สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำลียง

COMPUTER INTEGRATED MANUFACTURING: CONVEYOR SYSTEM SECTION

T119377

นายนภดล แซ่เตีย

นางสาวนันทพัทธ์ โชคนันทเกียรติ

นายประยุทธ์ ไตรพารา

นายปรีชา ทุมวงศ์

เลขหมู่.....
เลขทะเบียน 119377
วัน,เดือน,ปี - 7 ส.ค. 2554

b. 10266481
i.....

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTER INTEGRATED MANUFACTURING:

CONVEYOR SYSTEM SECTION

MR. NOPPADOL SAETIA

MISS NANTHAPHAT CHOKNATHAKIET

MR. PRAYUTE THRAIPARA

MR. PREECHA THUMWONG

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ACADEMIC YEAR 2010

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำเลียง

COMPUTER INTEGRATED MANUFACTURING: CONVEYOR SYSTEM SECTION

ผู้จัดทำ นายนภดล แซ่เตีย 50010244

นางสาวนันทพัทธ์ โชคนันทเกียรติ 50010801

นายประยุทธ์ ไตรพารา 50010911

นายปรีชา ทุมวงศ์ 50010930

..... อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร. วันชัย ริ้วรุจา)

..... อาจารย์ที่ปรึกษา

(ดร. วรรณดี เพชรมณีล้ำค่า)

..... อาจารย์ที่ปรึกษา

(อาจารย์ เทพจิตร์ เชยโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำเลียง

โดย

นายนภดล	แซ่เตีย	50010244
นางสาวนันทพัทธ์	โชคนันทเกียรติ	50010801
นายประยุทธ์	ไตรพารา	50010911
นายปรีชา	ทุมวงศ์	50010930

อาจารย์ที่ปรึกษา

รศ.ดร. วันชัย ริ้วรุจา

ดร. วรรณดี เพชรมณีล้ำค่า

อ.เทพจิตร์ เชยโภคา

ปีการศึกษา 2553

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอทฤษฎีและการจำลองกระบวนการผลิตที่ส่วนใหญ่ใช้ในอุตสาหกรรมการผลิต โดยมีโครงสร้างของกระบวนการที่ทำการจำลองประกอบด้วย ระบบการสังเกตการณ์และควบคุมทางมอนิเตอร์แสดงผล ระบบการติดต่อสื่อสาร ระบบลำเลียง และระบบแขนกล ซึ่งจุดมุ่งหมายของโครงงานนี้เพื่อทำการศึกษากระบวนการทำงานและการติดต่อสื่อสารกันระหว่างระบบต่างๆ เพื่อให้สามารถทำงานได้อย่างสอดคล้องกันอย่างเป็นระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTER INTEGRATED MANUFACTURING :

CONVEYOR SYSTEM SECTION

By

Mr. Noppadon Saetea 50010744

Miss. Nanthapat Chotnantakiat 50010930

Mr. Prayute Tripara 50010911

Mr. Preecha Thumwong 50010930

Advisors

Assoc.Prof. Dr. Vanchai Riewruja

Dr. Wandee Petchmaneelumka

Mr. Thepjit Cheypoca

Academic Year 2010

ABSTRACT

This thesis presents theory and simulation of process manufacturing industry. The structure of process simulation includes observation and control systems through monitor display, communication system, conveyor system and robot arm. The aim of this project is to educate processing and interfacing of all system for systematic operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก รศ.ดร. วันชัย ริ้วรุจา ดร.วรรณดี เพชรมณีล้ำค่า และอาจารย์เทพจิตร์ เชยโภคา ที่ได้กรุณาให้ความช่วยเหลือในการแก้ไขปัญหาต่างๆและคำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์เครื่องมือที่จำเป็น และความช่วยเหลืออื่นๆที่เป็นประโยชน์ต่อโครงงานนี้ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ นายอมรเทพ กาพย์แก้ว ที่ให้คำแนะนำต่างๆ เอื้อเฟื้ออุปกรณ์ที่จำเป็น และช่วยเหลืออื่นๆต่างๆ

ขอบคุณเพื่อนๆ ทุกคนที่กระตือรือร้นร่วมมือกันทำจนโครงงานนี้สำเร็จสมบูรณ์ขึ้นมาได้ รวมกันแก้ไขปัญหาต่างๆให้ผ่านพ้นไปได้ กำลังใจของทุกคนที่ให้เป็นแรงผลักดันซึ่งกันและกัน ช่วยสนับสนุนอุปกรณ์ที่ขาดเหลือ รวมทั้งคอยถามไถ่ความคืบหน้าของโครงงานอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงงานนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นายนภดล	แซ่เตีย
นางสาวนันทพัทธ์	โชคนันทเกียรติ
นายประยุทธ์	ไตรพารา
นายปรีชา	ทุมวงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IIII
สารบัญรูปภาพ	VII
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ของโครงงาน	2
1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน	2
1.4 รายละเอียดของโครงงาน	4
บทที่ 2 ภาพรวมของระบบจำลองกระบวนผลิตอัตโนมัติ	4
2.1 ส่วนลำเลียง	6
2.1.1 รถลำเลียงสินค้า	6
2.1.2 สถานี	6
2.2 แขนกล	7
2.3 หุ่นยนต์แมลง	7
2.4 หุ่นยนต์ 2 ขา	8
2.5 คอนโทรลเลอร์หลักที่กระจายคำสั่ง	8
2.6 โปรแกรมควบคุมการทำงาน	9
2.6.1 โหมดควบคุมด้วยมือ	10
2.6.2 โหมดควบคุมอัตโนมัติ	10
2.7 การติดต่อสื่อสารกันของระบบ	10
2.7.1 การติดต่อสื่อสารกันระหว่างรถลำเลียง กับ สถานี	10
2.7.2 การติดต่อสื่อสารกันระหว่าง คอนโทรลเลอร์หลัก กับ สถานี	11
2.7.3 การติดต่อสื่อสารกันระหว่างคอนโทรลเลอร์หลัก กับ หน้าจอแสดงผล	11
บทที่ 3 ทฤษฎีและความรู้ที่เกี่ยวข้อง	12
3.1 ทฤษฎีและความรู้ที่เกี่ยวข้อง	12
3.1.1 มอเตอร์กระแสตรง	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.1.1.1 หลักการทำงานของมอเตอร์กระแสตรง	12
3.1.1.2 การขับและกลับทิศทางของมอเตอร์กระแสตรง	13
3.1.1.3 คุณสมบัติของมอเตอร์กระแสตรง	14
3.1.1.4 โมเดลคณิตศาสตร์ของดีซีมอเตอร์	15
3.1.1.5 โมเดลอิเล็กโทรแมคคานิคอล	16
3.1.2 การสื่อสารแบบอนุกรม (serial Port)	22
3.1.2.1 การส่งข้อมูลแบบอะซิงโครนัส	23
3.1.2.2 การส่งข้อมูลแบบซิงโครนัส	24
3.1.2.3 การสื่อสารผ่าน Serial port	24
3.1.2.4 RS-232	25
3.1.3 พอร์ตขนาน	26
3.1.4 ไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 (ARDIUNO MEGA)	27
3.1.4.1 คุณสมบัติ ของ ET-EASY MEGA1280 (ARDIUNO MEGA)	27
3.1.4.2 หลักการทำงานของ Main controller	28
3.1.4.3 สาเหตุการเลือกใช้ Easy Mega1280	31
3.1.5 PIC16F690	31
3.1.6 TSOP34836	32
3.1.6.1 คุณสมบัติ	32
3.1.6.2 สาเหตุการเลือกใช้ TSOP34836	33
3.1.7 อินฟราเรดเซนเซอร์	33
3.1.8 การทำงานของวงจรเซ็นเซอร์แสงที่ใช้ในการตรวจจับเส้น	34
3.2 การออกแบบระบบจำลองกระบวนการผลิตอัตโนมัติ	35
3.2.1 กล้องดิจิตอลวีดีโอ	36
3.2.2 คอมพิวเตอร์	36
3.2.3 โครงสร้างทางกายภาพของระบบ	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.2.3.1 ส่วนรถลำเลียง	37
3.2.3.2 ส่วนสถานี	37
3.2.4 วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง	38
บทที่ 4 การทดลอง	41
4.1 การทดลองโปรแกรม และ ผลที่ได้จากการทำงาน	41
บทที่ 5 บทสรุป	45
5.1 สรุปผลของโครงงาน	45
5.2 ปัญหาและอุปสรรค และ แนวทางการแก้ไข	45
5.3 แนวทางการพัฒนา	45
ภาคผนวก ก โปรแกรม Visual Basic 2008	47
ภาคผนวก ข โปรแกรมการควบคุมรถลำเลียง	57
ภาคผนวก ค โปรแกรมสถานี (Station)	63
ภาคผนวก ง Main Controller	68
ภาคผนวก จ โปรสเตอร์ระบบโดยรวม และระบบลำเลียง	74
เอกสารอ้างอิง	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้า
1.1 องค์ประกอบของระบบจำลอง	2
2.1 องค์ประกอบโดยรวมของระบบจำลองกระบวนการผลิต	4
2.2 แผนผังการทำการของระบบโดยรวม	5
2.3 รถลำเลียง (Vehicle)	6
2.4 สถานี (Station)	6
2.5 แขนกล (Robot arm)	7
2.6 หุ่นยนต์แมลง	7
2.7 หุ่นยนต์ 2 ขา	8
2.8 กระบวนการทำงานของคอนโทรลเลอร์	9
2.9 หน้าตาโปรแกรมควบคุมการทำงาน	9
2.10 การติดต่อสื่อสารกันระหว่างรถลำเลียง กับ สถานี	10
2.11 การติดต่อสื่อสารกันระหว่าง คอนโทรลเลอร์หลัก กับสถานี	11
2.12 การติดต่อสื่อสารกันระหว่างคอนโทรลเลอร์หลัก กับ หน้าจอแสดงผล	11
3.1 โครงสร้างทั่วไปของมอเตอร์กระแสตรง	12
3.2 การกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์	13
3.3 การใช้ทรานซิสเตอร์เพื่อขับรีเลย์ให้ทำงาน	13
3.4 การใช้ทรานซิสเตอร์เป็นวงจรขับและกำหนดทิศทางของมอเตอร์กระแสตรง	14
3.5 วงจรภายในของมอเตอร์กระแสตรง	14
3.6 อินพุตและเอาต์พุตของโมเดลทางคณิตศาสตร์ของมอเตอร์	15
3.7 โมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น	16
3.8 แรงบิดต่างๆที่เกิดขึ้นต่อโหลดของมอเตอร์	17
3.9 บล็อกไดอะแกรมของดีซีมอเตอร์โมเดล	19
3.10 การรับ-ส่งข้อมูลแบบอนุกรม	22
3.11 การรับ-ส่งข้อมูลแบบอะซิงโครนัส	23
3.12 การรับ-ส่งข้อมูลแบบซิงโครนัส	24
3.13 การสื่อสารข้อมูลผ่านพอร์ตอนุกรม	25
3.14 การสื่อสารข้อมูลผ่านพอร์ตขนาน	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่ **หน้า**

3.15 บอร์ด ET-EASY MEGA1280 (ARDIUNO MEGA) 27

3.16 แผนภาพแสดงการทำงานของส่วน Main Controller 29

3.17 วงจรภายในตัวบอร์ด Et-easy mega128 30

3.18 สถาปัตยกรรมภายในของ AVR 31

3.19 ลักษณะของ PIC16F690 31

3.20 ลักษณะของ TSOP34836 32

3.21 อินฟราเรดเซนเซอร์ 33

3.22 การสะท้อนของแสงอินฟราเรดเมื่อกระทบกับวัตถุ 34

3.23 แผนผังรวมของระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์ 35

3.24 ลักษณะของรถลำเลียง 37

3.25 ลักษณะของสถานี 37

3.26 ลายวงจรของส่วนรถลำเลียง 38

3.27 วงจรภายในส่วนของรถลำเลียง 39

3.28 วงจรภายในของส่วนสถานี 40

3.29 ภาพวงจรของส่วนสถานี 40

4.1 การเลือก com port และการ เชื่อมต่อ 41

4.2 การจับภาพจากกล้อง Webcam 42

4.3 การเลือกโหมดการทำงาน 42

4.4 รายละเอียดในโหมดควบคุมด้วยมือ 43

4.5 การเลือกโหมดควบคุมอัติโนมัติ 43

4.6 ส่วนแสดงผลสถานีการทำงาน 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

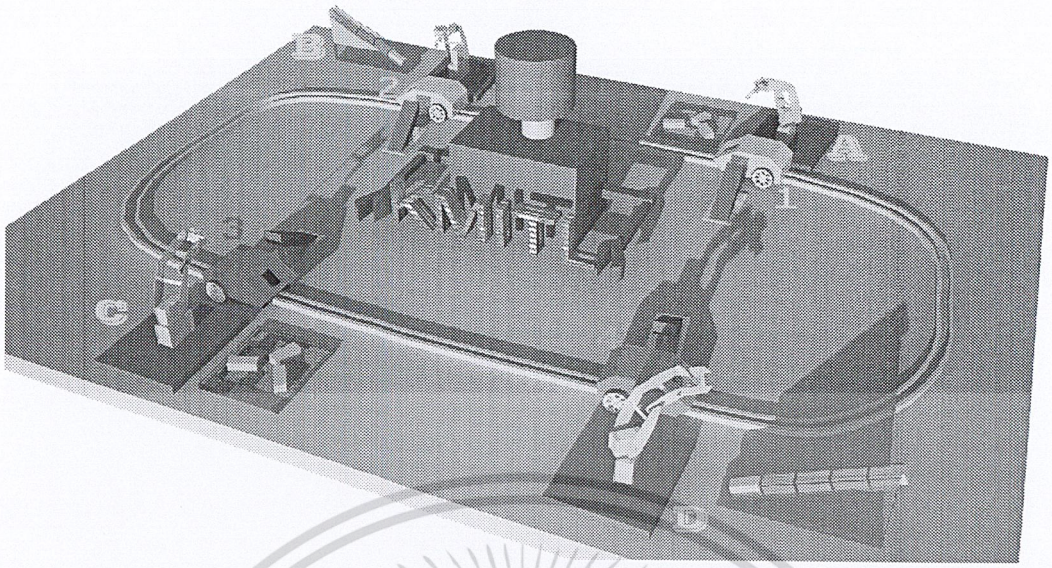
บทนำ

1.1 กล่าวนำ

ในปัจจุบัน วงการอุตสาหกรรมในหลายๆประเทศเกิดการแข่งขันกันอย่างมาก เพื่อให้สินค้าของตนเองสามารถผลิตได้อย่างรวดเร็วและมีประสิทธิภาพ สามารถส่งออกสู่ท้องตลาดทั้งในและนอกประเทศได้เป็นจำนวนมาก ส่งผลให้เทคโนโลยีที่ต่างๆที่เกี่ยวข้องกับวงการอุตสาหกรรมมีการพัฒนาขึ้นอย่างรวดเร็ว จากการผลิตด้วยแรงงานคนงาน มีการปรับปรุงพัฒนาขึ้นมาและเริ่มมีการใช้ระบบการควบคุมแบบอัตโนมัติ และในปัจจุบันมีการใช้ระบบคอมพิวเตอร์เข้ามาใช้ในการควบคุมกระบวนการผลิตแขนงต่างๆในโรงงานอุตสาหกรรมการผลิตมากขึ้นจนอาจกล่าวได้ว่าอุตสาหกรรมบางประเภทจำเป็นต้องใช้คอมพิวเตอร์ ไม่สามารถดำเนินการอุตสาหกรรมโดยไม่มีคอมพิวเตอร์ได้ ตัวอย่างเช่นอุตสาหกรรมยานยนต์ ซึ่งมีการแข่งขันกันสูงที่สุด จำเป็นต้องใช้คอมพิวเตอร์ในแทบทุกขั้นตอนการผลิต เพื่อให้การผลิตสินค้าของตนเองมีกระบวนการผลิตที่รวดเร็ว มีประสิทธิภาพในขั้นตอนการผลิตส่งผลให้เกิดความคล่องตัวของกระบวนการ และช่วยลดต้นทุนการผลิต จากการเกิดสินค้าที่เสียหายในขั้นตอนการผลิตที่ผิดพลาด เนื่องจากกระบวนการผลิตอัตโนมัติจากระบบคอมพิวเตอร์นั้น สินค้าที่ผลิตออกมาจะเป็นแบบเดียวกันเกือบจะทุกชิ้น โดยการนำระบบคอมพิวเตอร์มาช่วยในงานอุตสาหกรรมโดยให้มีการทำงานร่วมกัน เพื่อเพิ่มประสิทธิภาพของการผลิตงานอุตสาหกรรม ซึ่งเป็นที่รู้จักกันในชื่อ การผลิตผสมผสานด้วยคอมพิวเตอร์ (Computer Integrated Manufacturing, CIM)

ในโครงงานนี้จึงการศึกษาและจำลองระบบการทำงานของกระบวนการผลิตแบบอัตโนมัติโดยมีการสั่งการและตรวจสอบด้วยคอมพิวเตอร์ เป็นระบบที่มีใช้กันส่วนใหญ่ในวงการอุตสาหกรรม ซึ่งในระบบจะประกอบด้วย แขนกล รถลำเลียงสินค้า สถานีที่ทำการติดต่อสื่อสารระหว่างรถ-ลำเลียงสินค้ากับตัวควบคุมการส่งสัญญาณ และโปรแกรมที่ทำการสั่งการและตรวจสอบ โดยในโครงงานนี้ต้องการที่จะตรวจสอบและสั่งการระบบทั้งหมดให้สามารถทำงานผ่านทางคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 องค์ประกอบของระบบจำลอง

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. ทำการศึกษาและทดลองการใช้งานอุปกรณ์ต่างๆในระบบควบคุม
2. ทำการศึกษาและจำลองระบบ (CIM) ซึ่งประกอบด้วย แขนกลทำหน้าที่จับสิ่งของ หุ่นยนต์ลำเลียงสินค้า สถานีที่ทำหน้าที่ติดต่อสื่อระหว่างหุ่นยนต์ลำเลียงสินค้ากับตัวควบคุมหลัก และคอมพิวเตอร์ที่ทำหน้าที่สั่งการและตรวจสอบการทำงาน ซึ่งในโครงงานนี้ใช้กล้องเว็บแคม (Webcam) โดยโครงงานนี้ใช้คอมพิวเตอร์และโปรแกรมวิชวลเบสิก (Visual Basic)
3. ทำการศึกษาการฟังก์ชันการทำงานไมโครคอนโทรลเลอร์ (Micro Controller) ตระกูลต่างๆ
4. ศึกษาการติดต่อสื่อสารข้อมูลแบบอนุกรม และแบบขนาน

1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน

การศึกษาและจำลองระบบการผลิตผสมผสานด้วยคอมพิวเตอร์นั้น จำเป็นเป็นต้องทำการศึกษาองค์ประกอบของระบบก่อนว่า โดยทั่วไปของระบบๆหนึ่งนั้นประกอบด้วยอะไรบ้าง หลังจากที่ได้ทำการศึกษา ค้นคว้าดูแล้วพบว่า โดยส่วนใหญ่นั้นในกระบวนการผลิตแห่งหนึ่งๆนั้นจะมีการทำงานของหลายๆส่วนเข้าด้วยกัน ซึ่งจะประกอบด้วย ส่วนที่ทำหน้าที่หยิบจับชิ้นงาน ส่วนที่ทำหน้าที่ขนส่งชิ้นงานไปยังหน่วยการทำงานต่อไป ส่วนที่ทำการกระจายคำสั่งและเก็บข้อมูลจากแต่ละหน่วยการทำงานไว้ มอนิเตอร์ทำหน้าที่แสดงผลการทำงานและทำหน้าที่สั่งการจากห้องควบคุม โดยที่แต่ละส่วนสามารถติดต่อสื่อสารกันและทำงานได้อย่างสอดคล้องกัน โดยผ่านการสั่งงานจากมอนิเตอร์และจากนั้นก็ให้ส่วนกระจายคำสั่งและเก็บข้อมูลทำหน้าที่กระจายคำสั่งที่ได้รับมาจากมอนิเตอร์กระจายคำสั่งไปให้แต่ละส่วนที่ทำหน้าที่รับผิดชอบการทำงานนั้นๆต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวมาแล้วข้างต้นโครงงานนี้จึงมีการแบ่งกลุ่มกันไปทำการศึกษาและทำการจำลองการทำงานของแต่ละส่วน โดยในปริญญานิพนธ์ฉบับนี้ได้ทำการศึกษาและจำลองในส่วนของแสดงผล กับส่วนเก็บข้อมูลและขนส่งชิ้นงาน โดยในส่วนของแสดงผลนั้น ให้มีการแสดงผลที่คอมพิวเตอร์ ซึ่งควรที่จะใช้งานได้อย่างสะดวกแต่มีฟังก์ชันการทำงานได้อย่างหลากหลาย เพื่อที่จะให้ผู้ที่ทำการควบคุมระบบอยู่นั้นไม่เกิดความสับสนและง่ายต่อการใช้งาน ใช้โปรแกรมวิชวลเบสิก มีภาษาที่เรียบง่ายสามารถประยุกต์ได้ง่ายและมีเครื่องมือคำสั่งต่างๆ (Tool) มาให้สำเร็จรูป มีพัฒนาเป็นแอพพลิเคชันโดยออกแบบให้เป็นมอนิเตอร์เป็นปุ่มกดเพื่อความงาน เมื่อเรียกใช้งานจึงเพียงแค่คลิกปุ่มคำสั่งที่ต้องการเท่านั้น

ในส่วนเก็บข้อมูลและขนส่งชิ้นงานจะประกอบด้วย รถลำเลียง สถานีที่ทำการติดต่อสื่อสารระหว่างรถลำเลียงกับคอนโทรลเลอร์ที่ทำหน้าที่เก็บข้อมูลและกระจายคำสั่ง คอนโทรลเลอร์ที่ทำหน้าที่เก็บขอมูลและกระจายคำสั่ง

การทำงานของทั้งสามนี้คือเมื่อคอนโทรลเลอร์ได้รับคำสั่งจากมอนิเตอร์แล้วจะทำการตรวจสอบเงื่อนไข ถ้าคำสั่งให้ทำงาน คอนโทรลเลอร์จะทำการสั่งการไปที่สถานีหรือแขนกลตามเงื่อนไข และเมื่อมอนิเตอร์ต้องการตรวจสอบข้อมูลแล้วคอนโทรลเลอร์จะทำการส่งข้อมูลที่เก็บไว้ไปแสดงผลที่มอนิเตอร์ ส่วนสถานีเมื่อได้รับคำสั่งมาก็จะทำการติดต่อสื่อสารผ่านทางสัญญาณอินฟราเรด (Infrared) ไปยังรถลำเลียง แล้วส่งค่าข้อมูลกลับไปยังคอนโทรลเลอร์เพื่อให้เก็บข้อมูลไว้ ในส่วนแขนกลเมื่อได้รับคำสั่งมาก็จะทำการหยิบชิ้นงาน โดยศึกษาการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์ภายนอกรวมถึงวงจรที่ใช้ในการควบคุมซึ่งรายละเอียดจะกล่าวในบทต่อไป

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวนำถึงวัตถุประสงค์ หลักการใหม่ ขั้นตอนการศึกษา และการจัดทำโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์ของแต่ละบท

บทที่ 2 ภาพรวมของระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์

บทที่ 3 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องในการประมวลผลภาพ การใช้งานพอร์ตขนาน การใช้งานพอร์ตอนุกรม วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้องและหลักการออกแบบ นำเสนอการประกอบโครงสร้างของระบบ รวมถึงแนวคิดในการออกแบบระบบจำลอง

บทที่ 4 การทดลอง เป็นส่วนการทดสอบองค์ประกอบต่างๆ ในระบบ ตลอดจนการทดลองระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์

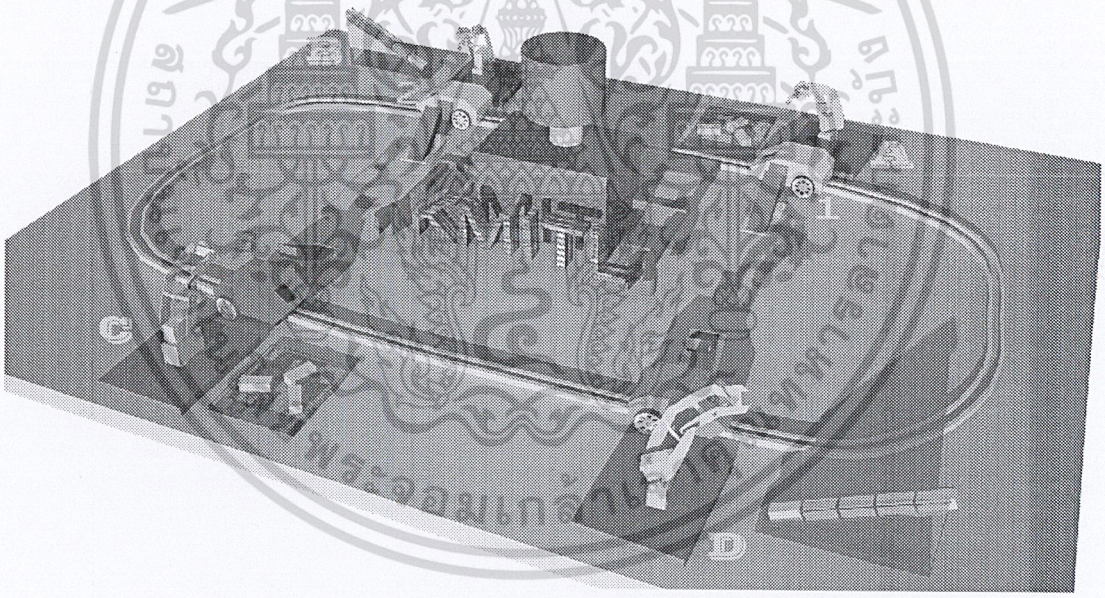
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงงานนี้ต่อไป

