแกรมจัดการทางด้านการฟิกได้เช่นต้องการวาดรูปเส้นตรงก็เขียนโค้ดดังนี้

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawLine(Pens.Blue, 10, 10,this.Width,this.Height);
}
```

5. จากนั้นทดสอบrunดูจะแสดงผลลัพธ์ในตัวอย่างเป็นการสร้างออปเจกต์ g ของคลาส Graphics จากนั้นใช้ออปเจกต์นี้เรียก method DrawLineเพื่อวาดรูปใน method นี้มีพารามิเตอร์ 4 ตัวคือสีของเส้นในที่นี้ใช้สีน้ำเงินตำแหน่งการวาดรูป x1, y1, x2, y2
ตัวอย่างคำสั่งที่ใช้โปรแกรม Visual C# ในการโหลดรูปมาเพื่อทำเป็นภาพเคลื่อนไหวตัวอย่างในการใช้คำสั่งดังนี้

```
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void panel1_Paint(object sender, PaintEventArgs e)
        {
            Image img = Image.FromFile("D:/ชื่อโฟลเดอร์/ชื่อรูปภาพ.jpg"); // เลือกส่วนที่
            เก็บรูปภาพ
            e.Graphics.DrawImage(img, 0, 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2.1 การใช้งาน Pen และ Brush

Pen ทาหน้าที่ในการวาดภาพต่างๆเช่นเส้นตรงเส้นโค้งและอื่นๆโดยการสร้าง pen มีคำสั่งดังนี้

```
Pen p = new Pen(Brush brush); //คำอธิบายรูปแบบนี้ต้องกำหนดรูปแบบของ Brush
หรือ
```

```
Pen p = new Pen(Color color); //คำอธิบายรูปแบบนี้เป็นการกำหนดสีของเส้นที่
ต้องการจะวาด
```

หรือ

```
Pen p = new Pen(Brush brush.float width); //คำอธิบายรูปแบบนี้ต้องกำหนดชนิด
ของ Brush และขนาดของเส้นที่ต้องการวาด
```

หรือ

```
Pen p = new Pen(Color color.float width); //คำอธิบายรูปแบบนี้ต้องกำหนดสีของ
เส้นและขนาดของเส้น
```

ตัวอย่างการใช้งาน Pen

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics; Pen p1 = new Pen(Color.Blue, 4);
    g.DrawLine(p1, 10, 10, 80, 80); Pen p2 = new Pen(Color.Red, 4);
    g.DrawLine(p2, 80, 30, 100, 100);
}
```

ผลลัพธ์จะได้เส้นตรงที่มีความกว้าง 4px สองเส้นคือเส้นสีน้ำเงินและแดง

2.8.2.2 การใส่สีพื้นหลัง

ใช้ออปเจกต์ของคลาส Graphics เรียก method FillRectangleดังนี้

```
g.FillRectangle(Brushes.Yellow,this.ClientRectangle);ในFillRectangleนี้จะมี
```

พารามิเตอร์คือสีที่ต้องการใส่ในตัวอย่างนี้ใช้สีเหลืองและthis.ClientRectangleหมายถึงให้ใส่สีพื้นหลังทั้งหมดในกรอบของ window ที่แสดงผล

ตัวอย่างการใส่สีพื้นหลัง

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics; g.FillRectangle(Brushes.Yellow, this.ClientRectangle);
    Pen p1 = new Pen(Color.Blue, 4); g.DrawLine(p1, 10, 10, 80, 80);
    Pen p2 = new Pen(Color.Red, 4); g.DrawLine(p2, 80, 30, 100, 100);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น 2.8.2.3 การใช้งาน Brush เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Brush ใช้กำหนดสี, รูปร่างต่างๆให้กับออปเจกต์ที่เราต้องการ Brush มีคลาสต่างๆดังนี้

- HatchBrushสำหรับใส่ pattern – PathGradientBrushสำหรับกำหนด path แบบไล่สี

- LinearGradientBrushสำหรับกำหนดเส้นตรงแบบไล่สี – SolidBrushสำหรับกำหนดสีแบบ Solid (สีทึบ) – TextureBrushสำหรับใส่ภาพ
รูปแบบการใช้งาน Brush ต่างๆ

1. SolidBrushมีคำสั่งสร้างคลาสสำหรับ Brush ชนิดนี้ดังนี้

```
SolidBrush s = new SolidBrush(Color color);
```

ตัวอย่างการใช้งานSolidBrush

```
private void Form1_Paint(object sender, PaintEventArgs e)
```

```
{
```

```
    Graphics g = e.Graphics;
```

```
    g.FillRectangle(Brushes.Teal, this.ClientRectangle);
```

```
    SolidBrush s = new SolidBrush(Color.Plum);
```

```
    g.FillEllipse(s, 40, 40, 100, 100);
```

```
}
```

Brush ต่างจาก Pen ตรงที่ว่าเป็นการกระทำกับทั้งรูปหรือ shape เช่นไล่สีทั้งรูปส่วน Pen จะกระทำกับเส้น (Line)

2. TextureBrush Brush ชนิดนี้เป็นการใส่รูปภาพต่างๆให้กับ shape คำสั่งในการสร้าง TextureBrushคือ

```
TextureBrush t =new TextureBrush(Image);
```

Image นี้เป็นรูปภาพที่จะใส่ให้กับTextureBrushซึ่งจะต้องกำหนด image ก่อนที่จะใช้ brush ดังนี้

```
Bitmap bmp =new Bitmap("picture1.gif");
```

```
TextureBrush t =new TextureBrush(bmp);
```

ตัวอย่างการใช้งานTextureBrush

```
private void Form1_Paint(object sender, PaintEventArgs e)
```

```
{
```

```
    Graphics g = e.Graphics;
```

```
    g.FillRectangle(Brushes.Teal, this.ClientRectangle);
```

```
    g.FillRectangle(t, 40, 40, 100, 100); }
```

3. LinearGradientBrushมีรูปแบบดังนี้

```
LinearGradientBrush l =new LinearGradientBrush(Point1,Point2,Color1,Color2) ;
```

โดย Point1 เป็นจุดเริ่มไล่สีและ Point2 เป็นจุดสุดท้าย Color1 เป็นสีเริ่มต้นที่จะทำการไล่สี Color2 เป็นสีสุดท้าย

ตัวอย่างการใช้งานLinearGradientBrush

```
private void Form1_Paint(object sender, PaintEventArgs e)
```

```
{
```

```
    Graphics g = e.Graphics;
```

```
    g.FillRectangle(Brushes.Teal, this.ClientRectangle);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Bitmap bmp =new Bitmap("star.gif");
Point p1 =new Point(30,30);
Point p2 =new Point(130,130);
LinearGradientBrush l = new LinearGradientBrush(p1, p2, Color.Peru,
Color.BurlyWood);
g.FillRectangle(l, 0, 0, this.Width, this.Height); }

```

4. การใช้งานHatchBrushมีรูปแบบดังนี้

```
HatchBrush h =new HatchBrush(HatchStylestyle,Colorforecolor);
```

หรือ

```
HatchBrushh=new HatchBrush(HatchStylestyle,Colorforecolor,Colorbackcolor);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การปรับปรุงและพัฒนา

3.1 โครงสร้างโดยรวมของระบบลำเลียง

สำหรับระบบลำเลียงนั้น ได้แบ่งส่วนประกอบต่างๆ ออกเป็นส่วนย่อยๆ ซึ่งในแต่ละส่วนจะมีหน้าที่ หลักการทำงาน และการสื่อสารที่แตกต่างกันโดยจะสามารถแบ่งส่วนประกอบต่างๆ ของระบบลำเลียงออกได้เป็น 3 ส่วน ดังนี้

3.1.1 ส่วนของระบบลำเลียงขาขึ้น (Part A)

ในส่วนของระบบลำเลียง (Part A) นั้นมีวงจรถับ เพื่อให้สายพานลำเลียงหมุน วงจรถวจจับทำหน้าที่ตรวจจับวัตถุที่เข้ามา โดยใช้ไมโครคอนโทรลเลอร์ (ET-EASY 168 STAMP) ในการสั่งการและส่งค่า ลักษณะของระบบลำเลียงเป็นดังรูปที่ 3.1

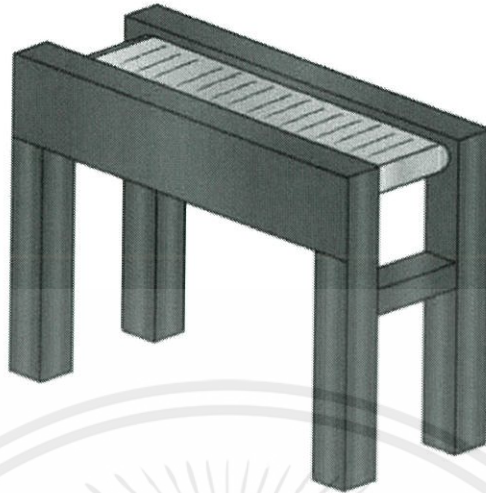


รูปที่ 3.1 ลักษณะของระบบลำเลียง (Part A)

3.1.2 ส่วนของระบบลำเลียง (Part B, D และ E)

ในส่วนของระบบลำเลียง (Part B, D และ E) นั้นมีวงจรถับ เพื่อให้สายพานลำเลียงหมุน วงจรถวจจับทำหน้าที่ตรวจจับสินค้าเพื่อให้สายพานหมุนและหยุดหมุน โดยใช้ไมโครคอนโทรลเลอร์ (ET-EASY 168 STAMP) ในการสั่งการและส่งค่า ลักษณะของระบบลำเลียงเป็นดังรูปที่ 3.2

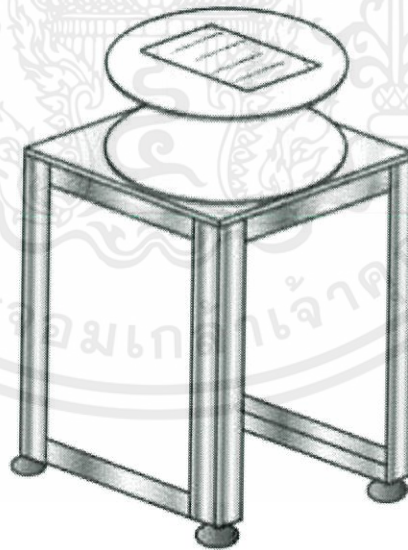
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ลักษณะของระบบลำเลียง (Part B, D และ E)

3.1.3 ส่วนคัตแยก (Part C)

ในส่วนคัตแยกนั้น มีวงจรถับเพื่อให้สายพานหมุน วงจรถองจับทำหน้าที่ตรวงจบบเพื่อคัตแยกวัตถุส่งไปยัง Part D และ E โดยใช้ไมโครคอนโทรลเลอร์ (ET-EASY 168 STAMP) ในการรับสัญญาณและสั่งการ ลักษณะของส่วนคัตแยกเป็นดังรูปที่ 3.3

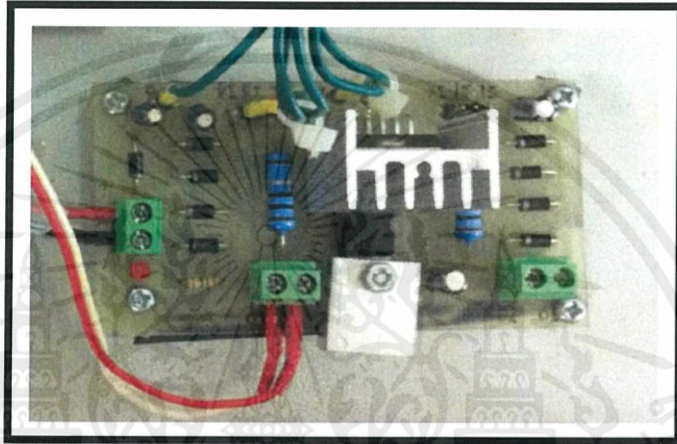


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.3 ลักษณะของส่วนคัตแยก (Part C) ให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจรไฟฟ้าโดยรวมของระบบลำเลียง

3.2.1 วงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง

การออกแบบวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงเพื่อให้สามารถควบคุมความเร็วของมอเตอร์และทิศทางของมอเตอร์โดยมีอุปกรณ์ที่ช่วยในการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงคือไอซี เบอร์ L298 เนื่องจากไอซีเบอร์ดังกล่าว สามารถทนกระแสไฟฟ้าได้มาก จึงเหมาะสำหรับระบบสายพานลำเลียงที่มีการเปิด-ปิดบ่อยครั้ง เพื่อให้มอเตอร์ทำงานได้หลายๆครั้งติดต่อกัน

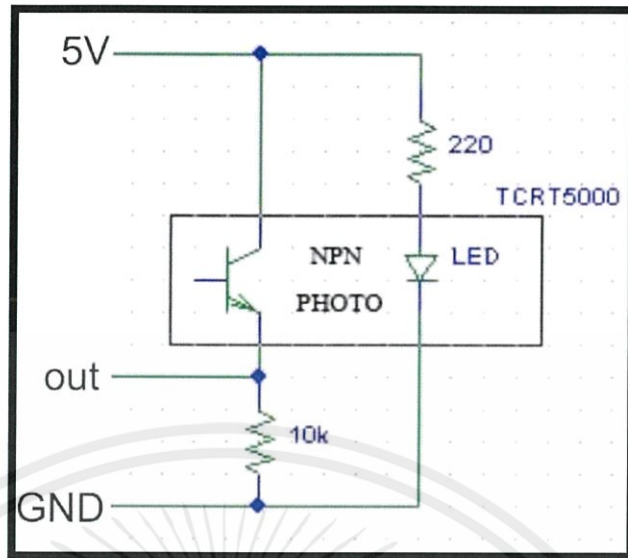


รูปที่ 3.4 วงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง

3.2.2 วงจรควบคุมการทำงานของเซนเซอร์และวงจรเซนเซอร์

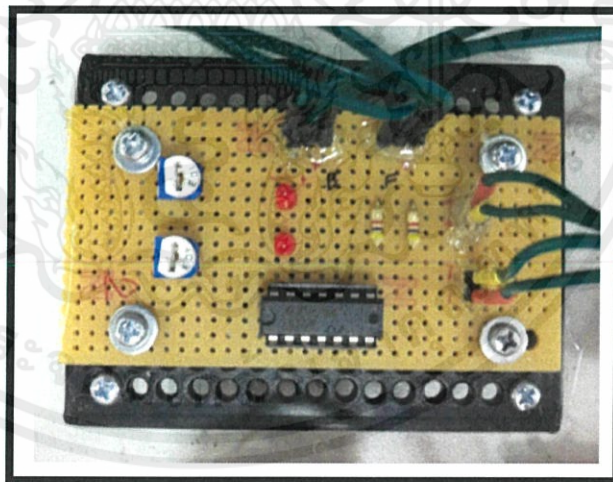
เซนเซอร์ที่นำมาใช้เป็น IC เบอร์ TCRT5000 ซึ่งเป็น Reflective optical sensor with transistor output คือภายในตัว IC มีตัวส่งแสงหรือที่รูปสัญลักษณ์ที่เป็นหลอด LED จะทำหน้าที่ส่งแสงออกไปตลอดเวลาแสงที่ส่งออกไปก็จะไปกระทบกับพื้นแล้วสะท้อนกลับมาที่ตัวรับแสงหรือสัญลักษณ์ในที่นี้คือโฟโตทรานซิสเตอร์ (Photo transistor) ถ้าพื้นนั้นเป็นสีขาวแล้วแสงที่กระทบกับพื้นก็จะสะท้อนกลับมาได้ดีและถ้าแสงนั้นเป็นสีดำแสงที่สะท้อนออกมาก็แทบไม่มีเพราะสีดำมีการดูดแสงสิ่งที่ต้องคำนวณคือค่า R ที่ต้องใช้เพื่อจำกัดกระแสทางด้านฝั่งตัวส่งและฝั่งตัวรับตัวส่งจะต้องดูค่า Forward current, Forward voltage โดยดูจากกราฟและต้องกำหนดแรงดันจากแหล่งจ่ายหรือที่เรียกว่า Vcc แล้วใช้หลักการ KVL ในการคำนวณหาค่า R ตัวรับจะต้องดูค่า Collector current, Collector emitter voltage โดยดูจากกราฟซึ่งจะใช้อยู่แค่ 2 สถานะคือตอนที่จับสีขาว (Saturated) และตอนที่จับสีดำ (Cut off) โดยการคำนวณ R จะคำนวณจาก Saturate นอกจากนี้ยังต้องดูว่าระยะห่างจากพื้นถึงเซนเซอร์สัมพันธ์กับสัญญาณอย่างไรโดยดูจากกราฟอีกเช่นกันจากการคำนวณในที่นี้เราใช้ค่า $R = 220 \Omega$ ของตัวรับและ 10k ของตัวส่ง

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การเชิงพาณิชย์หรือการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 วงจรเซนเซอร์ TCRT5000 ที่ใช้งานจริง

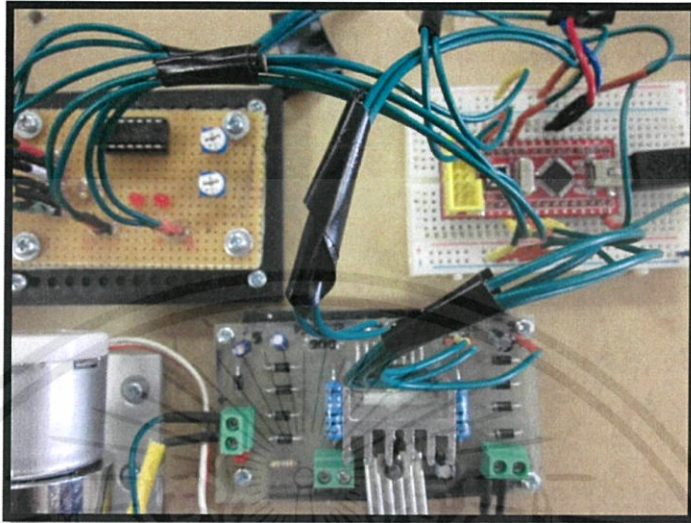
การออกแบบวงจรควบคุมการทำงานของเซนเซอร์ที่สามารถปรับระยะเซนเซอร์ได้ โดยเลือกใช้ไอซีเบอร์ LM339 เป็นอุปกรณ์ช่วยในการควบคุมการทำงานของเซนเซอร์โดยทำงานร่วมกับตัวต้านทานปรับค่าได้ ซึ่งตัวต้านทานปรับค่าได้นี้ จะทำให้เซนเซอร์สามารถปรับระยะได้



รูปที่ 3.6 วงจรควบคุมการทำงานของเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

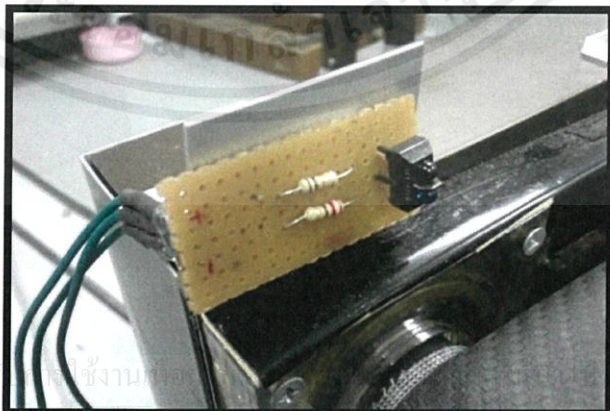
3.2.3 การต่อวงจรใช้งานจริงทั้งระบบ



รูปที่ 3.7 การต่อวงจรรวมของระบบ

3.3 การปรับปรุงฐานยึดเซนเซอร์

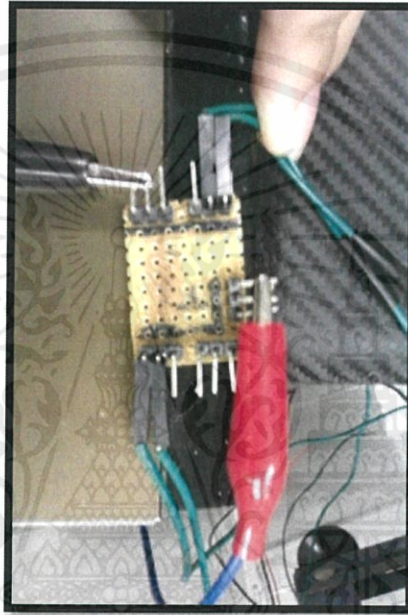
เนื่องจากในส่วนของระบบลำเลียงนั้นต้องอาศัยการตรวจจับของเซนเซอร์ในการนับการเข้าออกของวัตถุ จากโครงสร้างเดิมจะพบว่าการติดตั้งเซนเซอร์ที่ไม่มั่นคงสามารถหลุดได้ง่าย เมื่อมีการขยับเพียงเล็กน้อย เซนเซอร์จะตรวจจับค่าผิดพลาด จึงมีการปรับปรุงโดยใช้เหล็กฉากแทน ให้เซนเซอร์ติดอยู่กับที่ ไม่ให้มีการหลุดหรือหมุนออกนอกเส้นทางได้ เพื่อให้เซนเซอร์สามารถตรวจจับวัตถุได้อย่างเสถียรมากขึ้น ดังรูปที่ 3.8



รูปที่ 3.8 การติดตั้งเซนเซอร์ของ Part B และ Part D, E แบบใหม่

3.4 การพัฒนาการจ่ายไฟ

เนื่องจากการจ่ายไฟแบบเดิมมีการใช้ซัพพลาย 3-4 เครื่องในการจ่ายไฟให้ในแต่ละส่วนของระบบลำเลียงนี้ จึงมีการปรับปรุงให้ใช้ซัพพลายเพียงเครื่องเดียวโดยเพิ่มบอร์ดหนึ่งเข้ามาดังรูปที่ 3.6 เพื่อให้สะดวกในการเปิด - ปิดเครื่อง ในด้านการปฏิบัติงาน การใช้ซัพพลายเพียงเครื่องเดียวยังทำให้มีพื้นที่ในการปฏิบัติงานมากขึ้น ลดการเกิดอุบัติเหตุ เช่นการสะดุดสายไฟ สามารถเปิด-ปิดไฟได้จากที่เดียว ช่วยให้การปฏิบัติงานเป็นไปได้ง่ายขึ้น

