สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำเลียง

COMPUTER INTEGRATED MANUFACTURING : CONVEYOR SYSTEM SECTION

นางสาวกันยารัตน์ สาครไทย
นางสาวกาญจนา ทั่งพิทักษ์ไพศาล
นางสาวดุริยา โพธิ์ทอง

เลขหมู่.....
เลขทะเบียน 119324
วัน,เดือน,ปี - 7 ส.ค. 2554

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTER INTEGRATED MANUFACTURING :

CONVEYOR SYSTEM SECTION

MissKanyarat Sakhornthai

MissKanchana Thangpitakpaisal

MissDuriya Photong

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN CONTROL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำเลียง
COMPUTER INTEGRATED MANUFACTURING :
CONVEYOR SYSTEM SECTION

ผู้จัดทำ นางสาวกันยารัตน์ สาครไทย 50010080
นางสาวกาญจนา ทั่งพิทักษ์ไพศาล 50010083
นางสาวดุริยา โพธิ์ทอง 50010532

.....อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร. วันชัย ริ้วรุจา)

.....อาจารย์ที่ปรึกษา
(ดร. วรรณดี เพชรมณีล้ำค่า)

.....อาจารย์ที่ปรึกษา
(อาจารย์ เทพจิตร์ เชยโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำเลียง

โดย

นางสาวกันยารัตน์	สาครไทย	50010080
นางสาวกาญจนา	ทั่งพิทักษ์ไพศาล	50010083
นางสาวดุริยา	โพธิ์ทอง	50010532

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร. วันชัย ริ้วรุจา

ดร. วรรณดี เพชรมณีล้ำค่า

อาจารย์ เทพจิตร์ เชยโภคา

ปีการศึกษา 2553

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอแบบจำลองการทำงานของระบบลำเลียงในอุตสาหกรรม โดยโครงสร้างของระบบประกอบด้วย ตัวควบคุมหลัก รถลำเลียงสินค้า สถานี แขนกล และมอนิเตอร์ จุดมุ่งหมายของโครงงานนี้คือต้องการให้ระบบทำงานได้จริงเปรียบเสมือนระบบลำเลียงสินค้าในโรงงาน โดยเราต้องการขนส่งสินค้าด้วยรถขนส่งและแขนกล โดยมีการสั่งงานจากคอมพิวเตอร์ เริ่มจากการออกแบบและประกอบโครงสร้างของระบบลำเลียงสินค้า แล้วทำการเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษา C โดยมี Arduino เป็นตัวรับ-ส่งสัญญาณในการควบคุมระบบ ส่วนของรถลำเลียงใช้ PIC16F690 สถานีใช้ AVR ATmega8L PU แขนกลใช้ ET-EASY168STAMP มอนิเตอร์ใช้ ET-EASYMEGA 1280 มาทำงานรวมกับวงจรอิเล็กทรอนิกส์และโปรแกรม เพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพ และท้ายสุดทำการทดลอง จากการทดลองพบว่าระบบควบคุมที่ได้ออกแบบนั้น สามารถควบคุมให้ระบบทำงานได้ตามที่เราสั่งการ โดยมีความผิดพลาดอยู่ในระดับที่ยอมรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTER INTEGRATED MANUFACTURING :

CONVEYOR SYSTEM SECTION

By

MissKanyarat Sakhornthai 50010080

MissKanchana Thangpitakpaisal 50010083

MissDuriy Photong 50010532

Advisors

Assoc.Prof. Dr. Vanchai Riewruja

Dr. Wandee Petchmaneelumka

Mr. Thepjit Cheypoca

Academic Year 2010

ABSTRACT

This thesis presents about conveyor system model in factory industry. The structure consists of a main controller, transport cars, stations, robot arms and monitor. The main purpose of this project is to implement the simulated system that can work as a conveyor system in factory industry. Goods transportation by the cars and robot arms is controlled via computer. Beginning of operation is design and assembly of the conveyor system structure. Then C language and Arduino programs are used for control main system. PIC16F690, AVR ATmega8L PU, ET-EASY168STAMP and ET-EASYMEGA 1280 board are used to control in the part of transport car, station, robot arm and monitor, respectively. Electronic circuits and program are used to control the operation of the system efficiently. From experiment, we found that the system could control the operation under the command with acceptable error.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดีจาก รศ.ดร.วันชัย ริ้วรุจา , ดร.วรรณดี เพชรมณีล้ำค่า , อ.เทพจิตร์ เชยโภคา ที่กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่น ๆที่เป็นประโยชน์ต่อโครงงาน ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอบคุณเพื่อน ๆ ทุกคน ที่รวมแรง ร่วมใจกันทำโครงงานชิ้นนี้ ให้ความช่วยเหลือในการแบ่งปันข้อมูล การเสียสละ และ การให้อภัยซึ่งกัน รวมถึงการช่วยเหลือของบุคคลภายนอกที่เข้ามาช่วยในส่วนของชิ้นงาน หรือ การให้ข้อมูลความรู้ที่เป็นประโยชน์ในการทำโครงงาน

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงงานนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นางสาวกันยารัตน์ สาครไทย

นางสาวกาญจนา ทั่งพิทักษ์ไพศาล

นางสาวดุริยา โพธิ์ทอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

Abstract II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญภาพ VII

สารบัญตาราง IX

บทที่ 1 บทนำ 1

1.1 ความเป็นมา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตการศึกษา 2

1.4 รายละเอียดของปริญญานิพนธ์ 2

บทที่ 2 ภาพรวมของระบบ 3

2.1 Computer Integrated Manufacturing (CIM) 3

2.2 ระบบฐานข้อมูลในระบบ CIM 4

2.3 ฮาร์ดแวร์และซอฟต์แวร์ของระบบ CIM 6

2.4 คอมพิวเตอร์ที่ใช้กับระบบ CIM 6

2.5 ระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์ในโครงงานนี้ 8

2.6 ขั้นตอนการทำงานของระบบโดยรวม 9

2.7 ส่วนประกอบภายในระบบ 10

บทที่ 3 ระบบลำเลียง 13

3.1 ระบบลำเลียง 14

3.2 มอเตอร์กระแสตรง 14

3.2.1 หลักการทำงานของมอเตอร์กระแสตรง 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.2.2 การขับและการกลับทิศทางของมอเตอร์กระแสตรง 16