บทที่ 2

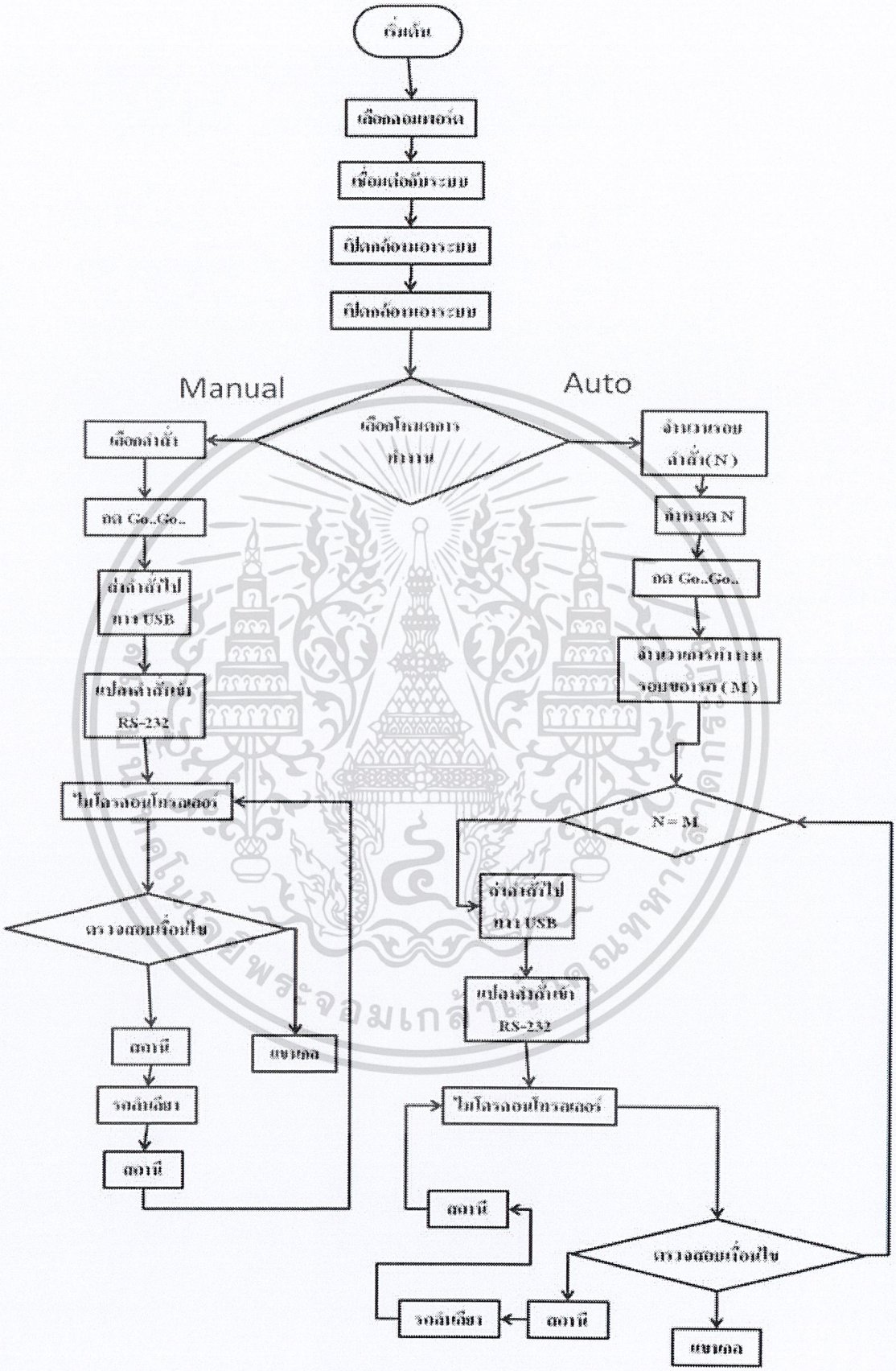
ภาพรวมของระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์

จากที่ได้ทำการศึกษาระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์มาแล้วนั้น จึงได้มีออกแบบและจำลองระบบโดยให้กระบวนการผลิตนี้ ประกอบด้วย รถลำเลียงสินค้าที่ติดต่อสื่อสารกับสถานี ซึ่งสถานีที่รับคำสั่งมาจากตัวคอนโทรลเลอร์ที่ทำหน้าที่กระจายคำสั่งที่มอนิเตอร์ส่งผ่านมาให้ โดยรถลำเลียงสินค้าจะวิ่งบนรางไฟฟ้า ทำหน้าที่ในการลำเลียงสินค้าจากหน่วยการทำงานหนึ่งไปยังอีกหน่วยการทำงานหนึ่ง และเมื่อแขนกลได้รับคำสั่งจากตัวคอนโทรลเลอร์จะทำการหยิบจับสินค้า เพื่อนำสินค้าเข้า-ออกระหว่างรถลำเลียงกับหน่วยการทำงาน ซึ่งจะมีหุ่นยนต์แมลงมารับสินค้าจากแขนกลไปส่งต่อให้กับหุ่นยนต์ 2 ขาเพื่อไปเก็บในคลังสินค้าต่อไปดังรูปที่ 2.1 โดยให้สามารถสั่งการและตรวจสอบการทำงานจากทางมอนิเตอร์ได้



รูปที่ 2.1 องค์ประกอบโดยรวมของระบบจำลองกระบวนการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



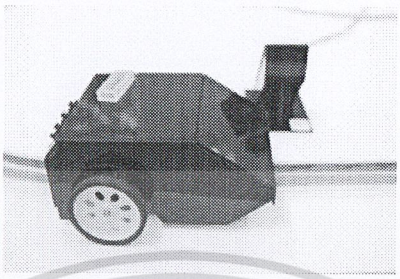
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 แผนผังการทำการของระบบโดยรวม

2.1 ส่วนลำเลียง

แบ่งออกเป็น 2 ส่วน คือ

2.1.1 รถลำเลียงสินค้า (Vehicle)



รูปที่ 2.3 รถลำเลียง (Vehicle)

หลักการทำงานของส่วนรถลำเลียงสินค้านั้น เราใช้หลักการการส่งข้อมูลผ่านทางสัญญาณอินฟราเรดด้วยความถี่ 40 KHz โดยการทำงานจะเริ่มเมื่อตัวรถนั้นเคลื่อนที่เข้ามาจอดที่สถานี จากนั้น วงจร รับ-ส่ง จะเริ่มทำงาน โดยสิ่งที่ส่งไปจะเป็นหมายเลขประจำของตัวรถ ซึ่งส่วนที่ทำหน้าที่เป็นตัวรับสัญญาณที่ส่งไปคือ ส่วนสถานีนั่นเอง

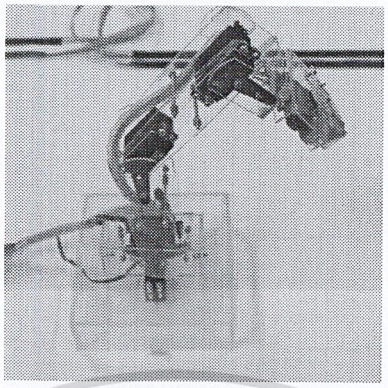
2.1.2 สถานี (Station)

รูปที่ 2.4 สถานี (Station)

หลักการทำงานของส่วนสถานีนั้น เราใช้หลักการการส่งข้อมูลผ่านทางสัญญาณอินฟราเรดเช่นเดียวกับหลักการของส่วนรถลำเลียง โดยการทำงานจะเริ่มเมื่อ รถเคลื่อนที่เข้ามาจอด และสถานีได้รับสัญญาณอินฟราเรดที่เป็นหมายเลขของรถแล้ว จากนั้นสถานีจะทำการส่งข้อมูลหมายเลขของรถที่ได้รับมานั้นไปยังส่วนตัวคอนโทรลเลอร์เพื่อทำการเก็บรวบรวมและกระจายคำสั่งมีที่รับมาจากมอนิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 แขนกล (Robot arm)

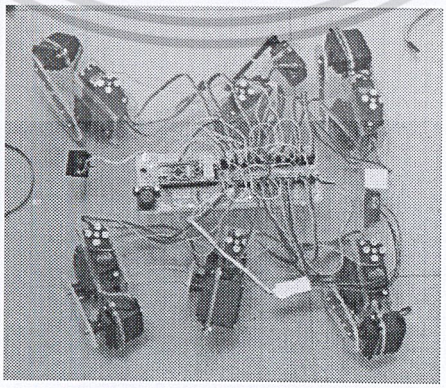


รูปที่ 2.5 แขนกล (Robot arm)

ส่วนประกอบของแขนกล จะแบ่งออกเป็น 4 ส่วนหลักๆ คือ

1. ส่วนควบคุม ทำหน้าที่ควบคุมการทำงานของแขนกลทั้งระบบ ซึ่งในส่วนนี้จะรับค่าจากปุ่ม
2. ส่วนประมวลผล ทำหน้าที่ในการประมวลผลคำสั่งที่ได้จากส่วนควบคุมจากSTAMP 16 แล้วทำการประมวลผลค่าที่ได้มา จากนั้นทำการสั่งการไปยังเซอร์โวมอเตอร์ให้ทำงาน
3. ส่วนเซอร์โวมอเตอร์ เป็นส่วนที่ขับเคลื่อนตัวแขนกลให้เคลื่อนไหวไปยังตำแหน่งที่ต้องการ ซึ่งจะถูกติดตั้งยังข้อต่อของตัวแขนกล
4. ส่วนที่ประกอบเป็นแขนกล จะแบ่งออกเป็น 3 ส่วน คือ คลิปเปอร์ แขน และฐาน ทุกชิ้นส่วนทำจากแผ่นอะคิลิค ซึ่งออกแบบจากโปรแกรม Solid works แล้วกัดชิ้นงานด้วยเครื่อง CNC และนำมาประกอบเข้ากับเซอร์โวมอเตอร์เป็นแขนกล

2.3 หุ่นยนต์แมลง



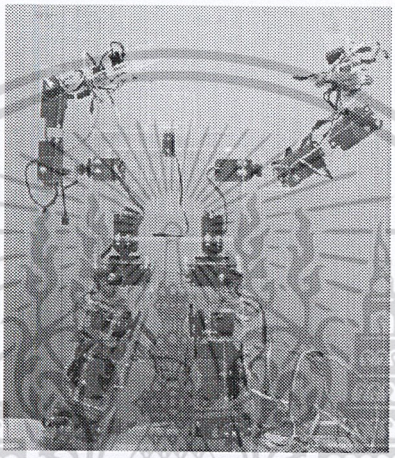
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 2.6 หุ่นยนต์แมลง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการควบคุมแบบอัตโนมัติ ซึ่งจะรับชิ้นงานจากแขนกลซึ่งหยิบมาจากรถขนส่งภายในระบบลำเลียงสินค้าเพื่อจะนำชิ้นงานมาเก็บไว้ที่คลังสินค้า (Store) หรือ สถานีต่อไปในระบบ และสามารถควบคุมด้วยสวิตซ์ควบคุม เมื่อต้องการใช้กับการขนส่งแบบอื่น ๆ ที่ต้องการตามความเหมาะสมของงานนั้น ๆ

2.4 หุ่นยนต์ 2 ขา



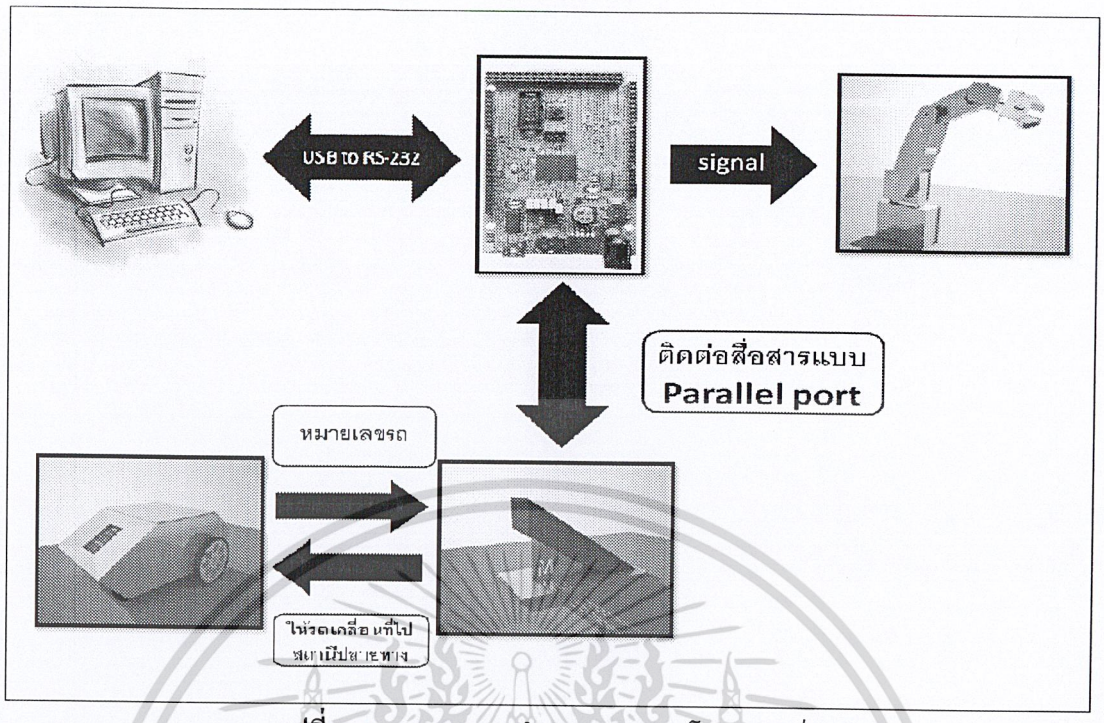
รูปที่ 2.7 หุ่นยนต์ 2 ขา

หุ่นยนต์ 2 ขา ประกอบด้วย Servo Motor จำนวน 23 ตัว ควบคุม Servo Motor โดยเชื่อมต่อสายสัญญาณกับ Pin D22 – D44 ที่กำหนดให้ปล่อยค่า Duty Cycle Pulse ตามที่เขียนโปรแกรมไว้ ในการควบคุมหุ่นยนต์ 2 ขา ในเบื้องต้น ในการจัดท่าหุ่นยนต์ ได้ทำการประยุกต์ใช้วงจรควบคุมการจัดท่าของหุ่นยนต์แขนกลมาใช้ในการจัดท่าการเดินของหุ่นยนต์ 2 ขา โดยใช้ Digital Pin D2-D9 ซึ่งแต่ละ Pin จะส่งสัญญาณ Logic 1 ออกมา เช่น สวิตซ์ S1 จะมี Pin D2 กับ D6 เชื่อมกันอยู่ เมื่อกดปุ่ม S1 Logic จาก D2 และ D9 จะเชื่อมถึงกัน ทำให้ Microcontroller รู้ว่ามีการกดปุ่ม S1 ตามที่เขียนโปรแกรมไว้

2.5 คอนโทรลเลอร์หลักที่กระจายคำสั่ง (Main Controller)

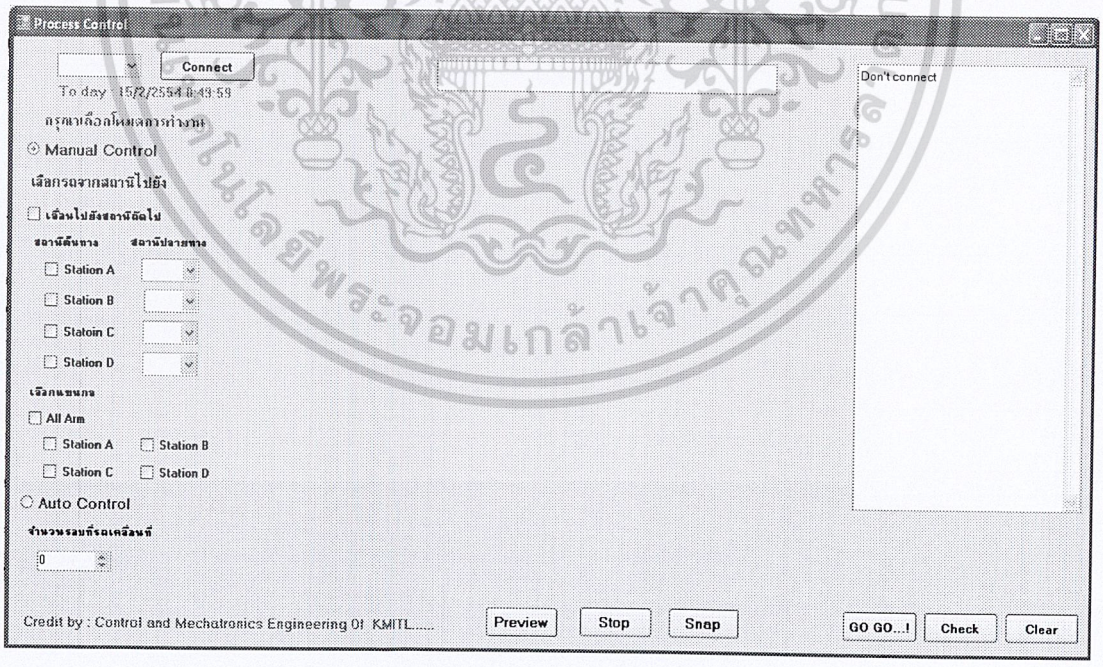
คอนโทรลเลอร์หลักจะติดต่อสื่อสารผ่านทางสายสัญญาณ RS-232 แล้วทำการแปลงเป็น USB ก่อนเข้าสู่คอมพิวเตอร์ เมื่อคอนโทรลเลอร์รับคำสั่งมาจะทำการกระจายคำสั่งไปยังแต่ละสถานีและแขนกลของแต่ละสถานี โดยคอนโทรลเลอร์ใช้สายแพในการเชื่อมต่อข้อมูลสถานีและติดต่อกับแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 กระบวนการทำงานของคอนโทรลเลอร์

2.6 โปรแกรมควบคุมการทำงาน



รูปที่ 2.9 หน้าตาโปรแกรมควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นโปรแกรมที่ทำหน้าที่ตรวจสอบและสั่งการทำงานบนคอมพิวเตอร์ โดยคอนโทรลเลอร์จะแบ่งระบบออก 2 โหมด คือ

2.6. 1 โหมดควบคุมด้วยมือ (Manual Mode)

ในการทำงานในโหมดควบคุมด้วยมือ นั้นเราจะเป็นผู้สั่งการได้ว่าจะให้รถที่สถานีที่เราต้องการไปจอดในสถานีที่เรากำหนด โดยเราต้องทำการเลือกสถานีต้นทาง และสถานีปลายทาง

2.6.2 โหมดคบคุมอัตโนมัติ (Auto Mode)

สำหรับการทำงานในโหมด Auto นั้นจะมีลำดับการทำงานดังนี้

1). รถทั้ง 4 คันจะเข้ามาจอดประจำสถานี

2). แขนกลจะทำงานโดยแบ่งการทำงานออกเป็น 2 แบบ คือแขนกลสถานี A กับ C จะหยิบของออกจากหลังรถ แขนกลสถานี B กับ D จะหยิบของมาวางไว้หลังรถ

3). เมื่อแขนกลทำงานเสร็จ รถทั้ง 4 คัน จะเคลื่อนที่ไปจอดสถานีต่อไป

2.7 การติดต่อสื่อสารกันของระบบ

สำหรับการติดต่อสื่อสารกันของภายในระบบที่ทำการจำลองนั้น สามารถแบ่งการติดต่อสื่อสารระหว่างกันออกได้เป็น 3 แบบคือ

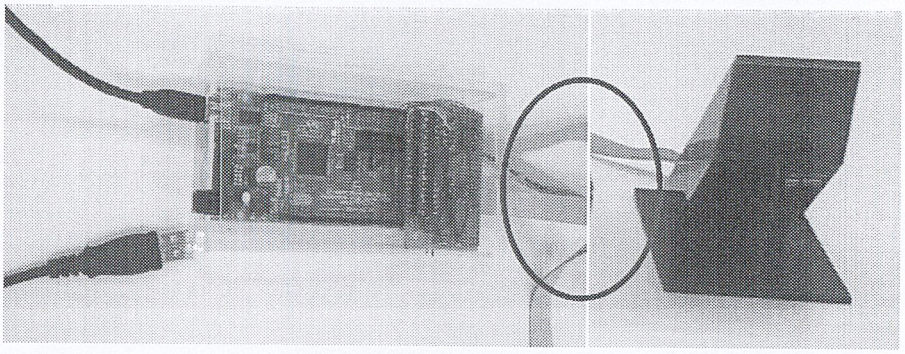
2.7.1 การติดต่อสื่อสารกันระหว่างรถลำเลียง กับ สถานี

รูปที่ 2.10 การติดต่อสื่อสารกันระหว่างรถลำเลียง กับ สถานี

สำหรับการลิงค์กันระหว่างรถลำเลียง กับ สถานีนั้น ใช้การติดต่อสื่อสารกันผ่าน โดยใช้ตัวรับ-ส่งอินฟราเรดเป็นตัวส่งสัญญาณ และใช้ Module TSOP34836 เป็นตัวรับสัญญาณอินฟราเรดระหว่างรถลำเลียงกับสถานี ซึ่งช่วงความถี่ที่ใช้ในการรับ-ส่งข้อมูลผ่านสัญญาณอินฟราเรดที่ใช้ คือ 40 KHz โดยที่จะติดตั้ง ตัวรับและส่ง สัญญาณนี้ไว้ที่รถลำเลียงและสถานี อย่างละ1ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

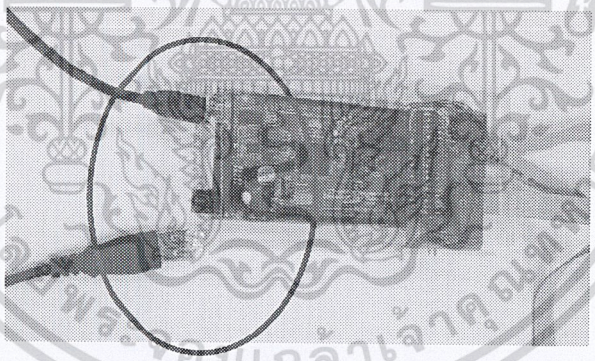
2.7.2 การติดต่อสื่อสารกันระหว่าง คอนโทรลเลอร์หลัก กับสถานี



รูปที่ 2.11 การติดต่อสื่อสารกันระหว่าง คอนโทรลเลอร์หลัก กับสถานี

สำหรับการลิงค์กันระหว่าง คอนโทรลเลอร์หลัก กับ สถานีแต่ละสถานีนั้นใช้สายแพรในการติดต่อข้อมูลแบบขนาด 8 Bit โดยใช้การรับ-ส่งข้อมูลแบบขนาน (Parallel Port) โดยส่วนคอนโทรลเลอร์หลักจะทำหน้าที่เป็นตัวส่งข้อมูลไปที่แต่ละสถานี เพื่อสั่งการว่าให้รถลำเลียงที่มาจอดอยู่ที่สถานีต้นทางเคลื่อนที่ไปสถานีปลายทางตามที่ต้องการ

2.7.3 การติดต่อสื่อสารกันระหว่างคอนโทรลเลอร์หลัก กับ หน้าจอแสดงผล



รูปที่ 2.12 การติดต่อสื่อสารกันระหว่างคอนโทรลเลอร์หลัก กับ หน้าจอแสดงผล

สำหรับการลิงค์กันระหว่างคอนโทรลเลอร์กับหน้าจอแสดงผลบนเครื่องคอมพิวเตอร์นั้นใช้สาย USB ในการรับ-ส่งข้อมูลโดยก่อนเข้าไปสู่คอนโทรลเลอร์นั้นมีการแปลงข้อมูลไปเป็นการติดต่อข้อมูลแบบอนุกรม (Serial Port) โดยใช้ไอซีที่ทำหน้าที่เปลี่ยนรูปแบบการติดต่อสื่อสารข้อมูลแบบ USB to RS232 โดยที่ส่วนหน้าจอแสดงผลนั้นจะรับคำสั่งจากผู้ใช้ผ่านหน้าโปรแกรม จากนั้นจะส่งข้อมูลไปที่ คอนโทรลเลอร์หลักเพื่อสั่งการระบบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ทฤษฎีและความรู้ที่เกี่ยวข้อง

3.1 ทฤษฎีและความรู้ที่เกี่ยวข้อง

จากที่ได้กล่าวในบทที่ 1 แล้วว่า ก่อนที่จะมีการออกแบบและจำลองระบบการผลิตผสมผสานด้วยคอมพิวเตอร์นั้น จำเป็นต้องศึกษาองค์ประกอบต่างๆที่จำเป็นของระบบที่จะทำการจำลองให้เข้าใจเสียก่อน พบว่าในกระบวนการผลิตที่จำลองนั้นในแต่ละส่วนนั้นมีส่วนที่สำคัญหลายส่วน ดังนั้นในบทนี้จะศึกษาและอธิบายถึงองค์ประกอบต่างๆ ที่จะนำไปใช้งานจริงในระบบที่ทำการจำลอง ซึ่งประกอบด้วยการประมวลผลภาพดิจิตอล มอเตอร์ไฟฟ้ากระแสตรง วงจรที่ใช้ในการรับส่งสัญญาณอินฟราเรด รวมถึงรายละเอียดของพอร์ตขนาน และพอร์ตอนุกรม และตัวควบคุมแบบต่างๆ

3.1.1 มอเตอร์กระแสตรง

3.1.1.1 หลักการทำงานของมอเตอร์กระแสตรง

รูปที่ 3.1 โครงสร้างทั่วไปของมอเตอร์กระแสตรง

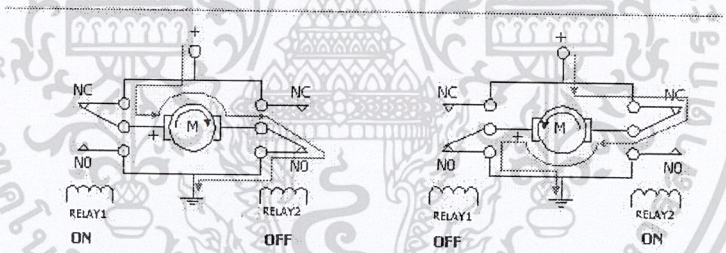
จากในรูปทางเดินของฟลั๊กซ์แม่เหล็ก และสนามแม่เหล็กจะเกิดจากแท่งแม่เหล็กเฟอร์ไรต์ 2 ชิ้นที่ขึ้นรูปเป็นแบบโค้งยึดติดกับตัวถังได้พอดี เพื่อที่จะให้เส้นแรงแม่เหล็กวิ่งเข้าสู่ใจกลางของมอเตอร์ได้ ดังนั้นความเข้มของแม่เหล็กจะขึ้นอยู่กับขนาดความหนาของแม่เหล็กด้วย ซึ่งส่งผลให้ฟลั๊กซ์แม่เหล็กวิ่งไปบนตัวถังโลหะ กระแสไฟฟ้าในขดลวดที่พันกับทุ่นโรเตอร์ก็จะทำให้เกิดสนามแม่เหล็กไฟฟ้า และต้านกับสนามแม่เหล็กถาวร จึงเกิดเป็นแรงบิดเพื่อที่จะหมุนทุ่นโรเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

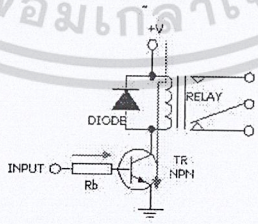
ให้ไปในทิศทางเดียวกันกับทิศทางของสนามแม่เหล็กที่มีแรงมากกว่า กระแสก็จะไหลผ่านไปยังทุ่นโรเตอร์ โดยผ่านแปรงถ่าน ซึ่งจะสัมผัสกับแหวนตัวนำในทุ่นโรเตอร์ และแหวนคอมมิวเตเตอร์ ซึ่งจะถูกแบ่งออกเป็น 3 เซกเมนต์เพื่อที่จะทำหน้าที่นำกระแสเข้าขดลวดนั้นเอง