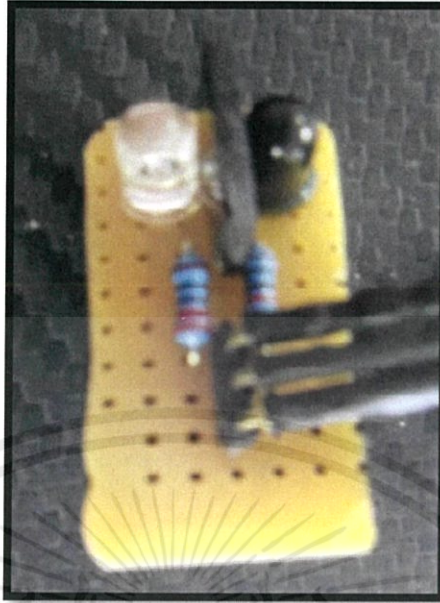


รูปที่ 3.9 โครงสร้างบอร์ดจ่ายไฟ

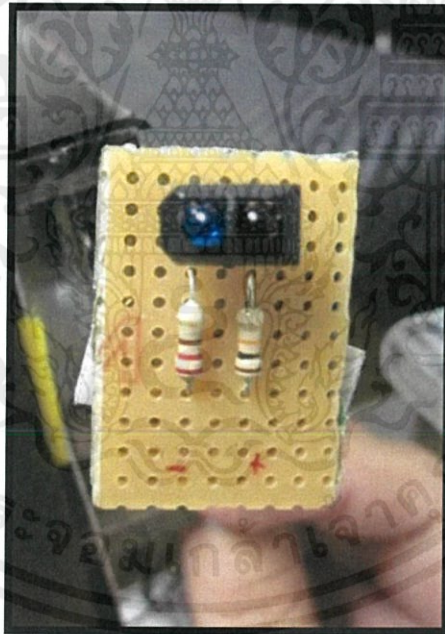
3.5 การปรับปรุงเซนเซอร์

ในส่วนของ Part A จะใช้เซนเซอร์อินฟราเรดในการตรวจจับวัตถุ ซึ่งเซนเซอร์อินฟราเรดนี้จะมีรัศมีการตรวจจับที่กว้างทำให้การตรวจจับนั้น มีการควบคุมที่ยาก จึงเปลี่ยนมาใช้เป็น TCRT 5000 ซึ่งเป็นเซนเซอร์อินฟราเรดเช่นกัน แต่จะมีลักษณะทางกายภาพที่ดีกว่า สามารถควบคุมการตรวจจับวัตถุได้ง่ายกว่า เนื่องจากมีรัศมีการตรวจจับที่แคบ และไกลกว่า จึงเหมาะสมกับการทำงานนี้มากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 วงจรเซนเซอร์ที่ใช้เซนเซอร์อินฟราเรด



รูปที่ 3.11 วงจรเซนเซอร์ที่ใช้ TCRT 5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การออกแบบในส่วนของโปรแกรม

การเขียนโปรแกรมนี้นี้เป็นการเขียนโปรแกรมสำหรับควบคุมการทำงานของระบบคลังสินค้า จะประกอบไปด้วย ส่วนลำเลียง และส่วนคัดแยก โดยตลอดการทำงานนับตั้งแต่ต้นจนจบกระบวนการลำเลียงสินค้า จะแสดงที่หน้าจอคอมพิวเตอร์ ซึ่งเป็นภาพกราฟิกจำลองการทำงานของระบบ จะทำงานสัมพันธ์กับการทำงานของระบบคลังสินค้าจริง โดยการออกแบบระบบจำลองคลังสินค้า ในเบื้องต้นนั้นต้องเข้าใจหลักการทำงานในภาพรวมการทำงานของระบบก่อนว่ามีหลักการทำงานอย่างไร แล้วจึงพิจารณาส่วนย่อยต่างๆอันประกอบไปด้วย หน้าจอแสดงผลและการสั่งการในส่วนลำเลียงและส่วนคัดแยก

ระบบคลังสินค้าแบ่งออกเป็น 5 Part (A, B, C, D, E และ F) โดยแบ่งการทำงานได้เป็น 3 ส่วน คือ

1. ส่วนลำเลียงสินค้าขึ้น (Part A)
2. ส่วนลำเลียงสินค้า (Part B, D และ E)
3. ส่วนคัดแยกสินค้า (Part C)

โดยการเขียนโปรแกรมจะแบ่งออกเป็น 2 ส่วน คือ โปรแกรม arduino และ โปรแกรม Visual C#

3.6.1 โปรแกรม arduino

ในส่วนนี้จะเป็นส่วนของการเขียนโปรแกรมเพื่อควบคุมการทำงานของระบบสายพานลำเลียงในแต่ละส่วน

ในส่วนของการลำเลียงสินค้าขึ้น (Part A) จะลำเลียงสินค้าทั้งหมดขึ้นสู่สายพานลำเลียง (Part A) และส่วนลำเลียงสินค้า (Part B) ก็จะลำเลียงสินค้าไปสู่ส่วนคัดแยก (Part C) โดยในกระบวนการนี้เกิดจากที่ส่วนลำเลียงสินค้าขึ้น (Part A) มีวงจรขับเพื่อทำให้สายพานหมุนและมีวงจรตรวจจับสินค้าที่จะคอยสั่งการให้สายพานหมุนและหยุดหมุนและคอยตรวจแยกสินค้าแล้วส่งค่าที่ได้ไปยังส่วนคัดแยกสินค้า (Part C) อีกที่หนึ่งเพื่อทำการคัดแยกสินค้า ในส่วนนี้ใช้ไมโครคอนโทรลเลอร์ (ET EASY 168 STAMP) ในการสั่งการและส่งค่า

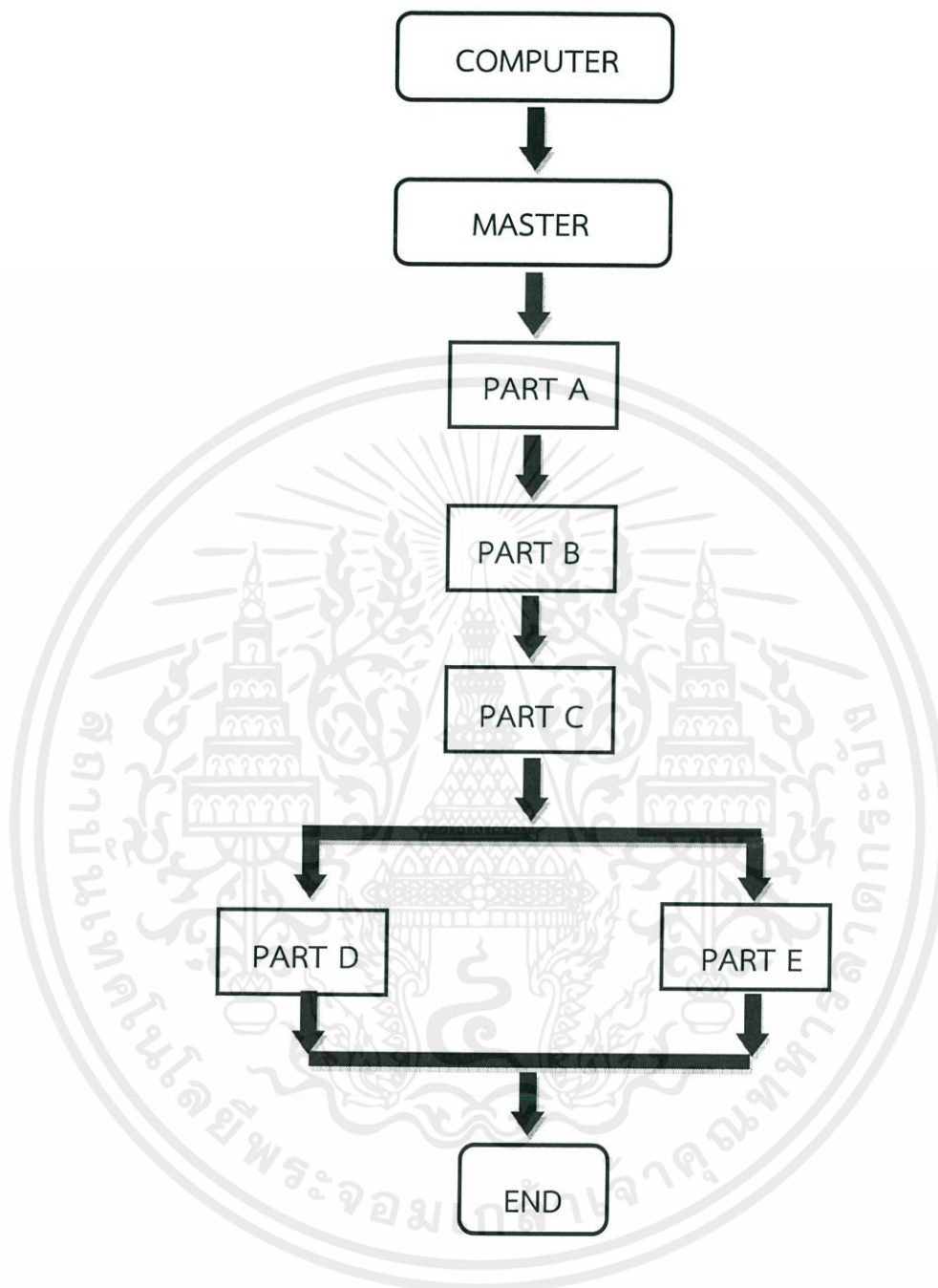
ในส่วนลำเลียงสินค้า (Part B, D และ E) จะเป็นส่วนที่จะลำเลียงสินค้าไปยังส่วนต่างๆคือ

1. ส่วนลำเลียงสินค้า (Part B) จะรับสินค้าจากส่วนลำเลียงสินค้าขึ้น (Part A) แล้วลำเลียงสินค้าไปยังส่วนคัดแยก (Part C)
2. ส่วนลำเลียงสินค้า (Part D) จะลำเลียงสินค้าจากส่วนคัดแยก (Part C) แล้วลำเลียงสินค้าไปยังส่วนจัดเก็บสินค้า
3. ส่วนลำเลียงสินค้า (Part E) จะรับสินค้าจากส่วนคัดแยก (Part C) แล้วลำเลียงสินค้าไปยังส่วนจัดเก็บสินค้า

ในส่วนคัดแยกสินค้า (Part C) จะเป็นส่วนที่รับสินค้าจากส่วนลำเลียง (Part B) และจะทำการคัดแยกสินค้าส่งไปยัง Part D หรือ Part E ตามที่ได้รับการตรวจจับจากส่วนคัดแยกสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

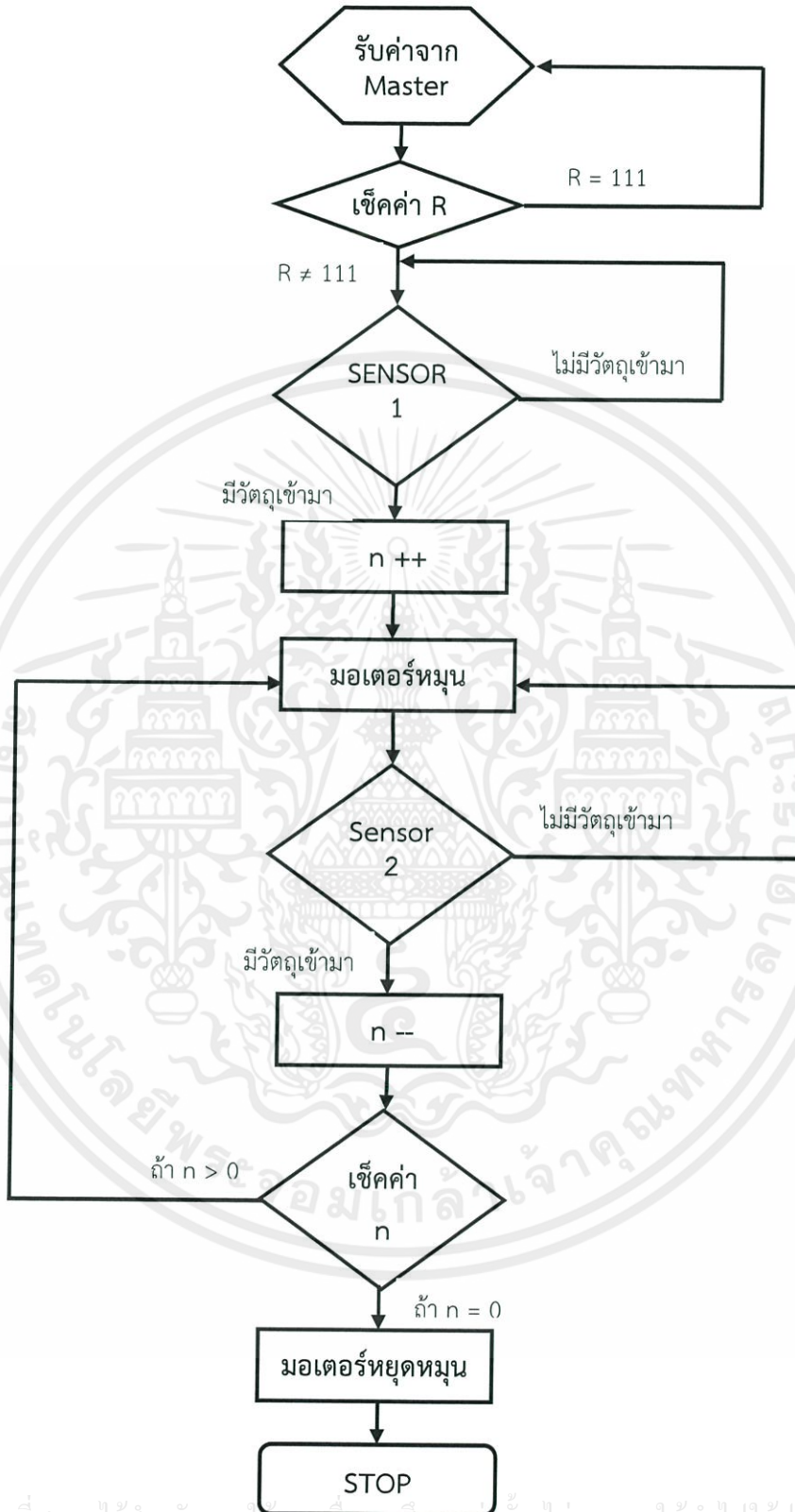
3.6.1.1 หลักการทำงานของระบบ



รูปที่ 3.12 ขั้นตอนการทำงานของทั้งระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

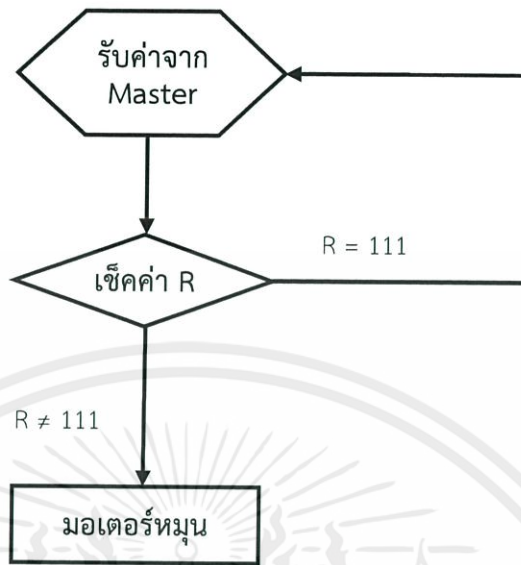
3.6.1.2 หลักการทำงานของโปรแกรมส่วนลำเลียงสินค้าขึ้น(Part A)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

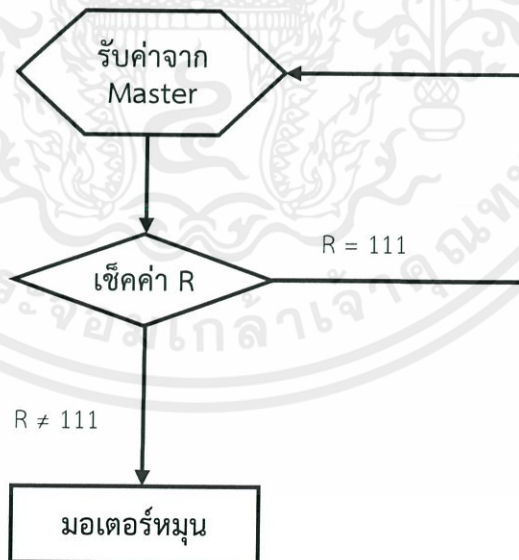
รูปที่ 3.13 ขั้นตอนการทำงานของสายพานลำเลียงขาขึ้น (Part A)

3.6.1.3 หลักการทำงานของโปรแกรมส่วนลำเลียงสินค้า (Part B)



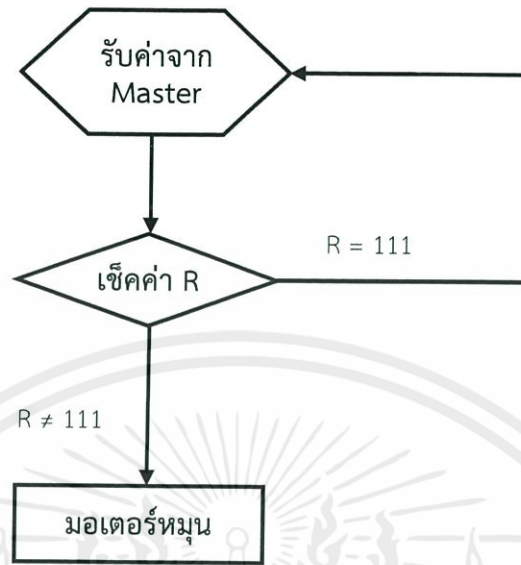
รูปที่ 3.14 ขั้นตอนการทำงานของสายพานลำเลียง (Part B)

3.6.1.4 หลักการทำงานของโปรแกรมส่วนลำเลียงสินค้า (Part D)



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.1.5 หลักการทำงานของโปรแกรมส่วนลำเลียงสินค้า (Part E)

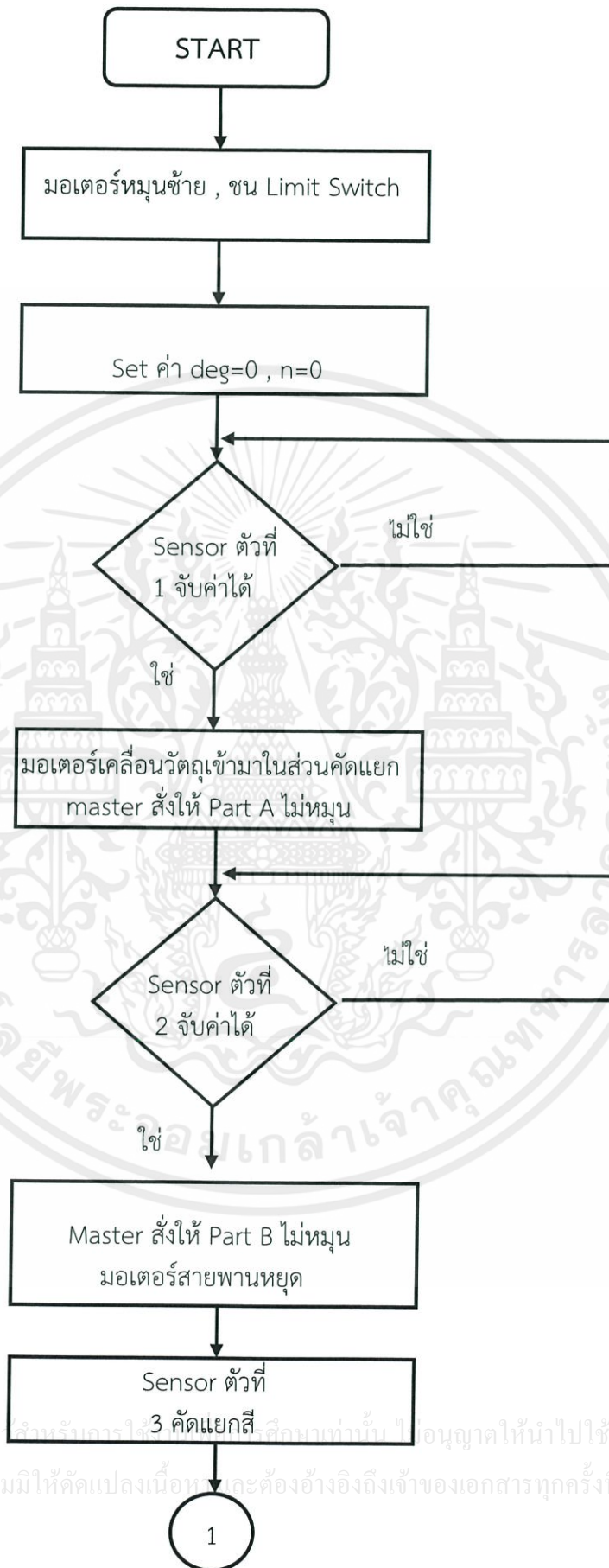


รูปที่ 3.16 ขั้นตอนการทำงานของสายพานลำเลียง (Part E)

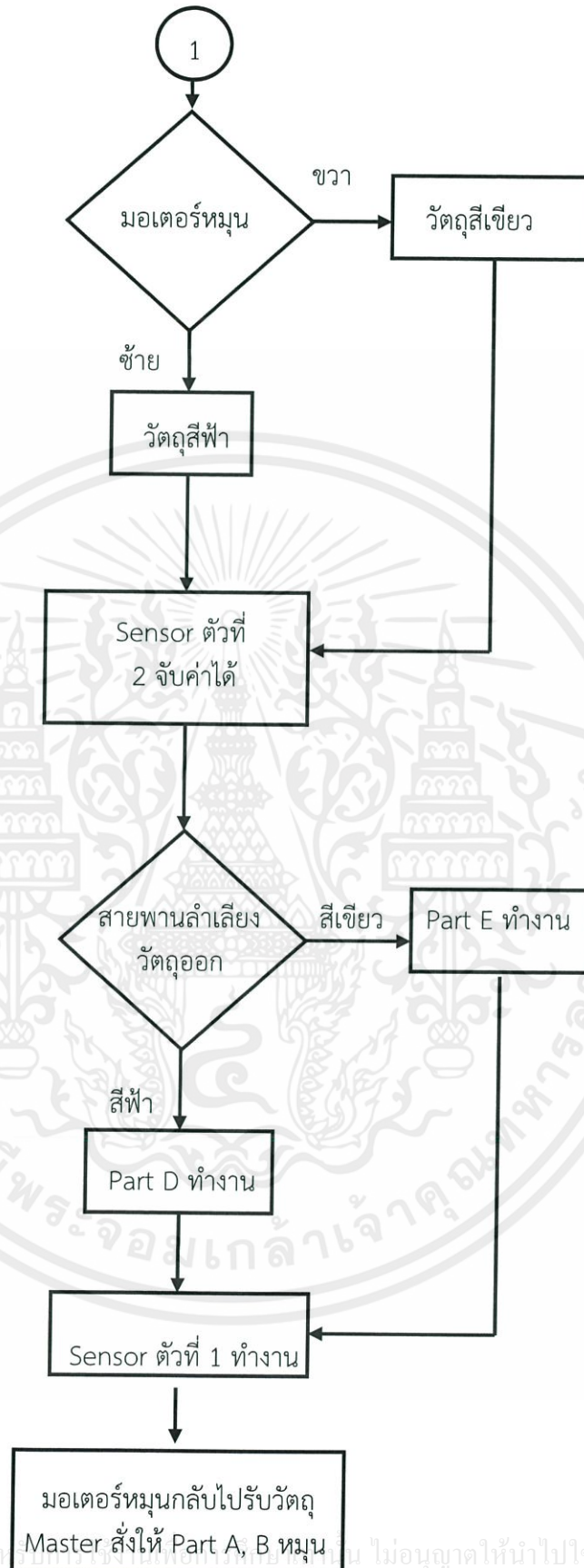
3.6.1.6 หลักการทำงานของโปรแกรมส่วนคัดแยก (Part C)