3.2.3 คุณสมบัติของมอเตอร์กระแสตรง 17

3.2.4 โมเดลคณิตศาสตร์ของดีซีมอเตอร์ 18

3.2.5 โมเดลอิเล็คโทรแมคคานิคอล 19

3.3 การออกแบบ 26

3.3.1 การเลือกใช้ 26

3.4 การติดต่อสื่อสารของระบบ 30

3.5 การทำงานของระบบ 31

3.6 ขั้นตอนการทำงาน 32

3.7 โครงสร้างของระบบ 33

บทที่ 4 การทดลอง 41

4.1 การทดลองโปรแกรม และผลที่ได้จากการทำงาน 41

บทที่ 5 บทวิจารณ์และสรุป 45

5.1 สรุปผลการทดลอง 45

5.2 ปัญหาที่พบและแนวทางแก้ไข 46

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา 46

ภาคผนวก 47

ภาคผนวก ก. 48

โครงสร้างหุ่นยนต์

ก.1 แขนกล 48

โครงสร้างแขนกล 48

หลักการทำงาน 49

ก.2 หุ่นยนต์สองขา 51

Servo motor ที่ใช้ในโรงงาน 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

องประกอบของหุ่นยนต์ 2 ขา 55

วงจรสวิตช์ควบคุม 57

ก.3 หุ่นยนต์แมลง 6 ขา 60

วงจรที่ใช้ในการควบคุม 62

ภาคผนวก ข. 64

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข.1 ET-EASY168 STAMP 64

ข.2 ET-EASY MEGA1280 68

ภาคผนวก ค. 77

โปรแกรมควบคุมส่วนต่าง ๆ

ค.1 โปรแกรมการควบคุมรถลำเลียง 77

ค.2 โปรแกรมควบคุมรถลำเลียงสินค้า 86

ค.3 โปรแกรมควบคุมMain Controller 96

ค.4 โปรแกรมควบคุมสถานีรับ-ส่งสัญญาณ 105

ค.5 โปรแกรม VB 113

ภาคผนวก ง. 128

โปสเตอร์ระบบลำเลียง 128

เอกสารอ้างอิง 129

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้า
รูปที่ 2.1 แสดงองค์ประกอบของระบบ CIM	5
รูปที่ 2.2 Flow Chart แสดงขั้นตอนการทำงาน	9
รูปที่ 2.3 แสดงระบบลำเลียง	10
รูปที่ 2.4 รถขนส่งสินค้า	10
รูปที่ 2.5 สถานี	11
รูปที่ 2.6 แขนกล	11
รูปที่ 2.7 ท่อส่งสินค้า	12
รูปที่ 2.8 คลังสินค้า	12
รูปที่ 2.9 หุ่นยนต์แมลง 6 ขา	13
รูปที่ 2.10 หุ่นยนต์ 2 ขา	13
รูปที่ 3.1 โครงสร้างทั่วไปของมอเตอร์กระแสตรง	14
รูปที่ 3.2 โครงสร้างโดยรวมของมอเตอร์กระแสตรง	15
รูปที่ 3.3 การกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์	16
รูปที่ 3.4 การใช้ทรานซิสเตอร์เพื่อขับรีเลย์ให้ทำงาน	16
รูปที่ 3.5 การใช้ทรานซิสเตอร์เป็นวงจรขับและกำหนดทิศทางของมอเตอร์กระแสตรง	17
รูปที่ 3.6 วงจรภายในของมอเตอร์กระแสตรง	17
รูปที่ 3.7 อินพุตและเอาต์พุตของโมเดลทางคณิตศาสตร์ของมอเตอร์	19
รูปที่ 3.8 โมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น	19
รูปที่ 3.9 แรงบิดต่างๆที่เกิดขึ้นต่อโหลดของมอเตอร์	21
รูปที่ 3.10 บล็อกไดอะแกรมของดีซีมอเตอร์โมเดล	22
รูปที่ 3.11 รางรถขนส่งสินค้า	26
รูปที่ 3.12 ลักษณะของ ET-EASY168 STAMP	27
รูปที่ 3.13 ลักษณะของ Easy Mega 1280	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูปที่	หน้า
รูปที่ 3.14 ลักษณะของ PIC16F690	28
รูปที่ 3.15 ลักษณะของ TSOP34836	29
รูปที่ 3.16 ภาพรวมการทำงานของระบบ	31
รูปที่ 3.17 ขั้นตอนการทำงานของระบบลำเลียง	32
รูปที่ 3.18 ลักษณะของ Easy Mega1280	33
รูปที่ 3.19 การจัดเรียงขาของ AT mega 1280	36
รูปที่ 3.20 แผนภาพแสดงการทำงานของส่วน Main Controller	37
รูปที่ 3.21 ลักษณะของ PIC16F690	37
รูปที่ 3.22 ลักษณะของ ET-EASY168 STAMP	38
รูปที่ 3.23 ลักษณะของบอร์ด ET-EASY168 STAMP	40
รูปที่ 4.1 การเลือก port เพื่อ connect	41
รูปที่ 4.2 การดูการทำงานจริงผ่านกล้อง webcam	42
รูปที่ 4.3 การเลือกโหมดการทำงาน	42
รูปที่ 4.4 การเลือกโหมดควบคุมด้วยมือ	43
รูปที่ 4.5 การเลือกโหมดอัตโนมัติ	43
รูปที่ 4.6 การตรวจสอบสถานะการทำงาน	44
รูปที่ ก.1.1 โครงสร้างแบบ Top ของแขนกล	48
รูปที่ ก.1.2 โครงสร้างแบบ Font ของแขนกล	48
รูปที่ ก.1.3 โครงสร้างแบบ Isometric ของแขนกล	49
รูปที่ ก.1.4 แขนกลที่ประกอบเสร็จสมบูรณ์	50
รูปที่ ก.2.1 โครงสร้างหุ่นยนต์ 2 ขา	51
รูปที่ ก.2.2 เซอร์โวมอเตอร์ รุ่น MG945	52
รูปที่ ก.2.3 Servo motor รุ่น S3003	53
รูปที่ ก.2.4 servo motor EK 2-0508	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูปที่	หน้า
รูปที่ ก.2.5 องค์ประกอบของหุ่นยนต์ 2 ขา	55
รูปที่ ก.2.6 วงจรของหุ่นยนต์ 2 ขา	56
รูปที่ ก.2.7 วงจรสวิตช์ควบคุม	57
รูปที่ ก.2.8 บอร์ดควบคุมเซอร์โวมอเตอร์	57
รูปที่ ก.2.9 แสดงการเชื่อมต่อ servo motor กับ pin ของ AVR MEGA 1280	58
รูปที่ ก.2.10 หุ่นยนต์ 2 ขา ที่ประกอบเสร็จสมบูรณ์	59
รูปที่ ก.3.1 โครงสร้างหุ่นยนต์แมลงหกขา ด้านหน้า	60
รูปที่ ก.3.2 โครงสร้างหุ่นยนต์แมลงหกขา ด้านข้าง	60
รูปที่ ก.3.3 โครงสร้างหุ่นยนต์แมลงหกขา ด้านข้าง	61
รูปที่ ก.3.4 โครงสร้างหุ่นยนต์แมลงหกขา มุม Isometric	61
รูปที่ ก.3.5 วงจรที่ใช้ในการควบคุมหุ่นยนต์แมลงหกขา	62
รูปที่ ก.3.6 Flowchart ของโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์แมลงหกขา	63
รูปที่ ข.1.1 ET-EASY168 STAMP	64
รูปที่ ข.1.2 การจัดสรรขาสัญญาณของบอร์ด ET-EASY168 STAMP	66
รูปที่ ข.1.3 ลักษณะของบอร์ด ET-EASY168 STAMP	66
รูปที่ ข.1.4 ตารางขาสัญญาณของ ET-EASY168 STAMP	67
รูปที่ ข.2.1 บอร์ด ET-EASY MEGA1280	68
รูปที่ ข.2.2 โครงสร้างของบอร์ด ET-EASY MEGA1280 (Duino MeGA)	70
รูปที่ ข.2.3 การทำงานของบอร์ด ET-EASY MEGA1280	72
รูปที่ ข.2.4 สัญญาณขาของ ET-EASY MEGA1280	72
รูปที่ ข.2.5 สัญญาณขาของ ET-EASY MEGA1280	73
รูปที่ ข.2.6 การเชื่อมต่อของสายแพร์กับบอร์ด ET-EASY MEGA1280	73
รูปที่ ข.2.7 สัญญาณของบอร์ด ET-EASY MEGA1280	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 3.1 ตารางแสดง การจัดสรรขาสัญญาณของ บอร์ด ET-EASY168 STAMP 40

ตารางที่ ข.2 ตารางขาสัญญาณของ ET-EASY168 STAMP 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

ระบบลำเลียงสินค้า มีความสำคัญอย่างมากกับกระบวนการผลิตในอุสาหกรรม ซึ่งระบบลำเลียงสินค้าในธุรกิจการผลิตก็เปรียบเสมือนระบบประสาทของร่ายกายมนุษย์ จะต้องทำงานได้อย่างเป็นปกติเสมอ ไม่เช่นนั้นส่วนอื่น ๆ ก็ไม่สามารถทำงานได้อย่างมีประสิทธิภาพ

โครงงานนี้เป็นแบบจำลองการลำเลียงสินค้าด้วยรถขนส่งและแขนกลแบบอัตโนมัติ โดยมีการสั่งงานจากคอมพิวเตอร์ มีการวางระบบโดยนำทฤษฎีและความรู้ทางด้านอิเล็กทรอนิกส์มาประยุกต์ใช้ในการสร้างวงจรควบคุม ในส่วนของชิ้นงานได้นำอุปกรณ์ต่าง ๆ มาใช้ในการควบคุม เช่น เซอร์โวมอเตอร์ ไมโครคอลโทลเลอร์ เป็นต้น มาทำงานรวมกับวงจรอิเล็กทรอนิกส์ และ โปรแกรมเพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพ หรือเรียกอีกชื่อว่า ระบบ Computer Integrated Manufacturing (CIM) ซึ่งเป็นระบบการผลิตที่ใช้คอมพิวเตอร์เข้ามาควบคุมกระบวนการผลิตทั้งหมด ทำให้มีการแลกเปลี่ยนข้อมูลระหว่างหน่วยงานแต่ละหน่วยได้ ทำให้แต่ละหน่วยรับรู้ความก้าวหน้าซึ่งกันและกัน ขอบข่ายหน้าที่การทำงานของระบบ CIM มีหลากหลายอย่าง เช่น ออกแบบ วิเคราะห์ วางแผน จัดซื้อ จัดการบัญชีต้นทุน ควบคุมคงคลัง และการกระจายผลิตภัณฑ์ เหล่านี้จะถูกเชื่อมโยงโดยคอมพิวเตอร์ไปยังอุปกรณ์ หรือหน่วยต่างๆ ภายในองค์กร ระบบ CIM จะทำให้สามารถควบคุมกระบวนการได้โดยตรงและสามารถแสดงการทำงานปัจจุบันของทุกกระบวนการทำงาน ซึ่งระบบ CIM จะประกอบด้วยคอมพิวเตอร์ที่มีการแลกเปลี่ยนข้อมูลระหว่างกัน 2 เครื่องขึ้นไป

1.2 วัตถุประสงค์

1. เพื่อจำลองระบบลำเลียงสินค้าในโรงงานอุตสาหกรรม
2. เพื่อพัฒนาทักษะทางด้านวงจรอิเล็กทรอนิกส์และการเขียนโปรแกรม
3. เพื่อพัฒนาความคิดสร้างสรรค์และความชำนาญในการออกแบบโครงสร้างของระบบลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตการศึกษา

1. การออกแบบระบบลำเลียง
2. การเขียนโปรแกรม ARDUINO , ภาษาซี.
3. การศึกษาการทำงานของ วงจรอิเล็กทรอนิกส์ และ เซอร์โวมอเตอร์
4. จำลองระบบเสมือนจริง

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวนำถึงวัตถุประสงค์ หลักการใหม่ ขั้นตอนการศึกษาและการจัดทำโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงทฤษฎีที่เกี่ยวข้องในระบบ Computer Integrated Manufacturing (CIM) และรายละเอียดของส่วนประกอบทั้งหมดในระบบ Computer Integrated Manufacturing (CIM)

บทที่ 3 หลักการและแนวคิดในการออกแบบ นำเสนอส่วนประกอบโครงสร้างของระบบลำเลียง

บทที่ 4 การทดลอง เป็นส่วนการทดสอบองค์ประกอบต่างๆ ในระบบ

บทที่ 5 บทวิจารณ์และสรุป สรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงงานนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การผลิตผสมผสานด้วยคอมพิวเตอร์

ปัจจุบันนี้คอมพิวเตอร์ได้ถูกนำมาใช้กับชีวิตประจำวันอย่างหลีกเลี่ยงไม่ได้ ไม่ทางตรงก็ทางอ้อม เหตุมาจากคอมพิวเตอร์ สามารถช่วยในการทำงานซ้ำ ๆ ซาก ๆ ปริมาณมาก ๆ แทนมนุษย์เราได้อย่างรวดเร็ว ถูกต้อง แม่นยำ และสามารถทำงานได้โดยไม่ต้องหยุดพัก และคอมพิวเตอร์ยังสามารถทำงานแทนมนุษย์ได้ในสภาพแวดล้อม หรือสถานทีที่เป็นอันตรายต่อสุขภาพร่างกายของมนุษย์ เช่น สถานที่ที่มีกัมมันตภาพรังสี หรือในงานที่มีความเสี่ยงสูงในโรงงานอุตสาหกรรม เป็นต้น

2.1 การผลิตผสมผสานด้วยคอมพิวเตอร์

การผลิตผสมผสานด้วยคอมพิวเตอร์เป็นระบบการผลิตที่ใช้คอมพิวเตอร์เข้ามาควบคุมกระบวนการผลิตทั้งหมด การผสมผสานของระบบ ทำให้มีการแลกเปลี่ยนข้อมูลระหว่าหน่วยงานแต่ละหน่วยได้ ทำให้แต่ละหน่วยรับรู้ความก้าวหน้าซึ่งกันและกัน ข้อดี คือ ระบบการผลิตจะมีความรวดเร็วและมีข้อผิดพลาดน้อย แต่ข้อดีหลัก ๆ ของ CIM คือ ความสามารถในการสร้างกระบวนการผลิตแบบอัตโนมัติ ซึ่งโดยทั่วไปแล้วระบบ CIM จะเป็นกระบวนการควบคุมแบบปิด (Closed-loop Control Processes) บนพื้นฐานของข้อมูล ณ ปัจจุบันที่ได้รับจากตัวตรวจรู้ (Sensor)