3.1.1.2 การขับและกลับทิศทางของมอเตอร์กระแสตรง (DC motor)

ในการใช้ไอซีไมโครคอนโทรเลอร์เป็นตัวควบคุมการหมุน และทิศทางของมอเตอร์กระแสตรงนั้น เราจะต้องมีส่วนของวงจร ที่เรียกว่า วงจรขับมอเตอร์ (Driver) ในส่วนของวงจรกลับทิศทางของมอเตอร์นั้น สามารถที่จะใช้รีเลย์ต่อวงจร สวิตซ์เพื่อกลับทิศทางของขั้วไฟกระแสตรง หรืออาจใช้อุปกรณ์สารกึ่งตัวนำที่เป็นวงจรขับกำลังเช่น ทรานซิสเตอร์ มอสเฟต แล้วแต่วิธีที่เราจะเลือกใช้งาน จากรูปเป็นการใช้รีเลย์ควบคุมการเปลี่ยนทิศทางการหมุนของมอเตอร์ โดยการควบคุมการปิด - เปิดที่รีเลย์ 2 ตัว ซึ่งจะทำหน้าที่กลับทิศทางของขั้วไฟที่ป้อนให้กับมอเตอร์ โดยการสลับการทำงานของรีเลย์ เช่นให้รีเลย์ตัวที่ 1 ทำงาน (ON) และรีเลย์ตัวที่ 2 หยุดทำงาน (OFF) จะทำให้มอเตอร์หมุนไปทางซ้าย และในทำนองเดียวกันถ้าหากรีเลย์ตัวที่ 1 หยุดทำงาน (OFF) และรีเลย์ตัวที่ 2 ทำงาน (ON) ก็จะทำให้มอเตอร์หมุนไปทางขวา



รูปที่ 3.2 การกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์

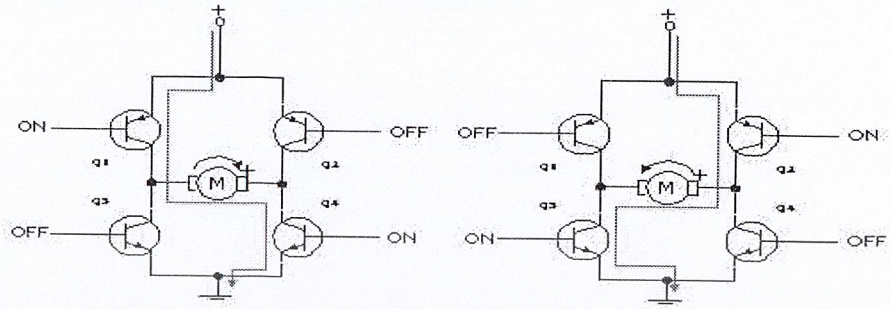


รูปที่ 3.3 การใช้ทรานซิสเตอร์เพื่อขับรีเลย์ให้ทำงาน

จากรูปที่3.2 เป็นวงจรขับรีเลย์โดยใช้ทรานซิสเตอร์ทำหน้าที่ขยายกระแส ด้วยเหตุผลเพราะไม่สามารถจะใช้ขา เอาต์พุตของไมโครคอนโทรลเลอร์ป้อนกระแสไฟที่ขดลวดของรีเลย์โดยตรงได้ เนื่องจากว่ากระแสที่จ่ายออกมาจากขา เอาต์พุตของไมโครคอนโทรลเลอร์มีค่าน้อยเกินไป ดังนั้น เราจึงต้องมีส่วนของวงจรทรานซิสเตอร์เพื่อที่จะทำการขยายกระแสให้เพียงพอในการป้อนให้กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขดลวดของรีเลย์ ส่วนไดโอดนำมาต่อไว้ สำหรับป้องกันแรงดันย้อนกลับที่เกิดจากการเหนี่ยวนำของสนามแม่เหล็กในขณะเกิดการยุบตัว ซึ่งอาจจะทำให้ทรานซิสเตอร์เสียหายได้

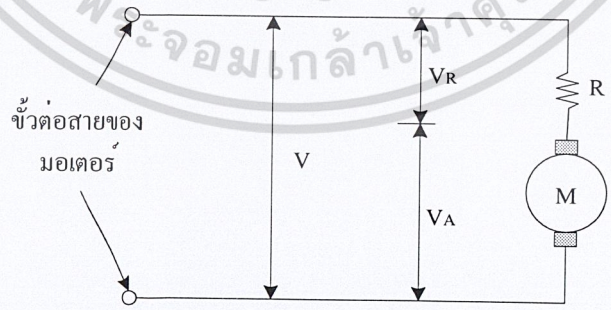


รูปที่ 3.4 การใช้ทรานซิสเตอร์เป็นวงจรขับและกำหนดทิศทางของมอเตอร์กระแสตรง

จากรูปเป็นวงจรลิเนียร์บริดจ์แอมป์ ซึ่งจะประกอบไปด้วยทรานซิสเตอร์กำลัง 4 ตัวที่ทำหน้าที่ขับ และควบคุมทิศทางการหมุนของมอเตอร์ ถ้าหากกำหนดให้ทรานซิสเตอร์ Q1 และ Q4 อยู่ในสภาวะทำงาน (Active) กระแสไฟฟ้าจะไหลผ่านทรานซิสเตอร์จากซ้ายไปขวา โดยผ่านมอเตอร์กระแสตรงทำให้มอเตอร์หมุนไปทางขวา ในทำนองเดียวกันถ้าหากเราทำให้ทรานซิสเตอร์ Q2 และ Q3 อยู่ในสภาวะทำงาน (Active) กระแสไฟฟ้าก็จะไหลจากทางขวาไปทางซ้ายซึ่งจะส่งผลให้มอเตอร์กลับทิศทางการ หมุนจากทางขวาไปทางซ้าย

3.1.1.3 คุณสมบัติของมอเตอร์กระแสตรง

ในการอธิบายคุณสมบัติของมอเตอร์กระแสตรงให้ละเอียดนั้นต้องพิจารณาแรงดันที่ป้อนและความต้านทานของโรเตอร์ด้วย วงจรภายในของมอเตอร์เขียนได้ดังรูปที่ 3.8



รูปที่ 3.5 วงจรภายในของมอเตอร์กระแสตรง

โดยสมมติให้ทุ่นโรเตอร์ไม่มีความต้านทานอยู่เลย อนุกรมกับความต้านทานซึ่งในที่นี้ก็คือความต้านทานของขดลวดนั่นเอง แรงดันที่ขั้วต่อสายของมอเตอร์ก็คือผลบวกระหว่างแรงดันที่ทุ่นโรเตอร์ (V_A) และ แรงดันตกคร่อมความต้านทานขดลวด (V_R)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดัน V_A ถูกเรียกว่า แรงเคลื่อนเหนี่ยวนำป้อนกลับ (BACK EMF) ซึ่งเกิดขึ้นในโรเตอร์ ขณะที่หมุนแรงดันที่เกิดขึ้นนี้เป็นไปตากกฎของการเหนี่ยวนำแม่เหล็กไฟฟ้าจากการเคลื่อนที่ของตัวนำในสนามแม่เหล็ก สัมพันธ์กับแรงเคลื่อนเหนี่ยวนำแม่เหล็ก และ ความเร็วในการเคลื่อนที่ของตัวนำ แรงดันที่เกิดขึ้นจะมีขั้วตรงกันข้ามกับแรงดันที่ป้อนให้กับมอเตอร์ และ แปรผันตรงกับความเร็วในการหมุน ผลบวกของแรงดันที่ทุ่นโรเตอร์ (V_A) และแรงดันตกคร่อมขดลวด (V_R) ต้องเท่ากับแรงดันที่ป้อนให้กับมอเตอร์ (V)

$$V = V_A + V_R \tag{3.1}$$

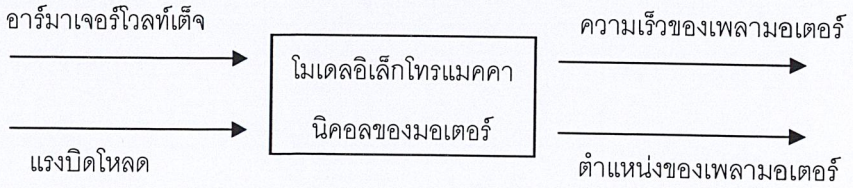
เมื่อพิจารณาตั้งแต่มอเตอร์หยุดนิ่ง ความเร็วมีค่าเป็นศูนย์ ดังนั้น $V_A = 0, V_R = V$ กระแสที่ไหลในมอเตอร์หาได้จาก

$$I = V_R / R \tag{3.2}$$

เมื่อมอเตอร์เริ่มหมุนจะมีความเร็ว และ V_A เพิ่มขึ้นเป็นเส้นตรงตามความเร็ว V_R ซึ่งมีค่าเท่ากับความแตกต่างระหว่าง V_A และ V จะเริ่มลดลงกระแส I ก็จะเริ่มลดลงเช่นกันขณะที่มอเตอร์ยังมีความเร่งอยู่ ความเร็วจะเพิ่มขึ้น แรงบิดจะลดลงจนกว่าจะถึงจุดซึ่งแรงบิดจองมอเตอร์รับภาระโหลดได้สมดุลพอดี ขณะที่มอเตอร์ไม่มีโหลด และ หมุนอย่างอิสระจะมีเพียงค่าความฝืดของแบริ่ง และ แรงต้านอากาศทำให้ V_A เกือบเท่ากับค่า V

3.1.1.4 โมเดลคณิตศาสตร์ของดีซีมอเตอร์

ดีซีมอเตอร์ที่ใช้ร่วมกับดีซีแอมพลิไฟล์ทั้งในระบบการบังคับตำแหน่งและการบังคับความเร็วมักจะได้รับการประยุกต์ใช้เป็นส่วนประกอบสร้างกำลังงานในระบบการนำร่องและระบบบังคับต่างๆ และเนื่องจากวิทยาการเกี่ยวกับสารแม่เหล็กและการขยายด้วยโซลิคสเตททำให้ดีซีมอเตอร์แบบแม่เหล็กถาวรได้รับความนิยมใช้เป็นส่วนประกอบการขับเคลื่อนในระบบการบังคับแบบปิดลูฟต่างๆ มากขึ้น การออกแบบและการชดเชยระบบดังกล่าวได้อย่างเหมาะสมจะต้องใช้โมเดลทางคณิตศาสตร์ของส่วนประกอบทั้งหมดในระบบ

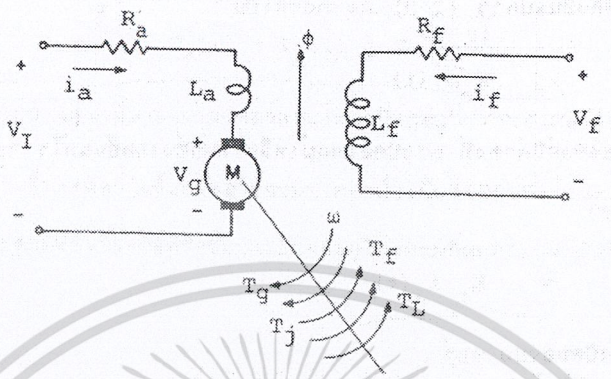


รูปที่ 3.6 อินพุตและเอาต์พุตของโมเดลทางคณิตศาสตร์ของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.5 โมเดลอิเล็กโทรแมคคานิคอล

ส่วนสำคัญของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้นมีโมเดลดังแสดงในรูปที่ 3.7



รูปที่ 3.7 โมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น

R_a: ความต้านทานของอาร์เมเจอร์

L_a: อินดัคแต้นซ์ของอาร์เมเจอร์

V_g: โวลท์เต็จกำเนิดในอาร์เมเจอร์(โวลท์เต็จย้อนกลับ)

R_f: ความต้านทานของฟิลด์

L_f: อินดัคแต้นซ์ของฟิลด์

ϕ : ช่องว่างอากาศของเส้นแรงสนามแม่เหล็ก

ω : ความเร็วของเพลาอาร์เมเจอร์

T_g: แรงบิดที่พัฒนาขึ้นในมอเตอร์

T_f: แรงบิดเสียดทานของมอเตอร์

T_J: แรงเฉื่อยของมอเตอร์

T_L: แรงบิดโหลดบนเพลาของมอเตอร์

ขั้นแรกเราจะหาสมการพื้นฐานโมเดลของดีซีมอเตอร์ได้จากลูฟของอาร์เมเจอร์

$$V_i(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + V_g(t) \tag{3.3}$$

เทอมโวลท์เต็จ $V_g(t)$ ในสมการ (1) คือโวลท์เต็จย้อนกลับของมอเตอร์ซึ่งเกิดขึ้นเมื่อเส้นลวดตัวนำของอาร์มาเจอร์หมุนตัดเส้นแรงแม่เหล็กซึ่งเกิดขึ้นในกระแสของฟิลด์ (i_f) ตามกฎของฟาราเดย์ลูฟของเส้นลวดตัวนำหมุนในฟิลด์แม่เหล็กคงที่จะมีการเหนี่ยวนำโวลท์เต็จขึ้นในขดลวดนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

$$V_i(t) = \frac{d\lambda(t)}{dt} \tag{3.4}$$

เมื่อ $\lambda(t)$ คือเส้นแรงแม่เหล็กที่ลิงเคจ(linkages) ไปยังขดลวดและ t คือเวลาในการหมุนของคี อมนิวเทเตอร์ของมอเตอร์ การควบคุมวงจรของแต่ละส่วนของตัวนำในโรเตอร์จะเกิดโวลท์เต็จขึ้น ในส่วนของตัวนำนั้นตามสมการ (2) เมื่อ $\frac{d\lambda(t)}{dt}$ จะเป็นสัดส่วนต่อเส้นแรงแม่เหล็กในช่องว่าง อากาศและความเร็วเชิงมุม $\omega(t)$ เราจะได้ว่า

$$V_g(t) = K\phi(t)\omega(t) \tag{3.5}$$

สมมติให้กระแสของฟิลด์มีค่าคงที่และไม่คิดถึงส่วนการเปลี่ยนแปลงในเส้นแรงฟิลด์เนื่องจาก อาร์เมเจอร์รีแอคชั่นเส้นแรงฟิลด์ก็จะมีค่าคงที่ดังนั้นสมการ (3.5) ก็จะเป็น

$$V_g(t) = K_e\omega(t) \tag{3.6}$$

เมื่อเราสมมติให้เส้นแรงของฟิลมีค่าคงที่ แรงบิดของแม่เหล็กไฟฟ้าซึ่งเกิดขึ้นแก่โรเตอร์ของ มอเตอร์จะเป็นสัดส่วนกับกระแสของอาร์มาเจอร์

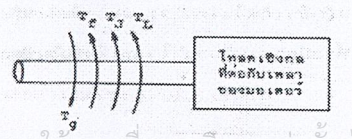
$$T_g(t) = K_t\, i_a(t) \tag{3.7}$$

เมื่อ K_t คือ ค่าคงที่ของแรงบิดของมอเตอร์

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์คือผลคูณของแรงบิดที่เกิดขึ้นและความเร็วเชิงมุม

$$P_g(t) = T_g(t)\,\omega(t) \tag{3.8}$$

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์ทั้งหมดนี้จะจ่ายไปยังโหลดที่ต่ออยู่กับเพลาของมอเตอร์แต่ กำลังงานนี้บางส่วนจะสูญเสียไปในมอเตอร์ การสูญเสียจากแรงเสียดทาน หมายถึงความหน่วง เนื่องจากลมที่มีต่อโรเตอร์ แรงเสียดทานตัวรองรับโรเตอร์ กระแสที่ไหลวนในเหล็กของโรเตอร์ และฮีสเทรีซิส โดยแรงบิดต่างๆแสดงดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 แรงบิดต่างๆที่เกิดขึ้นต่อโหลดของมอเตอร์

119377

$T_g(t)$: แรงบิดของมอเตอร์

$T_f(t)$: แรงบิดที่ต้องชนะการสูญเสียเนื่องจากการเสียดทาน

$T_J(t)$: แรงบิดเพื่อใช้เพิ่มอัตราเร่งแก่ความเฉื่อยของโหลด

$T_L(t)$: แรงบิดโหลด

ในช่วงเวลาใดๆก็ตาม แรงบิดของมอเตอร์จะต้องเท่ากับและมีทิศทางตรงข้ามกับผลรวมของแรงบิด $T_f(t)$ $T_J(t)$ และ $T_L(t)$ ดังนั้น

$$T_g(t) = T_f(t) + T_L(t) + J\frac{d\omega(t)}{dt} \tag{3.9}$$

เมื่อ J คือผลรวมของโมเม้นท์แรงเฉื่อยของโรเตอร์และโหลดที่ต่ออยู่ที่เพาของมอเตอร์

ผลรวมของแรงบิดเสียดทานที่ประกอบกันขึ้นที่เพลาของมอเตอร์ซึ่งเป็นลิเนียร์ฟังก์ชันกับความเร็วเชิงมุมของโรเตอร์เรียกว่า ส่วนประกอบของวิสคอสฟริคชั่นและมักจะอยู่ในเทอมที่แยกออกจากฟริคชั่นอื่นๆ ซึ่งแสดงได้ด้วยสมการต่อไปนี้

$$T_g(t) = T_f(t) + T_L(t) + J\frac{d\omega(t)}{dt} + B\omega(t) \tag{3.10}$$

เมื่อ B คือสัมประสิทธิ์ของวิสคอสฟริคชั่นของมอเตอร์และโหลดที่ต่ออยู่กับเพลาของมอเตอร์ $T_f(t)$ คือผลรวมของฟริคชั่นของโหลดและมอเตอร์ทั้งหมด มีแรงต้านของลมและการสูญเสียกำลังในเหล็กของเพลามอเตอร์ยกเว้นวิสคอสฟริคชั่น

สมการ (3.5) (3.6) (3.7) และ (3.8) เป็นชุดสมการพื้นฐานของดีซีมอเตอร์โมเดลและสมการเหล่านี้เราสามารถจะหาทรานสเฟอร์ฟังก์ชั่นของดีซีมอเตอร์ได้ โดยใส่ลาปลาสทรานสฟอร์มทั้งสองข้างของชุดสมการพื้นฐานได้เป็น

$$V_i(s) - V_g(s) = (R_a + sL_a)\, I_a(s) \tag{3.11}$$

$$V_g(s) = K_e\omega(s) \tag{3.12}$$

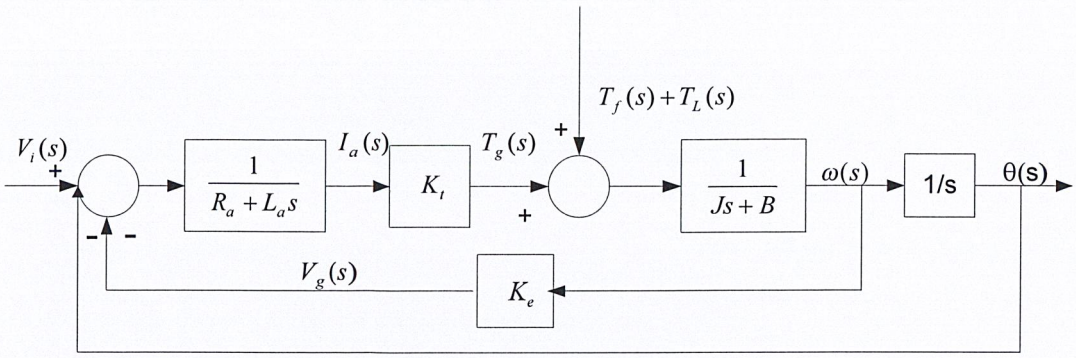
$$T_g(s) = K_t\, I_a(s) \tag{3.13}$$

$$T_g(s) - T_f(s) - T_L(s) = (B + sJ)\,\omega(s) \tag{3.14}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

สามารถเขียนเป็นบล็อกไดอะแกรมที่แสดงสมการพื้นฐานเหล่านี้ได้ดังนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 บล็อกไดอะแกรมของดีซีมอเตอร์โมเดล

<u>ข้อสังเกต</u>

สมมติว่าโวลท์เต็จที่ป้อนให้กับวงจรอาร์มาเจอร์ของมอเตอร์มีค่าคงที่ดังนั้นมอเตอร์จะหมุนด้วยความเร็วคงที่คือทำงานอยู่ที่สภาวะสงบนิ่งด้วยโหลดที่คงที่ กำลังงานเชิงกลที่เกิดขึ้นโดยโรเตอร์ จะได้สมการดังนี้

$$P_g = T_g\omega = K_t I_a \omega \tag{3.15}$$

เมื่อทุกเทอมในสมการสุดท้ายมีค่าคงที่เนื่องจากมอเตอร์ทำงานอยู่ที่สภาวะสงบนิ่งกำลังไฟฟ้าที่ถูกดูดกลืนโดยอาร์เมเจอร์ต้องเท่ากับ

$$P = V_g I_a = K_e \omega\ I_a \tag{3.16}$$

ดังนั้นเราจะได้ว่ากำลังงานเชิงกลที่เกิดขึ้นต้องเท่ากับกำลังงานไฟฟ้าที่ถูกดูดกลืนในโรเดอร์คือ สรุปได้ว่า $K_e = K_t$

ทรานสเฟอร์ฟังก์ชั่นของดีซีมอเตอร์

บล็อกไดอะแกรมของรูปที่ 3.9 แสดงถึงระบบที่มีสองอินพุท และมีเอาต์พุทเป็นความเร็วเชิงมุม $\omega(s)$ และการเคลื่อนที่แบบเชิงมุม $\omega(s)$ จากรูปที่ 3.8 ความเร็วเอาท์พุทของระบบเขียนได้เป็น

$$\omega(s) = G_1(s)V_i(s) + G_2(s)[T_f(s) + T_L(s)] \tag{3.17}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ

$$G_1(s) = \left.\frac{\omega(s)}{V_i(s)}\right|_{T_f(s)+T_L(s)=0} \tag{3.18}$$

$$G_2(s) = \left.\frac{\omega(s)}{T_f(s)+T_L(s)}\right|_{V_i(s)=0} \tag{3.19}$$

$G_1(s)$ คือทรานสเฟอร์ฟังก์ชั่นระหว่างโวลท์เต็จและความเร็ว

$$G_1(s) = \frac{\omega(s)}{V_i(s)} = \frac{K_t}{(L_a s + R_a)(Js + B) + K_t K_e} = \frac{K_m}{\alpha S^2 + \beta s + 1} \tag{3.20}$$

เมื่อ $K_m = \dfrac{K_t}{R_a B + K_t K_e}$

$$\alpha = \frac{L_a J}{R_a B + K_t K_e}$$

$$\beta = \frac{R_a J + L_a B}{R_a B + K_t K_e}$$

สมการ (3.20) เป็นโวลท์เต็จทรานสเฟอร์ฟังก์ชั่นของดีซีมอเตอร์ในเมื่อสมมติว่า T_f และ T_L มีค่าเป็นศูนย์ สมการ (3.20) สามารถเขียนใหม่ได้เป็น

$$G_1(s) = \frac{K_t}{R_a B(1+\tau_e s)(1+\tau_m s) + K_t K_e}$$

เมื่อ $\tau_e = \dfrac{L_a}{R_a}$ = ไทม์คอนสแต้นท์ทางไฟฟ้า

$\tau_m = \dfrac{J}{B}$ = ไทม์คอนสแต้นท์ทางเชิงกล

ถ้าอินดัคแต้นท์ของอาร์มาเจอร์มีค่าน้อย ไทม์คอนสแต้นท์ทางไฟฟ้าสามารถตัดทิ้งได้และได้สมการเป็น

$$G_V(s) = \frac{\omega(s)}{V_i(s)} = \frac{K_t}{R_a(Js+B) + K_t K_e} = \frac{K_m}{\tau s + 1} \tag{3.21}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $\tau = \dfrac{R_a J}{R_a B + K_t K_e}$

ในสมการ (3.21) ค่าคงที่ K_m อาจเรียกได้ว่าเป็นค่าคงที่ของมอเตอร์

ทรานสเฟอร์ฟังก์ชั่นแรงบิดโหลด $G_2(s)$ หาได้เป็น

$$G_2(s) = \frac{\omega(s)}{T_f(s) + T_L(s)} = \frac{-\dfrac{1}{Js+B}}{1 + \dfrac{K_t K_e}{(Js+B)(L_a s + R_a)}} = \frac{-\dfrac{R_a}{K_t} K_m \left[\dfrac{L_a}{R_a} s + 1\right]}{\alpha s^2 + \beta s + 1} \tag{3.22}$$

ซึ่งถ้าอินดัคแต้นท์ของอาร์มาเจอร์ไม่นำมาคิด จะทำให้ได้สมการ ดังนี้

$$G_L(s) = \frac{\omega(s)}{T_f(s) + T_L(s)} = \frac{-\dfrac{R_a}{K_t} K_m}{\tau s + 1}$$

ซึ่งจากสมการที่ (3.22) เมื่อให้ค่าของ ว่า T_f และ T_L มีค่าเป็นศูนย์จะทำให้ค่าทรานสเฟอร์ฟังก์ชั่นมีค่าดังนี้

$$\omega(s) = G_v(s) V_i(s) = \frac{K_m}{\tau s + 1} V_i(s) \tag{3.22}$$

โมเดลคณิตศาสตร์ในการควบคุมตำแหน่งของมอเตอร์ จาก Transfer function ในโมเดลคณิตศาสตร์ของดีซีมอเตอร์ ซึ่งจะได้ model ของการควบคุมความเร็วของมอเตอร์ในรูปของสมการอันดับหนึ่งเป็นดังนี้ คือ

$$\omega(s) = G_v(s) V_i(s) = \frac{K_m}{\tau s + 1} V_i(s) \tag{3.23}$$

จากสมการข้างต้นดังกล่าวนั้นจะเห็นถึงความสัมพันธ์ระหว่างความเร็วเชิงมุม (output) และ ค่าแรงดันที่ป้อน (input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