แนวคิดการออกแบบการทำงานของเครื่องคัดแยกสินค้าอัตโนมัติเพื่อให้เครื่องคัดแยกสินค้าอัตโนมัติสามารถเคลื่อนที่หมุนไปมาได้ตามทิศทางที่ต้องการได้ โดยนำคอมพิวเตอรส์ส่วนบุคคลเข้ามาใช้ในการควบคุมการเคลื่อนที่ของเครื่องคัดแยกสินค้า เริ่มจากการส่งข้อมูลจากคอมพิวเตอร์ แล้วทำการประมวลผล แล้วส่งข้อมูลจากการประมวลผลไปเข้าไมโครคอนโทรลเลอร์ในที่นี่ใช้ AVR168 Easy-Stamp ทำหน้าที่ประมวลผล ก่อนส่งข้อมูลไปยังบอร์ดควบคุมเซนเซอร์ซึ่งมีเซนเซอร์ทั้งหมด 3 ตัวและบอร์ดควบคุมมอเตอร์ซึ่งมีมอเตอร์ทั้งหมด 2 ตัว คือ มอเตอร์ที่นำวัตถุเข้าออกสายพานลำเลียงและมอเตอร์ที่ใช้ในการหมุนไปตามทิศทางที่กำหนด ซึ่งระบบการทำงานของเครื่องคัดแยกนี้ จะมีอยู่ด้วยกัน 6 ขั้นตอน คือ ขั้นที่ 1 วัตถุเข้ามาในส่วนคัดแยก เซนเซอร์ตัวที่ 1 สามารถตรวจจับได้ มอเตอร์เคลื่อนวัตถุเข้ามาในส่วนคัดแยก ขั้นที่ 2 วัตถุเข้ามาถึงเซนเซอร์ตัวที่ 2 จับได้ ขั้นที่ 3 เซนเซอร์ประมวลผลจับค่าได้แล้วทำการเคลื่อนที่ไปยังทิศทางที่กำหนด ขั้นที่ 4 เซนเซอร์ตัวที่ 2 จับค่าได้ มอเตอร์เคลื่อนวัตถุออกไปยังสายพานลำเลียง Part D หรือ Part E ขั้นที่ 5 เมื่อวัตถุเคลื่อนออกผ่านเซนเซอร์ตัวที่ 1 ให้มอเตอร์ที่ลำเลียงวัตถุหยุดการทำงาน ขั้นที่ 6 ประมวลผลการทำงานว่า จะต้องเคลื่อนที่ไปยังทิศทางไหนเพื่อให้สายพานของส่วนคัดแยกกลับไปอยู่ในตำแหน่งที่สามารถรับวัตถุจากสายพานลำเลียง Part B ได้ โดยสามารถแสดงหลักการทำงานได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในวงจำกัดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.17 ขั้นตอนการทำงานของส่วนคัดแยก (Part C)

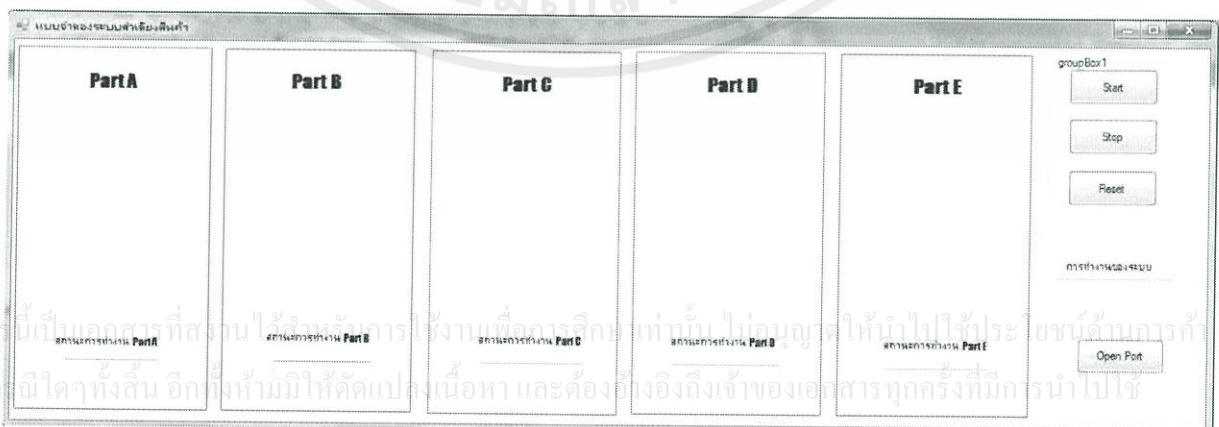
3.6.2 โปรแกรม Visual C#

โปรแกรม Visual C# เป็นโปรแกรมที่สร้างขึ้นมาเพื่อให้การเชื่อมต่อระหว่างผู้ใช้งานกับระบบแบบจำลองคลังสินค้าเป็นสิ่งที่ง่ายขึ้น โดยสามารถเข้าถึงระบบต่างๆ ผ่านทางหน้าจอคอมพิวเตอร์ได้ เช่น ปุ่มกด START เพื่อนำไปสั่งงานระบบให้เริ่มต้นการทำงาน ,มีการให้กรอกค่าเพื่อให้ระบบทำงานตามที่ใช้กรอก และยังสามารถแสดงผลการควบคุมเป็นภาพกราฟฟิกได้อีกด้วย

โดยขอบเขตของการแสดงโปรแกรม Visual C# จะมีหน้าต่างให้ผู้ใช้งานสามารถเข้าถึงการทำงานของโปรแกรม และสามารถแสดงผลการควบคุมเป็นภาพกราฟฟิกผ่านหน้าจอคอมพิวเตอร์ เขียนโปรแกรมจากโปรแกรมวิซวลซีชาร์ป (Visual C#) โดยออกแบบหน้าต่างโปรแกรมซึ่งมีส่วนที่แสดงภาพกราฟฟิกการทำงานของแบบจำลองระบบคลังสินค้า และปุ่มควบคุมการทำงานของระบบ ซึ่งมีการกำหนดการทำงานของแต่ละปุ่มดังนี้

- ปุ่ม START ระบบเริ่มทำงาน
- ปุ่ม STOP ระบบหยุดทำงาน
- ปุ่ม RESET ระบบเริ่มทำงานใหม่ทั้งหมดอีกครั้ง
- ปุ่ม Open Port เป็นตัวเปิดพอร์ตการทำงาน หากยังไม่เปิดพอร์ตโปรแกรมจะมีการเตือนผู้ใช้งานและมีการเขียนโปรแกรมกำหนดให้ภาพกราฟฟิกมีการทำงานที่สัมพันธ์กันกับปุ่มควบคุม โดยมีรูปแบบการทำงานดังต่อไปนี้
 - ส่วนลำเลียงขาขึ้น (Part A) ภาพกราฟฟิกแสดงภาพใน pictureBoxA และแสดงข้อความสถานะการทำงาน
 - ส่วนลำเลียง (Part B) ภาพกราฟฟิกแสดงภาพใน pictureBoxB และแสดงข้อความสถานะการทำงาน
 - ส่วนคัดแยก (Part C) ภาพกราฟฟิกแสดงภาพใน pictureBoxC และแสดงข้อความสถานะการทำงาน
 - ส่วนลำเลียง (Part D) ภาพกราฟฟิกแสดงภาพใน pictureBoxD และแสดงข้อความสถานะการทำงาน
 - ส่วนลำเลียง (Part E) ภาพกราฟฟิกแสดงภาพใน pictureBoxE และแสดงข้อความสถานะการทำงาน

3.6.2.1 ภาพรวมของโปรแกรม Visual C#



รูปที่ 3.18 หน้าจอแสดงผลการทำงานโปรแกรม Visual C#

3.6.2.2 การกำหนดคุณสมบัติของฟอร์มและคอนโทรลต่างๆ ในการออกแบบ Visual C#

ตารางที่ 3.1 การกำหนดคุณสมบัติของฟอร์มและคอนโทรลต่างๆ ในการออกแบบ Visual C#

ออบเจ็กต์/คอนโทรล	คุณสมบัติ	ค่าที่กำหนด
ฟอร์ม	Name	Form1
	Text	แบบจำลองระบบลำเลียงสินค้า
Label	Name	Label1
	Text	Part A
Label	Name	Label2
	Text	สถานะการทำงาน Part A
Label	Name	Label3
	Text	Part B
Label	Name	Label4
	Text	สถานะการทำงาน Part B
Label	Name	Label5
	Text	Part C
Label	Name	Label6
	Text	สถานะการทำงาน Part C
Label	Name	Label7
	Text	Part D
Label	Name	Label8
	Text	สถานะการทำงาน Part D
Label	Name	Label9
	Text	Part E
Label	Name	Label10
	Text	สถานะการทำงาน Part E
Label	Name	Label11
	Text	การทำงานของระบบ
Button	Name	btnStart
	Text	Start
Button	Name	btnStop
	Text	Stop
Button	Name	btnReset
	Text	Reset
Button	Name	btnPort
	Text	Open Port
TextBox	Name	txtA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการ
 ไม่่ว่ากรณีใดๆก็ตาม อีกรทั้งห้ามมิให้คัดลอก
 เพื่อการศึกษาเท่านั้น ไม่อ
 และต้องอ้างอิงถึงเจ้าของ
 ครั้งที่มีการนำไปใช้

ตารางที่ 3.1 การกำหนดคุณสมบัติของฟอร์มและคอนโทรลต่างๆ ในการออกแบบ Visual C# (ต่อ)

TextBox	Name	txtB
TextBox	Name	txtC
TextBox	Name	txtD
TextBox	Name	txtE
TextBox	Name	Txtwork



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองและผลการทดลองของการทำงานของแบบจำลองระบบ คลังสินค้า ซึ่งประกอบไปด้วยหน้าจอแสดงผลที่ติดต่อระหว่างผู้ใช้งานกับระบบ และ ส่วนของ โปรแกรมซึ่งควบคุมการทำงานของระบบทั้งหมด

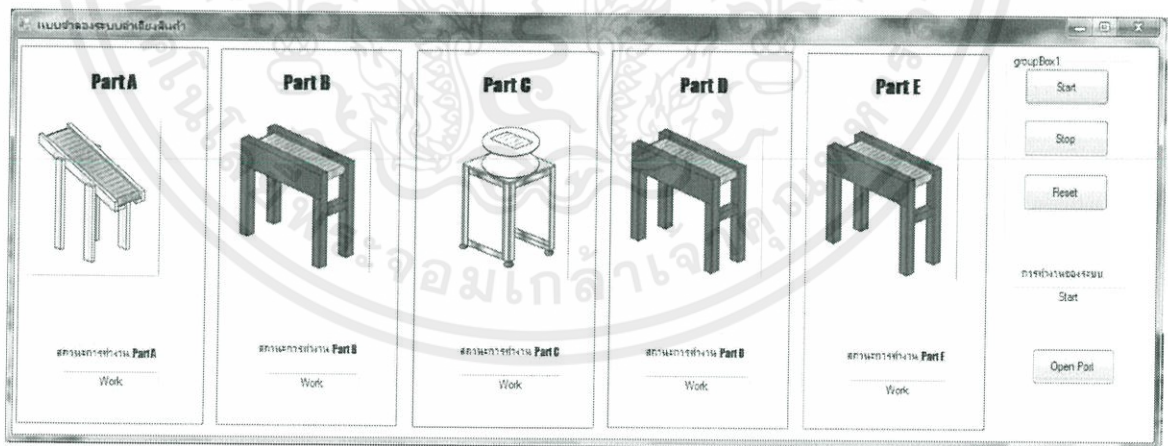
4.1 ลำดับขั้นตอนการทำงาน

- จ่ายแรงดันไฟฟ้าให้แก่ระบบคลังสินค้า 12 โวลต์
- สั่งให้ระบบเริ่มทำงานจากหน้าจอแสดงผลที่ติดต่อระหว่างผู้ใช้งานกับระบบ (Visual C#)
- ตรวจสอบการทำงานของระบบคลังสินค้าโดยให้วัตถุเคลื่อนที่ผ่านเข้าระบบอย่างต่อเนื่อง
- บันทึกข้อมูลการทำงานของระบบคลังสินค้าและตรวจสอบการทำงานของระบบว่าเป็นไปตามคำสั่งที่ออกแบบไว้หรือไม่

4.2 ผลการทดสอบการทำงานของระบบคลังสินค้า

4.2.1 เมื่อกดปุ่ม Start ที่หน้าจอติดต่อระหว่างผู้ใช้งานกับโปรแกรม

เมื่อผู้ใช้งานกดปุ่ม Start ที่หน้าจอแสดงผล จะทำให้ระบบเริ่มทำงานโดยหน้าจอแสดงผลจะแสดงดังรูปที่ 4.1



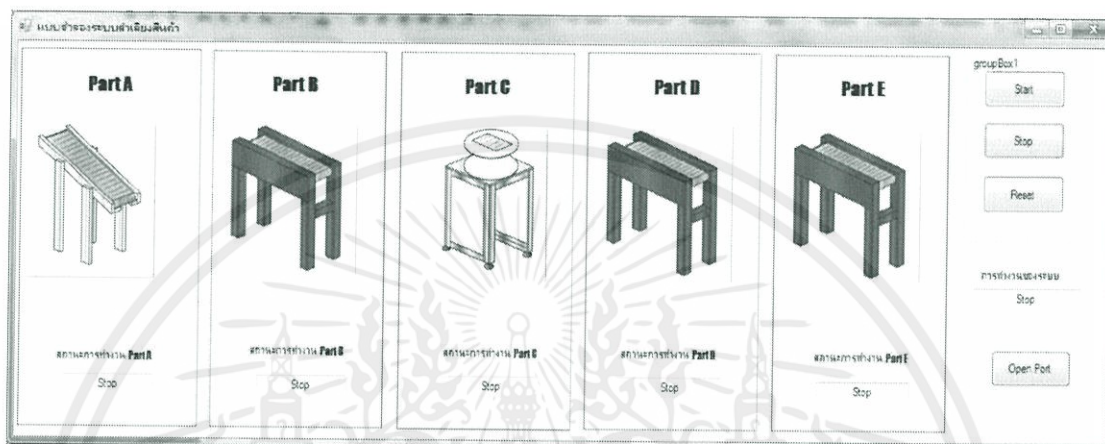
รูปที่ 4.1 หน้าจอแสดงผลการทำงานเมื่อกดปุ่ม Start

รับส่งข้อมูลระหว่าง Part C กับ Part A และระหว่าง Part C กับ Part B ทดลองโดยการวางกล่องบนสายพานลำเลียงสินค้า เมื่อมีกล่องเข้ามาใน Part C ไมคอนคอนโทรลเลอร์ของ Part C ที่เป็นตัว slave จะส่งข้อมูลไปยังไมโครคอนโทรลเลอร์หลักที่เป็นตัว master เพื่อให้ตัว master ส่งข้อมูลต่อไปยังตัว slave ของ Part A และ Part B ให้หยุดทำงาน และเมื่อ Part C ทำงานเสร็จก็จะส่ง

ข้อมูลกลับไปยังตัว master เพื่อให้ตัว master ส่งข้อมูลต่อไปยังตัว slave ของ Part A และ Part B ให้กลับมาทำงาน

4.2.2 เมื่อกดปุ่ม Stop ที่หน้าจอติดต่อระหว่างผู้ใช้งานกับโปรแกรม

เมื่อผู้ใช้งานกดปุ่ม Stop ที่หน้าจอแสดงผล จะทำให้ระบบหยุดทำงานโดยหน้าจอแสดงผลจะแสดงดังรูปที่ 4.2



รูปที่ 4.2 หน้าจอแสดงผลการทำงานเมื่อกดปุ่ม Stop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทวิจารณ์

โปรแกรมควบคุมและแสดงผลการทำงานของระบบคลังสินค้าที่ออกแบบได้นี้ ยังมีข้อจำกัดในการควบคุมผ่านทางหน้าจอคอมพิวเตอร์ เนื่องจากยังสามารถควบคุมระบบได้แค่เบื้องต้น ซึ่งในแต่ละส่วนของระบบมีไมโครคอนโทรลเลอร์ที่ถูกโปรแกรมไว้ทำงานเฉพาะส่วนแล้ว ทำให้ไม่สามารถควบคุม และแก้ไขขั้นตอนการทำงานในส่วนนั้นๆ ผ่านทางโปรแกรมที่หน้าจอคอมพิวเตอร์ได้เลย ระบบคลังสินค้านี้เป็นระบบอัตโนมัติที่มีการทำงานที่สัมพันธ์และต่อเนื่องกัน เมื่อมีข้อผิดพลาดในงานบางส่วนทำให้การทำงานทั้งหมดของระบบหยุดชะงัก และในส่วนของไมโครคอนโทรลเลอร์กลางที่ทำหน้าที่เป็นตัวบังคับการ (master) รับส่งข้อมูลกับไมโครคอนโทรลเลอร์ตัวอื่นๆ ในแต่ละส่วนของระบบที่เป็นตัวตอบสนองคำสั่ง (slave) ด้วยระบบ I²C Bus นี้มีความยุ่งยากในการทำงาน เนื่องจากการรับส่งข้อมูลต้องตั้งเป็นรหัสตามสถานะเซนเซอร์ที่ทำงาน เพื่อให้การรับส่งข้อมูลและการดึงข้อมูลมาใช้ไม่ทับซ้อนกันอีก ทั้งระบบ I²C Bus นี้เป็นระบบสายสัญญาณที่ต้องต่อสายระหว่างไมโครคอนโทรลเลอร์กลางที่เป็นตัวบังคับการ (master) กับไมโครคอนโทรลเลอร์อื่นๆที่เป็นตัวตอบสนองคำสั่ง (slave) ซึ่งระบบสายในปัจจุบันนี้ได้ล้าสมัยไปแล้ว อย่างไรก็ตามเมื่อนำทุกส่วนมาทดลองทำงานร่วมกันแล้วเห็นได้ว่าการทำงานร่วมกันได้เป็นอย่างดี

5.2 สรุปผล

โปรแกรมควบคุมและแสดงผลการทำงานของระบบคลังสินค้าที่สร้างขึ้นมานี้เป็นโปรแกรมที่ประกอบด้วยส่วนของการควบคุมการทำงานระบบคลังสินค้า และส่วนของภาพกราฟิกในการแสดงผลการทำงานของกระบวนการโดยใช้โปรแกรมสำเร็จรูปในการออกแบบ ได้แก่โปรแกรมวิซวลซีชาร์ป (Visual C#) และโปรแกรมอาเดียร์โน (Arduino) เป็นโปรแกรมหลักในการออกแบบโปรแกรมควบคุมและแสดงผลการทำงานของระบบคลังสินค้าที่ใช้ติดต่อกับผู้ใช้งาน โดยมีการสื่อสารกับไมโครคอนโทรลเลอร์ AVR รุ่น ET-EASY 168 STAMP ที่เป็นอุปกรณ์ภายนอกผ่านทางพอร์ตอนุกรม (serial port)

ระบบคลังสินค้าที่ออกแบบขึ้นมาี้ มีการทำงานในแต่ละส่วนที่สัมพันธ์และต่อเนื่องกันซึ่งมีการติดต่อสื่อสารกันด้วยระบบ I²C Bus ระหว่างตัวไมโครคอนโทรลเลอร์นี้เป็นตัวบังคับการ (master) ที่รับส่งข้อมูลกับไมโครคอนโทรลเลอร์ ในส่วนการทำงานอื่นๆที่เป็นตัวตอบสนองคำสั่ง (slave) อย่างเป็นระบบโดยอาศัยการตั้งรหัสสื่อสารตามสถานะของเซนเซอร์ เพื่อป้องกันการรับส่งข้อมูลและการดึงข้อมูลมาใช้เกิดการทับซ้อนกันในการเขียนโปรแกรมไมโครคอนโทรลเลอร์ตัวบังคับการ (master) นี้จะคอยตรวจสอบรหัสข้อมูลสถานะต่างๆ ของเซนเซอร์ในแต่ละส่วนการทำงานที่ไมโครคอนโทรลเลอร์อื่นๆ ที่เป็นตัวตอบสนองคำสั่ง (slave) ที่ส่งมาให้และเมื่อได้รับรหัสข้อมูลที่ตรงตามเงื่อนไขที่กำหนดไว้ ไมโครคอนโทรลเลอร์ซึ่งเป็นตัวบังคับการ (master) นี้จะส่งข้อมูลต่อไปยัง

ไมโครคอนโทรลเลอร์ที่เป็นตัวตอบสนองคำสั่ง (slave) ที่เป็นเป้าหมายในการทำงานขั้นต่อไปโดยไมโครคอนโทรลเลอร์ที่เป็นตัวตอบสนองคำสั่ง (slave) แต่ละตัวซึ่งถูกตั้งชื่อไว้เพื่อระบุตำแหน่งการควบคุมในส่วนของการทำงานของระบบ นอกจากนี้ยังมีการสื่อสารกับโปรแกรมควบคุมและแสดงผลการทำงานของระบบคลังสินค้าผ่านทางพอร์ตอนุกรม (serial port) ไปพร้อมๆ กันด้วยเพื่อให้โปรแกรมนี้สามารถนำค่าที่ได้สามารถนำไปแสดงผลด้วยภาพกราฟิกได้