Computer Integrated Manufacturing (CIM) เป็นทั้งกระบวนการผลิตและชื่อของระบบอัตโนมัติที่ควบคุมด้วยคอมพิวเตอร์ โดยมีหน้าที่สนับสนุนการทำงานและเป็นระบบการจัดการของระบบการผลิตที่ประกอบด้วยฝ่ายต่าง ๆ เช่น ฝ่ายงานวิศวกรรม ฝ่ายงานการผลิต ฝ่ายการตลาด และฝ่ายการสนับสนุนอื่น ๆ ขอบข่ายหน้าที่การทำงานของ CIM มีหลากหลายอย่าง เช่น ออกแบบ วิเคราะห์ วางแผน จัดซื้อ จัดการบัญชีต้นทุน ควบคุมคงคลัง และการกระจายผลิตภัณฑ์ เหล่านี้จะถูกเชื่อมโยงโดยคอมพิวเตอร์ไปยังอุปกรณ์ หรือหน่วยต่าง ๆ ภายในองค์กร CIM จะทำให้สามารถควบคุมกระบวนการได้โดยตรงและสามารถแสดงการทำงานปัจจุบันของทุกกระบวนการทำงาน

ข้อแตกต่าง 3 ประการที่ทำให้ CIM แตกต่างจากระบบการผลิตแบบอื่น ๆ คือ

1. สื่อที่ใช้ในการเก็บข้อมูลการคืนสภาพกระบวนการการควบคุมและการนำเสนอ
2. กลไกการตรวจจับและการดัดแปลงกระบวนการต่างๆ
3. อัลกอริทึมของการประมวลผลข้อมูล ที่ได้จากการตรวจจับและการดัดแปลงส่วนประกอบต่างๆ

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้นำไปใช้

CIM จะประกอบด้วยคอมพิวเตอร์ที่มีการแลกเปลี่ยนข้อมูลระหว่างกัน 2 เครื่องขึ้นไป เช่น ตัวควบคุมหุ่นยนต์อุตสาหกรรม (Industrial Robot) กับตัวคุมเครื่อง CNC สิ่งที่ต้องคำนึงถึงในการติดตั้งระบบ CIM คือ ปริมาณการผลิต ประสบการณ์ขององค์การและบุคลากร ระดับการผสมผสานของส่วนการผลิตและส่วนต่างๆ CIM มีประโยชน์มากที่สุดในองค์การที่มีระดับการใช้ข้อมูลสารสนเทศภายในองค์กรสูง

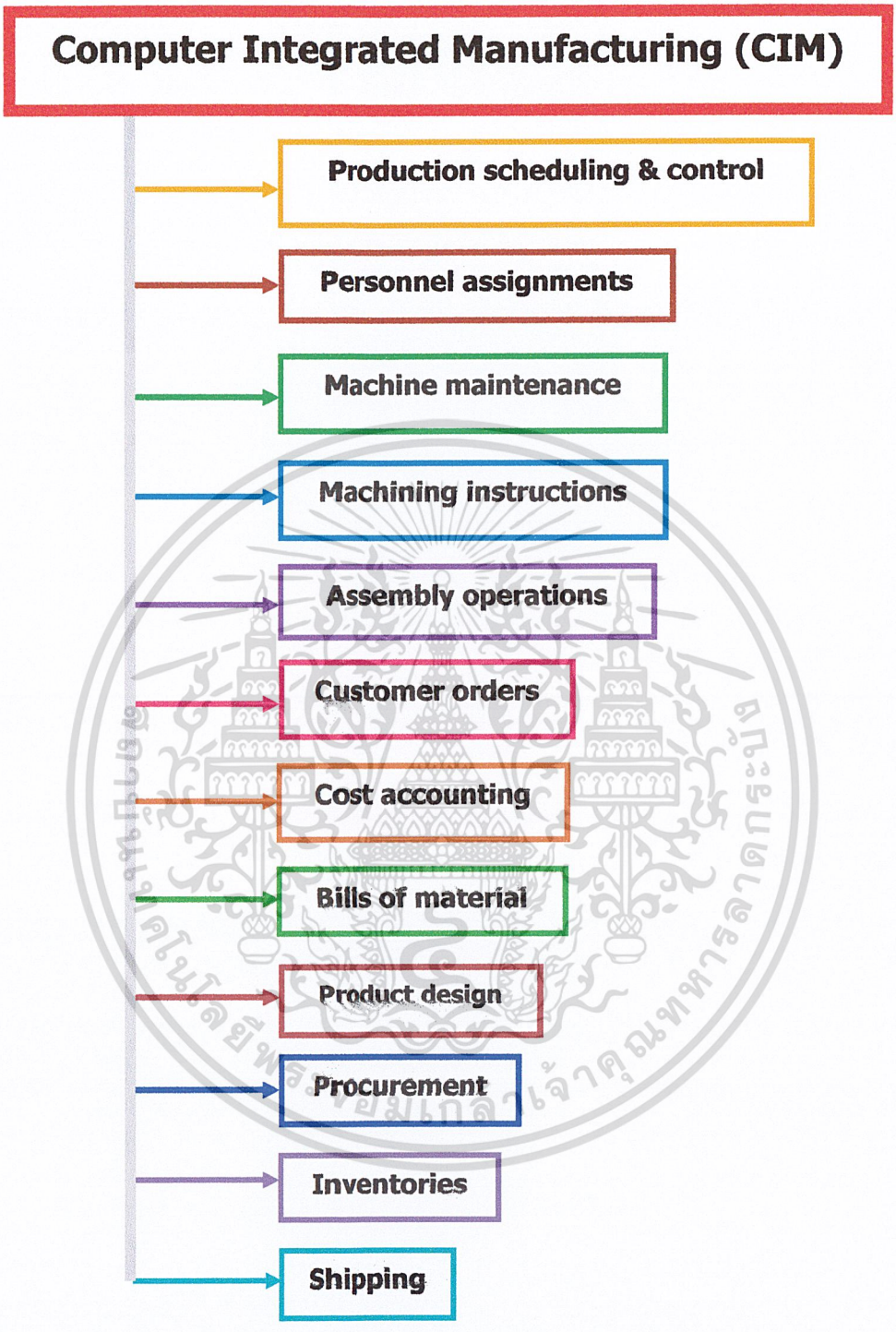
2.2 ระบบฐานข้อมูลในระบบCIM

CIM ต้องมีการจัดการด้วยระบบฐานข้อมูล ทั้งนี้สามารถแบ่งระบบของข้อมูลออก เป็น 2 ระบบหลักคือ

1. ระบบข้อมูลหลักขององค์กร จะถูกนำมาใช้ประโยชน์ในการจัดเตรียมและการจัดการสายการผลิต , การจัดเตรียมวัตถุดิบเมื่อได้รับใบสั่งซื้อ (order) , การควบคุมสต็อก และการจัดส่งสินค้า ฯลฯ ระบบข้อมูลหลักของ องค์กรมักจะประกอบไปด้วยข้อมูลทางด้านการตลาด , การออกแบบผลิตภัณฑ์และออกแบบทางวิศวกรรม , การวางแผน , การซื้อขาย , วิศวกรรมโรง งาน , ฮาร์ดแวร์ของโรงงานอัตโนมัติ , โกดังสินค้า , การวิจัยและพัฒนา , การวางแผนทางการเงิน , การจัดการข้อมูลข่าวสาร

2. ระบบควบคุมสายการผลิต ในการวางแผนสายการผลิตนั้นมักจะมีการแบ่งออก เป็นหน่วยงานย่อย ๆเพื่อทำการควบคุมสายงานการผลิตออกมาต่างหาก โรงงานที่มีเทคโนโลยีสมัยใหม่ทางด้านการควบคุมสายการผลิตอาจมีการวางแผนระบบการใช้เครื่องจักรให้มีความยืดหยุ่นของระบบการทำงานในกระบวนการผลิตได้ โดยจะใช้ระบบที่เรียกว่า FMS (Flexible Manufacturing System) ในหน่วยงานผลิตย่อยความเกี่ยวโยงกันของระบบ FMS กับ CAD/CAM/C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 องค์ประกอบของระบบ CIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ฮาร์ดแวร์และซอฟต์แวร์ของระบบ CIM

ระบบ CIM มีส่วนประกอบของฮาร์ดแวร์และซอฟต์แวร์ดังรายละเอียดดังต่อไปนี้

ฮาร์ดแวร์

1. เครื่องจักรและอุปกรณ์ของโรงงานได้แก่ สถานีงาน (workstations), ส่วนทำงาน (cells) , ระบบ DNC/FMS , อุปกรณ์เคลื่อนย้ายชิ้นงานและเครื่องมือ (work and tool handling devices) , อุปกรณ์กักเก็บ (storage devices) , เซนเซอร์ (sensors) และ อุปกรณ์จัดเก็บข้อมูลภายในโรงงาน (shopfloor data collection devices) เป็นต้น

2. คอมพิวเตอร์ , ตัวควบคุม , ระบบ CAD/CAM , สถานีงานและสถานีย่อย (terminals) , เครื่องพิมพ์ , พล็อตเตอร์ และอุปกรณ์ประกอบอื่น ๆ , โมเด็ม , สายเคเบิลและขั้วต่อ ฯลฯ

ซอฟต์แวร์

ซอฟต์แวร์มีอยู่หลากหลายชนิดตัวอย่างคือ โปรแกรม MIS (Management Information System) , บริหารการขาย , การตลาด , การเงิน , จัดการฐานข้อมูล , การออกแบบ , การวิเคราะห์ , การจำลองแบบ , Communication, Monitoring , ควบคุมการผลิต , Manufacturing area control , Job tracking , สินค้าคงคลัง , บาร์โค้ด , สั่งซื้อ , สายพานลำเลียง , วางแผนกระบวนการ , บริหารสาธารณูปโภคของโรงงาน