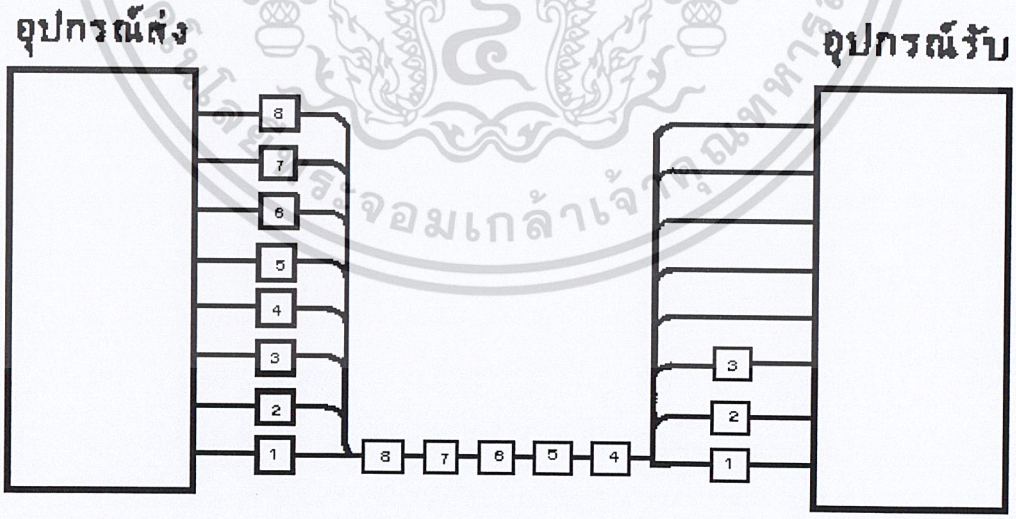
และในการควบคุมตำแหน่งจะมีการผ่านตัว Integrator (1/s) ทำให้ได้เอาท์พุตคือ $\Theta(s)$ ซึ่งจากรูปที่ 3.9และ transfer function ที่ได้ เมื่อทำการผ่าน Integrator (1/s) เข้าไปจะทำให้ได้ค่าของ output เป็นมุมในการเคลื่อนที่ของมอเตอร์ ซึ่งเขียนในรูปของสมการได้ดังนี้

$$\theta(s) = \frac{\omega(s)}{s} = \left[\frac{1}{s}\right]\left[\frac{K_m V_i(s)}{\tau s + 1}\right] \tag{3.24}$$

ส่วนในการหาโมเดลมอเตอร์ที่ใช้ในการทดลองนั้น เรากำหนดให้โมเดลของมอเตอร์เป็นโมเดลมอเตอร์อันดับหนึ่ง (First Order) ซึ่งเราได้ทำการทดลองเพื่อหาโมเดลที่เป็นลักษณะอันดับหนึ่ง

3.1.2 การสื่อสารแบบอนุกรม (Serial Port)

พอร์ตอนุกรม (Serial Port) คือ การถ่ายโอนข้อมูลโดยจะถูกส่งออกมาทีละบิต ระหว่างจุดส่งและจุดรับ การถ่ายโอนข้อมูลแบบอนุกรมต้องการตัวกลางสำหรับการสื่อสารเพียงช่องเดียวหรือสายเพียงคู่เดียว ค่าใช้จ่ายจะถูกสำหรับการส่งระยะทางไกลๆ โดยเฉพาะเมื่อเรามีระบบการสื่อสารทางโทรศัพท์ไว้ใช้งานอยู่แล้ว



รูปที่ 3.10 การรับ-ส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการส่งข้อมูล (Baud rate) คือความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิตต่อวินาทีเช่น 300, 1,200, 2,400, 4,800 , 9,600 ,14,400 ,19,200, 38,400 ,56,000 เป็นต้น

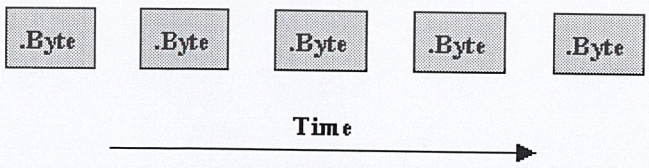
ประเภทของการสื่อสารแบบอนุกรมแบ่งตามลักษณะสัญญาณในการส่งแบ่งได้ 2 แบบ คือ

1. การสื่อสารแบบซิงโครนัส (Synchronous) เป็นการสื่อสารข้อมูลโดยใช้สัญญาณนาฬิกาในการควบคุมจังหวะของการรับส่งสัญญาณ

2. การสื่อสารแบบอะซิงโครนัส (Asynchronous) เป็นการสื่อสารที่ใช้สายข้อมูลเพียงตัวเดียว จะใช้รูปแบบของการส่งข้อมูล(Bit Pattern) เป็นตัวกำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล ส่วนไหนเป็นตัวข้อมูล ส่วนไหนจะเป็นตัวตรวจสอบความถูกต้องของข้อมูล และส่วนไหนเป็นส่วนปิดท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาคส่งและภาครับ

3.1.2.1 การส่งข้อมูลแบบอะซิงโครนัส

มักจะใช้กับเทอร์มินัลธรรมดา (dumb terminal) ไว้สำหรับรับข้อมูลจากคอมพิวเตอร์แม่และแสดงผลที่จอ โดยไม่สามารถเปลี่ยนแปลงข้อมูลได้ การส่งข้อมูลแบบนี้มักจะมีอัตราในการรับส่งข้อมูลที่แน่นอนมีหน่วยเป็นบิตต่อวินาที (bit per second) เมื่ออุปกรณ์อะซิงโครนัสจะส่งข้อมูล 1 ไบต์ ก็จะส่งบิตเริ่มต้น (start bit) ก่อน ซึ่งมักจะเป็น "0" และตามด้วยข้อมูลทั้ง 8 บิตใน 1 ไบต์ แล้วจึงจะส่งบิตหยุด (stop bit) ซึ่งมักจะเป็น "1" บิตทั้งหมดนี้ จะรวมกันเป็น 10 บิต ในการส่งข้อมูลเรียงตามลำดับดังนี้ 1 บิตเริ่มต้น 7 บิตข้อมูล (data bit) 1 บิต ภาวะเสมอมูล และ 1 บิตหยุด กระบวนการเหล่านี้จะห่างกัน 1 วินาที ที่จะส่งข้อมูลชุดต่อไป ซึ่งก็หมายถึงว่าเมื่อคอมพิวเตอร์แม่ได้รับบิตเริ่มต้น ก็คาดหวังว่าจะได้รับอีก 9 บิตภายในเวลา 1 วินาที



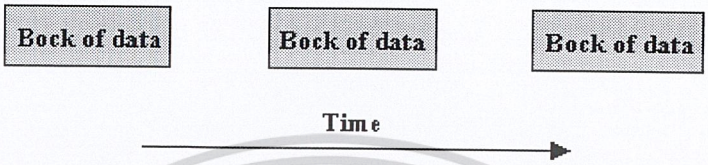
รูปที่ 3.11 การรับ-ส่งข้อมูลแบบอะซิงโครนัส (Asynchronous)

ในระบบนี้จะเกี่ยวข้องกับเวลาว่าเมื่อไรบิตต่อไปจะมาถึง ถ้าไม่ตรงตามที่กำหนดไว้ การส่งข้อมูลก็จะล้มเหลว ระบบนี้เหมาะในการส่งอักขระจากเทอร์มินัลมายังคอมพิวเตอร์แม่ทันที เคาะแป้นพิมพ์ของเทอร์มินัลก็จะรู้ทันทีว่าจะต้องส่งไบต์ใดโดยเติมบิตเริ่มต้นและบิตหยุดที่หัวและท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของข้อมูลไบต์นั้น ตามลำดับให้ครบ 10 บิตที่จะส่ง ในการส่งข้อมูลอัตราการส่งข้อมูลอาจจะเป็น 110, 300, 1,200, 2,400, 4,800, 9,600, 19,200 บิตต่อวินาที โดยที่ทางด้านส่งและด้านรับจะต้องมีการตั้งค่าความเร็วให้เท่ากัน

3.1.2.2การส่งข้อมูลแบบซิงโครนัส



รูปที่ 3.12 การรับ-ส่งข้อมูลแบบซิงโครนัส (Synchronous)

จะไม่ใช้บิตเริ่มต้นและบิตหยุด จะไม่มีการการหยุดชั่วขณะระหว่างอักขระ จะใช้วิธีให้จังหวะเวลาทั้งสองทางที่ติดต่อกัน มีอยู่สองวิธีที่ปฏิบัติคือ ใช้อักขระซิงก์ (since character) หรือใช้สัญญาณนาฬิกา (clock signal) การใช้อักขระซิงก์ไว้หน้าบล็อก (block) ของอักขระที่ใหญ่ โดยการใส่อักขระซิงก์ไว้หน้าบล็อกของข้อมูลอักขระซิงก์นี้เป็นบิตจำนวนหนึ่งที่ทางอุปกรณ์เครื่องรับสามารถใช้ในการกำหนดอัตราเร็วของข้อมูลให้ตรงกับทางอุปกรณ์เครื่องส่ง การใช้สัญญาณนาฬิกาของด้านส่ง และสัญญาณนาฬิกาของด้านรับจะใช้คนละสายหรือคนละช่องสัญญาณในการส่งข่าวสารเกี่ยวกับเวลาของข้อมูลที่จะส่ง โดยทั่วไปการส่งข้อมูลแบบซิงโครนัสจะทำงานภายใต้การควบคุมของโปรโตคอลในระบบนั้น ๆ และนิยมใช้กับเทอร์มินัลฉลาดและเทอร์มินัลอัจฉริยะ การส่งข้อมูลจะนำข้อมูล 1 ไบท์ มาส่งออกไปตามสายไฟฟ้าเรียงกันไปจนครบ 8 บิต ซึ่งเท่ากับ 1 ตัวอักษร

3.1.2.3 การสื่อสารผ่าน Serial port

การสื่อสารผ่าน Serial port สามารถแบ่งได้เป็น 3 ชนิด ได้แก่ Simplex, Half-Duplex และ Full-Duplex

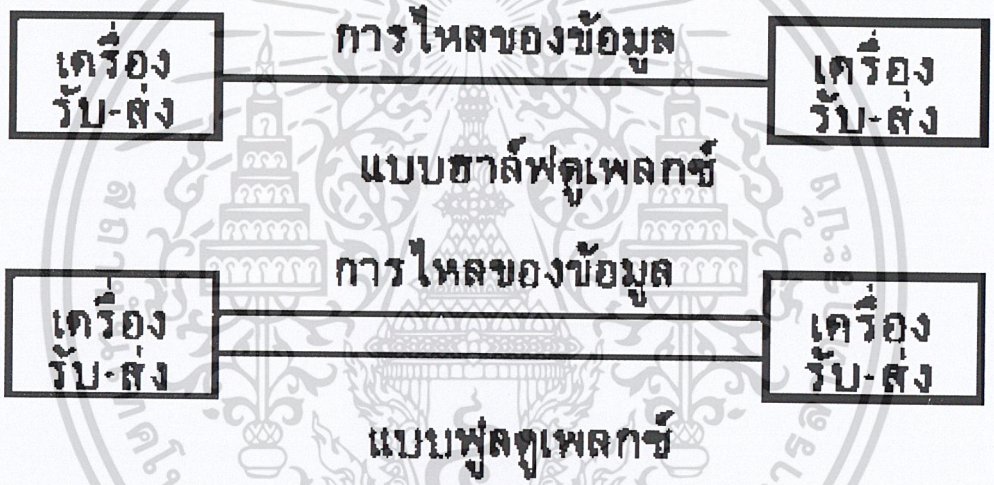
1. Simplex จะเป็นการสื่อสารแบบทางเดียว โดยที่จะมีฝ่ายส่งจะทำหน้าที่ในการส่งเพียงอย่างเดียว ไม่สามารถทำหน้าที่รับข้อมูลได้ ในขณะที่ฝ่ายรับก็จะทำหน้าที่ในการรับเพียงอย่างเดียว ไม่สามารถที่จะทำการส่งข้อมูลได้ อธิเช่น การที่วิทยุรับสัญญาณจากสถานีวิทยุเป็นต้น ซึ่งจะเห็นว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเครื่องรับก็จะมีหน้าที่รับข้อมูลเพียงอย่างเดียว ในขณะที่ฝ่ายส่ง (สถานีวิทยุ) ก็จะมีหน้าที่ส่งข้อมูลเพียงอย่างเดียว

2. Half-Duplex จะเป็นการส่งข้อมูลแบบสองทาง โดยที่ในช่วงเวลาใดๆ จะต้องมีผู้ส่งข้อมูลเพียงผู้เดียว จะทำการส่งพร้อมๆ กันมิได้ อธิเช่น การที่คนสองคนโทรศัพท์คุยกัน ซึ่งจะเป็นการผลัดกันพูด อันจะเห็นว่าทั้งสองฝ่ายสามารถที่จะเป็นได้ทั้งผู้ส่งและรับข้อมูล

3. Full-Duplex จะเป็นการส่งข้อมูลแบบสองทาง ที่สามารถจะส่งข้อมูล และรับข้อมูลได้พร้อมๆ กัน อธิเช่น การที่คนทั้งสองใช้โปรแกรมทำการ chat กัน ซึ่งทั้งสองฝ่ายสามารถที่จะส่งและรับข้อมูลได้พร้อมๆ กัน



รูปที่ 3.13 การสื่อสารข้อมูลผ่านพอร์ตอนุกรม

3.1.2.4 RS-232

RS-232 ย่อมาจาก Recommended Standard-232 (มาตรฐานแนะนำรุ่น 232) เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรม (Serial Port) จำกัดความยาวสายไว้ที่ 50 ฟุต (หรือประมาณ 15 เมตร) สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิตต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้น

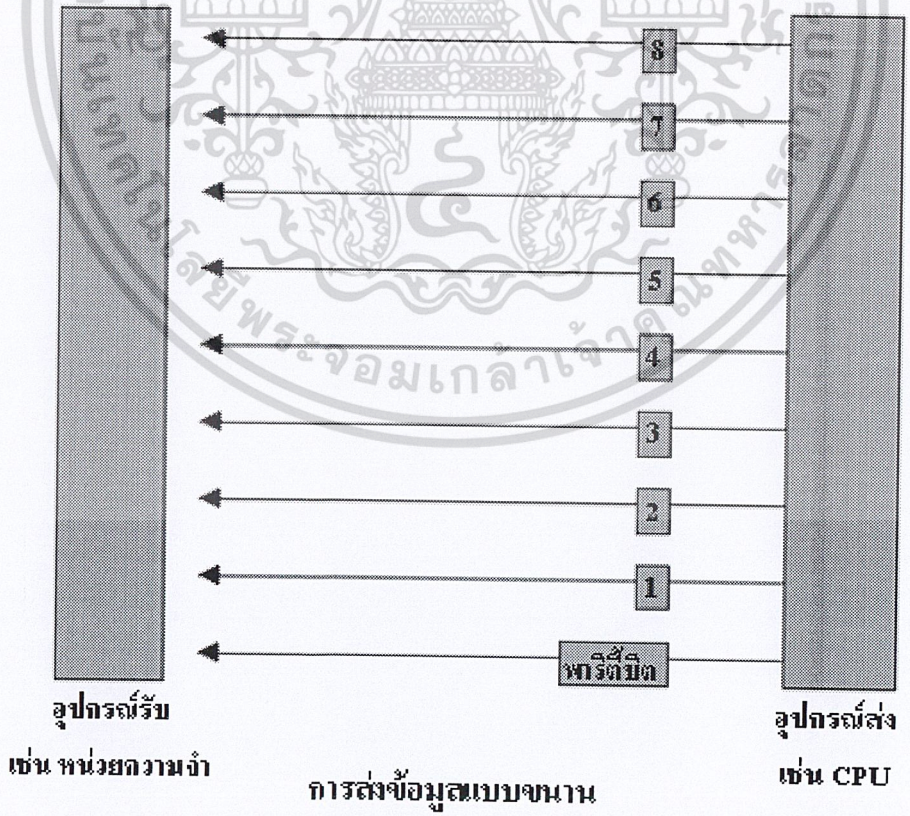
RS-232 มีจุดเริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่าง คอมพิวเตอร์กับโมเด็มในสมัยนั้น ตัวมาตรฐานจะกำหนดสิ่งที่เกี่ยวข้องกับการเชื่อมต่อนี้ด้วยกันทั้งหมด 4 หัวข้อหลักๆ ด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คุณสมบัติทางไฟฟ้าของสัญญาณ
2. คุณสมบัติทางกลของการเชื่อมต่อ ซึ่งหมายถึงตัวคอนเน็กเตอร์นั่นเอง
3. หน้าที่การทำงานของวงจรสำหรับแลกเปลี่ยนข้อมูล
4. มาตรฐานการเชื่อมต่อสำหรับระบบสื่อสารเฉพาะอย่าง

3.1.3 พอร์ตขนาน

พอร์ตขนานหรือ Parallel Port เป็นพอร์ตที่สามารถส่ง/รับข้อมูลได้ครั้งละหลายๆบิต และเป็นพอร์ตมาตรฐานที่มากับเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) ประโยชน์ของพอร์ตขนานนอกจากใช้สำหรับเชื่อมต่อกับเครื่องพิมพ์แล้ว ยังเป็นช่องทางเชื่อมต่ออีกทางเลือกหนึ่งสำหรับการเชื่อมต่อเครื่อง PC กับอุปกรณ์ภายนอก โดยที่ไม่จำเป็นที่จะต้องมีความรู้ทางด้านฮาร์ดแวร์มากนัก เพราะเราอาศัยเครื่องคอมพิวเตอร์เป็นส่วนประมวลผล การเขียนโปรแกรมจึงเหมือนกับการเขียนโปรแกรมทั่วไป และใช้พอร์ตขนานเป็นส่วนเชื่อมต่อกับวงจรที่เป็นส่วนนำเข้า (input) หรือส่วนส่งออก (output) หน้าที่ที่เหลือก็เป็นเพียงการเขียนโปรแกรมเพิ่มเติมเพื่อทำการรับและส่งข้อมูลกับพอร์ตขนานเท่านั้น

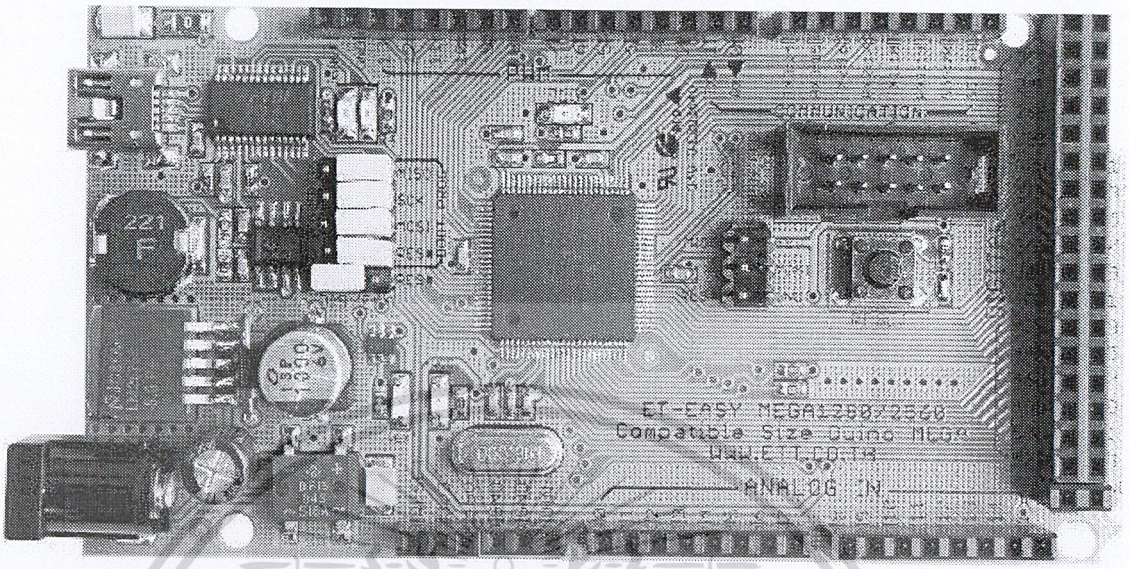


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 3.14 การสื่อสารข้อมูลผ่านพอร์ตขนาน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 ไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 (ARDIUNO MEGA)



รูปที่ 3.15 บอร์ด ET-EASY MEGA1280 (ARDIUNO MEGA)

3.1.4.1 คุณสมบัติของ ET-EASY MEGA1280 (ARDIUNO MEGA)

• เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ประสิทธิภาพสูงแต่ใช้พลังงานต่ำ ในตระกูล AVR®

• สถาปัตยกรรมแบบ RISC

– มีชุดคำสั่ง 135 คำสั่ง และส่วนใหญ่คำสั่งเหล่านี้จะใช้เพียง 1 สัญญาณนาฬิกาในการประมวลผลคำสั่ง

– มีเรจิสเตอร์สำหรับใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว

• หน่วยความจำ

– หน่วยความจำแฟลชสำหรับโปรแกรมขนาด 128 กิโลไบต์ เขียน/ลบได้ 10,000 ครั้ง

– หน่วยความจำแบบ EEPROM ขนาด 4 กิโลไบต์ เขียน/ลบได้ 100,000 ครั้ง

– หน่วยความจำแรมชนิดเอสแรม (SRAM) ขนาด 8 กิโลไบต์

• มีระบบโปรแกรมตัวเองอยู่ในตัวชิพ

• สามารถทำการอ่านขณะเขียนได้จริง โดยสามารถล็อกการทำงานได้เพื่อความปลอดภัยของซอฟต์แวร์

• มีการเชื่อมประสานกับ JTAG (IEEE std. 1149.1 compliant)

• คุณสมบัติการเชื่อมต่อกับอุปกรณ์ภายนอก

– มีตัวตั้งเวลาและตัวนับขนาด 8 บิต จำนวน 2 ตัว ที่สามารถแยกโหมดการทำงานจากกันได้ 2 โหมด คือ Prescalar และ Compare

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

– มีตัวตั้งเวลาและตัวนับขนาด 16 บิต จำนวน 4 ตัว ที่แยกโหมดการทำงานได้ 3 โหมด คือ Prescaler, Compare- และ Capture

– มีตัวนับแบบเวลาจริง (Real Time Counter) ที่แยกวงจรกำหนดความถี่ได้

– มี PWM จำนวน 12 ช่องสัญญาณที่สามารถกำหนดความละเอียดได้ 16 บิต

– มีตัวปรับผลการเปรียบเทียบของเอาต์พุต

– มีตัวแปลงสัญญาณอนาล็อกให้เป็นดิจิตอลขนาด 10 บิต จำนวน 16 ช่องสัญญาณ

– มีพอร์ตสื่อสารอนุกรมที่สามรารถกำหนดอัตราการรับ/ส่งได้จำนวน 4 พอร์ต

– เชื่อมประสานอนุกรมแบบ SPI ได้ทั้งการเป็นมาสเตอร์และสเลฟ (Master/Slave)

– มีการเชื่อมประสานแบบอนุกรมด้วยสายสัญญาณ 2 เส้นแบบ ส่งข้อมูลแบบเรียงไบต์ (Byte Oriented)

– มีตัวตั้งเวลาแบบวอตช์ด็อกที่สามารถกำหนดการทำงานได้โดยสามารถแยกสัญญาณนาฬิกาได้ จากตัวชิพ

– มีตัวเปรียบเทียบสัญญาณแบบอนาล็อกอยู่ในตัว

– มีการรองรับการขัดจังหวะและการเวก-อัพ (Wake-up) เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับขาของชิพ

• คุณสมบัติพิเศษ

– มีระบบเริ่มระบบเมื่อมีการรีเซ็ตและมีระบบตรวจจับการเกิด บราวน์เอาท์ (Brown-out) ที่สามารถ กำหนดการทำงานได้

– มีตัวตรวจหาความเที่ยงตรงของออสซิเลเตอร์อยู่ในตัว (Internal Calibrated Oscillator)

– มีแหล่งการขัดจังหวะทั้งภายในและภายนอก (External and Internal Interrupt Sources)

– มีโหมดการทำงานสลีป 6 แบบ คือ Idle, ADC Noise Reduction, Power-save, Power-down, Standby, และ Extended Standby

• การใช้พลังงาน

– โหมดการทำงาน: ที่ 1 MHz ต้องการแรงดัน 1.8V กระแส 500μA

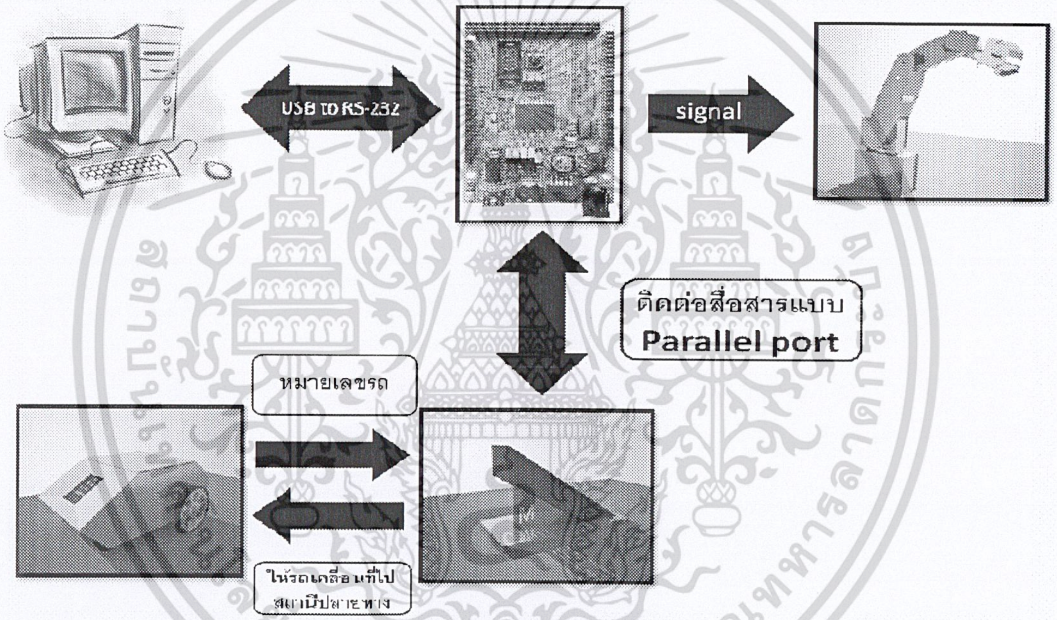
– โหมดเพาเวอร์ดาวน์ (Power-down) ต้องการกระแสเพียง 0.1 μA ที่แรงดัน 1.8V

3.1.4.2 หลักการทำงาน ส่วนคอนโทรลเลอร์หลัก

Board Et-easy mega128 นั้นนอกจากจะทำหน้าที่เป็นเหมือนกลไกหลักในการตัดสินใจการกระทำต่างๆของระบบแล้ว ตัว Controller ยังทำหน้าที่เป็นตัวเชื่อมสัญญาณต่างๆของระบบเข้าด้วยกันอีกด้วย เราสามารถยกตัวอย่างการทำงานเช่น เมื่อผู้ใช้งานต้องการตรวจสอบหมายเลขของรถที่เข้ามาจอดแต่ละสถานี การทำงานจะเริ่มเมื่อผู้ใช้กดคำสั่งตรวจสอบหมายเลขรถ สถานีแต่ละสถานีจะส่งสัญญาณมาทาง USB to Serial Port เพื่อแปลงสัญญาณจาก RS-232 เข้า USB Port