5.3 ปัญหาที่พบและแนวทางแก้ไข

ตารางที่ 5.1 ปัญหาที่พบและแนวทางแก้ไขพัฒนา

ปัญหา	แนวทางแก้ไขพัฒนา
<ul style="list-style-type: none"> ขาดอุปกรณ์และเครื่องมือ เครื่องใช้ในการทำงาน เช่น เลื่อย คีม บัดกรี อีกทั้งยังขาดความรู้ความชำนาญในการใช้เครื่องมือต่างๆ ทำให้การดำเนินการซ่อมแซมเป็นไปได้ช้ากว่าที่ควร 	<ul style="list-style-type: none"> หาซื้ออุปกรณ์ที่ขาดหายไป และยืมบางเครื่องมือจากเพื่อนที่มี ศึกษาวิธีการใช้ก่อนเพื่อความปลอดภัย และป้องกันการเสียหาย
<ul style="list-style-type: none"> การศึกษาการเขียนโปรแกรม Arduino ซึ่งขาดความชำนาญในการเขียนโปรแกรมทำให้ยากต่อการเข้าใจคำสั่งและใช้ระยะเวลาในการวิเคราะห์โปรแกรมเก่านั้นนาน 	<ul style="list-style-type: none"> ศึกษาเกี่ยวกับการเขียนโปรแกรมเพิ่มเติม เพื่อเพิ่มความรู้ความชำนาญในการทำงาน แบ่งกันศึกษาคนละส่วน เพื่อให้ได้งานเร็วขึ้น
<ul style="list-style-type: none"> ส่วนที่ยึดกับเซนเซอร์ไม่มั่นคง ทำให้การตรวจจับวัตถุมีความคลาดเคลื่อน 	<ul style="list-style-type: none"> ทำการยึดเซนเซอร์ใหม่ในทุกส่วนให้ยึดกับที่ และมั่นคง
<ul style="list-style-type: none"> มีสายไฟเกะกะไม่เป็นระเบียบเชื่อมต่อกันในแต่ละพาร์ท ทำให้สับสนเมื่อทำการไล่สาย และปฏิบัติงาน 	<ul style="list-style-type: none"> เก็บสายไฟให้เป็นระเบียบ และทำสัญลักษณ์ เช่น สายไฟ 5 V ,สายกราวด์
<ul style="list-style-type: none"> เนื่องจากการเป็นารต่อยอดจากโปรเจคของรุ่นที่ ห้างจระ เซนเซอร์ หรือสายพาน จึงอาจมีการเสื่อมสภาพไปหรือขาดหายไป 	<ul style="list-style-type: none"> ตรวจทดสอบทุกพาร์ทว่ายังสามารถใช้งานได้หรือไม่ ส่วนใดที่เสียหรือขาดหายไปก็ทำการซ่อมแซมเปลี่ยนใหม่ตามสมควร
<ul style="list-style-type: none"> โค้ดที่มีไม่สามารถทำงานได้ตามที่ต้องการ 	<ul style="list-style-type: none"> ปรับเปลี่ยนบางส่วนเพื่อให้สามารถทำงานได้ตามที่ต้องการ ที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาค้นคว้าเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาต

5.4 ข้อเสนอแนะและแนวทางในการศึกษาค้นคว้าพัฒนา

ควรตรวจสอบสภาพของแต่ละพาร์ทก่อนว่าครบถ้วนสมบูรณ์หรือไม่ก่อนการทดสอบ และควรศึกษาทำความเข้าใจในการเขียนโปรแกรม Arduino ก่อน เพื่อให้สามารถอ่าน ทำความเข้าใจคำสั่งได้อย่างรวดเร็ว และเพื่อการพัฒนาในระบบในครั้งต่อไปอาจมีการพัฒนาอุปกรณ์อิเล็กทรอนิกส์ให้สามารถแบ่งแยกลักษณะของวัตถุให้ได้ละเอียดยิ่งขึ้นเช่น บอกลี บอกขนาด เป็นต้น เพื่อสะดวกในการเก็บข้อมูลและสามารถนำข้อมูลไปใช้งานได้จริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] วรพงศ์ ตั้งศรีรัตน์. เซนเซอร์และทรานสดิวเซอร์. พิมพ์ครั้งที่ 6. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยีไทย (ไทย-ญี่ปุ่น), 2548.
- [2] เอกชัยมะการ. เรียนรู้เข้าใจใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วยArduino. พิมพ์ครั้งที่ 1. กรุงเทพฯ : พิมพ์โดยบริษัทอ็ททีทีจำกัด, 2552.
- [3] Benjamin. 2012. "Alldatasheet." [Online].
Available : <http://www.alldatasheet.com/view.jsp.Searchword=LM339>
- [4] Pcmán. 2008. "เริ่มต้นกับarduino." [Online].
Available : <http://www.logicthai.net/node/13>
- [5] Futurlec. 2012. "Technical Information - ST Microelectronics L298 Datasheet." [Online]. Available : <http://www.futurlec.com/Others/L298.shtml>
- [6] Exteen blog. 2009. "Sensor." [Online].
Available : <http://sensor-of-fluid.exteen.com/20090815/sensor>
- [7] Vishay Siliconix. "TCRT5000 Datasheet." [Online].
Available : <http://www.soiseek.com/VISHAY/TCRT5000>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมการควบคุม

ก.1 โปรแกรมส่วน Main Controller (Master)

```
#include <Wire.h>

int A=0;
int B=0;
int i,n;
int

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  Serial.println("Master Connected");
}

void loop()
{
  if(Serial.available())
  {
    B=Serial.read();

    if(B==50) //ถ้ากด Stop ส่ง 111 ไปหยุด ABCDE
    {
      for(int j=1;j<6;j++) //
      {
        Wire.beginTransmission(j);
        Wire.write(111);
        Wire.endTransmission();
      }
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    Wire.endTransmission();
}
else if(Y==204)
{
    for(int k=1;k<3;k++) //ส่งค่าให้ A B หยุดการทำงาน
    {
        Wire.beginTransmission(k);
        Wire.write(111);
        Wire.endTransmission();
    }
    Wire.beginTransmission(5); // ส่งให้ E ทำงาน
    Wire.write(999);
    Wire.endTransmission();
}
else if(Y==206)
{
    for(int k=1;k<3;k++) //ส่งค่าให้ A B เริ่มการทำงาน
    {
        Wire.beginTransmission(k);
        Wire.write(999);
        Wire.endTransmission();
    }
}
} //End of else # End of ถ้ากดปุ่ม Start

else if(B==51) //ถ้ากดปุ่ม Reset
{
    Wire.beginTransmission(3); //ส่งให้ Part C เคลียร์ค่าและหยุดการทำงาน
    Wire.write(212);
    Wire.endTransmission();
} //End of else # End of ถ้ากดปุ่ม Reset
} //End of void loop()

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆก็ตาม กรุณาแจ้งให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 โปรแกรมส่วนสายพานลำเลียงขาขึ้น (Part A)

```
#include <Wire.h>
#define in1 3
#define in2 4
#define in3 5
#define enb 6
#define in4 7
```

```
int R=0;
int n;
int A;
```

```
void setup()
{
  pinMode(in1,INPUT); // sensor 1
  pinMode(in2,INPUT); // sensor 2
  pinMode(in3,OUTPUT); // ขาไฟเข้า
  pinMode(in4,OUTPUT); // ขาไฟเข้า
  pinMode(enb,OUTPUT);
  Wire.begin(1);
  Wire.onRequest(requestEvent);
  Wire.onReceive(receiveEvent);
  Serial.begin(9600);
  Serial.println("AVR Mecah is conect");
}
```

```
void loop()
{
  int val = map(analogRead(enb),0,1023,0,255);
  analogWrite(enb,val);
  motor_stop();
  while(R!=111) //ถ้าไม่หยุดการทำงาน
  {
    if(n>0) //ถ้ามีวัตถุ Motor หมุน
    {
      motor_left();
    }
    if(digitalRead(in1)==0) //ถ้า Sensor หน้าจับวัตถุได้
  }
```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

motor_stop();
delay(1000);
  while(digitalRead(in1)==0)
  {
    motor_left();
  }
n++;
Serial.print("n = ");
Serial.println(n);
}

if(digitalRead(in2)==0) //ถ้า Sensor หลังจับวัตถุได้
{
  motor_left();
  delay(1000);
  while(digitalRead(in2)==0);
  {
    motor_stop();
  }
n--;
Serial.print("n = ");
Serial.println(n);
}
}
} //End void loop()
void motor_left()
{
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  digitalWrite(enb,HIGH);
}
void motor_stop()
{
  digitalWrite(in3,LOW);
  digitalWrite(in4,LOW);
  digitalWrite(enb,HIGH);
}
void receiveEvent(int A)
{

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

A = Wire.read();
R=A;
}
void requestEvent() //Master ร้องขอ
{
// Wire.write(code);
}

```

ก.3 โปรแกรมส่วนสายพานลำเลียง (Part B)

```

#include <Wire.h>
#define in1 3
#define in2 4
#define in3 5
#define enb 6
#define in4 7
int R=0;
int S1=0;
int S2=0;
int T=1;
int i=0;
void setup()
{
pinMode(in1,INPUT); // sensor ตัวที่ 1
pinMode(in2,INPUT); // sensor ตัวที่ 2
pinMode(in3,OUTPUT); // ขาไฟเข้า
pinMode(in4,OUTPUT); // ขาไฟเข้า
pinMode(enb,OUTPUT);
Serial.begin(9600);
Wire.begin(2);
Wire.onRequest(requestEvent);
Wire.onReceive(receiveEvent);
}
void loop()
{
int val = map(analogRead(enb),0,1023,0,255);
analogWrite(enb,val);
S1 = digitalRead(in1);
S2 = digitalRead(in2);

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีข้อผิดพลาดประการใด ขออภัยและต้องอภัยถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(R!=111) //ถ้าไม่หยุดการทำงาน
    {
        motor_left();
        delay(2000);
    }
}

void motor_left()
{
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    digitalWrite(enb,HIGH);
}

void motor_stop()
{
    digitalWrite(in3,LOW);
    digitalWrite(in4,LOW);
    digitalWrite(enb,HIGH);
}

void receiveEvent(int A)
{
    A = Wire.read();
    R=A;
}

```

ก.4 โปรแกรมส่วนสายพานลำเลียง (Part D)

```

#include <Wire.h>
#define in1 2
#define in2 3
#define in3 4
#define enb 6
#define in4 5
int R=0;
int S1=0;
int S2=0;
int T=1;
int i=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void setup()
{
  pinMode(in1,INPUT); // sensor ตัวที่ 1
  pinMode(in2,INPUT); // sensor ตัวที่ 2
  pinMode(in3,OUTPUT); // ขาไฟเข้า
  pinMode(in4,OUTPUT); // ขาไฟเข้า
  pinMode(enb,OUTPUT);
  Serial.begin(9600);
  Wire.begin(4);
  // Wire.onRequest(requestEvent);
  Wire.onReceive(receiveEvent);
}
void loop()
{
  int val = map(analogRead(enb),0,1023,0,255);
  analogWrite(enb,val);
  S1 = digitalRead(in1);
  S2 = digitalRead(in2);
  if(R!=111) //ถ้าไม่หยุดการทำงาน
  {
    motor_left();
    delay(2000);
  }
}

void motor_left()
{
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  digitalWrite(enb,HIGH);
}

void motor_stop()
{
  digitalWrite(in3,LOW);
  digitalWrite(in4,LOW);
  digitalWrite(enb,HIGH);
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void receiveEvent(int A)
{
  A = Wire.read();
  R=A;
}

```

ก.5 โปรแกรมส่วนสายพานลำเลียง (Part E)

```

#include <Wire.h>
#define in1 2
#define in2 3
#define in3 5
#define enb 6
#define in4 4
int R=0;
int S1=0;
int S2=0;
int T=1;
int i=0;
void setup()
{
  pinMode(in1,INPUT); // sensor ตัวที่ 1
  pinMode(in2,INPUT); // sensor ตัวที่ 2
  pinMode(in3,OUTPUT); // ขาไฟเข้า
  pinMode(in4,OUTPUT); // ขาไฟเข้า
  pinMode(enb,OUTPUT);
  Serial.begin(9600);
  Wire.begin(5);
  // Wire.onRequest(requestEvent);
  Wire.onReceive(receiveEvent);
}
void loop()
{
  int val = map(analogRead(enb),0,1023,0,255);
  analogWrite(enb,val);
  S1 = digitalRead(in1);
  S2 = digitalRead(in2);
  if(R!=111) // ถ้าไม่หยุดการทำงาน
  {

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ขอสงวนสิทธิ์ในสิ่งที่ปรากฏ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

  if(R!=111) // ถ้าไม่หยุดการทำงาน
  {

```

```

    motor_left();
    delay(2000);
  }
}

void motor_left()
{
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  digitalWrite(enb,HIGH);
}

void motor_stop()
{
  digitalWrite(in3,LOW);
  digitalWrite(in4,LOW);
  digitalWrite(enb,HIGH);
}

void receiveEvent(int A)
{
  A = Wire.read();
  R=A;
}

```

ก.6 โปรแกรมส่วนตัดแยก (Part C)

```
#include <Wire.h>
```

```

int g=0,turn=0;// g=ลำดับเหตุการณ์ เข้า=0
int val,val_sum,cal;
int code,num=1,non=1;
int senB = 7;

```

```
void setup()
```

```
{
```

```

pinMode(2,INPUT); //สำหรับขา //Sensor_Motor_2(Rotate) ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
pinMode(3,INPUT); //Sensor_Motor_3(Rotate) เจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
pinMode(4,INPUT); //Sensor_IR_1(Front กล่องชนด้านใน)
pinMode(5,INPUT); //Sensor_IR_2(Center กล่องเข้าออก)

```

```

pinMode(6,INPUT);           //Sensor_IR_3(Check_colour)
pinMode(7,OUTPUT);         //to B stop
pinMode(9,OUTPUT);         //Motor_Line
pinMode(10,OUTPUT);        //Motor_Line
pinMode(11,OUTPUT);        //Motor_Rotate
pinMode(12,OUTPUT);        //Motor_Rotate
Serial.begin(9600);
Serial.println("AVR Mecah is conect");
Wire.begin(3);
Wire.onRequest(requestEvent);
Wire.onReceive(receiveEvent);
}

void loop()
{
int deg=0,n=0,i=0;
int d=0,g=5;non=num=1;
check_code(g,turn);
delay(1000);
while(digitalRead(6)==HIGH)
{
digitalWrite(12,HIGH);// rotate to limit switch
digitalWrite(11,LOW);
}
digitalWrite(11,HIGH);
digitalWrite(12,HIGH);
deg=deg-565; // มุมจาก limit switch ถึงตรงกลางต่อ part B
g=0;
while(non==1)// ทำงานตลอด
{

if(d==0)
{
Serial.println(" num");
Serial.println(num);
d=1;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
check_code(g,turn);

if((digitalRead(5)==LOW)&&(g==0))//motor enter from B to C when sensorfront work
{
  digitalWrite(9,HIGH);
  digitalWrite(10,LOW);
  g++;
}

else if((digitalRead(4)==LOW)&&(g==1))// motor stop when when sensorcenter work
{
  digitalWrite(senB,HIGH);
  digitalWrite(9,HIGH);
  digitalWrite(10,HIGH);
  delay(1000);
  val_sum = analogRead(6); // then sensor colour work
  cal=0;
  while(cal<=100)
  {
    val = analogRead(6);
    val_sum = (val_sum+val)/2;
    cal++;
  }
  Serial.println("what is");
  Serial.println(val_sum);
  if(val_sum>71&&g==1) // window blackgreen
  {
    turn=1;
    g++;
  }
  else if(val_sum<70&&g==1) // table lightblue
  {
    turn=2;
    g++;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if((digitalRead(4)==LOW)&&(g==3))// C goto D/E when sensor center work
{
  delay(2000);
  digitalWrite(10,HIGH);
  digitalWrite(9,LOW);
  g++;
}

```

```

else if((digitalRead(5)==LOW)&&(g==4))// sensor front work then
{
  delay(1000);
  if((digitalRead(5)==HIGH)) // box go to D/E sensor don't work
  {
    delay(3000);
    digitalWrite(9,HIGH); // motor conveyer stop
    digitalWrite(10,HIGH);
    g++; // g==4 now g==5
    delay(2000);
  }
}

```

```

else if((g==2)&&(turn==1)) // blkgreen
{
  delay(2000);
  Serial.println("This deg is -90");
  deg=deg-490; // to window part D
  g++;
}

```

```

else if((g==5)&&(turn==1))// already send to part D back to center
{
  Serial.println("This deg is 0");
  deg=deg+475;
  g=0;
  num++;
  turn=0;
  d=0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if((g==2)&&(turn==2)) // to table blue
{
    delay(2000);
    Serial.println("This deg is 90");
    deg=deg+470;
    g++;
}

```

```

else if((g==5)&&(turn==2)) //already send to part E back to center
{
    Serial.println("This deg is 0");
    deg=deg-550;
    g=0;
    num++;
    turn=0;
    d=0;
}

int m=0;
while(m==0)
{
    if(n<deg)
    {
        digitalWrite(12,HIGH);// go to table
        digitalWrite(11,LOW);
    }
    else if(n>deg)
    {
        digitalWrite(11,HIGH);
        digitalWrite(12,LOW);
    }
    if(n==deg)
    {
        digitalWrite(12,HIGH);// go to window
        digitalWrite(11,HIGH);
        n=0;
        deg=0;
    }
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        m=1;
    }
    else if((digitalRead(2)==1)&&(digitalRead(3)==0)&&(i==1)&&(n<deg))// encoder
    {
        n++;
        i=0;
    }
    else if((digitalRead(2)==0)&&(digitalRead(3)==1)&&(i==0)&&(n<deg))
    {
        n++;
        i=1;
    }
    else if((digitalRead(2)==1)&&(digitalRead(3)==0)&&(i==1)&&(n>deg))
    {
        n--;
        i=0;
    }
    else if((digitalRead(2)==0)&&(digitalRead(3)==1)&&(i==0)&&(n>deg))
    {
        n--;
        i=1;
    }
} // ปิด while(m==0)

} // close while(non==1)
} // close void loop

void requestEvent()
{
    Wire.write(code);
}

void receiveEvent(int X)
{
    X=Wire.read();
    int A=X;
}

int check_code(int g,int turn)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    if(g==0)
    {
        code=999;
    }
    else if(g==1)
    {
        code=202;
    }
    else if(g==3&&turn==1) // part D ทำงาน window blackgreen
    {
        code=203;
    }
    else if(g==3&&turn==2) //Part E ทำงาน table blue
    {
        code=204;
    }
    /*else if(g==3)
    {
        code=204;
    }
    else if(g==4)
    {
        code=205;
    }*/
    else if(g==5) // part A, B ทำงาน
    {
        code=206;
    }
    return code;
}

```

ก.7 โปรแกรมส่วนการแสดงผล (Visual C#)