2.4 คอมพิวเตอร์ที่ใช้กับระบบ CIM

การเลือกใช้เครื่องคอมพิวเตอร์จะเป็นขนาดใดนั้นจะขึ้นอยู่กับขนาดของข้อมูลและความต้องการความเร็วในการประมวลผล โดยปกติแล้วในระบบCIM สามารถเลือกใช้คอมพิวเตอร์ได้ทั้งสามประเภทคือ คอมพิวเตอร์ส่วนบุคคล , มินิคอมพิวเตอร์ และเมนเฟรมคอมพิวเตอร์ โดยประเภทแรกมักจะอยู่ตามสถานีงานย่อยๆ และมีการใช้งานในลักษณะโต้ตอบกับผู้ใช้งาน (users interact) ใช้ขณะที่สองประเภทหลังจะใช้เป็นคอมพิวเตอร์ในการดูแลและควบคุม (supervisory computers)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปแล้วเราต้องการเครื่องคอมพิวเตอร์ที่จะ นำมาใช้กับระบบ CIM ที่มีคุณสมบัติ ดังนี้

1. ความสามารถที่จะใช้ได้กับซอฟต์แวร์ทูล ชนิดพิเศษได้หลาย ๆ อย่าง เช่น Oracle, Ingres, Cybase , Informix , Progress และอื่น ๆ
2. ใช้ได้กับโปรแกรมหลาย ๆ ภาษา (language compatibility) เช่น ภาษา C , C + + , Cobol Frotran และอื่น ๆ
3. มีสมรรถนะสูง , เชื่อถือและวางใจได้ สามารถประมวลผลหรือกระทำการตามสั่งได้อย่างรวดเร็ว แม้จะมีข้อมูลอย่างมากมาย
4. เข้ากันได้กับคอมพิวเตอร์เครื่องอื่นๆ เมื่อถูกต่อเข้ากับระบบเครือข่ายและระบบการสื่อสาร
5. แสดงภาพกราฟิกได้อย่างรวดเร็วและสวยงาม
6. ใช้ได้กับระบบปฏิบัติการทั้ง UNIX และ MS-DOS และสามารถแชร์ไฟล์กัน ได้ทั้ง MS-DOS และ UNIX programs เป็นต้น

ในแวดวงอุตสาหกรรมโลกทุกวันนี้ต้นทุนของวัตถุดิบ , แรงงาน และค่าโสหุ้ยจิปาถะมี การปรับตัวสูงมากขึ้นเรื่อย ๆ ประกอบกับการแข่งขันทางด้านการตลาดที่มีอยู่สูง ทำให้โรงงาน ผู้ผลิตจำเป็นจะต้องเพิ่มผลผลิตหรืออัตราการผลิตให้มากขึ้น ทั้งนี้ก็เพื่อต้องการลดราคาต้นทุนต่อหน่วยของสินค้าให้ต่ำลง ทางเลือกหนึ่งที่ทำได้ก็คือการหันมาใช้เครื่องจักรหรือระบบที่มีการควบคุมหรือประสานด้วยคอมพิวเตอร์ ทั้งนี้ก็เพื่อให้ได้กำลังผลิตสูงสุด , ผลผลิตเพิ่มขึ้น , รวดเร็วแน่นอน , ลดจำนวนของเสียได้มากและสามารถคืนทุนได้ในระยะเวลาอันสั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสำเร็จในการดำเนินธุรกิจนั้นนอกจากการจะต้องชนะใจลูกค้าเพื่อให้หันมาสนใจ และซื้อสินค้าของบริษัทแล้ว ยังจะต้องเอาชนะคู่แข่งขันให้ได้อีกด้วยแนวทางในการเอาชนะคู่แข่งขันที่ฝ่ายผลิตจะต้องรับผิดชอบได้แก่

- ประสิทธิภาพของต้นทุน (cost efficiency)
- คุณภาพผลิตภัณฑ์ (product quality)
- ความเชื่อถือได้ (dependability)
- ความยืดหยุ่นของระบบการผลิต (flexibility)

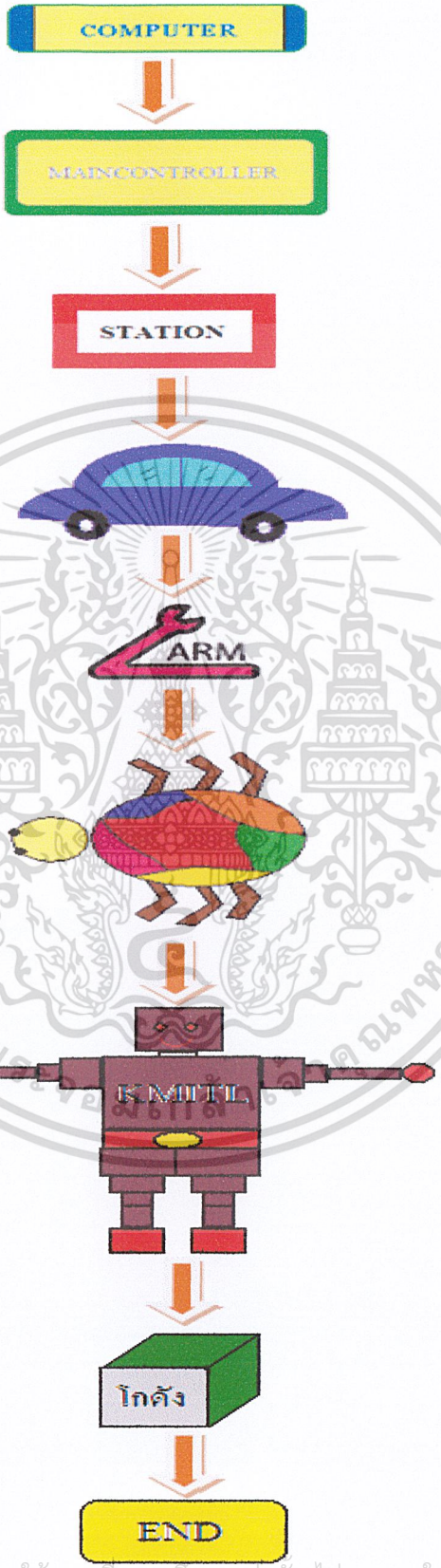
แนวทางทั้งสี่ข้อนี้มีอิทธิพลอย่างมากในการชิงความได้เปรียบในเชิงการแข่งขันการเลือกใช้ระบบการผลิตแบบประสานด้วยคอมพิวเตอร์หรือ CIM ก็เป็นอีกทางเลือกหนึ่งในการตัดสินใจเชิงกลยุทธ์ในการบริหารการผลิตและ CIM ก็สามารถตอบสนองความต้องการทั้งสี่ข้อดังกล่าวได้เป็นอย่างดี

2.5 ระบบจำลองการผลิตผสมผสานด้วยคอมพิวเตอร์ในโครงงานนี้

ในการทำโครงงานนี้ ได้ออกแบบจำลองการลำเลียงสินค้าแบบอัตโนมัติที่ถูกควบคุมด้วยระบบคอมพิวเตอร์ ซึ่งในแบบจำลองนี้จะประกอบไปด้วยส่วนต่าง ๆ ของระบบลำเลียงสินค้า คือ รถขนส่งที่ทำตามคำสั่งผ่านทางมอนิเตอร์ โดยรถขนส่งจะวิ่งบนรางทองแดงลำเลียงสิ้นค้าจากสถานีหนึ่ง ไปอีกสถานีหนึ่ง ซึ่งสถานีจะมีตัวตรวจจับความแตกต่างของรถแต่ละคัน เพื่อส่งข้อมูลไปยัง main controller ให้สั่งการไปที่แขนกล แต่ละสถานี ให้หยิบจับชิ้นงาน และจะมีหุ่นยนต์แมลง 6 ขา มารับสิ้นค้าจากแขนกลตามสถานี ไปส่งต่อให้กันหุ่นยนต์ 2 ขา นำชิ้นงานไปเก็บในคลังสินค้า ซึ่งในส่วนของ หุ่นยนต์แมลง 6 ขา และหุ่นยนต์ 2 ขา จะมีการเคลื่อนที่ได้อย่างอิสระเพื่อสะดวกในการเคลื่อนที จับเก็บสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ขั้นตอนการทำงานของระบบโดยรวม



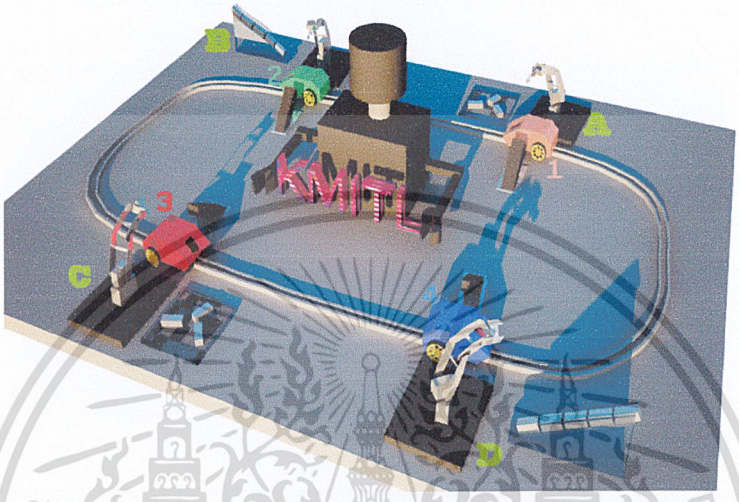
รูปที่ 2.2 Flow Chart แสดงขั้นตอนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ลอกเลียนแบบ ดัดแปลง เผยแพร่ หรือนำไปใช้ในเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ส่วนประกอบภายในระบบ

- ระบบลำเลียง

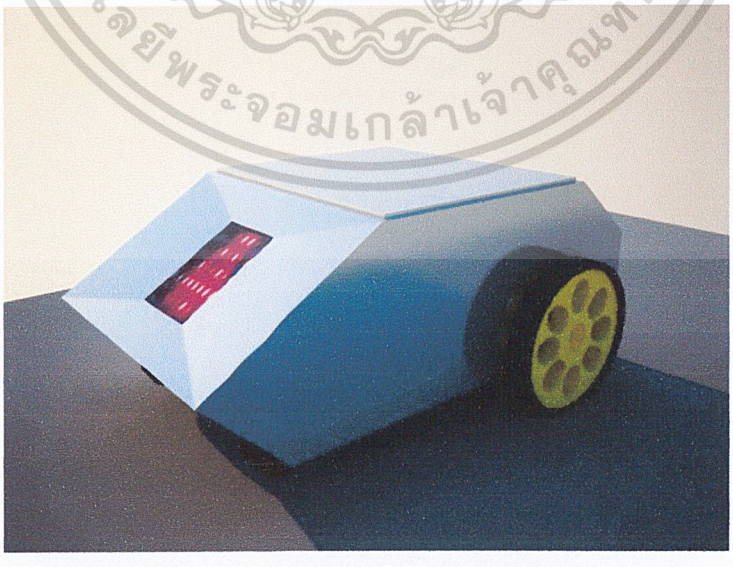
ระบบขนส่งสินค้าภายในโรงงาน ถูกควบคุมด้วยระบบคอมพิวเตอร์ มีการใช้หุ่นยนต์ในการทำงานของระบบเป็นส่วนใหญ่ โดยที่ Main Controller จะสั่งงานไปตามอุปกรณ์ต่างๆ ตามค่าของระบบ