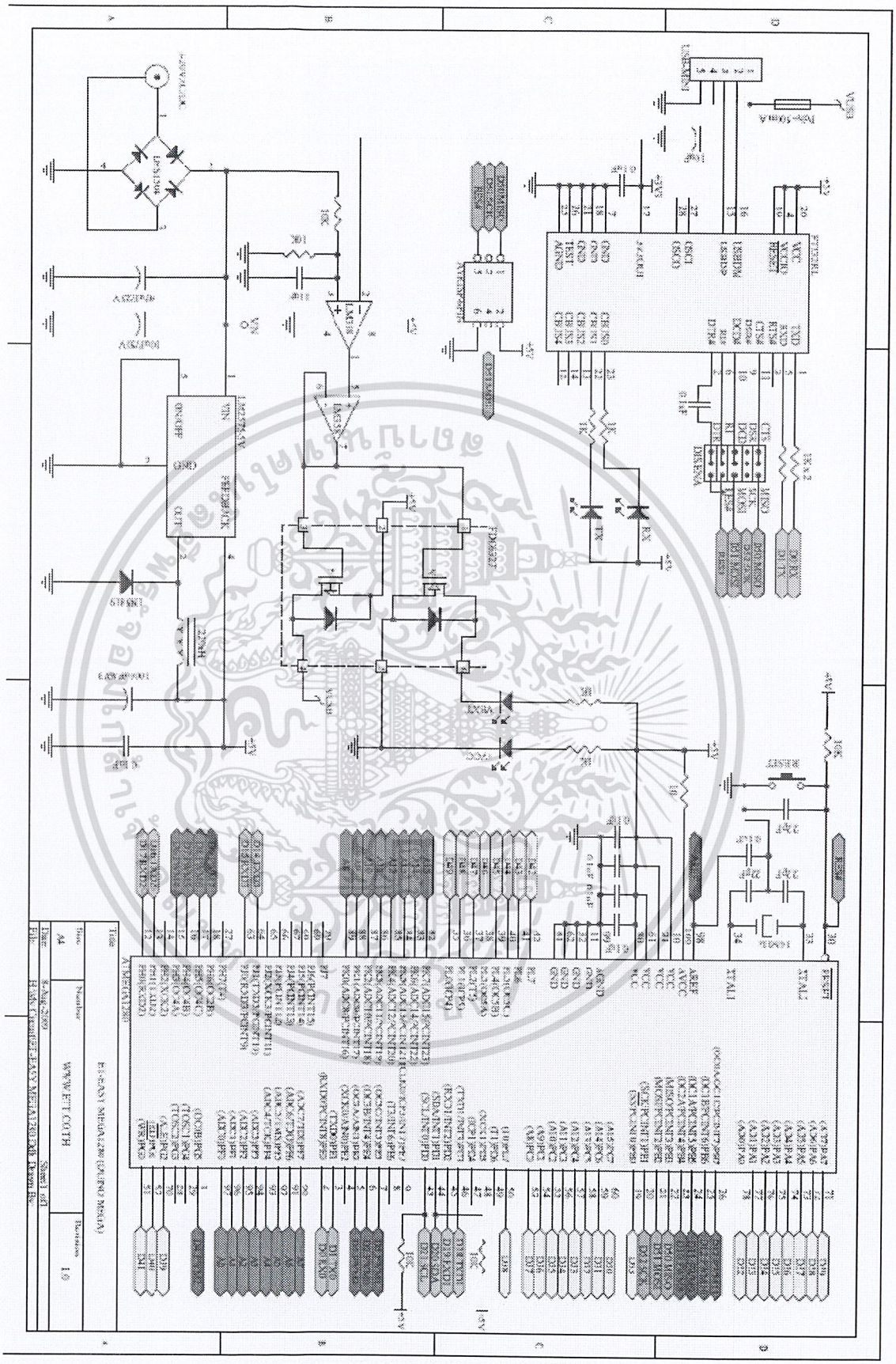
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นบอร์ดจะติดต่อสื่อสารเข้าไปที่สถานี โดยส่งสัญญาณผ่านทาง Parallel Port เพื่อให้สถานีส่งหมายเลขรถที่จอดอยู่กลับมา เมื่อตัว Controller ได้หมายเลขรถมาแล้วจะทำการส่งค่ากลับไปหาผู้ใช้งานเพื่อให้ผู้ใช้ทราบถึงตำแหน่งรถแต่ละคัน และเมื่อมีคำสั่งมาจากผู้ใช้ว่าต้องการให้รถเคลื่อนที่ไปยังสถานีปลายทางไหน ตัว Controller จะรับสัญญาณจากการที่ผู้ใช้กดปุ่มสั่งการ จากนั้นจะทำการส่งสัญญาณไปสั่งที่สถานีเพื่อให้สถานีส่งสัญญาณไปสั่งที่ตัวรถอีกทีหนึ่งให้รถเคลื่อนที่ไปสถานีปลายทางตามที่ได้รับสัญญาณที่ผู้ใช้สั่งมานอกจากนั้นตัว Controller เรายังทำหน้าที่ติดต่อสื่อสารกับตัว Controller ของส่วนแขนกลเพื่อควบคุมการทำงานของแขนกลเมื่อรถเคลื่อนที่เข้ามาจอดอีกด้วย โดยวงจรภายในตัวบอร์ดเป็นดังรูปที่3.16 และหลักการทำงานของส่วนคอนโทรลเลอร์หลัก



รูปที่ 3.16 แผนภาพแสดงการทำงานของส่วน Main Controller

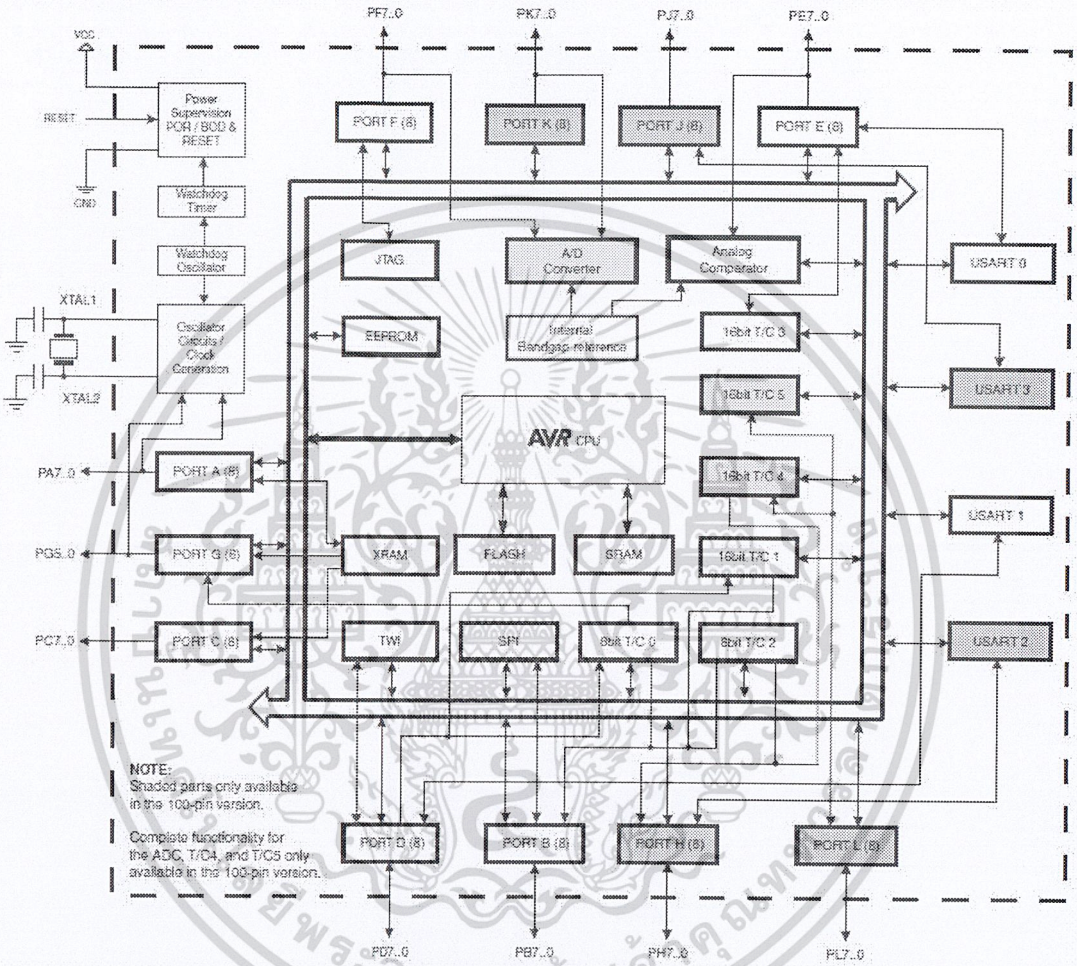
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 วงจรภายในตัวบอร์ด Et-easy mega128

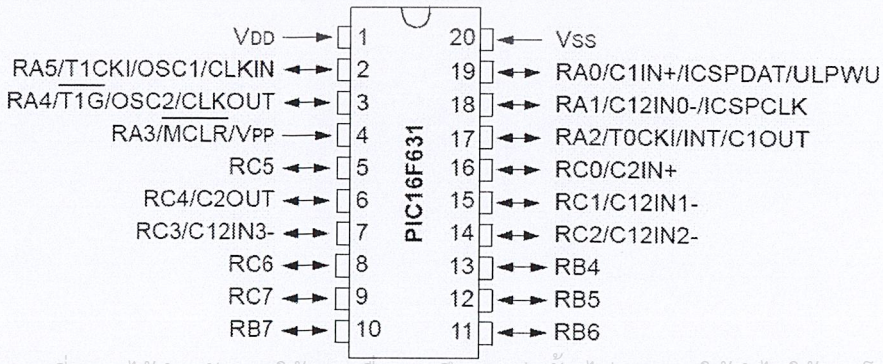
3.1.4.3 สาเหตุการเลือกใช้ Easy Mega1280

- เป็นบอร์ดที่มี พอร์ต ดิจิตอลและ อนาลอกเป็นจำนวนมากเหมาะสำหรับระบบใหญ่
- ความสะดวกในการใช้งานเพราะเราสามารถเชื่อมต่อตัวบอร์ดเข้ากับเครื่องคอมพิวเตอร์โดยผ่านทางสาย USB



รูปที่ 3.18 สถาปัตยกรรมภายในของ AVR

3.1.5 PIC16F690



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 3.19 ลักษณะของ PIC16F690

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

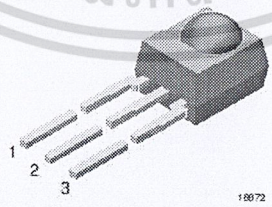
เป็นไมโครคอนโทรลเลอร์ที่นิยมใช้โดยทั่วไปพื้นฐานการทำงานก็คือ ระบบดิจิตอลโดยค่าเอาท์พุตที่ได้จากไมโครคอนโทรลเลอร์ จะเป็น 0 กับ 1 แต่ก็สามารถนำมาประยุกต์เชื่อมต่อกับอุปกรณ์ภายนอกได้มากมาย

ลักษณะโดยทั่วไปจากรูปที่ 3.19 จากรูปขาของ PIC16F690 แต่ละขาจะมีหน้าที่แตกต่างกันไป ซึ่งแยกออกเป็น PORT A , PORT B , PORT C โดยพื้นฐานแต่ละพอร์ตสามารถทำงานเป็นอินพุตและเอาท์พุต สำหรับไมโครคอนโทรลเลอร์ที่ใช้ต้องมีสัญญาณนาฬิกาภายในตัวเพื่อสร้างพัลซ์มาป้อนให้กับมอเตอร์เพื่อที่จะคุมให้มอเตอร์ทำงานตามที่เราต้องการซึ่งใน PIC16F690 มีสัญญาณนาฬิกาภายในตัวยู่แล้วโดยที่หนึ่งไซเคิล(Clock Bus) ของซีพียูจะประกอบไปด้วย สัญญาณนาฬิกาภายนอกจำนวน 4 ไซเคิล คือ Q1,Q2,Q3 และ Q4 ดังนั้นคามถี่ซีพียูประมวลผลต่อหนึ่งคำสั่งเท่ากับความถี่ของสัญญาณนาฬิกาภายนอกหารด้วย 4 หรือหากจะพิจารณาคามเร็วของไมโครคอนโทรลเลอร์ตระกูล PIC สามารถประมวลผลต่อหนึ่งคำสั่งเท่ากับ 1/4 เท่าของความถี่ออสซิเลเตอร์ภายนอก

3.1.6 TSOP34836

3.1.6.1 คุณสมบัติ

เป็นอุปกรณ์ขนาดเล็กที่ใช้ในการรับสัญญาณอินฟราเรดโดยข้อมูลจะถูกส่งผ่านหลอดอินฟราเรด ซึ่งการส่งข้อมูลผ่านอินฟราเรดเราต้องทำการเลือกความถี่ช่วงสัญญาณ เพื่อที่จะทำการเลือกตัวรับได้ว่า จะให้รับสัญญาณอินฟราเรดในช่วงใด ซึ่ง TSOP34836 จะสามารถรับสัญญาณอินฟราเรดได้ในช่วง 38KHz โดยลักษณะของ TSOP34836 เป็นดังรูปที่ 3.20



รูปที่ 3.20 ลักษณะของ TSOP34836

รังสีอินฟราเรด (อังกฤษ: Infrared (IR)) มีชื่อเรียกอีกชื่อว่า รังสีใต้แดง หรือรังสีความร้อน เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นอยู่ระหว่างคลื่นวิทยุและ แสงมีความถี่ในช่วง 1011 – 1014 เฮิรตซ์ มีความถี่ในช่วงเดียวกับไมโครเวฟ มีความยาวคลื่นอยู่ระหว่างแสงสีแดงกับคลื่นวิทยุสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

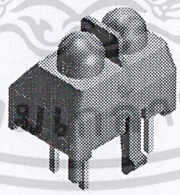
ทุกชนิดที่มีอุณหภูมิ อยู่ระหว่าง -200 องศาเซลเซียสถึง 4,000 องศาเซลเซียส จะปล่อยรังสีอินฟราเรดออกมา คุณสมบัติเฉพาะตัวของรังสีอินฟราเรด เช่น ไม่เบี่ยงเบนในสนามแม่เหล็กไฟฟ้าที่แตกต่างกันก็คือ คุณสมบัติที่ขึ้นอยู่กับความถี่ คือยิ่งความถี่สูงมากขึ้น พลังงานก็สูงขึ้นด้วย ดังนั้นในการใช้ประโยชน์ ใช้ในการควบคุมเครื่องใช้ระบบไกล (remote control) สร้างกล้องอินฟราเรดที่สามารถมองเห็นวัตถุในความมืดได้ เช่น อเมริกาสามารถใช้กล้องอินฟราเรดมองเห็นเวียตกงได้ตั้งแต่สมัยสงครามเวียดนาม และสัตว์หลายชนิดมีนัยน์ตารับรู้รังสีชนิดนี้ได้ ทำให้มองเห็นหรือล่าเหยื่อได้ในเวลากลางคืน

3.1.6.2 สาเหตุการเลือกใช้ TSOP34836

- ความต้องการกระแสต่ำ
- มีความม่นยำในการส่งข้อมูลสูง
- ราคาถูก

3.1.7 อินฟราเรดเซนเซอร์

โต้ทรานซิสเตอร์และแอลอีดีอินฟราเรดเป็นวงจรตรวจจับระยะ โดยมีลักษณะของวงจรคล้ายกับวงจรเซนเซอร์แสง ซึ่งโฟโต้ทรานซิสเตอร์จะทำหน้าที่คล้ายกับเป็นอุปกรณ์แอลดีอาร์ และเพิ่มแอลอีดีอินฟราเรดในการส่องแสงอินฟราเรดเพื่อสะท้อนวัตถุกลับมายังโฟโต้ทรานซิสเตอร์ ในการตรวจสอบระยะนั้นจะใช้หลักการทำงานของการสะท้อนแสงอินฟราเรด กับวัตถุ ซึ่งความเข้มแข็งของอินฟราเรดจะมีพลังงานสูงและสื่อได้ไกลกว่าแสงธรรมดาเนื่องจากมีแสงรบกวนต่ำ



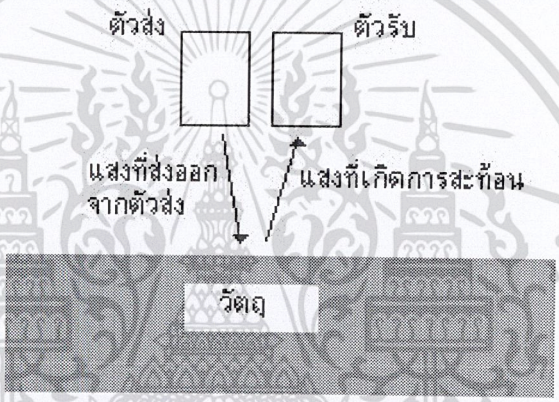
รูปที่ 3.21 อินฟราเรดเซนเซอร์

เมื่อจ่ายไฟให้วงจร LED จะส่งแสงอินฟราเรดในรัศมีที่กว้าง ถ้ามีวัตถุอยู่ด้านหน้าโฟโต้ทรานซิสเตอร์ จะได้รับแสงสะท้อนกลับมา ทำให้สามารถนำกระแสได้ เกิดแรงดันตกคร่อมขึ้นที่ขา S (Vout) โดยค่าแรงดันจะมาหรือน้อย ก็ขึ้นอยู่กับความเข้มของแสงอินฟราเรดที่ตกกระทบ โฟโต้ทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.8 การทำงานของวงจรเซ็นเซอร์แสงที่ใช้ในการตรวจจับเส้น

เมื่อทำการจ่ายไฟให้กลับวงจรเซ็นเซอร์แสงแล้ว ทำให้ตัวส่งสัญญาณอินฟาเรดที่เป็นตัวส่งจะเปล่งแสงออกไปตกกระทบกับพื้น เพื่อทำการวัดค่าของแสง เมื่อพื้นที่ตัวส่งสัญญาณอินฟาเรดเปล่งแสงออกไป มันจะสะท้อนแสงนั้นกลับมายังโฟโต้ไดโอดซึ่งเป็นตัวรับสัญญาณโดยที่ขาข้างหนึ่งของโฟโต้ไดโอดจะไปต่อยังขาอินพุตบวกของไอซี



รูปที่ 3.22 การสะท้อนของแสงอินฟราเรดเมื่อกระทบกับวัตถุ

เปรียบเทียบแรงดันเบอร์ LM339 เมื่อโฟโต้ไดโอดสามารถตรวจจับสัญญาณที่สะท้อนขึ้นมาจากพื้นได้ มันจะทำงานเหมือนกับเป็นทรานซิสเตอร์แสงตัวหนึ่ง เมื่อมีแสงมาตกกระทบก็จะมีเหมือนกับการที่มีกระแสไฟฟ้าไหลผ่านที่ขาเบส แล้วจะยอมให้กระแสไฟฟ้าไหลผ่านขาคอลเลกเตอร์ไปยังขาอีมิเตอร์เป็นการไบอัสแรงดันเข้าไปยังไอซี ซึ่งต่อขนานอยู่กับตัวทานต้าน 10 kΩ ได้เพื่อทำการเปรียบเทียบแรงดันที่ขาอินพุตบวกกับแรงดันที่ขาอินพุตลบของไอซีที่ต่ออยู่กับตัวต้านทานปรับค่าได้ 10 kΩ โดยปรับค่าของตัวต้านทานปรับค่าให้แรงดันน้อยกว่าแรงดันที่ขาอินพุตบวก ในขณะที่ทำการไบอัส

เมื่ออินฟาเรดเปล่งแสงออกไปตกกระทบกับพื้นสีขาว มันจะสะท้อนแสงกลับเข้ามาที่โฟโต้ไดโอดทำให้เกิดการไบอัสแรงดันขึ้นที่ขาอินพุตบวกของไอซี ซึ่งจะทำให้แรงดันที่ขาอินพุตบวกของไอซีจะมากกว่าแรงดันที่ขาอินพุตลบ จะทำให้ไอซีส่งสัญญาณเป็นลอจิก ''1'' หรือแรงดันไฟฟ้าขนาด 5 โวลต์ไปยังไมโครคอนโทรลเลอร์ แต่เมื่ออินฟาเรดเปล่งแสงออกไปตก

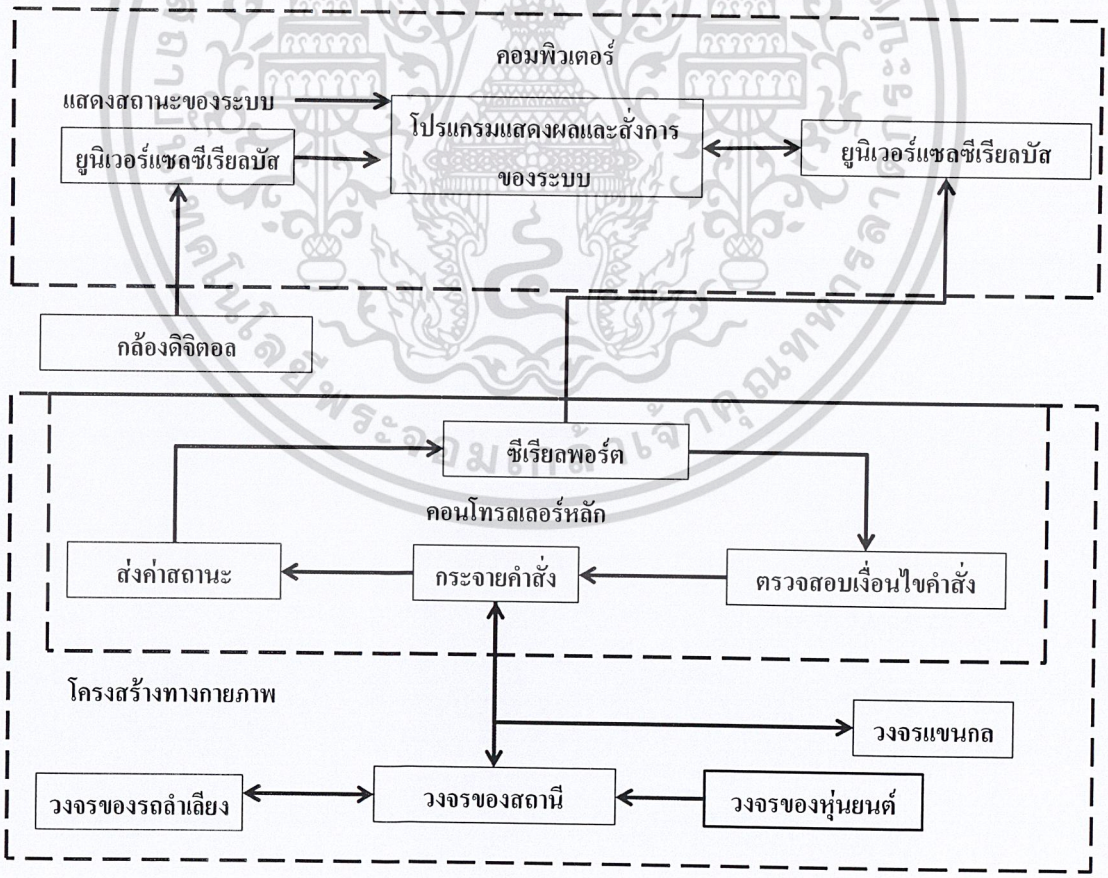
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระทบกับพื้นสีดำ แสงจะไม่สามารถที่จะสะท้อนกลับมาได้ ทำให้โฟโต้ไดโอดไม่เกิดการไบอัสแรงดัน ส่งผลทำให้แรงดันที่ขาอินพุตบวกของไอซีน้อยกว่าแรงดันที่ขาอินพุตลบ ทำให้ไอซีส่งลอจิก ''0'' หรือ แรงดัน 0 โวลต์ไปที่ไมโครคอนโทรลเลอร์ โดยไมโครคอนโทรลเลอร์จะรับสัญญาณลอจิกจากวงจรเซ็นเซอร์แสงทุกตัว แล้วทำการคำนวณประมวลผลจากการส่งสัญญาณลอจิกจากตัวเปรียบเทียบแรงดัน

3.2 การออกแบบจำลองระบบการผลิตผสมผสานด้วยคอมพิวเตอร์

การออกแบบระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์ ในเบื้องต้นนั้นต้องเข้าใจหลักการทำงานในภาพรวมของระบบกระบวนการ แล้วจึงพิจารณาส่วนย่อยต่างๆอันประกอบไปด้วย มอนิเตอร์แสดงผลและสั่งการ ส่วนลำเลียง แขนกล หุ่นยนต์ 2 ขาและหุ่นยนต์แมลง

หลักการทำงานในภาพรวมของระบบจำลองการผลิผสมผสานด้วยคอมพิวเตอร์ สามารถแสดงดังแผนผัง ในรูปที่ 3.23



รูปที่ 3.23 แผนผังรวมของระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.23 เห็นว่าระบบจำลองการผลิตอัตโนมัติประกอบด้วยส่วนหลักๆ ดังนี้คือ

- กล้องดิจิตอลวีดีโอ
- คอมพิวเตอร์
- โครงสร้างทางกายภาพของระบบ
- วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง

3.2.1 กล้องดิจิตอลวีดีโอ

กล้องดิจิตอลวีดีโอเป็นอุปกรณ์ที่สำคัญสำหรับโครงงานนี้ เนื่องจากเป็นอุปกรณ์ที่ใช้สำหรับตรวจสอบสถานะของรถในระบบว่าตรงกับการแสดงผลของข้อมูล

สิ่งจำเป็นสำหรับการใช้งานกล้องดิจิตอลวีดีโอ

- เครื่องไมโครคอมพิวเตอร์
- โปรแกรมระบบปฏิบัติการไมโครซอฟท์วินโดว์
- พอร์ตยูนิเวอร์แซลซีเรียลบัส
- โปรแกรมวิชวลเบสิก
- ต้องการไฟเลี้ยงจากคอมพิวเตอร์ 2 วัตต์
- ปรับโฟกัสได้ตั้งแต่ 1 นิ้ว ถึงอนันต์

3.2.2 คอมพิวเตอร์

คอมพิวเตอร์เป็นอุปกรณ์ที่ทำหน้าที่รับข้อมูลจากกล้องดิจิตอลวีดีโอ ผ่านทางยูนิเวอร์แซลซีเรียลบัส เพื่อแสดงภาพที่ได้จากหน้างานจริง แล้วนำมาเปรียบเทียบกับข้อมูลที่ไดรับมาจากคอนโทรลเลอร์ที่ทำหน้าที่เก็บรวบรวมข้อมูล และแสดงภาพจากกล้องดิจิตอลแล้ว ยังทำการแสดงผลข้อมูลของระบบจำลองที่คอนโทรลเลอร์หลักส่งค่าข้อมูลกับมาด้วย และยังทำหน้าที่สั่งการผลข้อมูลที่ได้กระบวนการทั้งหมดด้วยโดยให้คอนโทรลเลอร์หลักเป็นตัวกระจายคำสั่งที่ได้รับมาไปยังแต่ละหน่วยการทำงาน ทั้งสถานีและแขนกล