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Data;
```

```
using System.Drawing;
```

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันวิจัยและพัฒนาเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุผลเชิงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
```

```
namespace Writebymm
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
            serialPort1.Open();
```

```
        }
```

```
        private string sensor;
```

```
        private int A = 1;
```

```
        private void btnStart_Click(object sender, EventArgs e)
```

```
        {
```

```
            serialPort1.Write("1");
```

```
            txtwork.Text = "Start";
```

```
            timer1.Enabled = true;
```

```
            timer2.Enabled = true;
```

```
            timer3.Enabled = true;
```

```
            timer4.Enabled = true;
```

```
            timer5.Enabled = true;
```

```
            timer6.Enabled = true;
```

```
            pictureBoxA.ImageLocation = "C:/Project2012/MODEL/Part A/aniA.gif";
```

```
            pictureBoxB.ImageLocation = "C:/Project2012/MODEL/Part B/aniB.gif";
```

```
            pictureBoxC.ImageLocation = "C:/Project2012/MODEL/Part C/C0.jpg";
```

```
            pictureBoxD.ImageLocation = "C:/Project2012/MODEL/aniD.gif";
```

```
            pictureBoxE.ImageLocation = "C:/Project2012/MODEL/aniE.gif";
```

```
            txtA.Text = "Work";
```

```
            txtB.Text = "Work";
```

```
            txtC.Text = "Work";
```

```
            txtD.Text = "Work";
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        txtE.Text = "Work";

    }

private void btnStop_Click(object sender, EventArgs e)
{
    serialPort1.Write("2");
    if (A == 1)
    {
        txtwork.Text = "Stop";
        timer1.Enabled = false;
        timer2.Enabled = false;
        timer3.Enabled = false;
        timer4.Enabled = false;
        timer5.Enabled = false;
        timer6.Enabled = false;
        pictureBoxA.ImageLocation = "C:/Project2012/MODEL/Part A/A0.jpg";
        pictureBoxB.ImageLocation = "C:/Project2012/MODEL/Part B/B0.jpg";
        pictureBoxC.ImageLocation = "C:/Project2012/MODEL/Part C/C0.jpg";
        pictureBoxD.ImageLocation = "C:/Project2012/MODEL/Blue/B0.jpg";
        pictureBoxE.ImageLocation = "C:/Project2012/MODEL/Blue/B0.jpg";
        txtA.Text = "Stop";
        txtB.Text = "Stop";
        txtC.Text = "Stop";
        txtD.Text = "Stop";
        txtE.Text = "Stop";
    }
}

private void btnReset_Click(object sender, EventArgs e)
{
    serialPort1.Write("3");
    txtwork.Text = "Reset";
    timer1.Enabled = false;
    timer2.Enabled = false;
    timer3.Enabled = false;
    timer4.Enabled = false;
    timer5.Enabled = false;

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุผลแบบสงวนเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

timer6.Enabled = false;
txtA.Clear();
txtB.Clear();
txtC.Clear();
txtD.Clear();
txtE.Clear();
pictureBoxA.ImageLocation = "C:/Project2012/MODEL/Part A/A0.jpg";
pictureBoxB.ImageLocation = "C:/Project2012/MODEL/Part B/B0.jpg";
pictureBoxC.ImageLocation = "C:/Project2012/MODEL/Part C/C0.jpg";
pictureBoxD.ImageLocation = "C:/Project2012/MODEL/Blue/B0.jpg";
pictureBoxE.ImageLocation = "C:/Project2012/MODEL/Blue/B0.jpg";
txtA.Text = "Reset";
txtB.Text = "Reset";
txtC.Text = "Reset";
txtD.Text = "Reset";
txtE.Text = "Reset";
}

private void btnPort_Click(object sender, EventArgs e)
{
    if (serialPort1.DiscardNull)
    {
        serialPort1.Open();
        groupBox1.Enabled = true;
    }
    else
    {
        MessageBox.Show("ไม่ได้ต่อพอร์ท");
        groupBox1.Enabled = false;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข.1 คู่มือการใช้ LM298D



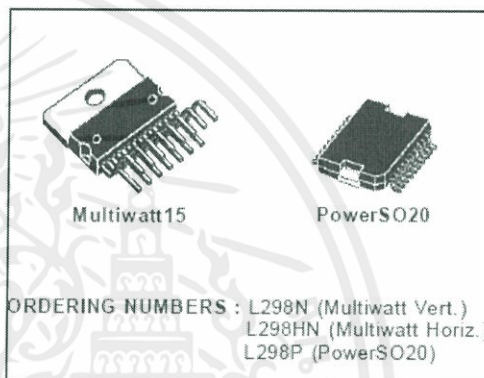
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

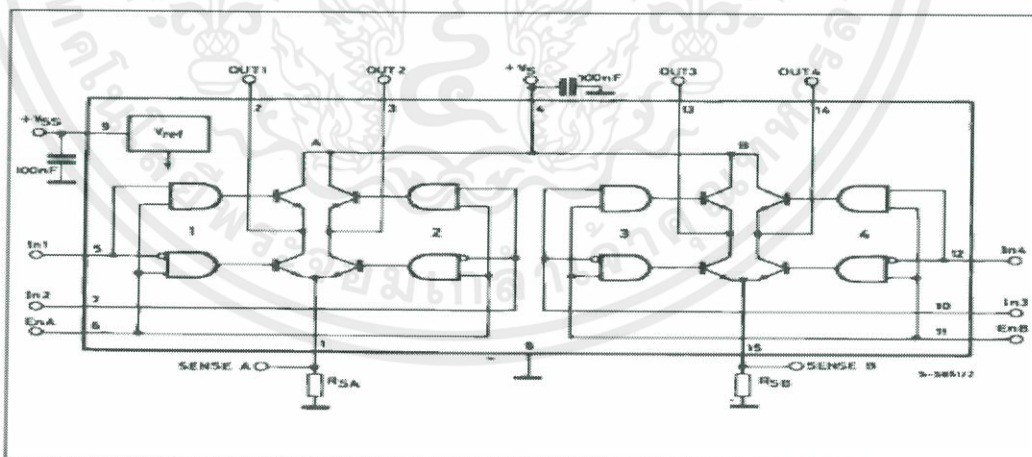
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

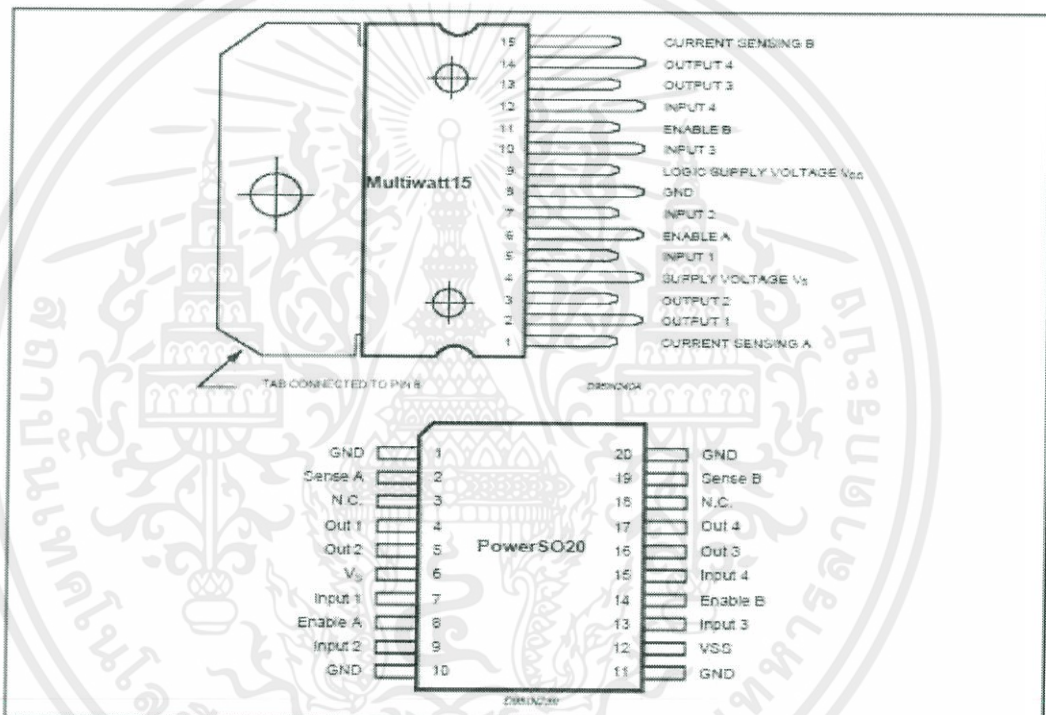


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{En}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100 μ s)	3	A
	- Repetitive (80% on -20% off; t _{on} = 10ms)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	°C
T_{stg}, T_J	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th(j-case)}$	Thermal Resistance Junction-case	Max.	-	3	°C/W
$R_{th(j-amb)}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN FUNCTIONS (refer to the block diagram)

MW. 15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2,3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5,7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS ($V_S = 42V$; $V_{SS} = 5V$, $T_j = 25^\circ C$; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	$V_{IH} + 2.5$		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	$V_{en} = H$; $I_L = 0$ $V_i = L$ $V_i = H$		13 50	22 70	mA mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	$V_{en} = H$; $I_L = 0$ $V_i = L$ $V_i = H$ $V_{en} = L$ $V_i = X$		24 7	36 12 6	mA mA mA
V_{IL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V_{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V_{SS}	V
I_{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	$V_i = L$			-10	μA
I_{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	$V_i = H \leq V_{SS} - 0.6V$		30	100	μA
$V_{en} = L$	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
$V_{en} = H$	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
$I_{en} = L$	Low Voltage Enable Current (pins 6, 11)	$V_{en} = L$			-10	μA
$I_{en} = H$	High Voltage Enable Current (pins 6, 11)	$V_{en} = H \leq V_{SS} - 0.6V$		30	100	μA
$V_{CEsat(H)}$	Source Saturation Voltage	$I_L = 1A$ $I_L = 2A$	0.85	1.35 2	1.7 2.7	V V
$V_{CEsat(L)}$	Sink Saturation Voltage	$I_L = 1A$ (5) $I_L = 2A$ (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	$I_L = 1A$ (5) $I_L = 2A$ (5)	1.80		3.2 4.9	V V
V_{Sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _I)	Source Current Turn-off Delay	0.5 V _I to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _I)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _I)	Source Current Turn-on Delay	0.5 V _I to 0.1 I _L (2); (4)		2		μs
T ₄ (V _I)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _I)	Sink Current Turn-off Delay	0.5 V _I to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _I)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _I)	Sink Current Turn-on Delay	0.5 V _I to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _I)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _I)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1V for t_s ≤ 50 μsec; In steady state V_{min} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

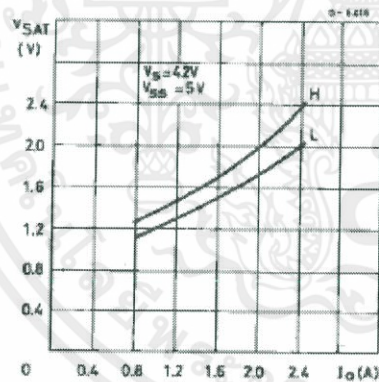
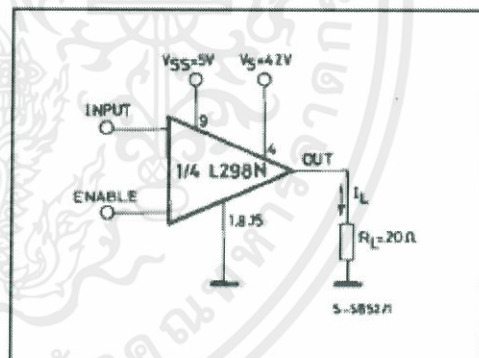


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

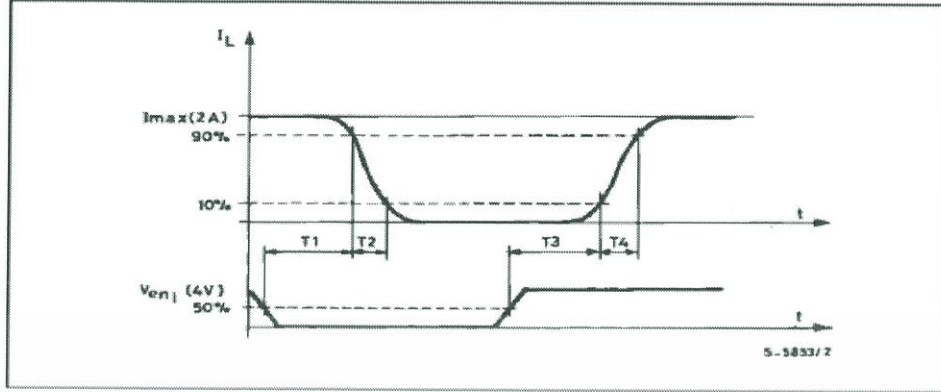
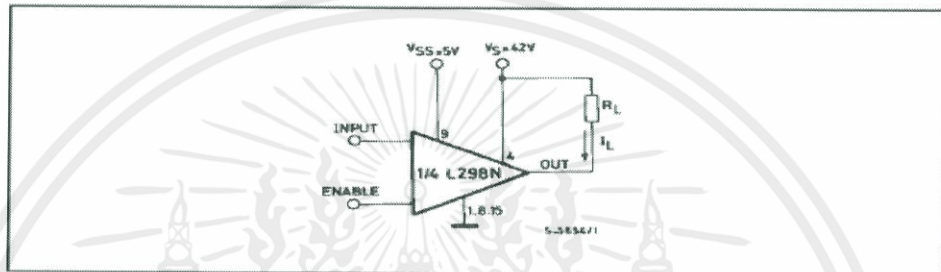
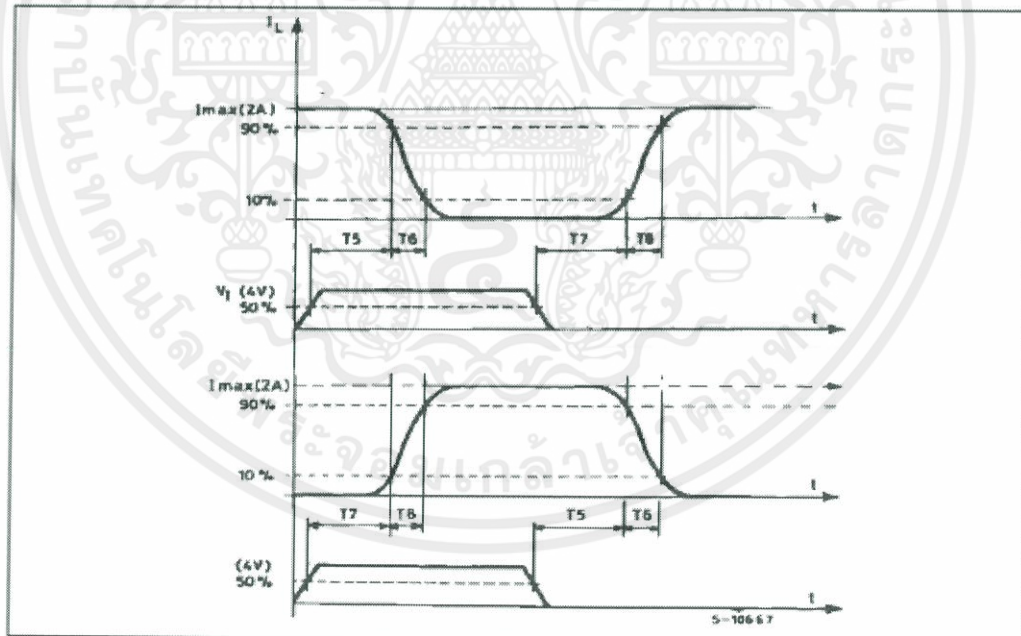


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.

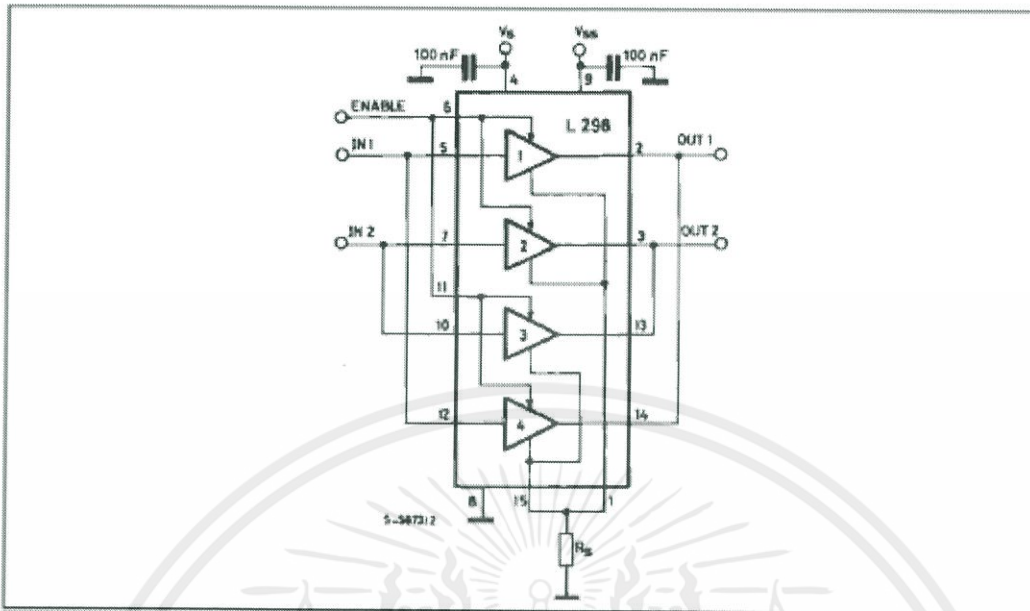
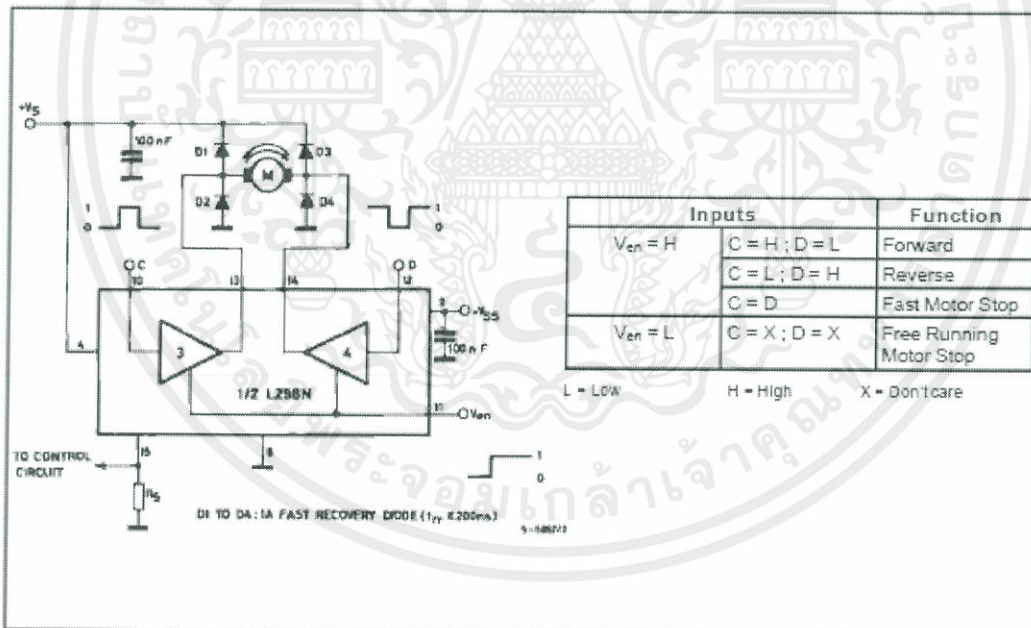


Figure 6 : Bidirectional DC Motor Control.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A, B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output: an external resistor (R_{SA} ; R_{SB} .) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1; In2; EnA and In3; In4; EnB. The In inputs set the bridge state when The En input is high; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

This solution can drive until 3 Amps in DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off: Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{tr} \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Schottky diodes would be preferred.

Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

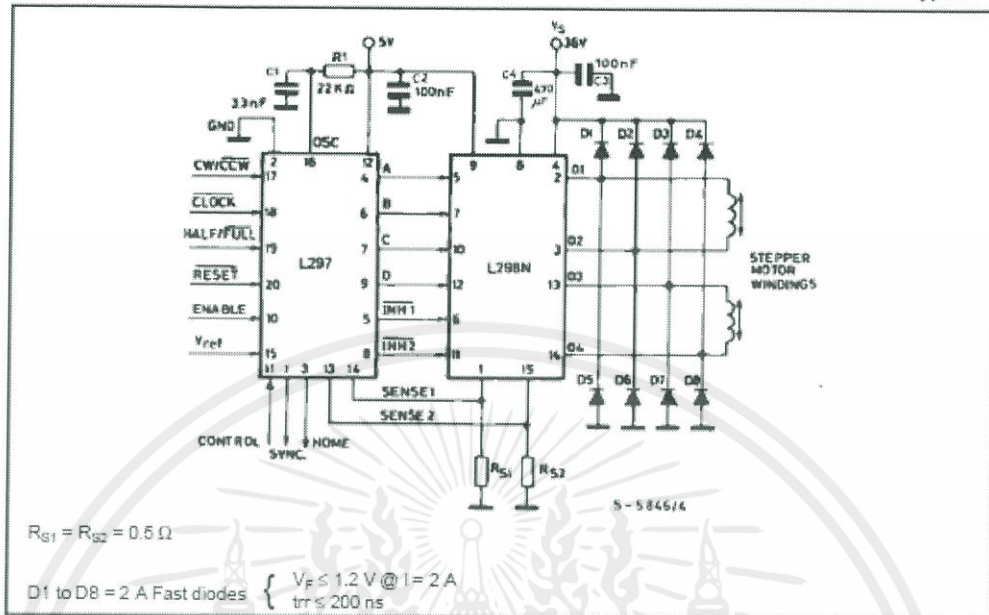
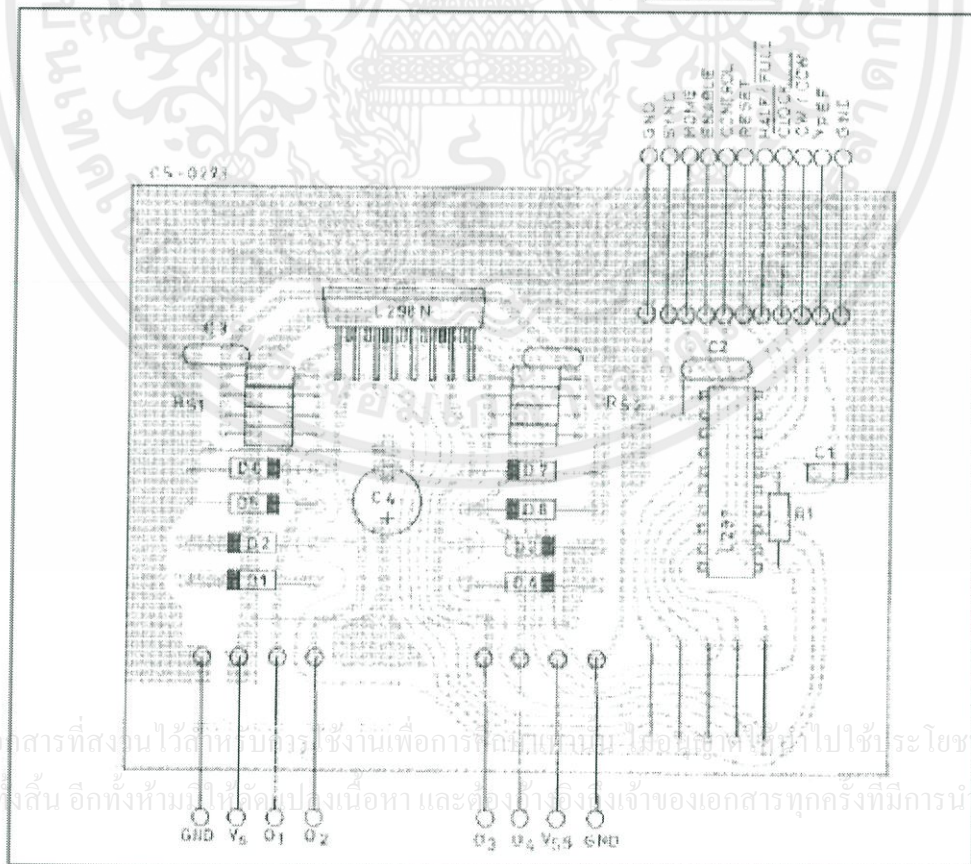
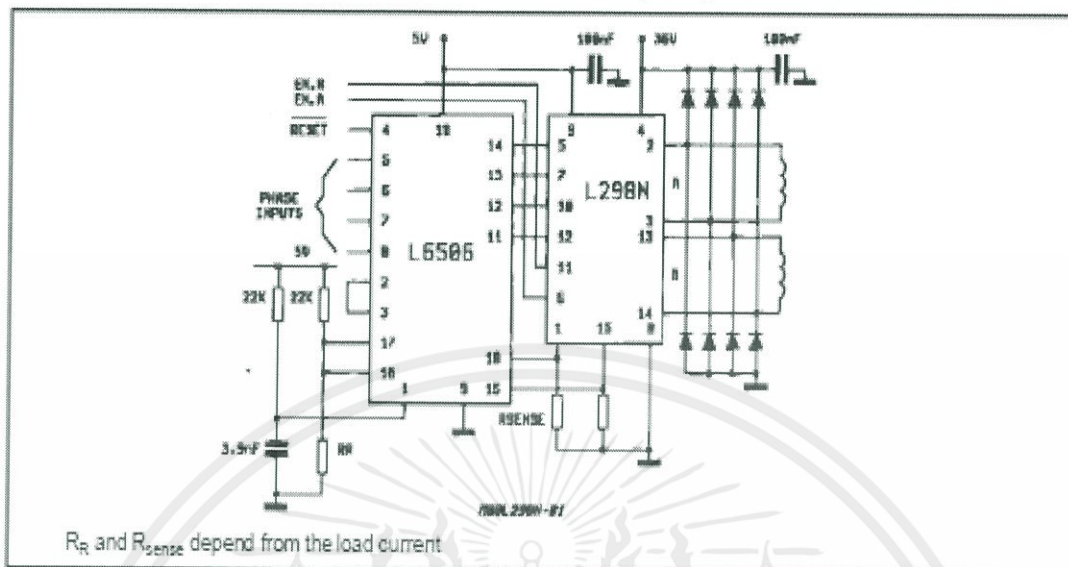


Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).



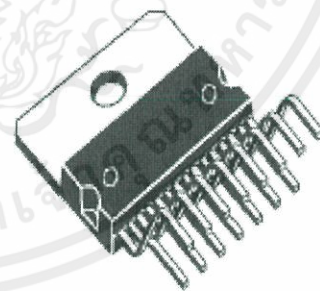
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาดูแลเรียน ไม่อนุญาตให้ผู้ใช้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องแจ้งเจ้าของเอกสารทุกครั้งที่มีกรณีไปใช้

Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.



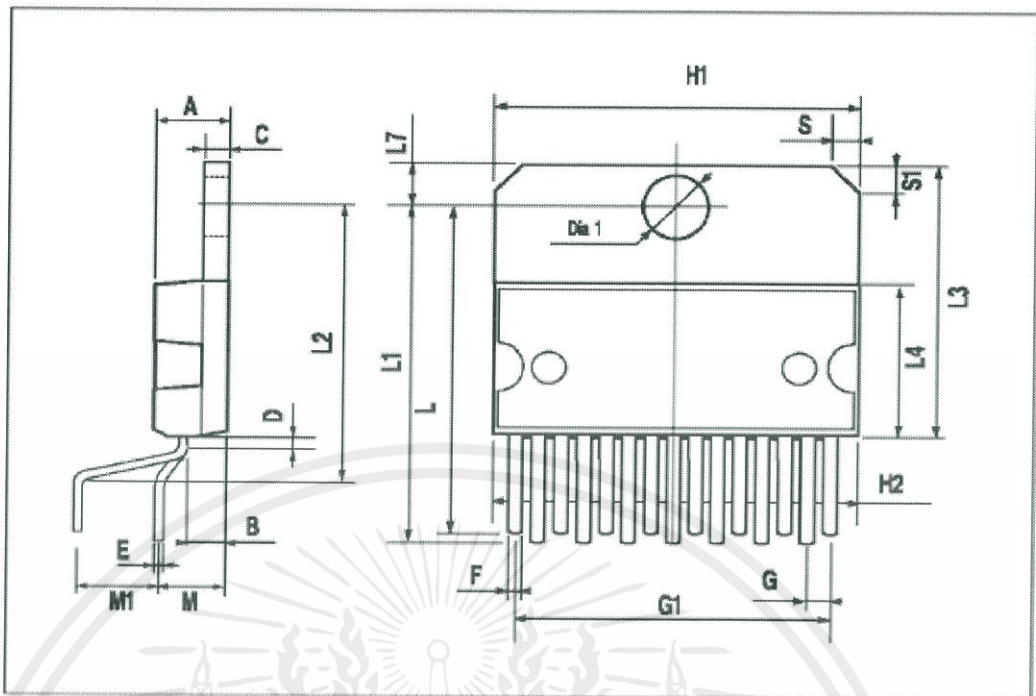
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.408	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia	3.65	3.85	4.05	0.144	0.152	0.160

OUTLINE AND MECHANICAL DATA



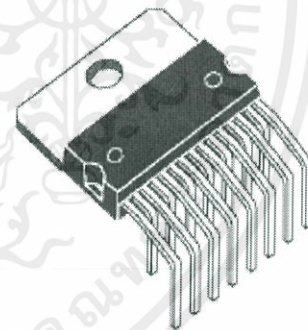
Multiwatt15 V