รูปที่ 2.3 ระบบลำเลียง

- รถขนส่งสินค้า

รถขนส่งสินค้าจะลำเลียงสินค้าจากสถานีหนึ่งไปยังอีกสถานีหนึ่ง

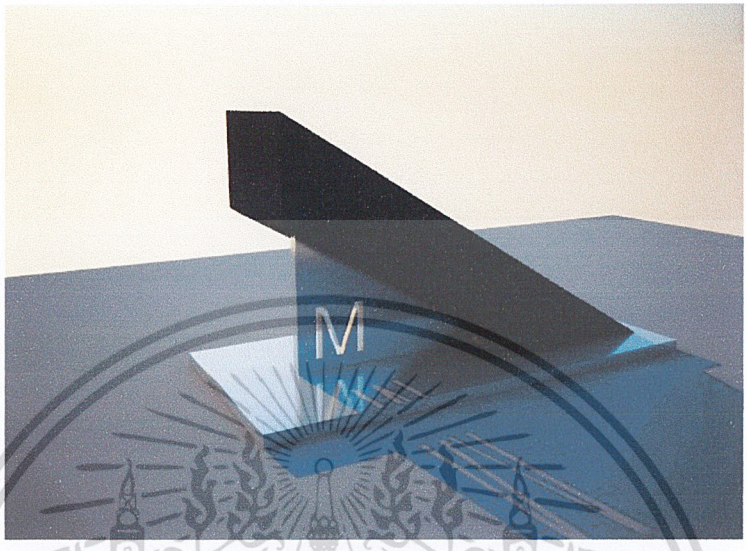


รูปที่ 2.4 รถขนส่งสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สถานี

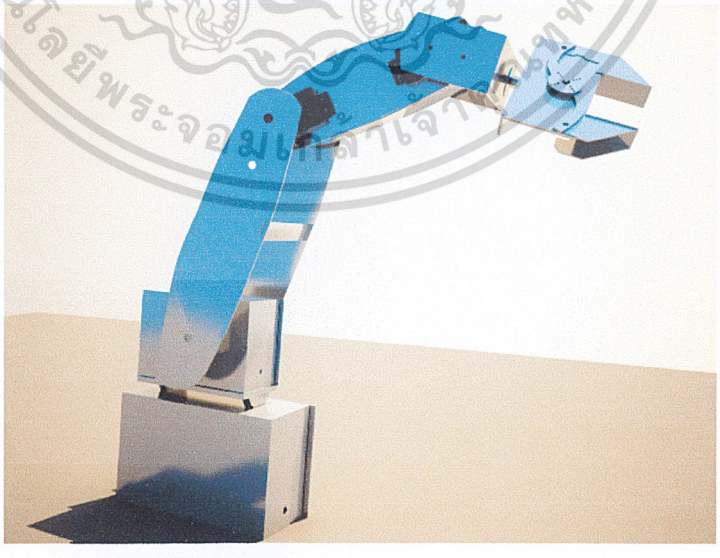
สถานี จะเป็นตัวตรวจจับข้อมูลของรถขนส่งสินค้า และรายงานผลไปยัง Main Controller เพื่อให้ Main Controller ประมวลผล และส่งคำสั่งกลับมา



รูปที่ 2.5 สถานี

- แขนกล

หยิบจับสินค้าจากรถ หรือ หยิบสินค้าจากคลังเก็บสินค้า

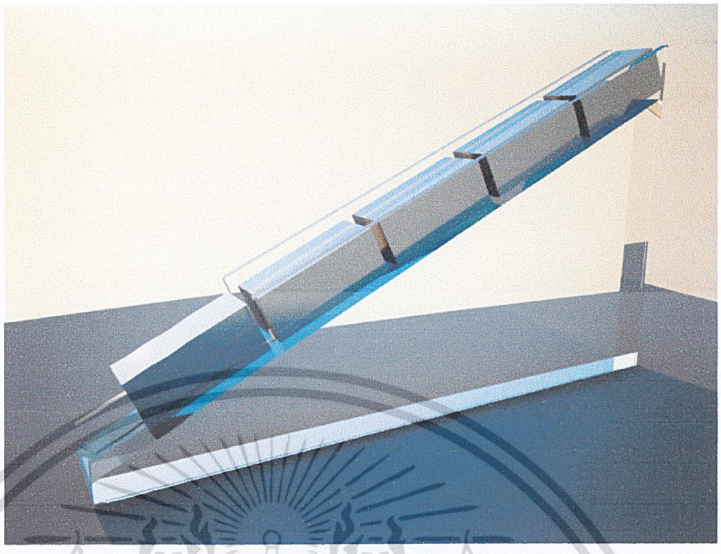


รูปที่ 2.6 แขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ท่อส่งสินค้า

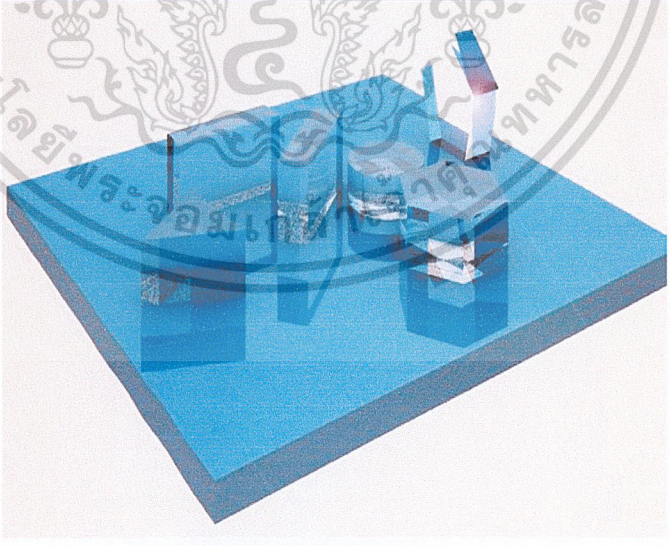
เพื่อความสะดวก มีระเบียบ ในการหยิบจับสินค้า



รูปที่ 2.7 ท่อส่งสินค้า

- คลังสินค้า

สถานที่เก็บสินค้าเพื่อความสะดวก ปลอดภัย และ ความมีระเบียบเรียบร้อย

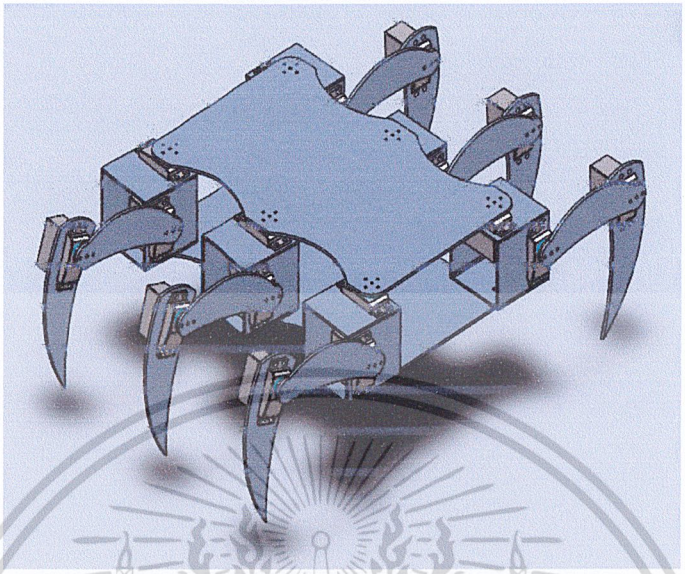


รูปที่ 2.8 คลังสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หุ่นยนต์แมลง 6 ขา

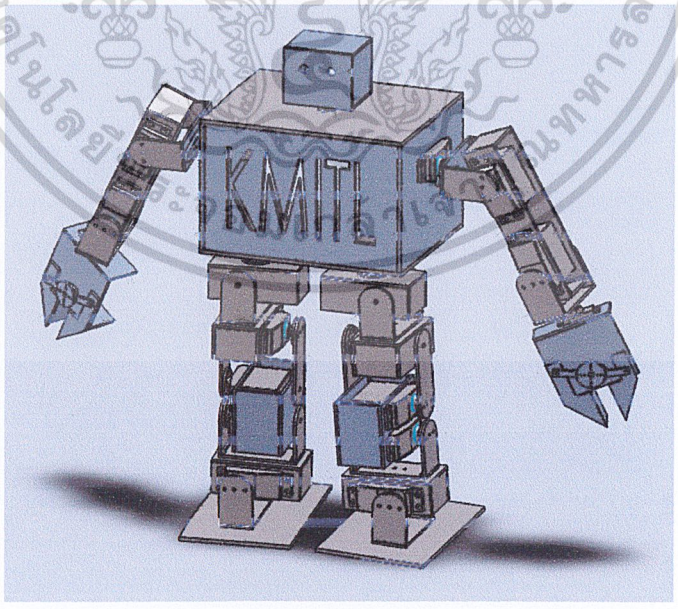
เคลื่อนย้ายสินค้าที่อยู่นอกเหนือเส้นทางของระบบได้อย่างอิสระ



รูปที่ 2.9 หุ่นยนต์แมลง 6 ขา

- หุ่นยนต์ 2 ขา

หยิบจับสินค้าเก็บเข้าคลังสินค้า หรือ เคลื่อนย้ายสินค้าได้อย่างอิสระ



รูปที่ 2.10 หุ่นยนต์ 2 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ระบบลำเลียง

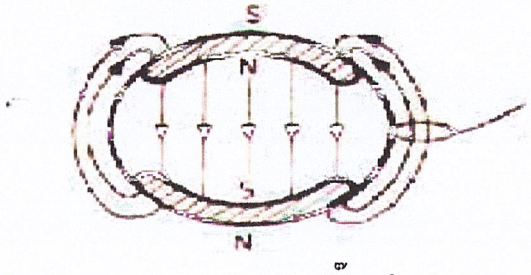
3.1 ระบบลำเลียงสินค้า

ระบบลำเลียงสินค้า(Conveyor System) ในธุรกิจการผลิตก็เปรียบเสมือนระบบประสาทของร่างกายมนุษย์ ซึ่งจะต้องทำงานเคียงเป็นนิตเสมอ เมื่อนั้นส่วนอื่นๆก็เมื่อเมื่อทำงานได้อย่างมีประสิทธิภาพ