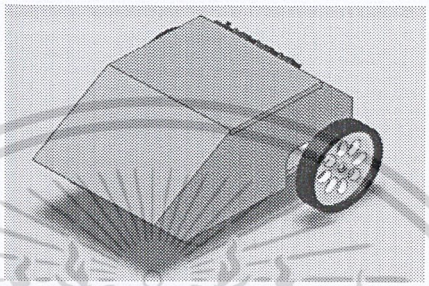
เนื่องจากการควบคุมในโครงงานนี้ใช้คอมพิวเตอร์เป็นอุปกรณ์ประมวลผลหลัก ดังนั้นการเขียนโปรแกรมจึงเป็นการดำเนินการบนระบบเวลาดีสครีต แทนการควบคุมแบบระบบเวลาต่อเนื่อง ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องมีความรัดกุม และทำงานได้รวดเร็ว ซึ่งคอมพิวเตอร์ที่ใช้จะต้องมีความเร็วพอที่จะทำให้คาบการสุ่มสัญญาณมีค่าน้อยเพียงพอ จนไม่มีผลต่อการควบคุมระบบทั้งหมดด้วย สำหรับโครงงานนี้เลือกใช้การเขียนโปรแกรมโดยภาษาวิชวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3โครงสร้างทางกายภาพของระบบ

3.2.3.1 ส่วนรถลำเลียง

ในส่วนของรถลำเลียงนั้นนอกจากจะมีวงจรขับมอเตอร์ (Drive motor) เพื่อขับให้รถวิ่ง และ วงจรตรวจจับ (Sensor) เพื่อสั่งให้รถหยุดแล้ว ส่วนรถลำเลียงยังมีวงจร รับและส่งสัญญาณอินฟราเรด เพิ่มเติมมาอีกหนึ่งวงจรด้วย โดยลักษณะรถลำเลียงเป็นดังรูป 3.24



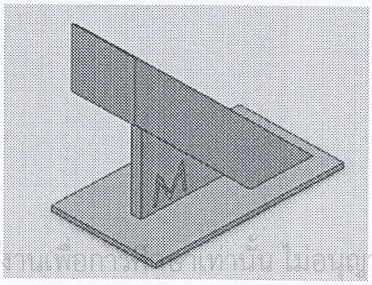
รูปที่ 3.24 ลักษณะของรถลำเลียง

หลักการการทำงาน ส่วนรถลำเลียง

หลักการทำงานของส่วนรถลำเลียงนั้น เราใช้หลักการการส่งข้อมูลผ่านทางสัญญาณอินฟราเรด โดยการทำงานจะเริ่มเมื่อ ตัวรถ นั้นเคลื่อนที่เข้ามาจอดที่สถานี จากนั้น วงจรรับ-ส่ง จะเริ่มทำงาน โดยสิ่งที่ส่งไปจะเป็นหมายเลขประจำของตัวรถ ซึ่งส่วนที่ทำหน้าที่เป็นตัวรับสัญญาณที่ส่งไปคือ ส่วนสถานี นั่นเอง

3.2.3.2 ส่วนสถานี

ในส่วนของสถานีนั้น นอกจากจะมีวงจร รับและส่งสัญญาณอินฟราเรด ซึ่งจะเหมือนกับส่วนของรถลำเลียงแล้ว ยังมีวงจรควบคุมการทำงนของสถานี อีก1วงจรด้วย โดยลักษณะสถานีเป็นดังรูป 3.25



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

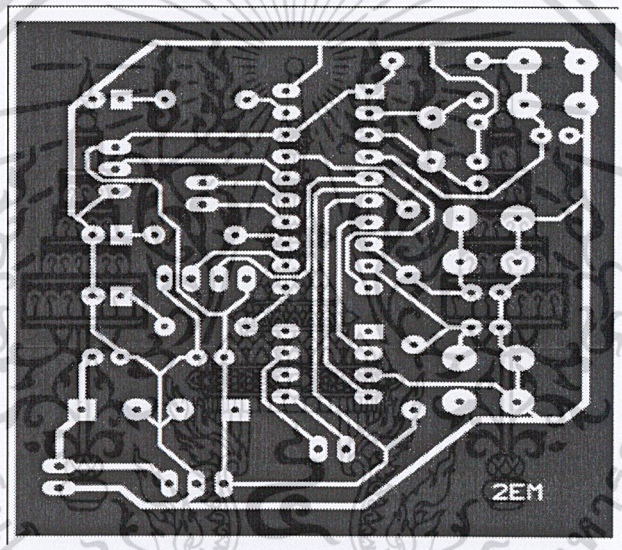
รูปที่ 3.25 ลักษณะของ Station

หลักการทำงาน ส่วน สถานี

หลักการทำงานของส่วนสถานีนั้น เราใช้หลักการการส่งข้อมูลผ่านทางสัญญาณอินฟราเรด เช่นเดียวกับหลักการของส่วนรถลำเลียง โดยการทำงานจะเริ่มเมื่อรถเคลื่อนที่เข้ามาจอด และส่วนของสถานีได้รับสัญญาณอินฟราเรดที่เป็นหมายเลขของรถแล้วจากนั้นสถานีจะทำการส่งหมายเลขของรถที่ได้รับมานั้นไปยังส่วนคอนโทรลเลอร์หลักเพื่อทำการประมวลผลและตัดสินใจ

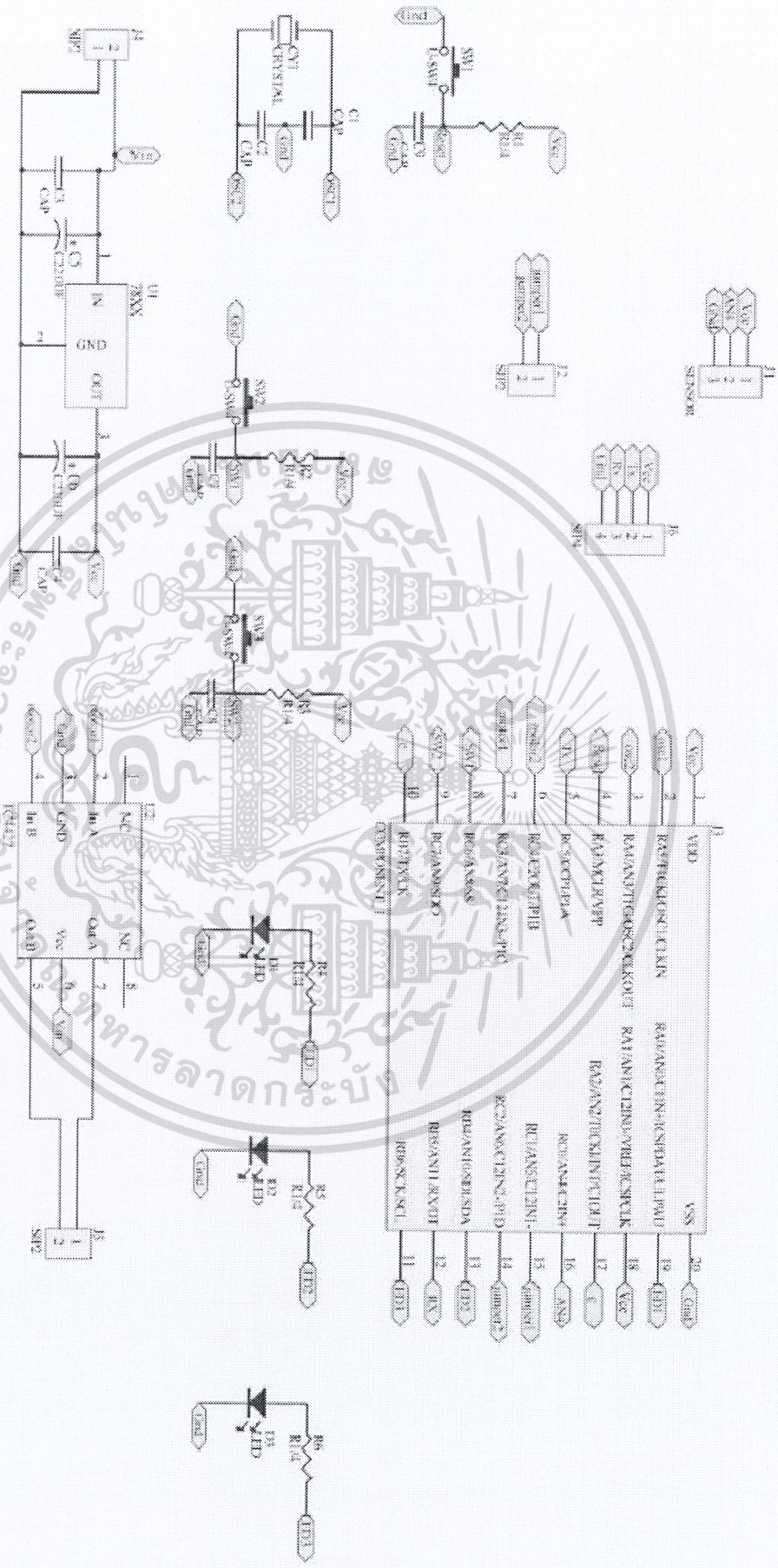
3.2.4วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง

โดยวงจรภายในส่วนของรถลำเลียงเป็นดังรูปที่ 3.26 และ ลายวงจรของส่วนรถลำเลียง เป็นดังรูปที่ 3.27



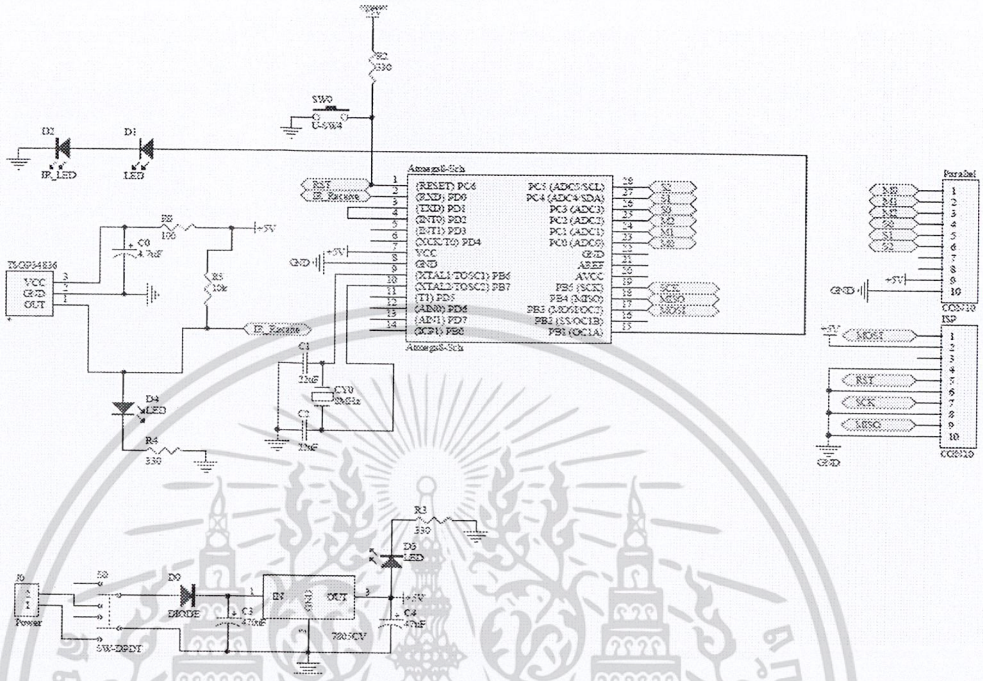
รูปที่ 3.26 ลายวงจรของส่วนรถลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 วงจรภายในส่วนของVehicle

โดยวงจรภายในส่วนของสถานี เป็นดังรูปที่ 3.28 และ ภาพวงจรของส่วนสถานี เป็นดังรูปที่ 3.29



รูปที่ 3.28 วงจรภายในของส่วนสถานี

รูปที่ 3.29 ภาพวงจรของส่วนสถานี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

ในบทนี้ จะกล่าวถึงการทดลองและผลการทดลอง ของโปรแกรม และการทำงานของระบบลำเลียง เมื่อมีการสั่งการจากโปรแกรม โดยมีรายละเอียดของการทดลองดังนี้

4.1 การทดลองโปรแกรม และผลที่ได้จากการทำงาน

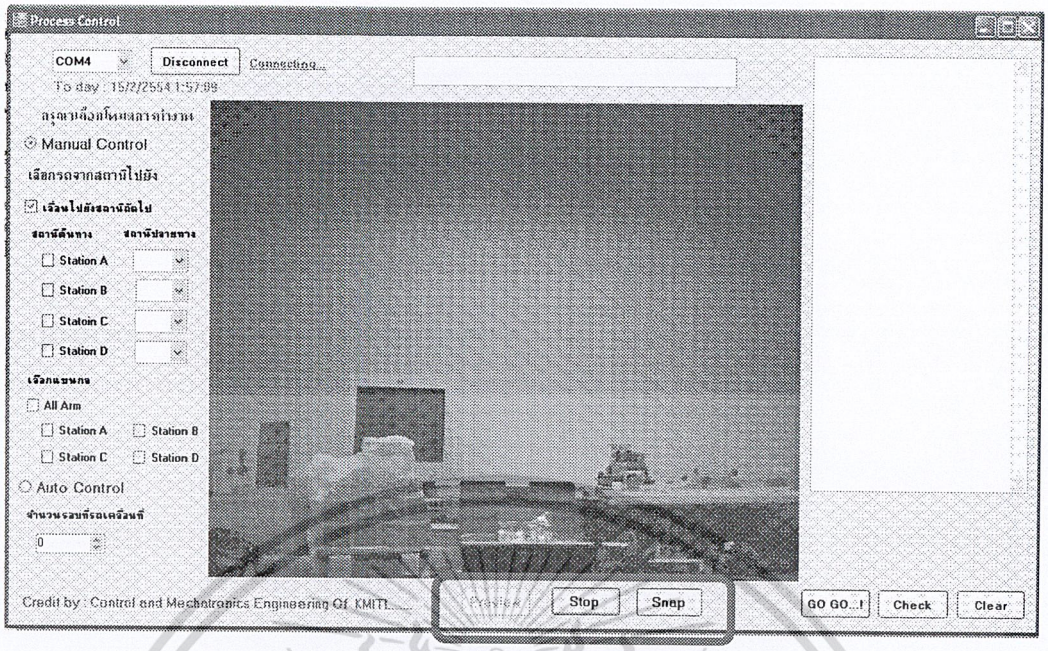
ในส่วนนี้ เป็นการศึกษาเฉพาะส่วนโปรแกรม Visual basic 2008 และ ผลที่ได้จากการสั่งจากโปรแกรม ซึ่งผลการทดลองมีดังนี้

หลังจากทำการรันโปรแกรมจาก Visual basic 2008

รูปที่ 4.1 การเลือก com port และการ เชื่อมต่อ

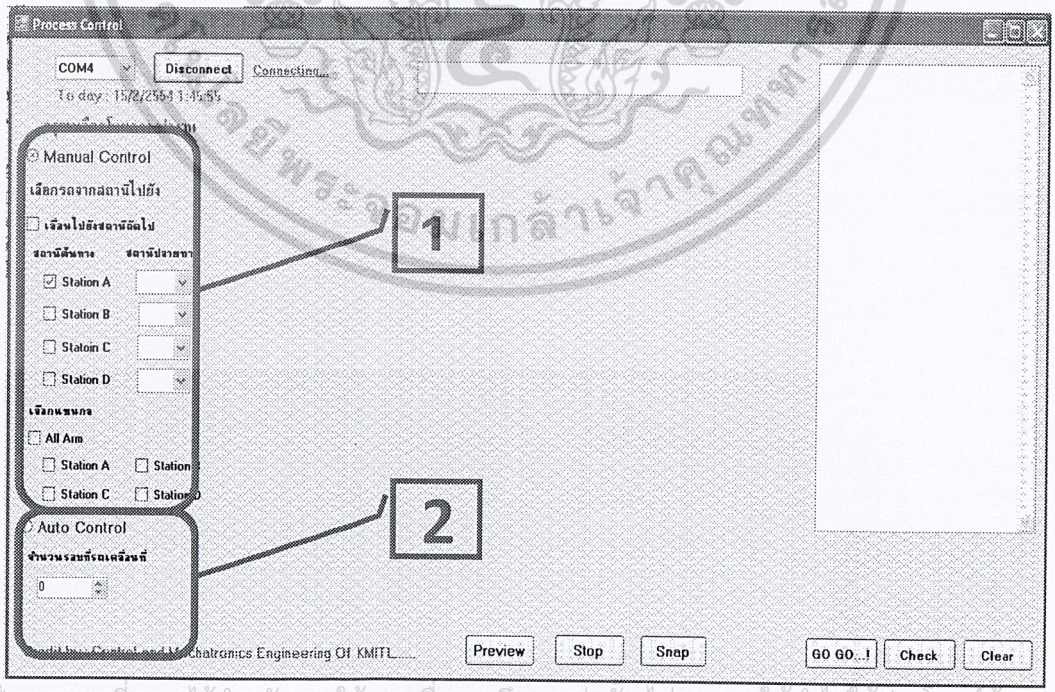
ทำการเลือก Com port และกดปุ่ม Connect เพื่อทำการเชื่อมต่อระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การจับภาพจากกล้อง Webcam

เราสามารถดูการทำงานจริงของระบบโดยผ่านทางกล้อง Webcam โดยกดปุ่ม Preview และ เราสามารถทำการ save รูปภาพในขณะนั้นได้โดยการกดปุ่ม Snap และทำการปิดกล้องได้โดยการกดปุ่ม Stop



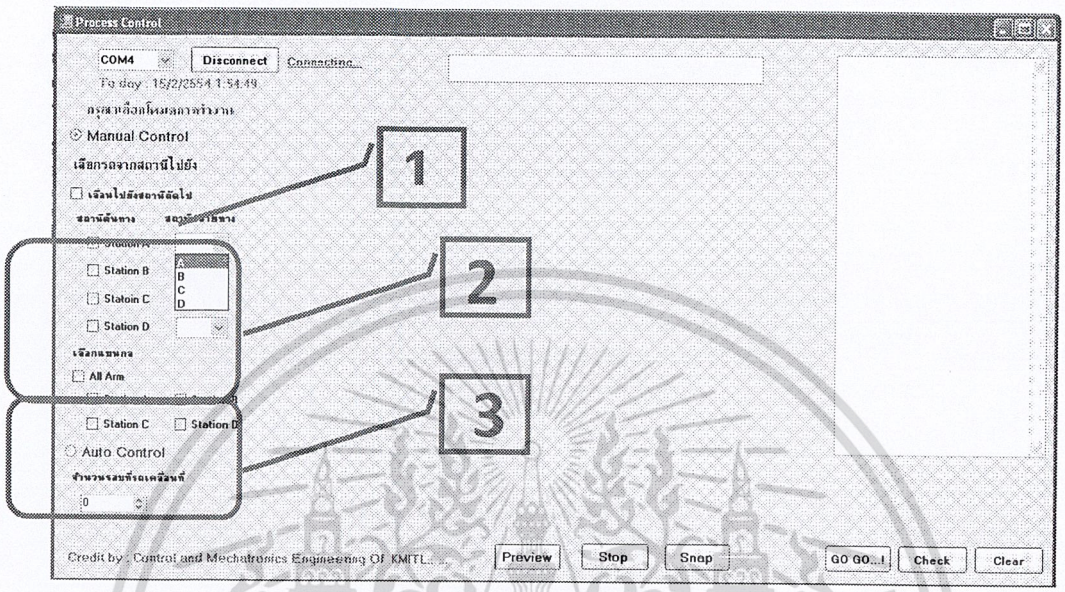
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 การเลือกโหมดการทำงาน

เมื่อเราเข้ามายังตัวโปรแกรมแล้วเราสามารถเลือกการทำงานของระบบลำเลียงได้ 2 แบบ คือ

1.แบบ Manual Control

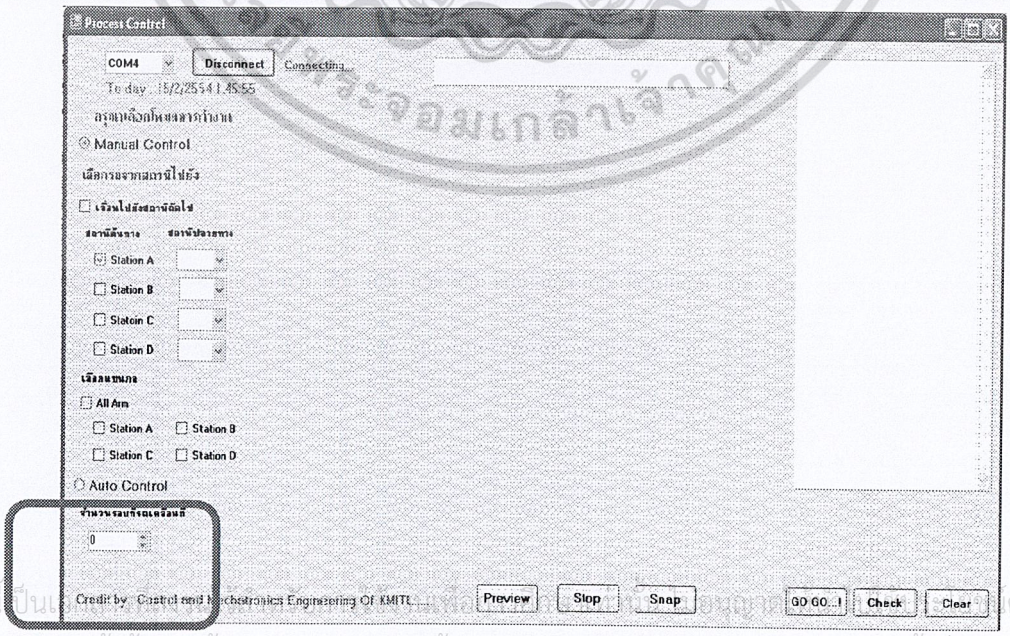
2.แบบAuto Control



รูปที่ 4.4 รายละเอียดในโหมดควบคุมด้วยมือ

ใน Mode Manual Control นั้น เราอาจแบบเป็น 3 ส่วนหลักๆ คือ

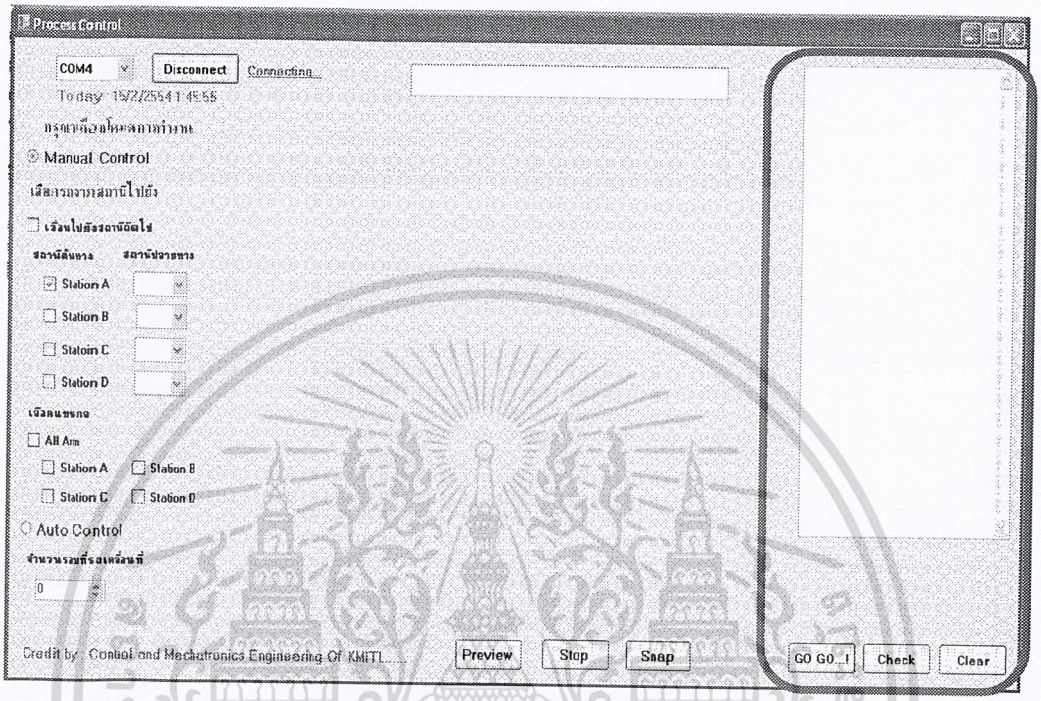
1. เป็นการสั่งการให้รถเคลื่อนที่ไปยังสถานีถัดไป โดยไม่มีการวิ่งข้ามสถานี
2. เป็นการสั่งการให้รถเคลื่อนที่ไปยังสถานีที่ต้องการได้
3. เป็นกานสั่งการให้แขนกลในสถานีต่างให้ทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 การเลือกโหมดควบคุมอัติโนมัติ

และใน Mode Auto Control นั้น เราสามารถกำหนดจำนวนรอบการทำงานได้ ว่าต้องการให้ระบบของเราทำงานได้กี่รอบ โดยเมื่อรถเคลื่อนที่มาถึงยังสถานี แขนกลจะทำการหยิบจับสิ่งของเองโดยอัตโนมัติ ไม่จำเป็นต้องรอคำสั่ง



รูปที่ 4.6 ส่วนแสดงผลสถานีการทำงาน

และเราสามารถตรวจสอบสถานะของการทำงาน ทั้งระบบได้ด้วยการกดปุ่ม Check ไม่ว่าจะเป็นการตรวจสอบว่าสถานีไหนมีรถคันใดจอดอยู่บ้าง และเราสามารถทำการลบข้อมูลได้โดยการกดปุ่ม Clear

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5

บทสรุป

5.1 สรุปผลของโครงงาน

จากการทดลองจำลองระบบกระบวนการผลิต เมื่อมีคำสั่งให้รถลำเลียงเคลื่อนที่จากสถานีหนึ่งไปยังสถานีหนึ่งหรือสถานีที่ต้องการ ผ่านทางคอมพิวเตอร์โดยสั่งได้ทั้งแบบ Manual หรือ Auto รถลำเลียงจะเคลื่อนที่ไปยังสถานีตามคำสั่งและจอดที่สถานี ซึ่งสามารถตรวจเช็คได้ว่ามีรถคันใดอยู่ที่สถานีใดบ้าง จากหน้าจอมอนิเตอร์ และเมื่อแขนกลได้รับคำสั่งจากตัวคอนโทรลเลอร์จะทำการหยิบจับสินค้า เพื่อนำสินค้าเข้า-ออกระหว่างรถลำเลียงกับหน่วยการทำงาน