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.68		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.796
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



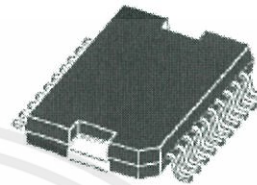
Multiwatt15 H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

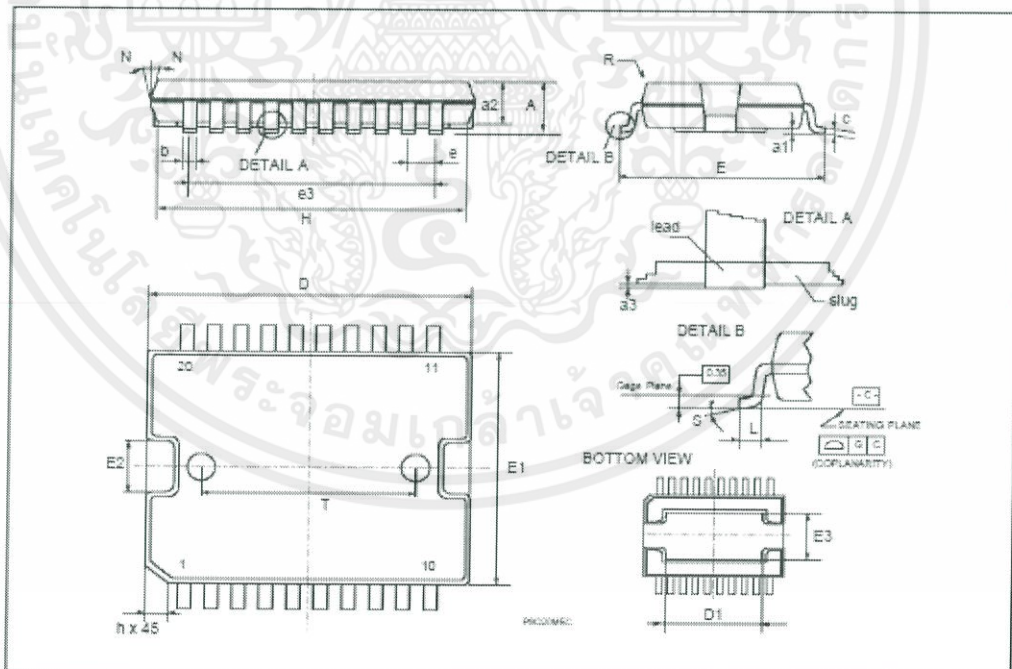
(1) "D and F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2คู่มือการใช้ TCRT5000



TCRT5000(L)

Vishay Semiconductors

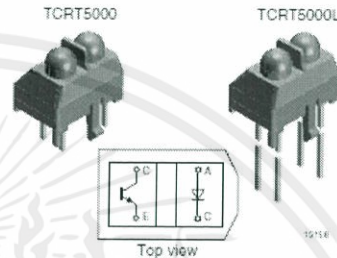
Reflective Optical Sensor with Transistor Output

Description

The TCRT5000 and TCRT500L are reflective sensors which include an infrared emitter and phototransistor in a leaded package which blocks visible light. The package includes two mounting clips. TCRT5000L is the long lead version.

Features

- Package type: Leaded
- Detector type: Phototransistor
- Dimensions:
L 10.2 mm x W 5.8 mm x H 7.0 mm
- Peak operating distance: 2.5 mm
- Operating range: 0.2 mm to 15 mm
- Typical output current under test: $I_C = 1 \text{ mA}$
- Daylight blocking filter
- Emitter wavelength 950 nm
- Lead (Pb)-free soldering released
- Lead (Pb)-free component in accordance to RoHS 2002/95/EC and WEEE 2002/96/EC



Applications

- Position sensor for shaft encoder
- Detection of reflective material such as paper, IBM cards, magnetic tapes etc.
- Limit switch for mechanical motions in VCR
- General purpose - wherever the space is limited

Order Instructions

Part Number	Remarks	Minimum Order Quantity
TCRT5000	3.5 mm lead length	4500 pcs, 50 pcs/tube
TCRT5000L	15 mm lead length	2400 pcs, 48 pcs/tube

Absolute Maximum Ratings

$T_{amb} = 25 \text{ }^{\circ}\text{C}$, unless otherwise specified

Input (Emitter)

Parameter	Test condition	Symbol	Value	Unit
Reverse voltage		V_R	5	V
Forward current		I_F	60	mA
Forward surge current	$t_p \leq 10 \text{ } \mu\text{s}$	I_{FSM}	3	A
Power dissipation	$T_{amb} \leq 25 \text{ }^{\circ}\text{C}$	P_V	100	mW
Junction temperature		T_J	100	$^{\circ}\text{C}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCRT5000(L)

Vishay Semiconductors

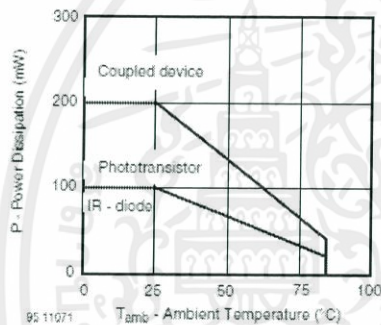


Output (Detector)

Parameter	Test condition	Symbol	Value	Unit
Collector emitter voltage		V_{CEO}	70	V
Emitter collector voltage		V_{ECO}	5	V
Collector current		I_C	100	mA
Power dissipation	$T_{amb} \leq 55^\circ\text{C}$	P_V	100	mW
Junction temperature		T_J	100	$^\circ\text{C}$

Sensor

Parameter	Test condition	Symbol	Value	Unit
Total power dissipation	$T_{amb} \leq 25^\circ\text{C}$	P_{tot}	200	mW
Operation temperature range		T_{amb}	-25 to +85	$^\circ\text{C}$
Storage temperature range		T_{stg}	-25 to +100	$^\circ\text{C}$
Soldering temperature	2 mm from case, $t \leq 10$ s	T_{sd}	260	$^\circ\text{C}$



Electrical Characteristics

$T_{amb} = 25^\circ\text{C}$, unless otherwise specified

Input (Emitter)

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Forward voltage	$I_F = 60$ mA	V_F		1.25	1.5	V
Junction capacitance	$V_R = 0$ V, $f = 1$ MHz	C_J		17		pF
Radiant intensity	$I_F = 60$ mA, $t_P = 20$ ms	I_E			21	mW/sr
Peak wavelength	$I_F = 100$ mA	λ_P	940			nm
Virtual source diameter	Method: 63 % encircled energy	\varnothing		2.1		mm

Output (Detector)

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Collector emitter voltage	$I_C = 1$ mA	V_{CEO}	70			V
Emitter collector voltage	$I_E = 100$ μA	V_{ECO}	7			V
Collector dark current	$V_{CE} = 20$ V, $I_F = 0$, $E = 0$	I_{CEO}		10	200	nA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



TCRT5000(L)

Vishay Semiconductors

Sensor

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Collector current	$V_{CE} = 5\text{ V}$, $I_F = 10\text{ mA}$, $D = 12\text{ mm}$	I_C ^{1,2)}	0.5	1	2.1	mA
Collector emitter saturation voltage	$I_F = 10\text{ mA}$, $I_C = 0.1\text{ mA}$, $D = 12\text{ mm}$	V_{CEsat} ^{1,2)}			0.4	V

¹⁾ See figure 3

²⁾ Test surface: Mirror (Mfr. Spindler a. Hoyer, Part No 340005)

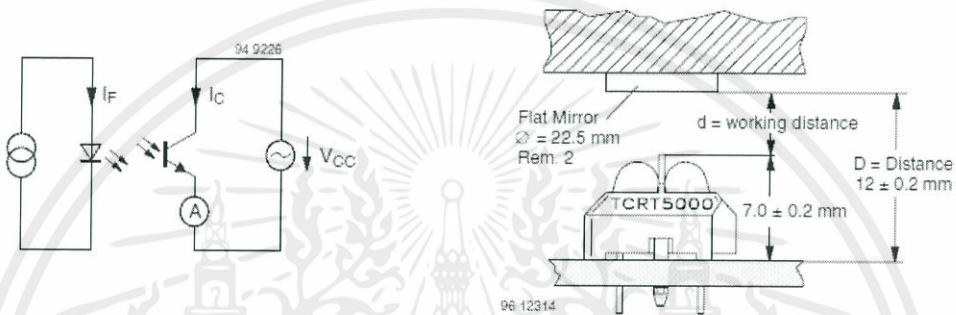


Figure 2. Test Circuit

Figure 3. Test Circuit

Typical Characteristics

$T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified

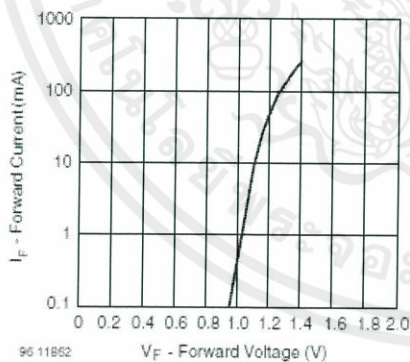


Figure 4. Forward Current vs. Forward Voltage

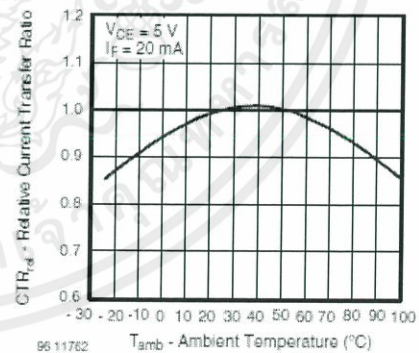
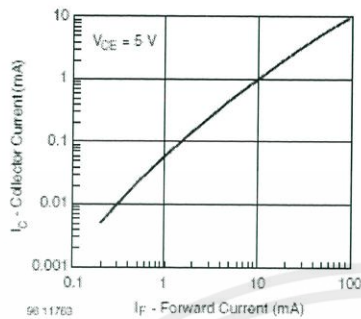


Figure 5. Relative Current Transfer Ratio vs. Ambient Temperature

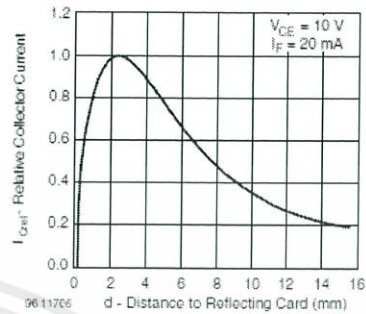
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCRT5000(L)

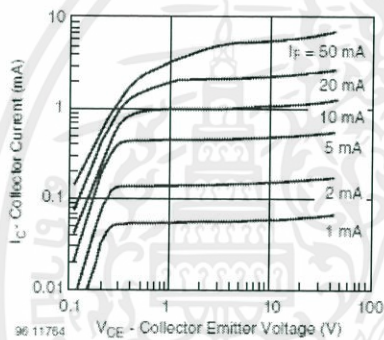
Vishay Semiconductors



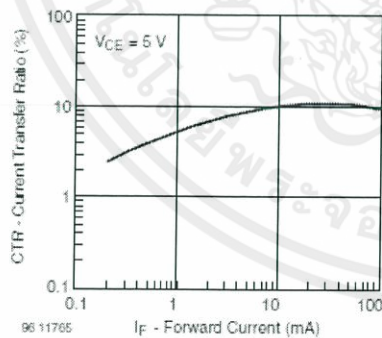
96 11763
Figure 6. Collector Current vs. Forward Current



96 11766
Figure 9. Relative Collector Current vs. Distance



96 11764
Figure 7. Collector Emitter Saturation Voltage vs. Collector Current



96 11765
Figure 8. Current Transfer Ratio vs. Forward Current

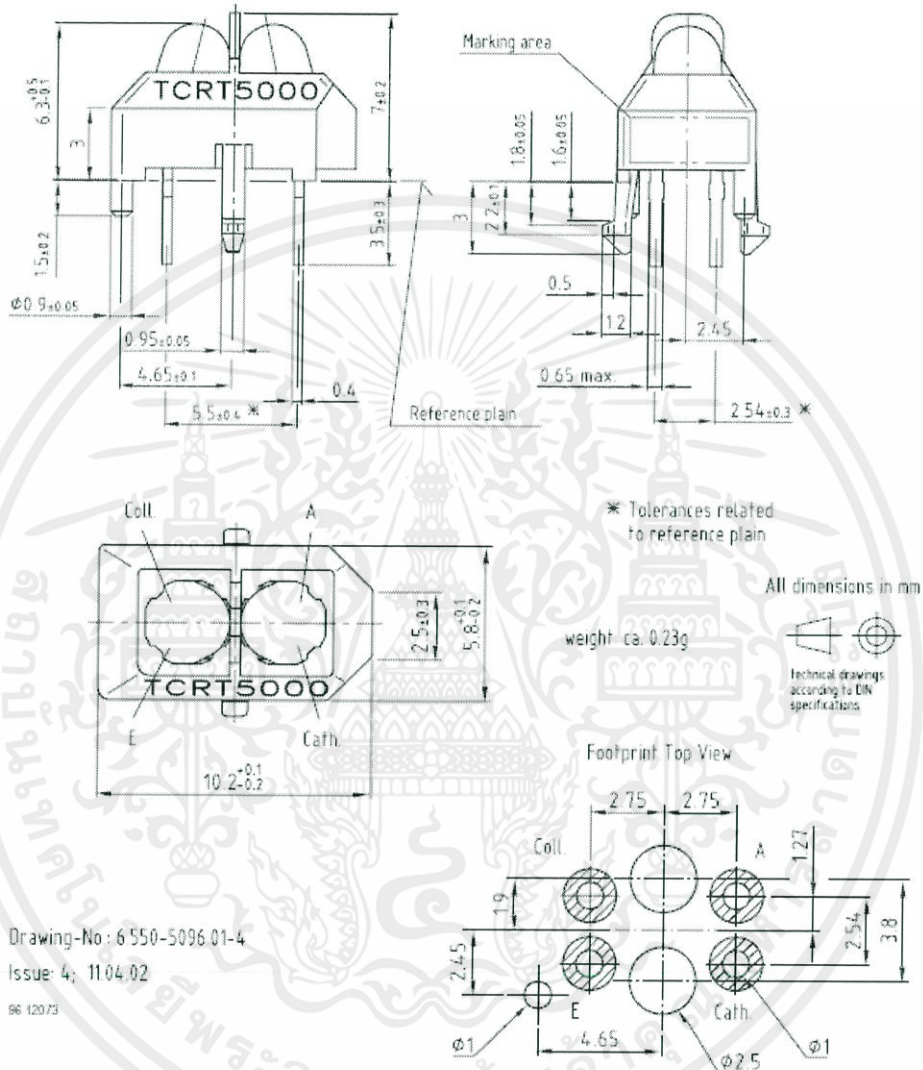
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



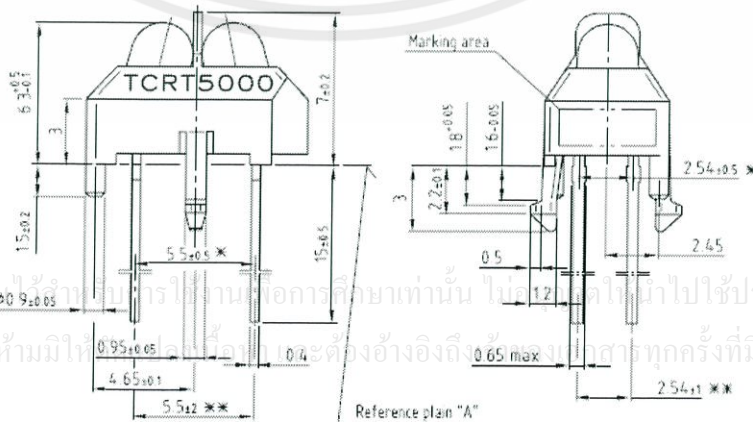
TCRT5000(L)

Vishay Semiconductors

Package Dimensions in mm



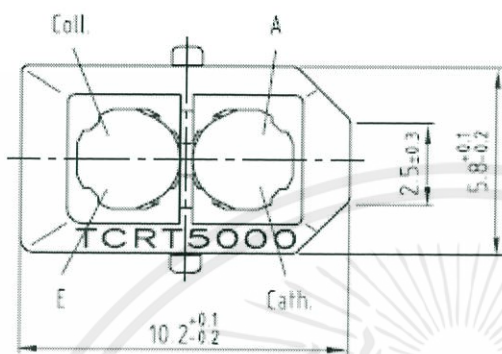
Drawing-No: 6550-5096.01-4
 Issue: 4; 11.04.02
 96 12073



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ผู้อื่นนำเอกสารนี้ไปเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาต

TCRT5000(L)

Vishay Semiconductors



* Tolerances related to reference plain "A"

** Tolerances related on lead end



Technical drawings according to DIN specifications

All dimensions in mm

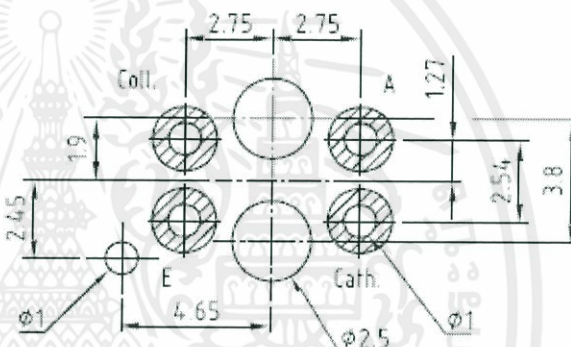
weight: ca. 0.23g

Drawing-No: 6.550-S146.01-4

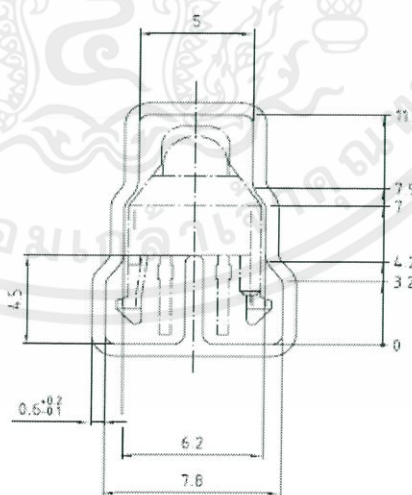
Issue: 4; 11.04.02

95 11267

Footprint Top View



TCRT5000, Tube Dimensions



With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm
All dimensions in mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานำไปใช้ประโยชน์ด้านการค้า

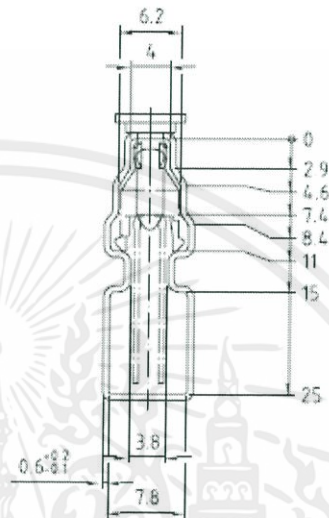
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอก และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Drawing-No: 9.700-5139.01-4
Issue: 1; 10.05.00
2000



TCRT5000(L)
Vishay Semiconductors

TCRT5000L, Tube Dimensions



With stopper pins
Tolerance: ± 0.5 mm
Length: 575 \pm 1 mm
All dimensions in mm

Drawing-No.: 9700-5178-01-4
Issue: 1, 25.02.00
200200

Ozone Depleting Substances Policy Statement

It is the policy of Vishay Semiconductor GmbH to

1. Meet all present and future national and international statutory requirements.
2. Regularly and continuously improve the performance of our products, processes, distribution and operating systems with respect to their impact on the health and safety of our employees and the public, as well as their impact on the environment.

It is particular concern to control or eliminate releases of those substances into the atmosphere which are known as ozone depleting substances (ODSs).

The Montreal Protocol (1987) and its London Amendments (1990) intend to severely restrict the use of ODSs and forbid their use within the next ten years. Various national and international initiatives are pressing for an earlier ban on these substances.

Vishay Semiconductor GmbH has been able to use its policy of continuous improvements to eliminate the use of ODSs listed in the following documents.

1. Annex A, B and list of transitional substances of the Montreal Protocol and the London Amendments respectively
2. Class I and II ozone depleting substances in the Clean Air Act Amendments of 1990 by the Environmental Protection Agency (EPA) in the USA
3. Council Decision 88/540/EEC and 91/690/EEC Annex A, B and C (transitional substances) respectively.

Vishay Semiconductor GmbH can certify that our semiconductors are not manufactured with ozone depleting substances and do not contain such substances.

เอกสารนี้เป็นเอกสารของบริษัท Vishay Semiconductor GmbH และสงวนลิขสิทธิ์ไว้สำหรับใช้
ไม่ว่ากรณีใดๆ ห้ามทำซ้ำโดยไม่ได้รับอนุญาต

TCRT5000(L)

Vishay Semiconductors



We reserve the right to make changes to improve technical design and may do so without further notice.

Parameters can vary in different applications. All operating parameters must be validated for each customer application by the customer. Should the buyer use Vishay Semiconductors products for any unintended or unauthorized application, the buyer shall indemnify Vishay Semiconductors against all claims, costs, damages, and expenses, arising out of, directly or indirectly, any claim of personal damage, injury or death associated with such unintended or unauthorized use.