สายพานลำเลียงคือ การลำเลียงสินค้า เพื่อให้ง่ายยิ่งขึ้น รูปแบบสายพานลำเลียงช่วยให้ทุ่นแรงในการลำเลียงสินค้า และทุ่นแรงงานอีกทั้งยังง่ายต่อ เพราะมีการใช้เครื่องสายพานลำเลียงนั้นทำให้ขนย้ายสินค้าในแนวราบได้ ทำให้เป็นอุปกรณ์ในผู้ผลิตที่ในปัจจุบันนี้มีการต่อสู้กันสูง สายพานลำเลียงจึงเป็นที่สำคัญในหมู่ผู้ผลิตขณะนี้ในด้านความเรียบร้อย โดยใช้สายพานลำเลียงซึ่งจะถูกควบคุมด้วยคอมพิวเตอร์หรือเรียกว่า CIM

3.2 มอเตอร์กระแสตรง

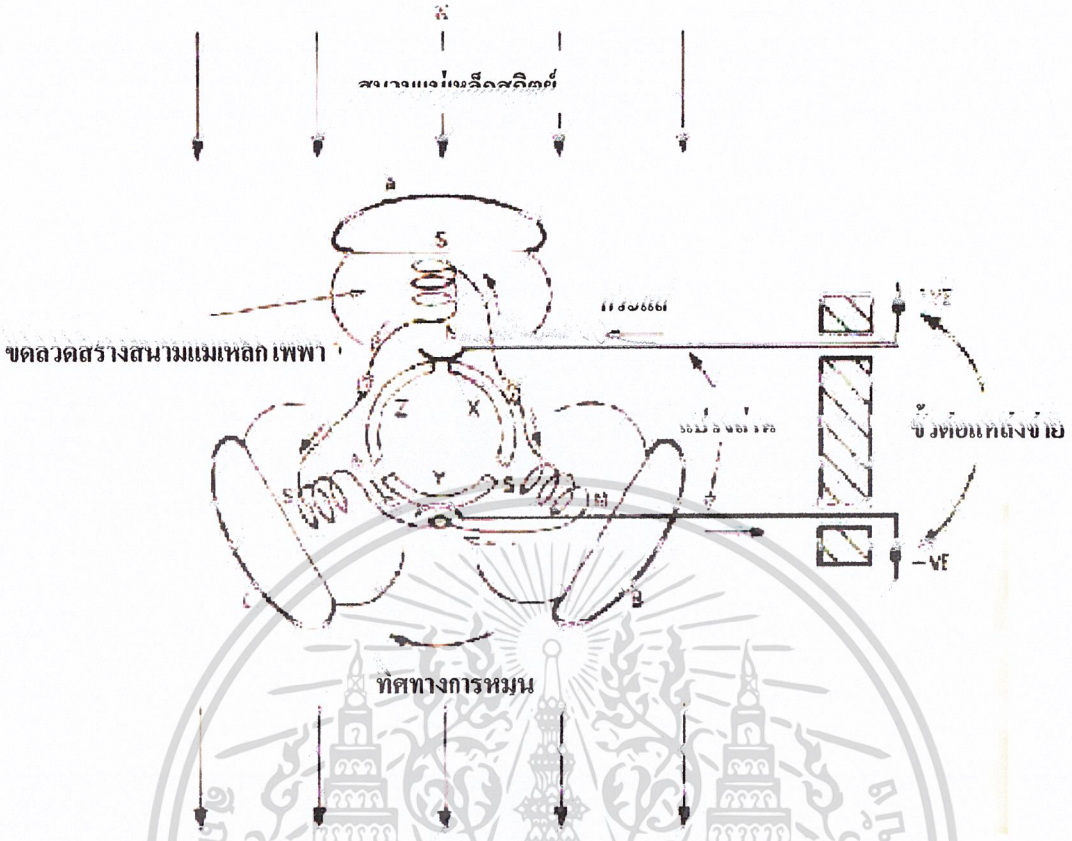
3.2.1 หลักการทำงานของมอเตอร์กระแสตรง



แม่เหล็กถาวรจะมีทางเดินของฟลักแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 โครงสร้างทั่วไปของมอเตอร์กระแสตรง



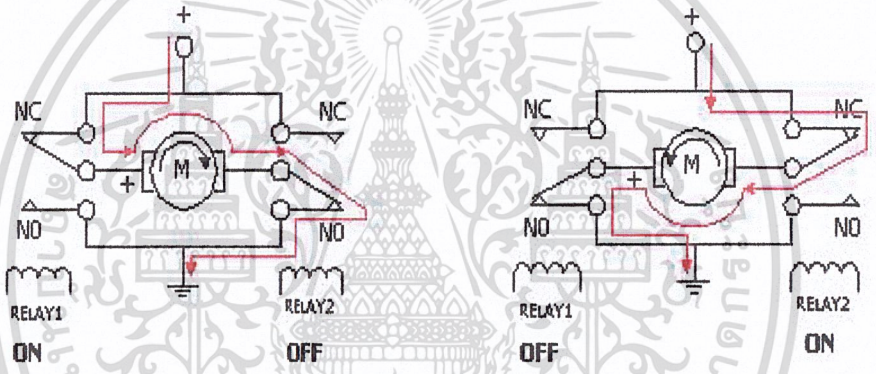
รูปที่ 3.2 โครงสร้างทั่วไปของมอเตอร์กระแสตรง

จากในรูปทางเดินของฟลั๊กซ์แม่เหล็ก และสนามแม่เหล็กจะเกิดจากแท่งแม่เหล็กเฟอร์ไรต์ 2 ชิ้นที่ขึ้นรูปเป็นแบบโค้งยึดติดกับตัวถังได้พอดี เพื่อที่จะให้เส้นแรงแม่เหล็กวิ่งเข้าสู่ใจกลางของมอเตอร์ได้ ดังนั้นความเข้มของแม่เหล็กจะขึ้นอยู่กับขนาดความหนาของแม่เหล็กด้วย ซึ่งส่งผลให้ฟลั๊กซ์แม่เหล็กวิ่งไปบนตัวถังโลหะ กระแสไฟฟ้าในขดลวดที่พันกับทุ่นโรเตอร์ก็จะทำให้เกิดสนามแม่เหล็กไฟฟ้า และต้านกับสนามแม่เหล็กถาวร จึงเกิดเป็นแรงบิดเพื่อที่จะหมุนทุ่นโรเตอร์ให้ไปในทิศทางเดียวกันกับทิศทางของสนามแม่เหล็กที่มีแรงมากกว่า กระแสก็จะไหลผ่านไปยังทุ่นโรเตอร์ โดยผ่านแปรงถ่าน ซึ่งจะสัมผัสกับแหวนตัวนำในทุ่นโรเตอร์ และแหวนคอมมิวเตเตอร์ ซึ่งจะถูกแบ่งออกเป็น 3 เซกเมนต์เพื่อที่จะทำหน้าที่นำกระแสเข้าขดลวดนั้นเอง

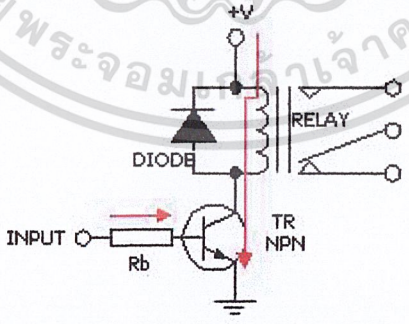
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การขับและกลับทิศทางของมอเตอร์กระแสตรง

ในการใช้ไอซีไมโครคอนโทรเลอร์เป็นตัวควบคุมการหมุน และทิศทางของมอเตอร์กระแสตรงนั้น เราจะต้องมีส่วนของวงจร ที่เรียกว่าวงจรขับมอเตอร์ (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 3.3 การกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์

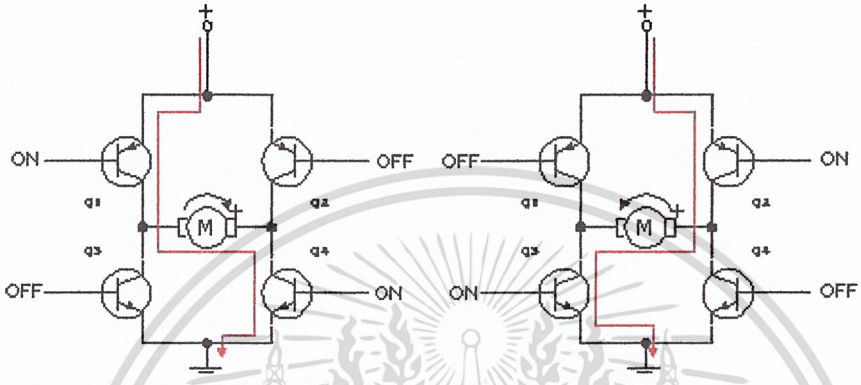


รูปที่ 3.4 การใช้ทรานซิสเตอร์เพื่อขับรีเลย์ให้ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

จากรูปเป็นวงจรขับรีเลย์โดยใช้ทรานซิสเตอร์ทำหน้าที่ขยายกระแส ด้วยเหตุผลเพราะไม่สามารถจะใช้ขา เอาต์พุตของไมโครคอนโทรลเลอร์ป้อนกระแสไฟที่ขดลวดของรีเลย์โดยตรงได้ เนื่องจากว่ากระแสที่จ่ายออกมาจากขา เอาต์พุตของไมโครคอนโทรลเลอร์มีค่าน้อยเกินไป ดังนั้นเราจึงต้องมีส่วนของวงจร ทรานซิสเตอร์เพื่อที่จะทำการขยายกระแสให้เพียงพอในการป้อนให้กับขดลวดของรีเลย์ ส่วนไดโอดนำมาต่อไว้ สำหรับป้องกันแรงดันย้อนกลับที่เกิดจากการเหนี่ยวนำของสนามแม่เหล็กในขณะเกิดการยุบตัว ซึ่งอาจจะทำให้ทรานซิสเตอร์เสียหายได้