5.2 ปัญหาและอุปสรรคและแนวทางการแก้ไข

1. จากการทำงานของแขนกล เมื่อรถเคลื่อนที่ทำให้ตำแหน่งสินค้าที่อยู่บนรถมีการขยับ ทำให้เมื่อแขนกลไม่สามารถหยิบจับสินค้าได้ ดังนั้นจึงต้องมีการใช้วัสดุกันลื่นติดไว้บนรถ ป้องการการขยับของสินค้า

2.จากการทำงานของเซนเซอร์ เมื่อรางรถสกปรกหรือมีสีดำ จะส่งผลให้เซนเซอร์จับเจอสีดำทำให้รถจอดไม่ตรงสถานีที่จะจอดจริง ดังนั้นต้องมีการแก้ไขส่วนที่มีสีดำ

3. จากการวิ่งของรถ บางครั้งทองแดงที่อยู่ใต้รถมีการงอ ไม่สัมผัสกับทองแดงที่ราง ทำให้ไม่มีไฟเข้าที่ตัวรถ ส่งผลให้รถหยุดการเคลื่อนที่ ดังนั้นจึงต้องมีการเปลี่ยนทองแดงใหม่และใส่พลาสติกกั้นระหว่างทองแดงที่ขั้วบวกและขั้วลบ เพื่อป้องกันการลัดวงจร

5.3 แนวทางการพัฒนา

แนวทางการพัฒนาเกี่ยวกับความสามารถในการทำงานของโครงงาน

1.พัฒนาในส่วนของเซนเซอร์ให้สามารถทำงานได้แม่นยำมากขึ้น เพื่อทำงานได้ในสถานที่จริงอย่างมีประสิทธิภาพที่สุด

2.พัฒนาปรับปรุงในส่วนของโปรแกรมให้รองรับกับสถานที่จริงที่มีความหลากหลายและกระบวนการของการผลิตแตกต่างกัน เพื่อที่จะทำงานได้อย่างถูกต้องมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรม Visual Basic 2008

```
Imports System
Imports System.Threading
Imports System.IO.Ports
```

```
Public Class Form1
    Dim str_input As Integer, c As Integer
    Dim b As String
    Dim st As String
    Dim intnum As Integer = 0, turnnum As Integer = 0 'ระยะเวลาที่นับถอยหลังในหนึ่งรอบ
    Dim i As Integer = 0
    '##############################  webcam ##############
    ' Create constant using attend in function of DLL file.
    ' กำหนดค่าคงที่ในการใช้ function ของ DLL file.
    Const WM_CAP As Short = &H400S
    Const WM_CAP_DRIVER_CONNECT As Integer = WM_CAP + 10
    Const WM_CAP_DRIVER_DISCONNECT As Integer = WM_CAP + 11
    Const WM_CAP_EDIT_COPY As Integer = WM_CAP + 30
    Const WM_CAP_SET_PREVIEW As Integer = WM_CAP + 50
    Const WM_CAP_SET_PREVIEWRATE As Integer = WM_CAP + 52
    Const WM_CAP_SET_SCALE As Integer = WM_CAP + 53
    Const WS_CHILD As Integer = &H40000000
    Const WS_VISIBLE As Integer = &H10000000
    Const SWP_NOMOVE As Short = &H2S
    Const SWP_NOSIZE As Short = 1
    Const SWP_NOZORDER As Short = &H4S
    Const HWND_BOTTOM As Short = 1
    Dim iDevice As Integer = 0 '  Normal device ID  device ID ปัจจุบันในเครื่อง
    Dim hHwnd As Integer ' Handle value to preview window ค่า Handle สำหรับการแสดงภาพในแต่ละวิโดส์ว
    ' Delare function/ API function AVI capture DLL. เรียก API function จาก AVI capture DLL.
    Declare Function SendMessage Lib "user32" Alias "SendMessageA" (  ByVal hwnd As Integer, ByVal wMsg As Integer,_
ByVal wParam As Integer, ByVal lParam As Object) As Integer
    Declare Function SetWindowPos Lib "user32" Alias "SetWindowPos" (ByVal hwnd As Integer,ByVal hWndInsertAfter As_
Integer, ByVal x As Integer, ByVal y As Integer, ByVal cx As Integer, ByVal cy As Integer, ByVal wFlags As Integer) As Integer
    Declare Function DestroyWindow Lib "user32" (ByVal hndw As Integer) As Boolean
    Declare Function capCreateCaptureWindowA Lib "avicap32.dll" ( ByVal lpszWindowName As String, ByVal dwStyle As_
Integer, ByVal x As Integer,ByVal y As Integer, ByVal nWidth As Integer, ByVal nHeight As Short, ByVal hWndParent As_
Integer,ByVal nID As Integer) As Integer
    Declare Function capGetDriverDescriptionA Lib "avicap32.dll" (ByVal wDriver As Short,ByVal lpszName As String, ByVal_
cbName As Integer, ByVal lpszVer As String,ByVal cbVer As Integer) As Boolean
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Label1.Text = "Credit by : Control and Mechatronics Engineering Of  KMITL....."
    SRP.Close()
    Dim name As String
    Dim portname As String()
    portname = IO.Ports.SerialPort.GetPortNames
    Cb_port.Sorted = True
    For Each name In portname '  เป็นการกำหนดค่าของport โดย portname คือ port IO ที่มีการเชื่อมต่ออยู่
        Cb_port.Items.Add(name) '  แล้วเอามาเก็บที่name จากนั้นให้add port จาก name มาใส่ในportที่จะทำการติดต่อ
    Next
    Lb_con.Text = ""
    Tm_data.Stop()
    tim_robot_go.Stop()
    Timer1.Stop()
    Tim_auto.Stop()
    Time_today.Interval = 100 'set ค่าระยะเวลาต่อรอบที่นับ การแสดงเวลาบอกหน้าจอ
    Time_today.Enabled = True 'แสดงเวลาที่หน้าจอ
    '############      tooltip      #######
    ToolTip1.SetToolTip(Btn_con, "ทำการเชื่อมต่อกับ main controller")
    ToolTip1.SetToolTip(Rd_auto, "สั่งให้ระบบทำงานอัตโนมัติ")
    ToolTip1.SetToolTip(Rd_man, "กำหนดค่าให้รถเคลื่อนที่ตามคำสั่ง")
    ToolTip1.SetToolTip(Btn_go, "สั่งให้รถมีการเคลื่อนไปยังstation ที่กำหนด")
    ToolTip1.SetToolTip(Btn_check, "check รถที่จอดที่ station")
    ToolTip1.SetToolTip(Btn_reset, "ยกเลิกการเชื่อมต่อกับ main controller")
    ToolTip1.SetToolTip(Cb_a, "สถานีต้นทาง")
    ToolTip1.SetToolTip(Cb_b, "สถานีปลายทาง")
    Label6.Text = " "
    Label5.Text = " "
End Sub
```

```
'########## time_data ทำงาน แล้วทำการรับค่ามาจาก maincontroller ###
Private Sub Tm_data_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Tm_data.Tick
    Dim sr2 As String 'เก็บค่าที่ได้รับมาจาก serail port
    If SRP.IsOpen = True Then
        Lb_con.Text = "Connecting..."
        sr2 = SRP.ReadExisting
        TXT_status.Text += sr2
    End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'##############  function การทำงานของserial port  ############
Private Sub Btn_con_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_con.Click
```

```
    If Cb_port.Text = "" Then
      MessageBox.Show("เลือก comport ที่จะทำการเชื่อมต่อก่อนทำการ connect","เลือก Comport ก่อนนะ",_
MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
      Tm_data.Enabled = True
      If Btn_con.Text = "Connect" Then
        Btn_con.Text = "Disconnect"
        With SRP 'set parameter serial port
          .PortName = Cb_port.Text
          .BaudRate = 9600
          .DataBits = 8
          .StopBits = IO.Ports.StopBits.One
          .Parity = IO.Ports.Parity.None
          SRP.Open()
          ToolTip1.SetToolTip(Btn_con, "ยกเลิกการเชื่อมต่อกับ main controller")
        End With
      ElseIf Btn_con.Text = "Disconnect" Then
        If MessageBox.Show(" คุณต้องการยกเลิกการเชื่อมต่อกับ Process หรือ ไม่ ", "Worning",_
MessageBoxButtons.OKCancel,MessageBoxIcon.Question) = Windows.Forms.DialogResult.OK Then
          SRP.Close()
          Lb_con.Text = ""
          Btn_con.Text = "Connect"
          Tm_data.Enabled = False
        Else
          Lb_con.Text = "Connecting..."
          Btn_con.Text = "Disconnect"
        End If
      End If
    End If
  End Sub
```

```
  Private Sub Btn_reset_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_reset.Click
    TXT_status.Clear()
    Cb_a.Text = ""
    Cb_b.Text = ""
    Cb_c.Text = ""
    Cb_d.Text = ""
  End Sub
```

```
  Private Sub Btn_check_Click(ByVal sender As System.Object,ByVal e As System.EventArgs) Handles Btn_check.Click
    If SRP.IsOpen = True Then 'check หมายเลขรถแต่ละสถานี
      SRP.Write("c")
      SRP.Write("h")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SRP.Write("e")
        SRP.Write("c")
        SRP.Write("k")
        SRP.Write(" ")
    Else
        TXT_status.Text = "Don't connect"
    End If
End Sub
```

```
Sub output(ByVal a_out As String, ByVal b_out As String) 'ตัวแปรส่งค่าไปผ่านserialportเพื่อส่งไปให้สถานีทำงาน
    SRP.Write(a_out) 'ค่าที่จะส่งไปยังสถานีต้นทาง
    SRP.Write(b_out) 'ค่าที่จะสั่งให้รถเคลื่อนที่ไปยังปลายทาง
    SRP.Write(" ")
End Sub
```

```
Sub arm(ByVal num_arm As String) 'สั่งการแขนกลเพื่อให้ทำงาน
    SRP.Write("R")
    SRP.Write("M")
    SRP.Write(num_arm) 'เลือกชุดคำสั่งที่อยู่บน maincontroller
    SRP.Write(" ")
End Sub
```

```
Private Sub Btn_go_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_go.Click
    If SRP.IsOpen = True Then  'checkเงื่อนไขว่าport มีการเชื่อมต่อหรือไม่
        If Rd_man.Checked = True Then    'check condition ว่ามีการเลือกที่ manual controlหรือไม่
            If Ckb_allarm.Checked = True Then
                arm("0")
            End If
            If Ckb_arm1.Checked = True Then
                arm("3")
            End If
            If Ckb_arm2.Checked = True Then
                arm("4")
            End If
            If Ckb_arm3.Checked = True Then
                arm("1")
            End If
            If Ckb_arm4.Checked = True Then
                arm("2")
            End If
            If Ckb_stall.Checked = True Then
                output("A", "B") '5
                output("B", "C") '6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            output("C", "D") '7
            output("D", "A") '8
            'Ckb_allarm.CheckState = CheckState.Unchecked
        End If
        If Ckb_stA.Checked = True Then
            Select Case Cb_a.Text    'เลือกเงื่อนไขการป้อนตัวอักษรในช่องที่จะให้รถเคลื่อนไป
                Case "A"
                    output("A", "A") '1
                Case "B"
                    output("A", "B") '5
                Case "C"
                    output("A", "C") '9
                Case "D"
                    output("A", "D") '13
            End Select
            If Cb_a.Text = "" Then
                MessageBox.Show("ยังไม่ได้ทำการเลือกสถานีปลายทางของสถานี A", "Select the final station",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
            End If
        End If
        If Ckb_stB.Checked = True Then
            Select Case Cb_b.Text
                Case "B"
                    output("B", "B") '2
                Case "C"
                    output("B", "C") '6
                Case "D"
                    output("B", "D") '10
                Case "A"
                    output("B", "A") '14
            End Select
            If Cb_b.Text = "" Then
                MessageBox.Show("ยังไม่ได้ทำการเลือกสถานีปลายทางของสถานี B", "Select the final station",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
            End If
        End If
        If Ckb_stC.Checked = True Then
            Select Case Cb_c.Text
                Case "C"
                    output("C", "C") '3
                Case "D"
                    output("C", "D") '7
                Case "A"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                output("C", "A") '11
              Case "B"
                output("C", "B") '15
            End Select
            If Cb_c.Text = "" Then
              MessageBox.Show("ยังไม่ได้ทำการเลือกสถานีปลายทางของสถานี C", "Select the final station",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
            End If
          End If
          If Ckb_stD.Checked = True Then
            Select Case Cb_d.Text
              Case "D"
                output("D", "D") '4
              Case "A"
                output("D", "A") '8
              Case "B"
                output("D", "B") '12
              Case "C"
                output("D", "C") '16
            End Select
            If Cb_d.Text = "" Then
              MessageBox.Show("ยังไม่ได้ทำการเลือกสถานีปลายทางของสถานี D ", "Select the final station",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
            End If
          End If
        ElseIf Rd_auto.Checked = True Then
          tim_robot_go.Interval = 60000
          Timer1.Interval = 1000
          tim_robot_go.Enabled = True
          Timer1.Enabled = True
        End If
      Else
        TXT_status.Text = "Don't connect"
      End If
    End Sub
```

```
    Private Sub tim_robot_go_Tick_1(ByVal sender As System.Object, ByVal e As System.EventArgs)_
Handles tim_robot_go.Tick
      If Rd_auto.Checked = True Then
        If SRP.IsOpen = True Then
          If turnnum < NumericUpDown1.Value Then
            intnum = intnum + 1
            SRP.Write("A")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SRP.Write("U")
            SRP.Write("T")
            SRP.Write(" ")
            If intnum Mod 4 = 0 Then
              turnnum = turnnum + 1
              intnum = 0
            End If
          Else
            tim_robot_go.Enabled = False
            Timer1.Enabled = False
            NumericUpDown1.Value = 0
            intnum = 0
            turnnum = 0
          End If
        End If
      End If
    End Sub
```

```
  Private Sub Time_today_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Time_today.Tick
    Label4.Text = "To day : " & Today & " " & TimeOfDay 'show time of today
    If Ckb_stall.CheckState = CheckState.Checked Then
      If Ckb_stA.Checked = True Then
        Ckb_stA.CheckState = CheckState.Unchecked
      ElseIf Ckb_stB.Checked = True Then
        Ckb_stB.CheckState = CheckState.Unchecked
      ElseIf Ckb_stC.Checked = True Then
        Ckb_stC.CheckState = CheckState.Unchecked
      ElseIf Ckb_stD.Checked = True Then
        Ckb_stD.CheckState = CheckState.Unchecked
      End If
    End If
  End Sub
```

```
  Private Sub Rd_man_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)_
Handles Rd_man.CheckedChanged
    If SRP.IsOpen = True Then
      If Rd_man.Checked = True Then
        MessageBox.Show("กรุณาเลือกสถานีที่จะให้รถเคลื่อนที่ไป แล้วกดปุ่ม go", "เลือก Station", MessageBoxButtons.OK,_
MessageBoxIcon.Exclamation)
      End If
    Else
      TXT_status.Text = "Don't connect "
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

```
Private Sub Rd_auto_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)_
Handles Rd_auto.CheckedChanged
    If SRP.IsOpen = True Then
        If Rd_auto.Checked = True Then
            MessageBox.Show("กรุณากรอกตัวเลขเป็นตัวเลขครับ แล้วกดปุ่ม go", "กรอกจำนวนรอบ", MessageBoxButtons.OK,_
MessageBoxIcon.Exclamation)
        End If
    Else
        TXT_status.Text = "Don't connect "
    End If
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    If Rd_auto.Checked = True Then
        If SRP.IsOpen() = True Then
            i = i + 1
            If i = 60 Then
                i = 0
            End If
            Label5.Text = " รอบที่ : " & turnnum.ToString & " รถเคลื่อนที่ไป " & intnum.ToString & " station"
        Else
            i = 0
        End If
        Label6.Text = i.ToString
    End If
End Sub
```

****

" ขั้นตอนสุดท้าย, การปิดแสดงภาพใน โปรแกรมของเรา,เลิกทำการติดต่อ device และ destroy (ทำลาย Object) ที่ เราสร้างขึ้นมา ตอนต้นการเขียนโปรแกรม ที่ให้แสดงภาพ และปิดโปรแกรม

```
' ติดต่อ device ของกล้อง.
Private Sub LoadDeviceList()
    Dim strName As String = Space(100)
    Dim strVer As String = Space(100)
    Dim bReturn As Boolean
    Dim x As Integer = 0
    ' เรียกรายชื่อของ device กล้องทั้งหมดในเครื่องคอมของเรา และโชว์ใน List Box ที่ชื่อ lstDevices .
    Do
        ' เรียกชื่อ Driver และ version
        bReturn = capGetDriverDescriptionA(x, strName, 100, strVer, 100)
        ' ถ้าใช่ device ให้เพิ่มชื่อเข้าไปใน  list box
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If bReturn Then lstDevices.Items.Add(strName.Trim)
        x += 1
    Loop Until bReturn = False
End Sub
```

```
' แสดงผลของรูปภาพเคลื่อนไหวออกที่หน้าฟอร์มโปรแกรมที่เราสร้าง
Private Sub OpenPreviewWindow()
    Dim iHeight As Integer = picCapture.Height
    Dim iWidth As Integer = picCapture.Width
    ' เปิดแสดงผลที่ picturebox .
    ' สร้าง window ลูก ขึ้นมาด้วย ฟังชั่น capCreateCaptureWindowA ซึ่งคุณสามารถเห็นใน picturebox.
    hHwnd = capCreateCaptureWindowA(iDevice, WS_VISIBLE Or WS_CHILD, 0, 0, 640, _
        480, picCapture.Handle.ToInt32, 0)
    ' ติดต่อกับ device
    If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then
        ' ตั้งค่า preview scale
        SendMessage(hHwnd, WM_CAP_SET_SCALE, True, 0)
        ' ตั้งค่า preview rate ในระดับ milliseconds
        SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, 66, 0)
        ' เริ่มต้นการแสดงภาพ จากกล้อง
        SendMessage(hHwnd, WM_CAP_SET_PREVIEW, True, 0)
        ' หรับขนาด window ให้เท่ากับใน picturebox
        SetWindowPos(hHwnd, HWND_BOTTOM, 0, 0, picCapture.Width, picCapture.Height, _
                    SWP_NOMOVE Or SWP_NOZORDER)
        BtnSave.Enabled = True
        btnStop.Enabled = True
        btnStart.Enabled = False
    Else
        ' การติดต่อdevice Error  ให้ ปิด window
        DestroyWindow(hHwnd)
        BtnSave.Enabled = False
    End If
End Sub
```

```
' ใช้ฟังก์ชั่นชื่อ SendMessage ไป copy ข้อมูลไว้ใน clipboard ซึ่งย้ายภาพไปที่ picture box.
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnSave.Click
    Dim data As IDataObject
    Dim bmap As Image
    ' Copy ภาพ ไป clipboard
    SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
    ' นำ ภาพ จาก clipboard และ convert มันไปเป็น bitmap
    data = Clipboard.GetDataObject()
    If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)), Image)
        picCapture.Image = bmap
        ClosePreviewWindow()
        BtnSave.Enabled = False
        btnStop.Enabled = False
        btnStart.Enabled = True
        If sfdImage.ShowDialog = DialogResult.OK Then
          bmap.Save(sfdImage.FileName, Imaging.ImageFormat.Bmp)
        End If
        OpenPreviewWindow()
      End If
    End Sub
```

' ขั้นตอนสุดท้าย, การปิดแสดงภาพใน โปรแกรมของเรา,เลิกทำการติดต่อ deviceและ destroy (ทำลาย Object) ที่ เราสร้างขึ้นมาตอนต้นการเขียนโปรแกรม ที่ให้แสดงภาพ และปิดโปรแกรม

```
    Private Sub ClosePreviewWindow()
      ' เลิกติดต่อกับ device
      SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)
      ' ปิด window
      DestroyWindow(hHwnd)
    End Sub
```

```
    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStop.Click
      ClosePreviewWindow()
      btnStart.Enabled = True
    End Sub
```

```
    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStart.Click
      OpenPreviewWindow()
    End Sub
  End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมการควบคุมรถลำเลียง

โปรแกรมควบคุมรถลำเลียง เพื่อให้รถลำเลียงเคลื่อนที่ไปและสามารถหยุดที่สถานีต่างๆ ได้ตามคำสั่งที่ได้จากการติดต่อสื่อสารกับสถานีเป็นดังนี้

```
/* Include ------------------------------------------------------------*/
#include <16F690.h>
#device *=16
#device ICD=TRUE
#device adc=8
#FUSES NOWDT            //No Watch Dog Timer
#FUSES HS               //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOPROTECT        //Code not protected from reading
#FUSES NOBROWNOUT       //No brownout reset
#FUSES MCLR             //Master Clear pin enabled
#FUSES NOCPD            //No EE protection
#FUSES NOPUT            //No Power Up Timer
#FUSES IESO             //Internal External Switch Over mode enabled
#FUSES FCMEN            //Fail-safe clock monitor enabled
#define  KEYHIT_DELAY   500    // in milliseconds
#use delay(clock=20000000)
#use rs232(baud=2400,parity=N,xmit=PIN_B7,rcv=PIN_B5,bits=8, stream=AVR,timeout=KEYHIT_DELAY)
/* Define -------------------------------------------------------------*/
#define RobotNumber '2'         // RobotNumber 1 to 6
#define Analog0 4               // AnalogChannel 4
#define MotorDrive0 PIN_C3      // MotorDrivePin
#define MotorDrive1 PIN_C4      // MotorDrivePin
#define MPWMPin PIN_C6          // Manual PWM Pin_C2
#define ResMPWM 100             // Maximum Resolution PWM
#define IrFreq 528              // Ir Receive Freq 38 kHz.
#define NcPin0 PIN_A0           // Empty Pin0
#define NcPin1 PIN_B4           // Empty Pin1
#define NcPin2 PIN_C6           // Empty Pin2
#define NcPin3 PIN_B6           // Empty Pin3
#define NcPin4 PIN_C7           // Empty Pin4
#define Button0 !Input(NcPin4)  // Input0 use to memo
#define Button1 !Input(NcPin2)  // Input1 use to run, skip
#define Led0   NcPin0           // Output0 use to run status
#define Led1   NcPin1           // Output1 use to memo status
#define Led2   NcPin3           // Output2 use to sense
#define MaxSpeed 100            // Maximum Motor Speed.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MinSpeed 80                 // Minimum Motor Speed.
/* Private Variable ----------------------------------------------------*/
boolean toggle0 = 1;                // External Interrupt Toggle.
boolean ANL0;                       // Analog0[AN4] Logic
unsigned char ADC_Value;            // Analog to Digital Value
unsigned char Setpoint;             // ADC Setpoint Value.
unsigned char Range;                // Error Range.
char Loopi;                         // Counter Loop.
unsigned char Duty;                 // Set Duty Cycle Of Manual PWM.
unsigned char Mode;                 // Select MotorMode.
unsigned char skipStation = 1;      // Value of skiping station
/* Private Function ----------------------------------------------------*/
void MotorForward(char);
void MotorStop(void);
void MotorBackward(char);
int stacker(void);
unsigned char receive(void);
/* All Interrupt Service Routine ---------------------------------------*/
/*** @ brief RTCC Interrupt Service Routine.
 *      Read Sensor.  */
#int_RTCC
void  RTCC_isr(void){
  /* PWM Motor Control */
  if(Duty >= Loopi){
    if(Mode == 1){
      Output_High(MotorDrive0);
      Output_Low(MotorDrive1);
    }
    else if(Mode == 2){
      Output_High(MotorDrive0);
      Output_High(MotorDrive1);
    }
    else if(Mode == 3){
      Output_Low(MotorDrive0);
      Output_High(MotorDrive1);
    }
  }
  else if((Loopi >Duty) && (Loopi <= ResMPWM)){
    Output_Low(MotorDrive0);
    Output_Low(MotorDrive1);
  }
  else if(Loopi > ResMPWM){
    Loopi = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    Loopi++;
  }
  /*** @brief External Interrupt Service Routine.  */
  #int_EXT
  void  EXT_isr(void) {
    if(toggle0){
      SET_PWM1_DUTY(IrFreq/2);
      EXT_INT_EDGE(L_TO_H);
    }
    else if(!toggle0){
      SET_PWM1_DUTY(0);
      EXT_INT_EDGE(H_TO_L);
    }
    toggle0 = !toggle0;
  }
  /*** @brief TIMER1 Interrupt Service Routine.  */
  #int_TIMER1
  void  TIMER1_isr(void) {
    /* ADC Reading Control. ***************************************************/
    SET_ADC_CHANNEL(Analog0);
    ADC_Value =  READ_ADC();

    if(ADC_Value < Setpoint + 3*Range){
      ANL0 = 1;              // ANL0 = 1 When found Blackline.
      Output_High(Led2);
    }
    else{
      ANL0 = 0;              // ANL0 = 0 When not fount Blackline.
      Output_Low(Led2);
    }
  }
  //@brief Main Func.
  int main(){
    /* Initial Module *******************************************************/
    Setup_Adc_ports(sAN4|VSS_VDD);
    Setup_Adc(ADC_CLOCK_DIV_32);
    Setup_Spi(SPI_SS_DISABLED);
    Setup_Timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    Setup_Timer_1(T1_INTERNAL|T1_DIV_BY_1);
    Setup_Timer_2(T2_DIV_BY_1,131,1);
    Setup_Ccp1(CCP_PWM);
    Set_Pwm1_Duty(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Setup_Comparator(NC_NC_NC_NC);
Enable_Interrupts(INT_RTCC);
Enable_Interrupts(INT_EXT);
EXT_INT_EDGE(H_TO_L);
Enable_Interrupts(INT_TIMER1);
Enable_Interrupts(GLOBAL);
/* Begin Process ****************************************/
Output_High(led0);          // turn on running status led
Output_Low(led1);           // turn off memo status led
Output_Low(led2);           // turn off sense status led
MotorStop();
delay_ms(10);
Setpoint = Read_Eeprom(1);        // read memory
delay_ms(10);
Range = 5 + (float)Setpoint/15.00;    // adjust sensor
  /* Memory Value Sensor ************************************/
while(TRUE){
  if(Button1){              // push to memo
    Write_Eeprom(1,ADC_Value);
    delay_ms(10);
    Setpoint = Read_Eeprom(1);
    delay_ms(10);
    Range = 5 + (float)Setpoint/15.00;
    Output_High(Led1);          // turn on memo status led
  }
  else if(Button0){         // push to run
    Output_Low(Led0);           // turn off running status led
    delay_ms(500);
    break;
  }
  else{
    Output_Low(Led1);           // non-memo status led
  }
}
Output_High(Led0);          // turn on running status led
  /* Robot Running. ****************************************/
MotorForward(MaxSpeed);
delay_ms(200);              // Random error case.
  /* Robot Start Runing At Setpoint Line. ********************/
if(ANL0){
  MotorForward(100);
  while(ANL0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//* @brief Infinity Loop.
while(TRUE){
    /* Check Line 1 And Forward. *********/
  do{
    MotorForward(MaxSpeed);
  }while(!ANL0);
  while(ANL0);
    /* Slowly Then Check Line 2. *********/
  //do{
  //   MotorForward(MinSpeed);
  //}while(!ANL0);
  if(--skipStation == 0){          // Skip Check and Reduce
    /* Stop Motor. ****************************************/
    Output_Low(Led0);              // turn off running status led
    MotorStop();
    Disable_Interrupts(INT_RTCC);
    delay_ms(50);
    Disable_Interrupts(INT_TIMER1);
    delay_ms(50);
          // delay
    delay_ms(200);
    // send RobotNumber
    printf("%c",0x01);
    delay_ms(25);
    printf("%c",0x01);
    delay_ms(25);
        printf("[");              // to station protocol
    delay_ms(25);
        printf("%c",RobotNumber);
    delay_ms(25);
        delay_ms(25);
    Enable_Interrupts(INT_RTCC);
    delay_ms(25);
    Enable_Interrupts(INT_TIMER1);
        // receive skipStation
    skipStation = receive();
        Output_High(Led0);         // turn on running status led
}
  /* Running Again ****************************************/
do{
    Output_Low(Led0);              // turn on running status led
    MotorForward(MaxSpeed);
}while(ANL0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    return 0;
}
 //* @brief Forward Motor. */
void MotorForward(char i){
    Mode = 1;
    Duty = i;
}
/*** @brief Stop Motor.  */
void MotorStop(){
    Mode = 2;
    Duty = 100;
}
/*** @brief Backward Motor.  */
void MotorBackward(char i){
    Mode = 3;
    Duty = i;
    }
unsigned char receive(void){
    unsigned char value;
    unsigned char value2 = 0;
    unsigned char keep = '1';
    do{
        while(!kbhit());
        do{
            value=getc();
            if (value == '<'){
                value2 = value;
            }
        }while (RS232_ERRORS);  //RS232_ERRORS will be 0 when there is a timeout
    }while(value2 !='<');
    value2 = 0;
    while(!kbhit());
    do{
        value=getc();
        if((value == '1')||(value == '2')||(value == '3')||(value == '4')){
            keep = value;
        }
    }while (RS232_ERRORS);      //RS232_ERRORS will be 0 when there is a timeout
    keep = keep - 48;
    return keep;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โปรแกรมสถานี (Station)