Vishay Semiconductor GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany

Disclaimer

All product specifications and data are subject to change without notice.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, "Vishay"), disclaim any and all liability for any errors, inaccuracies or incompleteness contained herein or in any other disclosure relating to any product.

Vishay disclaims any and all liability arising out of the use or application of any product described herein or of any information provided herein to the maximum extent permitted by law. The product specifications do not expand or otherwise modify Vishay's terms and conditions of purchase, including but not limited to the warranty expressed therein, which apply to these products.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay.

The products shown herein are not designed for use in medical, life-saving, or life-sustaining applications unless otherwise expressly indicated. Customers using or selling Vishay products not expressly indicated for use in such applications do so entirely at their own risk and agree to fully indemnify Vishay for any damages arising or resulting from such use or sale. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

Product names and markings noted herein may be trademarks of their respective owners.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

การเริ่มต้นการใช้โปรแกรม Arduino

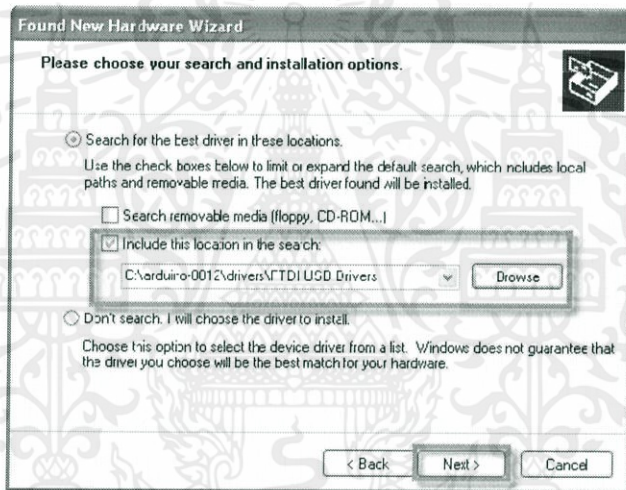
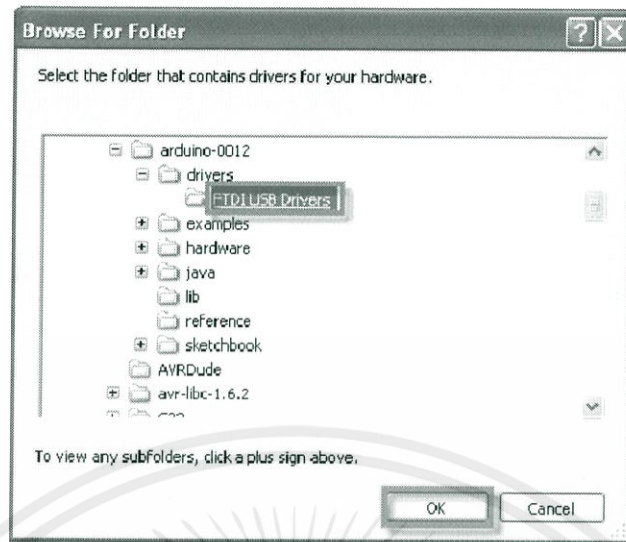
ค.1 การติดตั้ง Driver ของ USB Bridge ของบอร์ด ET-EASY168 STAMP

บอร์ด ET-EASY168 STAMP จะใช้ชิพ USB Bridge ของ FTDI เป็นตัวกลางในการเชื่อมต่อกับคอมพิวเตอร์ PC โดย USB Bridge ของ FTDI จะทำหน้าที่เป็นตัวกลางในการเชื่อมต่อและติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ PC กับ MCU ATmega168 ของบอร์ด ET-EASY168 STAMP ในรูปแบบของพอร์ตอนุกรม (Visual Com Port) โดยโปรแกรม Application ต่างๆที่ทำงานอยู่บนคอมพิวเตอร์ PC รวมทั้งโปรแกรม Arduino จะมองเห็น พอร์ต USB ที่เชื่อมต่อกับบอร์ด ET-EASY168 STAMP เป็นพอร์ตสื่อสารอนุกรม (Com Port) ช่องหนึ่งเท่านั้น ซึ่งถ้าเครื่องคอมพิวเตอร์ของผู้ใช้เคยทำการติดตั้ง Driver สำหรับ USB Bridge ของ FTDI ไว้ก่อนแล้ว เมื่อทำการเชื่อมต่อสาย USB ของบอร์ด ET-EASY168 STAMP เข้ากับ USB HUB คอมพิวเตอร์ PC ยังไม่เคยติดตั้ง Driver ของ FTDI ไว้ก่อนก็จะต้องทำการติดตั้ง Driver ให้กับบอร์ดให้เรียบร้อยเสียก่อนซึ่งมีลำดับขั้นตอนดังนี้

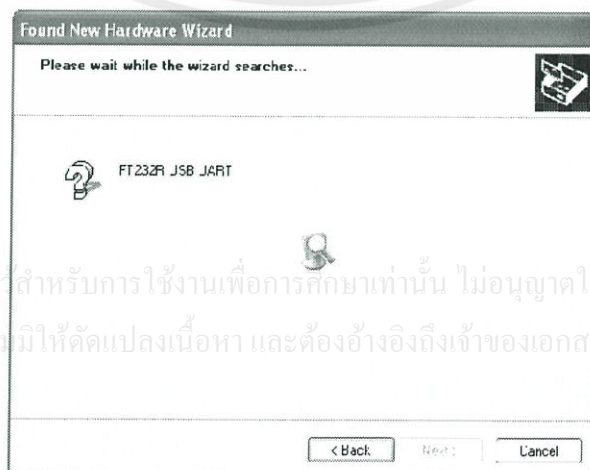
1. เตรียมแผ่น CD ROM ที่บรรจุ Driver ของ FTDI ไว้ให้พร้อม หรือ ในกรณีที่ผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduino ไว้เรียบร้อยแล้ว ภายในโฟลเดอร์ของโปรแกรม Arduino ก็จะมี Driver ของ FTDI จัดเตรียมไว้ให้เรียบร้อยแล้ว โดยจะอยู่ที่ “C:\arduino-0012\drivers\FTDI USB Drivers\”
2. ทำการเสียบสาย USB ของบอร์ด ET-EASY168 STAMP เข้ากับพอร์ต USB HUB ของเครื่องคอมพิวเตอร์ PC ซึ่ง Windows จะตรวจพบอุปกรณ์ใหม่ โดยเป็น “FT232R USB UART” และแจ้งให้ผู้ใช้ทำการติดตั้ง Driver ให้กับอุปกรณ์ ดังรูป



3. ให้เลือก Install from list or specific location (Advanced) แล้วเลือก Next ซึ่ง Windows ก็จะแจ้งให้ผู้ใช้ระบุตำแหน่งโฟลเดอร์ที่บรรจุไฟล์ Driver ของ FTDI ไว้ ก็ให้เลือกที่ Browse และเลือกไปยัง Drive และ โฟลเดอร์ที่เก็บไฟล์ Driver ไว้ ซึ่งถ้าผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduino ไว้แล้ว ก็ให้เลือกไปที่ “C:\arduino-0012\drivers\FTDI USB Drivers” แล้วเลือก Next ดังรูป



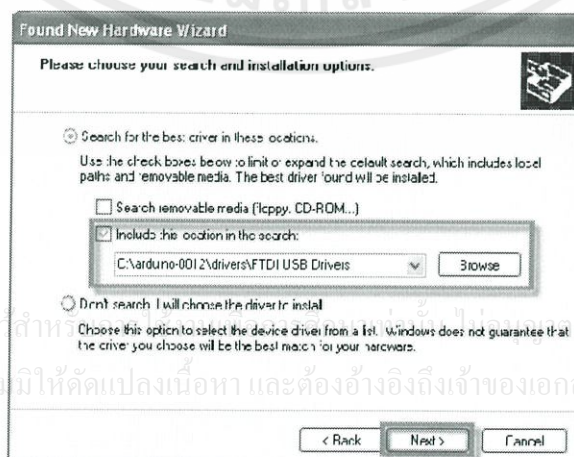
4. ในขั้นตอนนี้โปรแกรม Windows จะทำการค้นหาและติดตั้ง Driver ให้กับอุปกรณ์ ให้รอสักครู่จนการทำงานเสร็จเรียบร้อย แล้วเลือก Finish ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

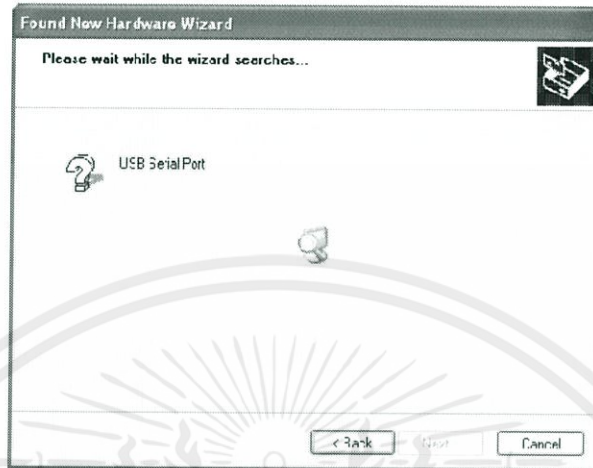


5. หลังจากทำการติดตั้ง Driver ของฮาร์ดแวร์เรียบร้อยแล้ว Windows ก็จะตรวจพบว่ามีอุปกรณ์ใหม่ ถูกเชื่อมต่ออยู่ โดยเป็นอุปกรณ์ประเภท “USB Serial Port” และแจ้งให้ผู้ใช้ทำการติดตั้ง Driver ให้กับอุปกรณ์ใหม่ที่ระบุเป็น “USB Serial Port” อีกครั้งหนึ่ง ซึ่งก็ให้เลือกระบุตำแหน่งไฟล์เดอไรที่เก็บไฟล์ Driver ไว้ ซึ่งให้เลือกเหมือนขั้นตอนในหัวข้อที่ 3 ดังรูป



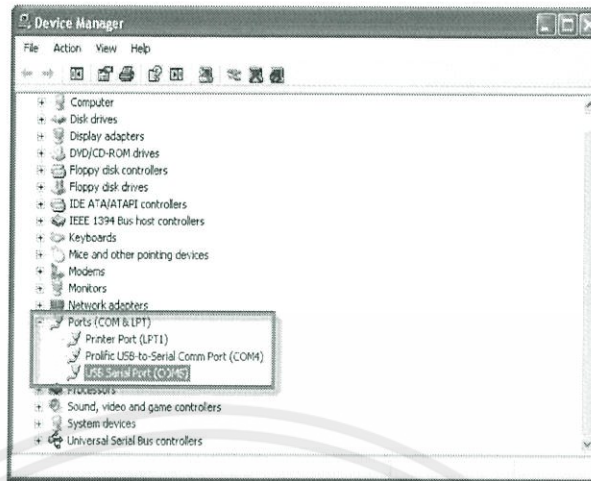
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... หน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ในขั้นตอนนี้โปรแกรม Windows จะทำการค้นหาและติดตั้ง Driver ให้กับอุปกรณ์ให้หรือสักครู่จนการทำงานเสร็จเรียบร้อยแล้วเลือก Finish ดังรูป



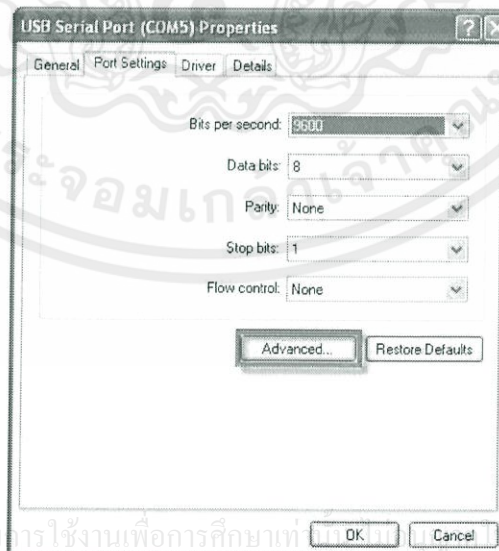
7. หลังจากทำการติดตั้ง Driver เรียบร้อยแล้ว ก็สามารถใช้งานอุปกรณ์ได้แล้ว แต่เพื่อความถูกต้องในครั้งแรกนี้ควรต้องเข้าไปทำการตรวจสอบและปรับแต่งค่าให้กับอุปกรณ์ก่อน โดยในขั้นตอนนี้ให้ไปที่ “My Computer → Control Panel → System → Hardware → Device Manager” แล้วทำการตรวจสอบที่ Ports (COM&LPT) แล้วดูที่ชื่อของ “USB Serial Port” ซึ่งให้ผู้ใช้จดจำหมายเลขของ Com Port ของอุปกรณ์ดังกล่าวไว้ เพื่อใช้อ้างอิงถึงในการเรียกใช้งาน ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

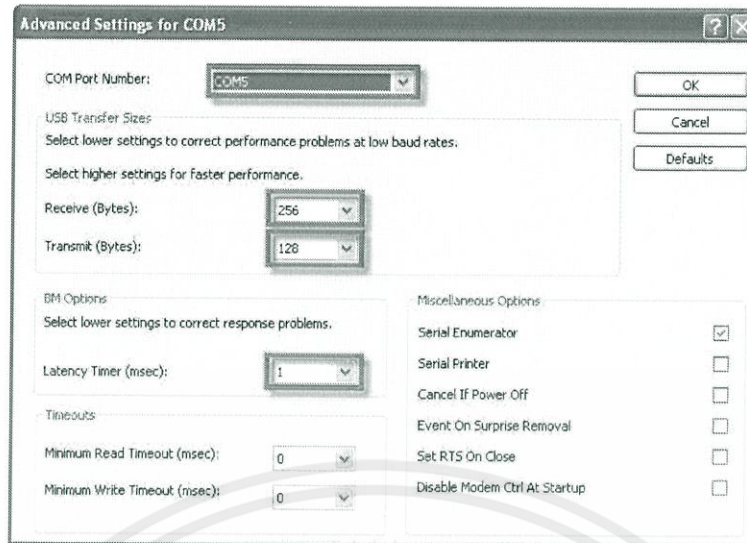


8. ในขั้นตอนนี้ให้คลิกเมาส์ที่เครื่องหมาย (+) หน้าหัวข้อ Ports(COM&LPT) แล้วมองหาอุปกรณ์ที่ชื่อ“USB Serial Port” ตามที่เราได้ทำการติดตั้ง Driver ไว้เรียบร้อยแล้ว หรือ ถ้าไม่แน่ใจว่าอุปกรณ์ดังกล่าวใช่อุปกรณ์ที่เป็นของบอร์ด “ET-EASY168 STAMP” หรือไม่ ให้ทดสอบด้วยการถอดสายUSB ออก รายชื่ออุปกรณ์ดังกล่าวจะต้องหายไป แต่เมื่อเสียบสาย USB กลับเข้ามาใหม่ รายชื่อของอุปกรณ์ดังกล่าวก็จะต้องปรากฏให้เห็นอีกครั้ง ถ้าทุกอย่างถูกต้อง ก็ให้ทำการคลิกเมาส์ที่ Tabรายชื่อของอุปกรณ์ดังกล่าว เมื่อปรากฏหน้าต่าง USB Serial Port Properties ขึ้นมาแล้วให้เลือกที่ Port Setting แล้วเลือก Advance เพื่อเข้าไปกำหนดค่าให้กับอุปกรณ์ให้เรียบร้อย ดังนี้

- a. USB Transfer Size → Receive (Bytes) ให้กำหนดเป็น 256
- b. USB Transfer Size → Transmit (Bytes) ให้กำหนดเป็น 128
- c. BM Option → Latency Timer (mSec) ให้กำหนดเป็น 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ค.2 การพัฒนาโปรแกรม ET-EASY168 STAMP แบบ Arduino Project

ตามปรกติแล้วบอร์ด ET-EASY168 STAMP จะทำการ ติดตั้งโปรแกรม Bootloaderไว้ให้เรียบร้อยแล้ว โดยใช้ Bootloaderชื่อ “BOOT_EASY168_AUTO_16MHZ.HEX” ซึ่งเป็น Bootloaderที่ทางทีมงานอีทีที ได้นำต้นฉบับจาก Arduinoมาปรับปรุงเงื่อนไขการทำงานใหม่ เพื่อให้การทำงานสอดคล้องกับระบบฮาร์ดแวร์ของบอร์ด ET-EASY168 STAMP ได้ดียิ่งขึ้น โดยโปรแกรม Bootloaderนี้จะใช้สำหรับติดต่อสื่อสารเพื่อสั่ง Upload Code จากคอมพิวเตอร์ PC ให้กับ MCU ในบอร์ดทำงาน โดยไม่ต้องใช้เครื่องโปรแกรมภายนอกให้ยุ่งยาก ซึ่ง คุณสมบัติของ Bootloaderที่ทาง อีทีที ปรับปรุงเพิ่มเติมขึ้น มีคุณสมบัติการทำงานเป็นดังนี้

- สื่อสารกับโปรแกรมภายนอกด้วย Protocol แบบ STK500 (STK500V1)
- ใช้ความเร็ว Baudrate 19200 โดยใช้ความถี่ XTAL 16 MHz
- โปรแกรม Bootloaderมีขนาด 2KByte ทำงานที่ตำแหน่ง 0x3800-0x3FFF
- ใช้ LED ที่ต่อกับขา Digital-13(PB5) เป็นตัวแสดงสถานะในขณะที่ Bootloaderทำงาน
- โปรแกรมใน Bootloaderจะทำงานโดยอัตโนมัติทุกครั้งหลังการรีเซ็ต โดย MCU จะเริ่มต้นทำงาน

ใน Bootloaderนี้ก่อนเสมอ ซึ่งถ้าไม่มีการติดต่อสื่อสารจากโปรแกรม Arduinoภายในเวลาประมาณ 3 วินาที MCU ก็จะออกจากการทำงานใน Bootloaderเพื่อไปเริ่มต้นทำงานตามคำสั่งที่เป็นของผู้ใช้ โดยอัตโนมัติ โดยในขณะที่ Bootloaderทำงาน ตอนเริ่มต้นจะเห็น LED ที่ต่อไว้กับขาสัญญาณ Digital-13(PB5) กระพริบ 3 ครั้ง แล้วติดค้าง เพื่อรอการติดต่อสื่อสารจากโปรแกรมสำหรับสั่งให้ทำการ Upload Code ให้กับ MCU ซึ่งถ้าไม่มีการติดต่อสื่อสารจากโปรแกรมของArduinoภายในระยะเวลาประมาณ 3 วินาที โปรแกรมก็จะกระโดดไปทำงานในตำแหน่งเริ่มต้นที่เป็นส่วนของโปรแกรมที่ผู้ใช้เขียนขึ้นทันที นอกจากนี้แล้วผู้ใช้อังยังสามารถเปลี่ยนไปใช้โปรแกรมBootloaderรุ่นมาตรฐาน ตามแบบโปรแกรม ของ Arduinoได้อีกด้วย โดยใช้โปรแกรม Bootloaderที่ชื่อ “ATmegaBOOT_168_diecimila.hex” โดยไฟล์ดังกล่าวจะถูกบรรจุไว้ใน Drive และโพลเดอร์เดียวกันกับที่ผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduinoไว้

แล้วคือเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ กรุณาใช้คำสั่ง `C:\arduino0012\hardware\bootloaders\atmega168\ATmegaBOOT_168_diecimila.hex` ใช้

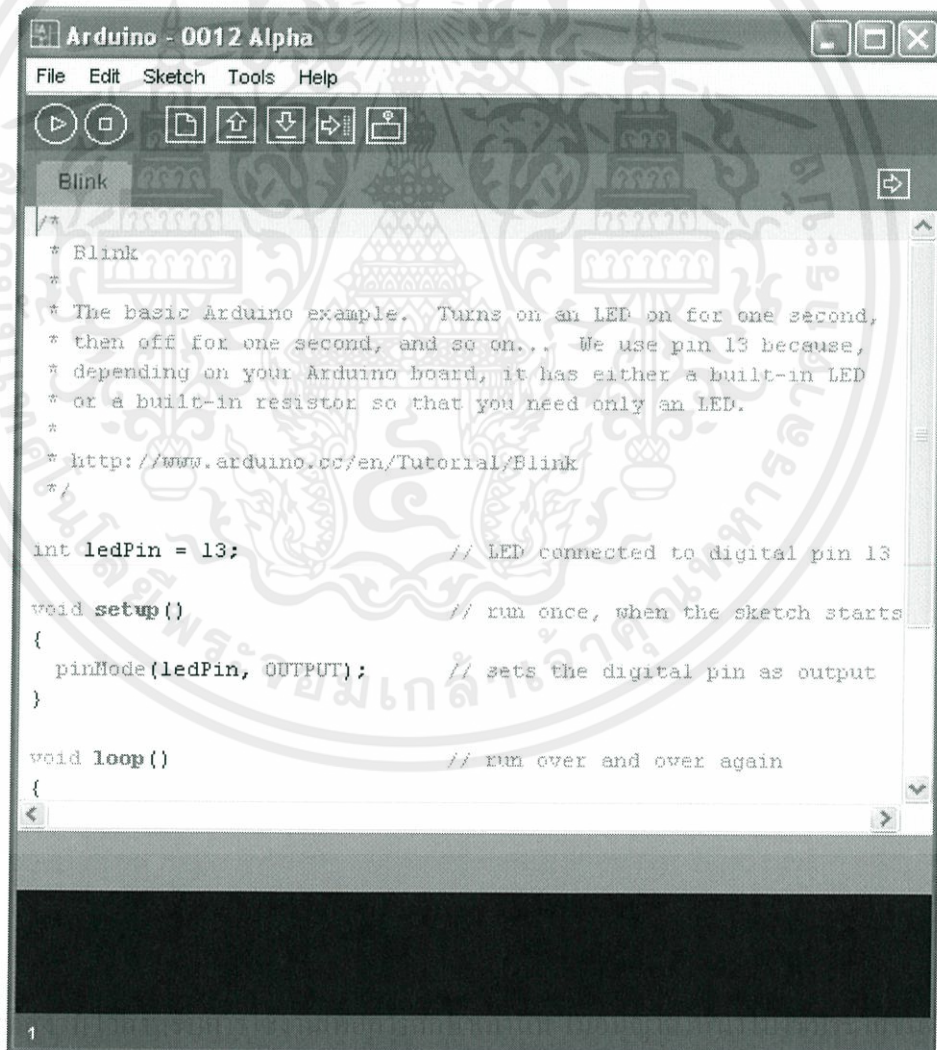
แต่อย่างไรก็ตามในการที่จะสามารถทำการติดตั้งโปรแกรม Bootloader Code ให้กับ MCU ได้นั้นผู้ใช้จำเป็นต้องมีเครื่องมือสำหรับทำหน้าที่ Program Code ให้กับ MCU อยู่ด้วย โดยใช้เครื่องโปรแกรมที่มีขั้วต่อตามมาตรฐานของ “AVRISP” ของ ATMEL แบบ IDE 10 PIN ได้ทันที

ค.3 การติดตั้งโปรแกรมArduino

หลังจากที่เราได้ทำการติดตั้ง USB Driver ให้กับบอร์ดเป็นที่เรียบร้อยแล้วก็เป็นอันเสร็จสิ้นขั้นตอนของการเตรียมการแล้วลำดับขั้นตอนต่อจากนี้เป็นต้นไปก็เป็นเรื่องของการใช้งานการเขียนโปรแกรมและการศึกษาเรียนรู้ต่างๆตามความต้องการแล้วแต่ก่อนอื่นเราจะต้องทำการติดตั้งโปรแกรมของArduinoเพื่อใช้เป็นโปรแกรมสำหรับศึกษาเรียนรู้ซึ่งมีลำดับขั้นตอนดังต่อไปนี้

1. ทำการ unzip ไฟล์ชื่อ “arduino-0012-win.zip” ไว้ในฮาร์ดดิสก์ซึ่งขอแนะนำให้ทำการ unzip ไว้ที่Root นอกสุดใน Drive C โดยหลังจากทำการ unzip เรียบร้อยแล้วจะได้โปรแกรมอยู่ที่ “c:\arduino-0012”

2. ทำการสั่ง Run โปรแกรม “arduino.exe” จะได้ผลดังรูป