```
/* Includes ------------------------------------------------------*/
#include<avr/io.h>
#include<avr/interrupt.h>
#include<compat/deprecated.h>
#define F_CPU 8000000UL
#include<util/delay.h>
#define FOSC 8000000// Clock Speed
#define BAUD 2400
#define MYUBRR FOSC/16/BAUD-1
/* Define ------------------------------------------------------*/
#define F_IR 38000          // Frequency of IR Receiver Module
#define TOP_PWM F_CPU/2/F_IR   // Maximum Value of Duty Cycle
#define data0 (PINC&0x01)
#define data1 (PINC&0x02)>>1
#define data2 (PINC&0b00000100)>>2
/* Private Variable ----------------------------------------------*/
unsigned char toggle = 1;        // Exti toggle
volatile unsigned char stTemp;          // stack temorary.
unsigned char dataTemp;          // data temporary.
volatile unsigned char dataStacked = '0';     // Data is stacked from serial
unsigned char dataBuffered;        // Data is bufferd from spi to serial
unsigned char keep = 0;
volatile char stFlag = 0;
/* Private Function ----------------------------------------------*/
/*** @brief Delay Millisecond.  */
void delay_ms(unsigned int i){
  for(;i>0;i--)
    _delay_ms(1);
}
/*** @brief Initialize PORT.  */
void Init_PORT(){
  sbi(DDRB,1);   // PWM
  //sbi(PORTB,1);
  //cbi(DDRD,2);   // INT0
  //cbi(DDRD,0);   // RXD
  //sbi(DDRD,1);   // TXD
}
/*** @brief Initialize UART.  */
static void Init_USART(unsigned int baud)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  // Set baud rate *.bps.
  UBRRH =(unsigned char)(baud>>8);
  UBRRL =(unsigned char)baud;
  // Enable receiver , tramsmitter , Enable Receiver Interrupt.
  UCSRB = (1<<RXEN) | (1<<TXEN) | (1<<RXCIE);
  // Set frame format: 8data, NoneParity, 1stop bit.
  UCSRC = (1<<URSEL)|(0<<USBS)| (1<<UCSZ1) | (1<<UCSZ0);
}
/*** @brief Receive RXEN  */
unsigned char receive(void){
  while(!(UCSRA&0x80));
  return UDR;
}
/*** @brief Transmit TXEN.  */
static void transmit(unsigned char c){
  while(!(UCSRA&0x20));
  UDR = c;
  UCSRA |= 0x20;
}
/*** @brief Initialize Interrupt.  */
void Init_Interrupt(){
  MCUCR = (1<<ISC01)|(0<<ISC00);   // INT0 Detect falling edge
  GICR  = (1<<INT0);           // INT Enable
  /* Set SREG for Global Interrupt */
  sei();
}
/*** @brief Initialize PWM.  */
void Init_PWM(){
  TCCR1A = 0xF2;
  TCCR1B = 0x11;
  ICR1 = TOP_PWM;
  OCR1A = TOP_PWM;
  OCR1B = TOP_PWM/2;
}
/*** @brief Initialize parallel  */
void parallelSetup(void){
  // PC3 , PC4, PC5 -> MISO
  // PC0 , PC1, PC2 -> MOSI
  // PB2 -> SCK
  // MOSI
  cbi(DDRC,0);
  cbi(DDRC,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  cbi(DDRC,2);
  // SCK
  cbi(DDRB,2);
  // MISO
  sbi(DDRC,3);
  sbi(DDRC,4);
  sbi(DDRC,5);
  // clear
  cbi(PORTC,3);
  cbi(PORTC,4);
  cbi(PORTC,5);
}
uint8_t parallelRead(void){
  uint8_t readBu[3];
    uint8_t readValue;
    readBu[0] = data0;
    readBu[1] = data1;
    readBu[2] = data2;
    readValue = (readBu[0]*1 + readBu[1]*2 + readBu[2]*4);
  return readValue;
}
void parallelWrite(uint8_t pw){
  switch(pw){
     case 0: cbi(PORTC,3);
         cbi(PORTC,4);
         cbi(PORTC,5);
       break;
     case 1: sbi(PORTC,3);
         cbi(PORTC,4);
         cbi(PORTC,5);
       break;
     case 2: cbi(PORTC,3);
         sbi(PORTC,4);
         cbi(PORTC,5);
       break;
     case 3: sbi(PORTC,3);
         sbi(PORTC,4);
         cbi(PORTC,5);
       break;
     case 4: cbi(PORTC,3);
         cbi(PORTC,4);
         sbi(PORTC,5);
       break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      case 5: sbi(PORTC,3);
          cbi(PORTC,4);
          sbi(PORTC,5);
        break;
      case 6: cbi(PORTC,3);
          sbi(PORTC,4);
          sbi(PORTC,5);
        break;
      case 7: sbi(PORTC,3);
          sbi(PORTC,4);
          sbi(PORTC,5);
        break;
  }
}
/*** @brief Main.  */
int main(void){
  Init_PORT();
  Init_PWM();
  Init_Interrupt();
  Init_USART(MYUBRR);
  parallelSetup();
  /*while(1){
    if(stFlag == 1){
      delay_ms(50);
      transmit(dataStacked);
      stFlag = 0;
      dataStacked = '0';
      transmit(dataStacked);
    }
  }*/
  /* Start Process */
  while(1){
    if((dataStacked<=47)&&(dataStacked>=55)){
      dataStacked = '0';
    }
    // Read when detect LOW
    if(!(PINB&0x04)){
        parallelWrite(dataStacked-48);
    }
    else{
      keep = parallelRead();
      if((keep==1)||(keep==2)||(keep==3)||(keep==4)){
        delay_ms(25);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            transmit(0x01);
            transmit(0x01);
            transmit('<');
            transmit(keep+48);
            delay_ms(500);
            transmit(0x01);
            transmit(0x01);
            transmit('<');
            transmit(keep+48);
            dataStacked = '0';
        }  }  }
  return 0;
}
/* Private Function ---------------------------------------------*/
/*** @brief  Interrupt Service Routine of " USART1->RX ".
 *   Send Value From RX1[Infrared] to TX0[Computer].  */
ISR(USART_RXC_vect){
    dataTemp = receive();
  if(dataTemp == '[') {     // from car protocol
      stTemp = 1;
    stFlag = 0;
  }
  if((stTemp == 1)&&(dataTemp !='[')){
      stTemp = 0;
    stFlag = 1;
      dataStacked = dataTemp;
    //transmit(dataStacked);
  }  }
/*** @brief  Interrupt Service Routine of " External Interrupt ".
 *   Modulate Signal from USART Port. */
ISR(INT0_vect){
  if(toggle){
    OCR1A = TOP_PWM/2;
    MCUCR = (1<<ISC01)|(1<<ISC00);  // INT0 Detect Rising edge
    GICR  = (1<<INT0);          // INT0 Enable
  }
  else if(!toggle) {
    OCR1A = TOP_PWM;
    MCUCR = (1<<ISC01)|(0<<ISC00);  // INT0 Detect falling edge
    GICR  = (1<<INT0);          // INT0 Enable
  }
  toggle = !toggle;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

Main Controller

```
/* Include ----------------------------------------*/
#include <avr/io.h>
#include <avr/interrupt.h>
#include <compat/deprecated.h>
unsigned char keeping;
unsigned char flag;
unsigned char stackBu[7];
uint8_t readingBu[4];
// Parallel start up pin
uint8_t A = 22;
uint8_t B = 29;
uint8_t C = 36;
uint8_t D = 43;
/* Function ----------------------------------------*/
/*** @brief Write Parallel */
uint8_t reading (uint8_t pinNum){
   uint8_t readBu[3];
   readBu[0] = digitalRead(pinNum);
   readBu[1] = digitalRead(pinNum+1);
   readBu[2] = digitalRead(pinNum+2);
   return (readBu[0]*1 + readBu[1]*2 + readBu[2]*4);
}
/*** @brief Write Parallel  */
void writing (uint8_t pinNum,uint8_t wr){
  uint8_t wrBu[3];
  switch(wr){
   case 0: wrBu[0] = 0; wrBu[1] = 0; wrBu[2] = 0;
        break;
   case 1: wrBu[0] = 1; wrBu[1] = 0; wrBu[2] = 0;
        break;
   case 2: wrBu[0] = 0; wrBu[1] = 1; wrBu[2] = 0;
        break;
   case 3: wrBu[0] = 1; wrBu[1] = 1; wrBu[2] = 0;
        break;
   case 4: wrBu[0] = 0; wrBu[1] = 0; wrBu[2] = 1;
        break;
   case 5: wrBu[0] = 1; wrBu[1] = 0; wrBu[2] = 1;
        break;
   case 6: wrBu[0] = 0; wrBu[1] = 1; wrBu[2] = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
    case 7: wrBu[0] = 1; wrBu[1] = 1; wrBu[2] = 1;
        break;  }
  for(uint8_t i = 0; i<=2 ; i++){
    if(wrBu[i] == 1){
      digitalWrite(pinNum+i, HIGH);
    }
    else{
      digitalWrite(pinNum+i, LOW);
    }  }  }
/*** @brief Parallel Control  */
uint8_t parallelCtrl(uint8_t pin,uint8_t data){
  uint8_t readValue;
  writing(pin,0);       //close slave read
  digitalWrite(pin+6,LOW);
  delay(20);
  readValue = reading(pin+3);
  writing(pin,data);
  delay(20);            //delay for ready
  digitalWrite(pin+6,HIGH);
  delay(20);
  writing(pin,0);       //close slave read
  return readValue;
}
/*** @brief Setup Arduino  */
void setup()   {
  Serial.begin(9600);
    // Station A port setup (D22-D28)
  pinMode(22,OUTPUT);     //MOSI
  pinMode(23,OUTPUT);     //MOSI
  pinMode(24,OUTPUT);     //MOSI
  pinMode(25,INPUT);      //MISO
  pinMode(26,INPUT);      //MISO
  pinMode(27,INPUT);      //MISO
  pinMode(28,OUTPUT);     //SCK
  digitalWrite(28,HIGH);  //Clock High
  writing(22,0);
  delay(50);
  // Station B port setup (D29-D35)
  pinMode(29,OUTPUT);     //MOSI
  pinMode(30,OUTPUT);     //MOSI
  pinMode(31,OUTPUT);     //MOSI
  pinMode(32,INPUT);      //MISO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  pinMode(33,INPUT);     //MISO
  pinMode(34,INPUT);     //MISO
  pinMode(35,OUTPUT);    //SCK
  digitalWrite(35,HIGH); //Clock High
  writing(29,0);
  delay(50);
  // Station C port setup (D36-D42)
  pinMode(36,OUTPUT);    //MOSI
  pinMode(37,OUTPUT);    //MOSI
  pinMode(38,OUTPUT);    //MOSI
  pinMode(39,INPUT);     //MISO
  pinMode(40,INPUT);     //MISO
  pinMode(41,INPUT);     //MISO
  pinMode(42,OUTPUT);    //SCK
  digitalWrite(42,HIGH); //Clock High
  writing(36,0);
  delay(50);
  // Station D port setup (D43-D49)
  pinMode(43,OUTPUT);    //MOSI
  pinMode(44,OUTPUT);    //MOSI
  pinMode(45,OUTPUT);    //MOSI
  pinMode(46,INPUT);     //MISO
  pinMode(47,INPUT);     //MISO
  pinMode(48,INPUT);     //MISO
  pinMode(49,OUTPUT);    //SCK
  digitalWrite(49,HIGH); //Clock High
  writing(43,0);
  delay(1000);
   // Go! Go! Go!
  Serial.println("command");
}
/*** @brief Super Loop  */
void loop(){
   // Keyboard input
     unsigned char iBu = 0;
   do{
     while(Serial.available() <= 0);
     stackBu[iBu] = Serial.read();
     Serial.print(stackBu[iBu],BYTE);
   }while((stackBu[iBu] != ' ')&&(iBu++<=7));
   // Filter data : just pass '1' to '4'
   if((stackBu[1] >= 54)&&(stackBu[0] <= 60)){
     stackBu[1] = '1';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    Serial.println("");
    // Selection Mode
    // Check All Parallel Data
if((stackBu[0]=='c')&&(stackBu[1]=='h')&&(stackBu[2]=='e')&&(stackBu[3]=='c')&&(stackBu[4]=='k')){
      Serial.println("Check All Data");
      Serial.print(" A=");
      readingBu[0] = parallelCtrl(A,0);
      Serial.print(readingBu[0], DEC);
      Serial.print(", B=");
      readingBu[1] = parallelCtrl(B,0);
      Serial.print(readingBu[1], DEC);
      Serial.print(", C=");
      readingBu[2] = parallelCtrl(C,0);
      Serial.print(readingBu[2], DEC);
      Serial.print(", D=");
      readingBu[3] = parallelCtrl(D,0);
      Serial.print(readingBu[3], DEC);
      Serial.print("\n");
    }
    // Send to A
    else if((stackBu[0] == 'A')&&((stackBu[1]>=65)&&(stackBu[1]<=68))){
     if(stackBu[1] == 'A'){
      keeping = '4';
     }
     else if(stackBu[1] == 'B'){
      keeping = '1';
     }
     else if(stackBu[1] == 'C'){
      keeping = '2';
     }
     else if(stackBu[1] == 'D'){
      keeping = '3';
     }
     flag = parallelCtrl(A,keeping-48);
     if(flag == 0){
      Serial.print("Parking Empty");
     }
     else{
      Serial.print("Robot ");
      Serial.print(flag, DEC);
      Serial.print(" go to station ");
      Serial.print(stackBu[1]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    Serial.print("\n");
  }
  // Send to B
  else if((stackBu[0] == 'B')&&((stackBu[1]>=65)&&(stackBu[1]<=68))){
    if(stackBu[1] == 'A'){
      keeping = '3';
    }
    else if(stackBu[1] == 'B'){
      keeping = '4';
    }
    else if(stackBu[1] == 'C'){
      keeping = '1';
    }
    else if(stackBu[1] == 'D'){
      keeping = '2';
    }
    flag = parallelCtrl(B,keeping-48);
    if(flag == 0){
      Serial.print("Parking Empty");
    }
    else{
      Serial.print("Robot ");
      Serial.print(flag, DEC);
      Serial.print(" go to station ");
      Serial.print(stackBu[1]);
    }
    Serial.print("\n");
  }
  // Send to C
  else if((stackBu[0] == 'C')&&((stackBu[1]>=65)&&(stackBu[1]<=68))){
    if(stackBu[1] == 'A'){
      keeping = '2';
    }
    else if(stackBu[1] == 'B'){
      keeping = '3';
    }
    else if(stackBu[1] == 'C'){
      keeping = '4';
    }
    else if(stackBu[1] == 'D'){
      keeping = '1';
    }
```

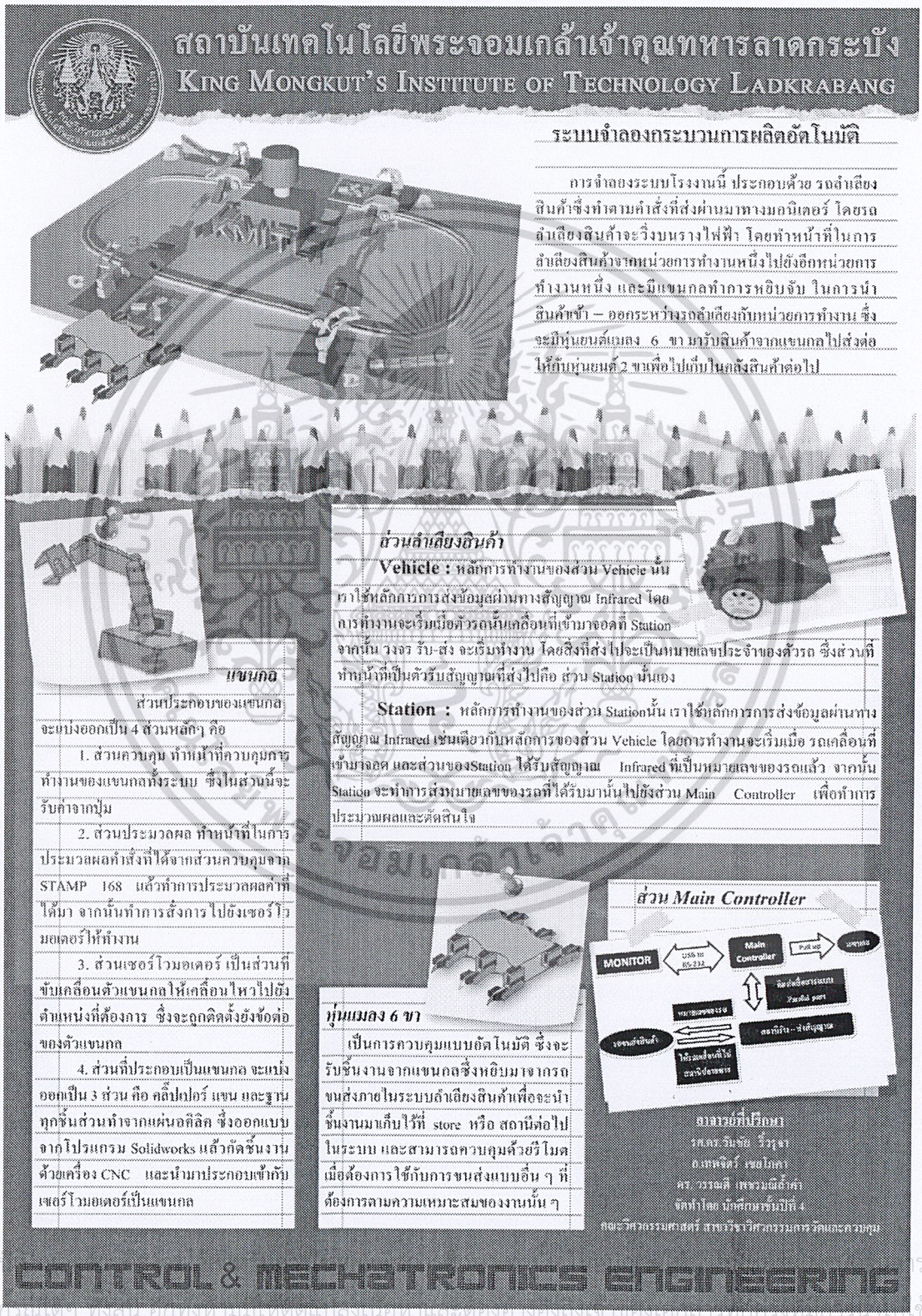
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  flag = parallelCtrl(C,keeping-48);
  if(flag == 0){
    Serial.print("Parking Empty");
  }
  else{
    Serial.print("Robot ");
    Serial.print(flag, DEC);
    Serial.print(" go to station ");
    Serial.print(stackBu[1]);
  }
  Serial.print("\n");
}
// Send to D
else if((stackBu[0] == 'D')&&((stackBu[1]>=65)&&(stackBu[1]<=68))){
  if(stackBu[1] == 'A'){
    keeping = '1';
  }
  else if(stackBu[1] == 'B'){
    keeping = '2';
  }
  else if(stackBu[1] == 'C'){
    keeping = '3';
  }
  else if(stackBu[1] == 'D'){
    keeping = '4';
  }
  flag = parallelCtrl(D,keeping-48);
  if(flag == 0){
    Serial.print("Parking Empty");
  }
  else{
    Serial.print("Robot ");
    Serial.print(flag, DEC);
    Serial.print(" go to station ");
    Serial.print(stackBu[1]);
  }
  Serial.print("\n");
}
// For Wrong Instruction
else{
  Serial.println("Wrong Instruction");
}
}
```

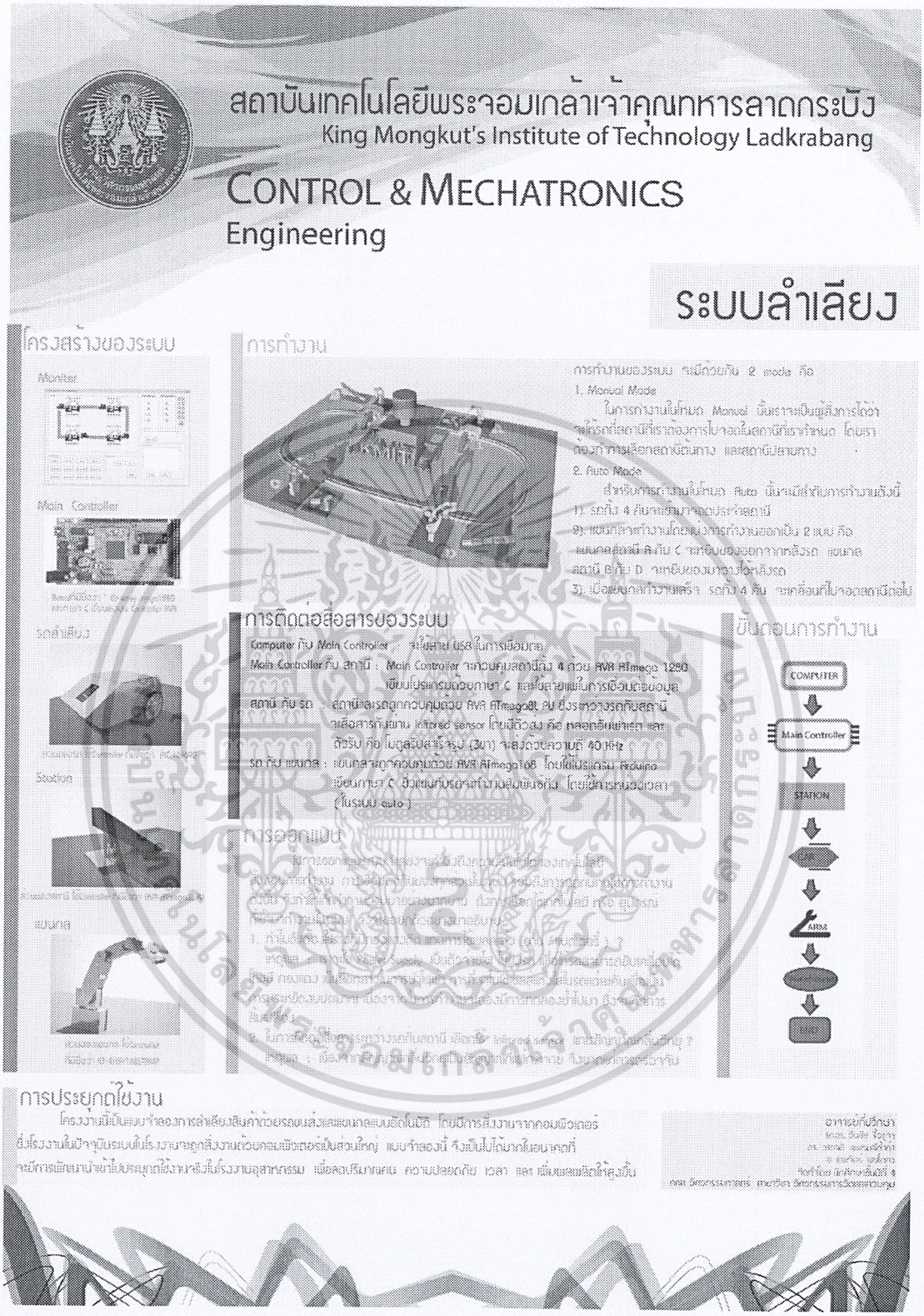
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

โปรสเตอร์ระบบโดยรวม และระบบลำเลียง



เอกสาร ... รค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ประจิน พลังสันติกุล. **เรียนรู้และใช้งาน CCS C คอมไพล์เลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC.** พิมพ์ครั้งที่1. กรุงเทพนคร : อินโนเวตีฟ เอ็กเพอริเมนต์.2547.

[2] นิรุธ อำนวยศิลป์. **คู่มือการเขียนโปรแกรม Microsoft Visual Basic Version 2008 ฉบับโปรแกรมเมอร์.** นนทบุรี : ไอดีซี **2552**

[3] ผศ.ดร. วรพงษ์ ตั้งศรีรัตน์ ,เซ็นเซอร์และทรานสดิวเซอร์ : ทฤษฎีการประยุกต์ใช้ในระบบการวัดและระบบควบคุม, กรุงเทพ, สำนักพิมพ์ ส.ส.ท. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้