```

Arduino - 0012 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()               // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()                // run over and over again
{

```

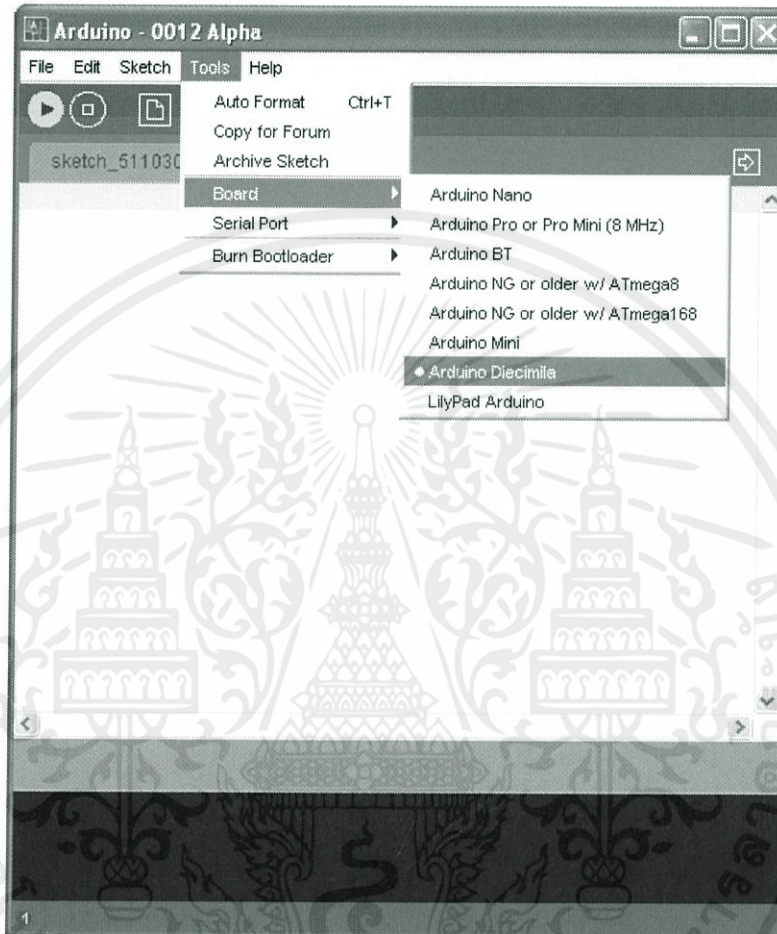
เอกสารนี้เป็นเอกสาร

ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

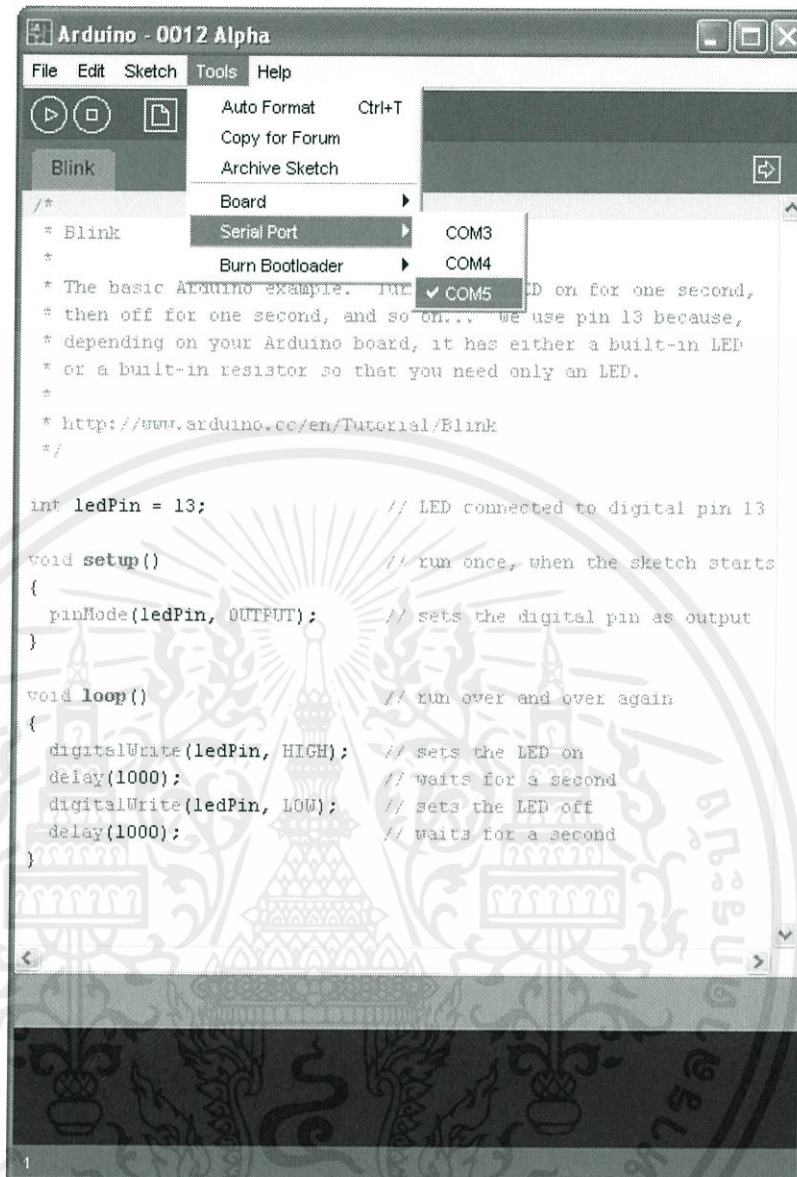
3. ในครั้งแรกของการเรียกใช้งานโปรแกรม ให้ทำการกำหนดระบบฮาร์ดแวร์ที่จะใช้งานกับโปรแกรมของ Arduino ให้เรียบร้อยเสียก่อน เนื่องจากในปัจจุบันนี้ มีการออกแบบวงจรและสร้างฮาร์ดแวร์บอร์ด

แบบต่างๆสำหรับนำมาใช้งานร่วมกับโปรแกรมพัฒนาของ Arduino ไว้มากมายหลายรุ่น โดยในกรณีของบอร์ด ET-EASY168 STAMP ให้ทำการเลือกกำหนดชื่อบอร์ดเป็น “Decimila” โดยคลิกเมาส์ที่ “Tools → Board → ArduinoDecimila” ดังรูป



4. เลือกกำหนดหมายเลขพอร์ต สำหรับติดต่อสื่อสารกับบอร์ด ให้ตรงกับหมายเลข Comport ที่ได้ทำการติดตั้ง Driver ของ USB ไวเินตอนแรก เช่น ถ้าตอนติดตั้ง Driver ของ USB แล้วได้หมายเลข Comport เป็น COM5 ให้คลิกเมาส์ที่ Tools → Serial Port → COM5 ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5. ทดสอบเขียนโปรแกรม โดยคลิกเมาส์ที่ File → New แล้วพิมพ์โปรแกรมทดสอบ หรืออาจใช้การ
 สั่งเปิดไฟล์ตัวอย่างที่สร้างไว้แล้วขึ้นมาแทนก็ได้ โดยในที่นี้ขอแนะนำให้ทดสอบด้วยโปรแกรมไฟกระพริบ
 โดยให้เลือก "File → sketchbook → Examples → Digital → Blink" ซึ่งจะได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

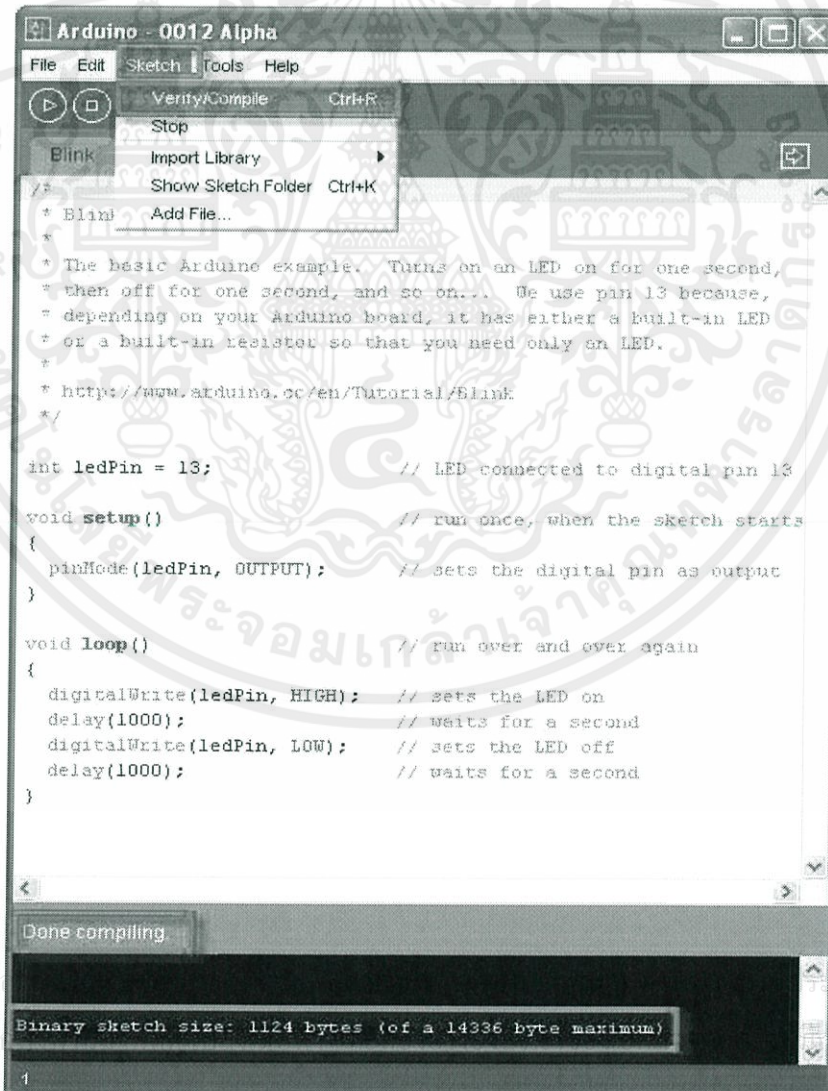
int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}

```

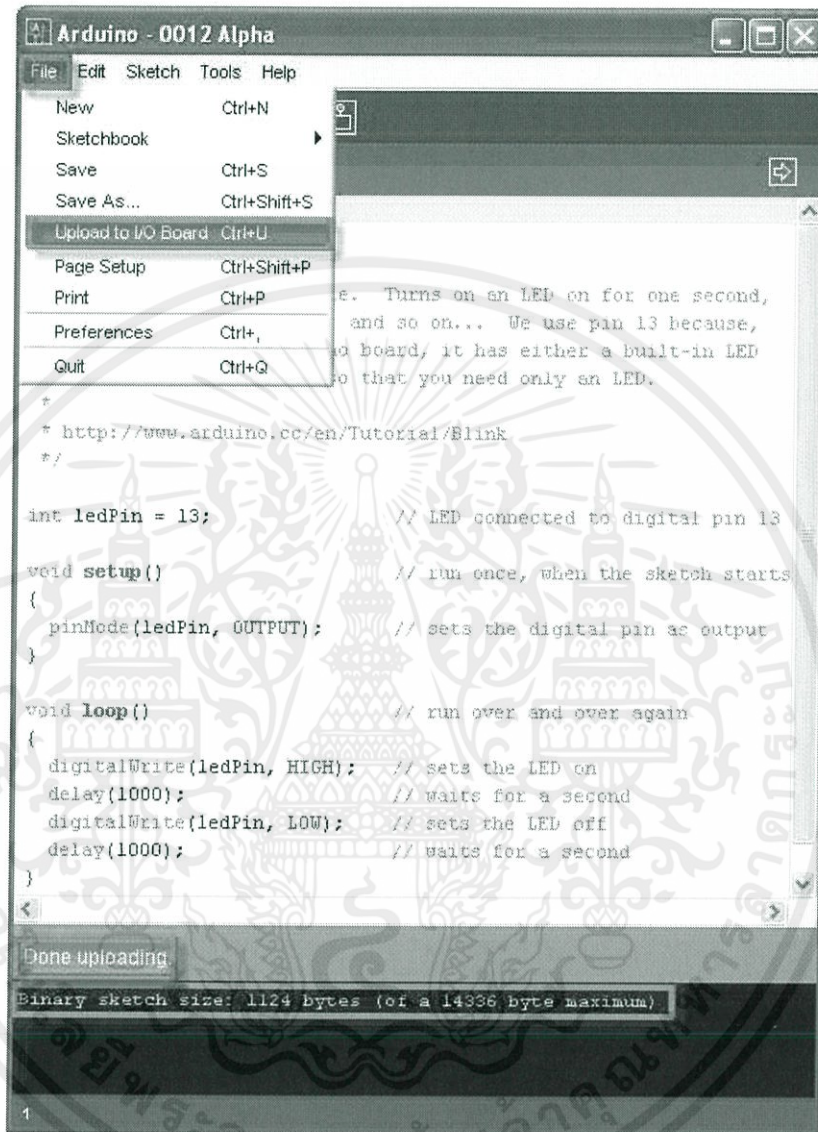
6. สั่งแปลโปรแกรมโดยคลิกเมาส์ที่ “Sketch → Verify/Compile” ดังตัวอย่าง



เอกสารนี้เป็นเอกสารที่
ไม่ว่ากรณีใดๆทั้งสิ้น

โยชน์ด้านการค้า
รณาไปใช้

7. สั่ง Download Code ให้กับบอร์ดโดยคลิกเมาส์เลือกที่ “File → Upload to I/O Board” แล้วรอสักครู่จนโปรแกรมทำงานเสร็จซึ่งควรได้ผลดังรูป



8. หลังจากที่ทำกร Upload Code ให้กับบอร์ดเป็นที่เรียบร้อยแล้ว บอร์ดก็จะเริ่มต้นทำงานตามคำสั่งที่เขียนไว้ในโปรแกรมทันที โดยจะสังเกตเห็น LED กระพริบ ติด และ ดับ สลับกันไปมา ด้วยความเร็วประมาณ 1 วินาที ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

โปสเตอร์แบบจำลองระบบคลังสินค้า

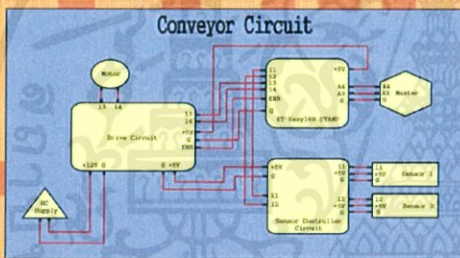
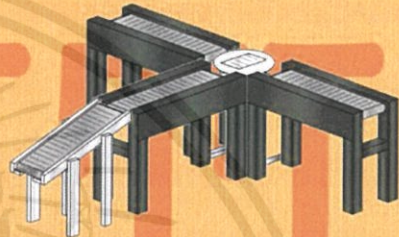


สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
King Mongkut's Institute of Technology Ladkrabang
Control Engineering

Warehouse System Models

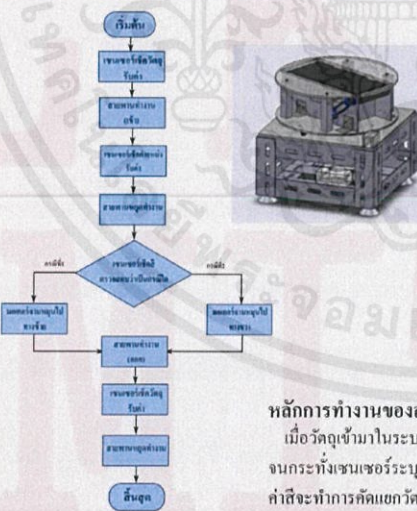
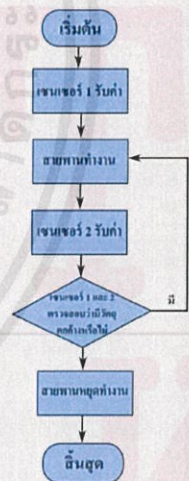
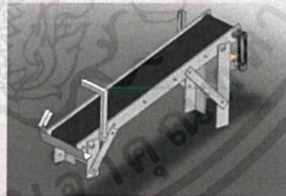
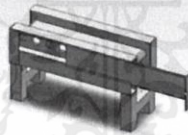
วัตถุประสงค์

1. เพื่อนำความรู้ทางวิศวกรรมระบบควบคุมมาใช้ในการออกแบบพัฒนาและปรับปรุงระบบการลำเลียง กัดแยกสินค้าให้เกิดประสิทธิภาพสูงสุด
2. เพื่อสร้างโปรแกรมสำเร็จรูปที่สามารถควบคุมการทำงานของกระบวนการต่าง ๆ ในระบบคลังสินค้าผ่านทางหน้าจอคอมพิวเตอร์



หลักการทำงานของสายพาน

เมื่อวัตถุเข้ามาในระบบ เซนเซอร์ 1 จะส่งค่าไปที่มอเตอร์สายพานทำงานไปจนกระทั่งเมื่อวัตถุออกจากระบบแล้ว เซนเซอร์ 2 จะส่งค่าให้มอเตอร์หน่วงเวลา ก่อนจะหยุด แต่หากว่ายังมีวัตถุค้างอยู่ในระบบ มอเตอร์จะยังคงหมุนต่อไปจนกว่าค่าจากเซนเซอร์ 1 และ 2 จะเท่ากัน



หลักการทำงานของส่วนคัดแยก

เมื่อวัตถุเข้ามาในระบบ เซนเซอร์ที่เป็นตัวเช็คของเข้า จะส่งค่าให้มอเตอร์สายพานหมุนเข้าจนกระทั่งเซนเซอร์ระบุตำแหน่งตรวจจับวัตถุได้ มอเตอร์สายพานจะหยุด และเซนเซอร์ตรวจค่าดีจะทำการคัดแยกวัตถุก่อนส่งค่าให้มอเตอร์จานหมุน หมุนไปทางด้านที่ต้องการ และมอเตอร์สายพานหมุนออก เพื่อส่งวัตถุไปยังส่วนต่อไป

การประยุกต์ใช้งาน

เนื่องจากปัจจุบันระบบลำเลียงสินค้าในโรงงานส่วนมากจะใช้ PLC เป็นตัวควบคุมระบบซึ่ง PLC มีราคาสูง ทำให้ต้นทุนสูง จึงเป็นไปได้ในอนาคตที่จะมีการพัฒนาโปรแกรมสำเร็จรูปนี้ และ นำเข้าไปประยุกต์ใช้งานจริงในโรงงานอุตสาหกรรม เพื่อลดต้นทุนในการผลิตให้น้อยลง

เอกสารนี้เป็นเอกสาร
ไม่ว่ากรณีใดๆ