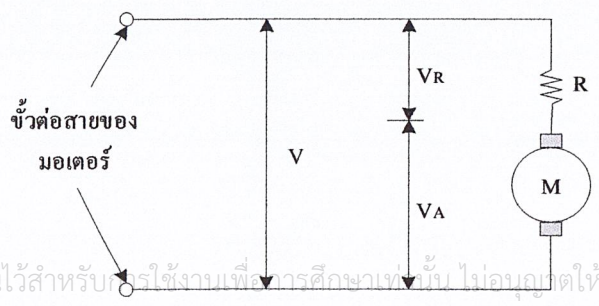


รูป 3.5 การใช้ทรานซิสเตอร์เป็นวงจรขับและกำหนดทิศทางของมอเตอร์กระแสตรง

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

3.2.3 คุณสมบัติของมอเตอร์กระแสตรง

ในการอธิบายคุณสมบัติของมอเตอร์กระแสตรงให้ละเอียดนั้นต้องพิจารณาแรงดันที่ป้อนและความต้านทานของโรเตอร์ด้วย วงจรภายในของมอเตอร์เขียนได้ดังรูปที่ 3.6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 วงจรภายในของมอเตอร์กระแสตรง

119324

โดยสมมติให้ทุ่นโรเตอร์ไม่มีความต้านทานอยู่เลย อนุกรมกับความต้านทานซึ่งในที่นี้ก็คือความต้านทานของขดลวดนั่นเอง แรงดันที่ขั้วต่อสายของมอเตอร์ก็คือผลบวกระหว่างแรงดันที่ทุ่นโรเตอร์ (V_A) และ แรงดันตกคร่อมความต้านทานขดลวด (V_R)

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

$$V = V_A + V_R \quad (V) \tag{3.1}$$

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

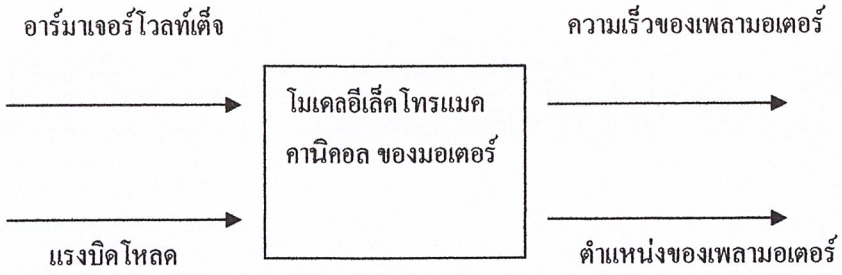
$$I = V_R / R \quad (A) \tag{3.2}$$

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

3.2.4 โมเดลคณิตศาสตร์ของดีซีมอเตอร์

ดีซีมอเตอร์ที่ใช้ร่วมกับดีซีแอมปลิไฟล์ทั้งในระบบการบังคับตำแหน่งและการบังคับความเร็วมักจะได้รับการประยุกต์ใช้เป็นส่วนประกอบสร้างกำลังงานในระบบการนำร่องและระบบบังคับต่างๆ และเนื่องจากวิทยาการเกี่ยวกับสารแม่เหล็กและการขยายด้วยโซลิดสเตททำให้ดีซีมอเตอร์แบบแม่เหล็กถาวรได้รับความนิยมใช้เป็นส่วนประกอบการขับเคลื่อนในระบบการบังคับแบบปิดลูฟต่างๆ มากขึ้น การออกแบบและการชดเชยระบบดังกล่าวได้อย่างเหมาะสมจะต้องใช้โมเดลทางคณิตศาสตร์ของส่วนประกอบทั้งหมดในระบบ

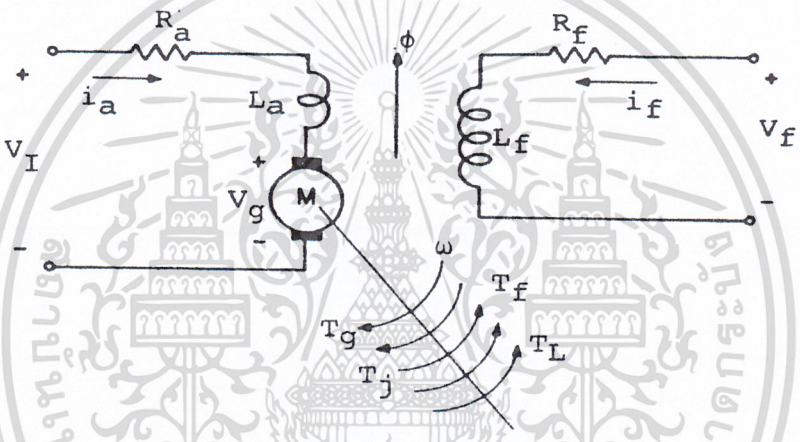
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 อินพุตและเอาต์พุตของโมเดลทางคณิตศาสตร์ของมอเตอร์

3.2.5 โมเดลอิเล็คโทรแมคคานิคอล

ส่วนสำคัญของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้นมีโมเดลดังแสดงในรูปที่ 3.8



รูปที่ 3.8 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

เมื่อ $\lambda(t)$ คือเส้นแรงแม่เหล็กที่ลิงเคจ(linkages) ไปยังขดลวดและ t คือเวลาในการหมุนของคอมนิวเทเตอร์ของมอเตอร์ การควบคุมวงจรของแต่ละส่วนของตัวนำในโรเตอร์จะเกิดโวลท์เต็จขึ้นในส่วนของตัวนำนั้นตามสมการ (3.4) เมื่อ $\frac{d\lambda(t)}{dt}$ จะเป็นสัดส่วนต่อเส้นแรงแม่เหล็กในช่องว่างอากาศและความเร็วเชิงมุม $\omega(t)$ เราจะได้ว่า

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

สมมติให้กระแสของฟิลด์มีค่าคงที่และไม่คิดถึงส่วนการเปลี่ยนแปลงในเส้นแรงฟิลด์เนื่องจากอาร์เมเจอร์รีแอคชั่นเส้นแรงฟิลด์ก็จะมีค่าคงที่ดังนั้นสมการ (3.5) ก็จะเป็น

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

เมื่อเราสมมติให้เส้นแรงของฟิลมีค่าคงที่ แรงบิดของแม่เหล็กไฟฟ้าซึ่งเกิดขึ้นแก่โรเตอร์ของมอเตอร์จะเป็นสัดส่วนกับกระแสของอาร์มาเจอร์

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

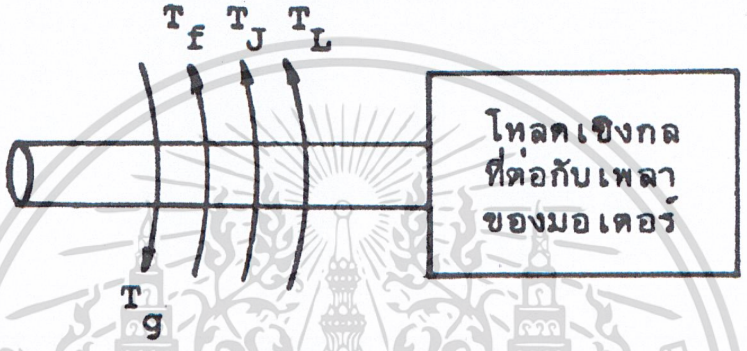
เมื่อ K_t คือ ค่าคงที่ของแรงบิดของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์คือผลคูณของแรงบิดที่เกิดขึ้นและความเร็วเชิงมุม

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

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์ทั้งหมดนี้จะจ่ายไปยังโหลดที่ต่ออยู่กับเพลาของมอเตอร์แต่กำลังงานนี้บางส่วนจะสูญเสียไปในมอเตอร์ การสูญเสียจากแรงเสียดทาน หมายถึงความหน่วงเนื่องจากลมที่มีต่อโรเตอร์ แรงเสียดทานตัวรองรับโรเตอร์ กระแสที่ไหลวนในเหล็กของโรเตอร์และฮีสเทรีซิส โดยแรงบิดต่างๆแสดงดังนี้



รูปที่ 3.9 แรงบิดต่างๆที่เกิดขึ้นต่อโหลดของมอเตอร์

$T_g(t)$: 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลรวมของแรงบิดเสียดทานที่ประกอบกันขึ้นที่เพลาของมอเตอร์ซึ่งเป็นลิเนียร์ฟังก์ชันกับความเร็วเชิงมุมของโรเตอร์เรียกว่า ส่วนประกอบของวิสคอสฟริคชั่นและมักจะอยู่ในเทอมที่แยกออกจากฟริคชั่นอื่นๆ ซึ่งแสดงได้ด้วยสมการต่อไปนี้

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

เมื่อ B คือสัมประสิทธิ์ของวิสคอสฟริคชั่นของมอเตอร์และโหลดที่ต่ออยู่กับเพลาของมอเตอร์ $T_f(t)$ คือผลรวมของฟริคชั่นของโหลดและมอเตอร์ทั้งหมด มีแรงต้านของลมและการสูญเสียกำลังในเหล็กของเพลามอเตอร์ยกเว้นวิสคอสฟริคชั่น

สมการ (1) (4) (5) และ (8) เป็นชุดสมการพื้นฐานของดีซีมอเตอร์โมเดลและสมการเหล่านี้เราสามารถจะหาทรานสเฟอร์ฟังก์ชั่นของดีซีมอเตอร์ได้ โดยใส่ลาปลาสทรานสฟอร์มทั้งสองข้างของชุดสมการพื้นฐานได้เป็น

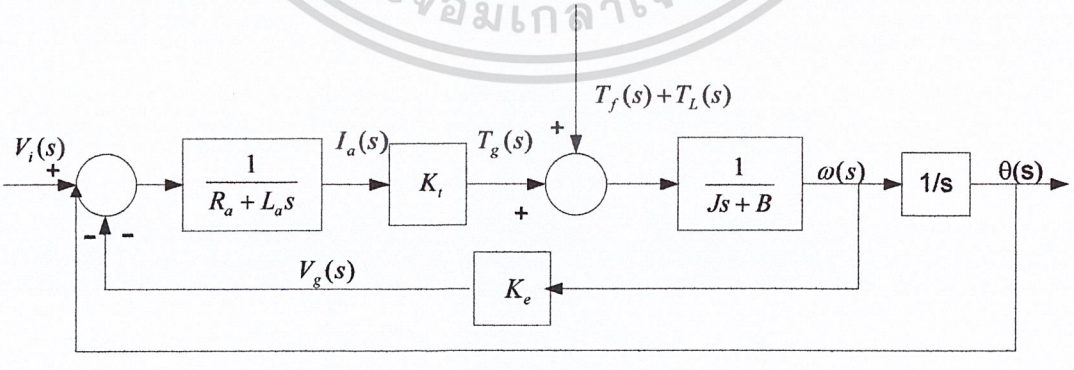
$$V_i(s) - V_g(s) = (R_a + sL_a)\,I_a(s) \qquad (3.11)$$

$$V_g(s) = K_e\omega(s) \qquad (3.12)$$

$$T_g(s) = K_t\,I_a(s) \qquad (3.13)$$

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

สามารถเขียนเป็นบล็อกไดอะแกรมที่แสดงสมการพื้นฐานเหล่านี้ได้ดังนี้



รูปที่ 3.10 บล็อกไดอะแกรมของดีซีมอเตอร์โมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสังเกต

สมมติว่าโวลท์เต็จที่ป้อนให้กับวงจรอาร์มาเจอร์ของมอเตอร์มีค่าคงที่ดังนั้นมอเตอร์จะหมุนด้วยความเร็วคงที่คือทำงานอยู่ที่สภาวะสงบนิ่งด้วยโหลดที่คงที่ กำลังงานเชิงกลที่เกิดขึ้นโดยโรเตอร์จะหาได้จากสมการ 3.8 จะได้

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

เมื่อทุกเทอมในสมการสุดท้ายมีค่าคงที่เนื่องจากมอเตอร์ทำงานอยู่ที่สภาวะสงบนิ่ง กำลังไฟฟ้าที่ถูกดูดกลืนโดยอาร์เมเจอร์ต้องเท่ากับ

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

ดังนั้นเราจะได้ว่ากำลังงานเชิงกลที่เกิดขึ้นต้องเท่ากับกำลังงานไฟฟ้าที่ถูกดูดกลืนในโรเตอร์คือสรุปได้ว่า $K_e = K_t$ ทรานสเฟอร์ฟังก์ชั่นของดีซีมอเตอร์บล็อกไดอะแกรมของรูปที่ 3.10 แสดงถึงระบบที่มีสองอินพุท และมีเอาต์พุทเป็นความเร็วเชิงมุม $\omega(s)$ และการเคลื่อนที่แบบเชิงมุม จากรูปที่ 3.10 ความเร็วเอาท์พุทของระบบเขียนได้เป็น

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

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

$$G_1(s) = \frac{\omega(s)}{V_i(s)} = \frac{K_t}{(L_a s + R_a)(Js + B) + K_t K_e}$$

$$= \frac{K_m}{\alpha S^2 + \beta s + 1} \tag{3.20}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ

$$K_m = \frac{K_t}{R_a B + K_t K_e}$$

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

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

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

$$G_1(s) = \frac{K_t}{R_a B(1+\tau_e s)(1+\tau_m s) + K_t K_e} \tag{3.21}$$

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

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

ถ้าอินดัคแต้นท์ของอาร์มาเจอร์มีค่าน้อย ไทม์คอนสแต้นท์ทางไฟฟ้าสามารถตัดทิ้งได้และได้สมการเป็น

$$G_V(s) = \frac{\omega(s)}{V_i(s)} = \frac{K_t}{R_a(Js+B) + K_t K_e} \tag{3.22}$$

$$= \frac{K_m}{\varpi s + 1} \tag{3.23}$$

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

ในสมการ (3.23) ค่าคงที่ K_m อาจเรียกได้ว่าเป็นค่าคงที่ของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานสเฟอร์ฟังก์ชั่นแรงบิดโหลด $G_2(s)$ หาได้เป็น

$$G_2(s)=\frac{\omega(s)}{T_f(s)+T_L(s)}=\frac{-\frac{1}{Js+B}}{1+\frac{K_tK_e}{(Js+B)(L_as+R_a)}}$$

$$=\frac{-\frac{R_a}{K_t}K_m\left[\frac{L_a}{R_a}s+1\right]}{\alpha s^2+\beta s+1} \qquad (3.24)$$

ซึ่งถ้าอินดัคแต้นท์ของอาร์มาเจอร์ไม่นำมาคิด จะทำให้ได้สมการ ดังนี้

$$G_L(s)=\frac{\omega(s)}{T_f(s)+T_L(s)}=\frac{-\frac{R_a}{K_t}K_m}{\varpi s+1} \qquad (3.25)$$

ซึ่งจากสมการที่ (3.16) เมื่อให้ค่าของ ว่า T_f และ T_L มีค่าเป็นศูนย์จะทำให้ค่าทรานสเฟอร์ฟังก์ชั่นมีค่าดังนี้

$$\omega(s)=G_v(s)V_i(s)=\frac{K_m}{\varpi s+1}V_i(s) \qquad (3.26)$$

โมเดลคณิตศาสตร์ในการควบคุมตำแหน่งของมอเตอร์ จาก transfer function โมเดลคณิตศาสตร์ของดีซีมอเตอร์ ซึ่งจะได้ model ของการควบคุมความเร็วของมอเตอร์ในรูปของสมการอันดับหนึ่งเป็นดังนี้ คือ

$$\omega(s)=G_v(s)V_i(s)=\frac{K_m}{\varpi s+1}V_i(s) \qquad (3.27)$$

จากสมการข้างต้นดังกล่าวนั้นจะเห็นถึงความสัมพันธ์ระหว่างความเร็วเชิงมุม (output) และ ค่าแรงดันที่ป้อน (input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในการควบคุมตำแหน่งจะมีการผ่านตัว Integrator (1/s) ทำให้ได้เอาท์พุตคือ $\Theta(s)$ ซึ่งจากรูปที่ 3.10 และ transfer function ที่ได้จากหัวข้อที่กล่าวข้างต้นนั้นเมื่อทำการผ่าน Integrator (1/s) เข้าไปจะทำให้ได้ค่าของ output เป็นมุมในการเคลื่อนที่ของมอเตอร์ ซึ่งเขียนในรูปของสมการได้ดังนี้

$$\theta(s) = \frac{\omega(s)}{s} = \left[\frac{1}{s}\right]\left[\frac{K_m V_i(s)}{\tau s + 1}\right] \tag{3.28}$$

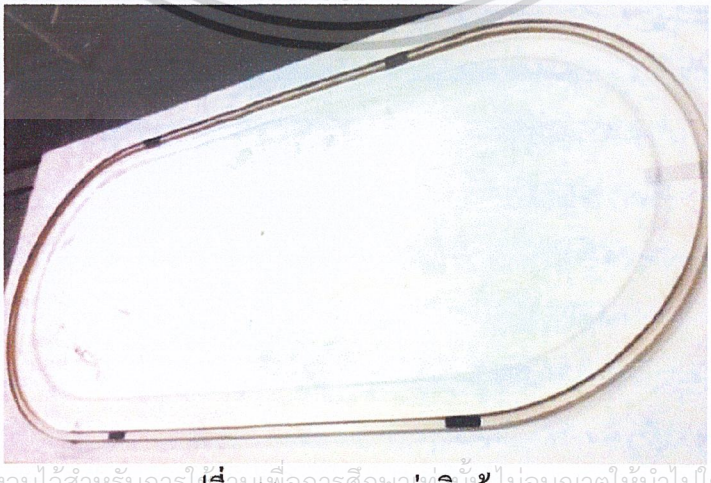
ส่วนในการหาโมเดลมอเตอร์ที่ใช้ในการทดลองนั้น เรากำหนดให้โมเดลของมอเตอร์เป็นโมเดลมอเตอร์อันดับหนึ่ง(First Order)ซึ่งเราได้ทำการทดลองเพื่อหาโมเดลที่เป็นลักษณะอันดับหนึ่ง

3.3 การออกแบบ

การออกแบบระบบลำเลียงสินค้าที่ดีนั้นจะต้องออกแบบแต่ละระบบเพื่อตอบสนองต่อความต้องการ การออกแบบผลิตภัณฑ์แต่ละตัวจะต้องมีประสิทธิภาพ ความปลอดภัย และใช้ประโยชน์สูงสุดในการดำเนินงาน โดยที่ระบบลำเลียงนั้นต้องรองรับต่อรูปแบบการใช้งานมากพอๆกับความสะดวกในการขยายระบบในอนาคต สำหรับการออกแบบระบบจำลองนี้จะคำนึงถึงความเป็นไปได้ของเทคโนโลยี ขั้นตอนการทำงาน การเชื่อมต่อกันของหน่วยต่างๆในระบบและเพื่อให้ประหยัดค่าใช้จ่ายแล้วจะต้องคำนึงถึงการเลือกใช้เทคโนโลยี อุปกรณ์ที่จะใช้ในอย่างเหมาะสม

3.3.1 การเลือกใช้

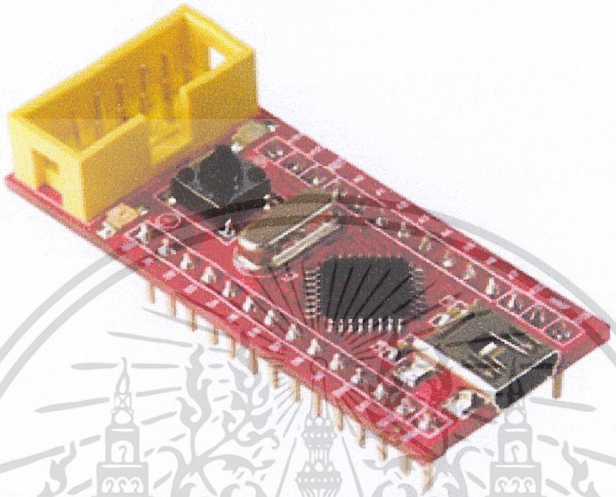
1.รางสำหรับรถขนส่งสินค้าจะใช้รางที่ติดทองแดงเพื่อเป็นสื่อกลางในการนำไฟฟ้าจาก Power Suppy ที่เป็นตัวจ่ายไฟฟ้าไปยังรถขนส่งสินค้า แทนการใช้พลังงานจากเซลล์แห้ง (ถ่าน / แบตเตอรี่) ซึ่งจะทำให้สินเปลืองเนื่องจากการในการทำงานจะต้องมีการทดสอบระบบหลายครั้ง



รูปที่ 3.11 รางรถขนส่งสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ในส่วนของแขนกล ET-EASY168 STAMP เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR8 ขนาดเล็ก และได้ทำการเลือกใช้ IC เบอร์ ATMEGA168เป็น MCU ประจำบอร์ด นอกจากนี้แล้วภายในตัวบอร์ดยังได้รวมเอาไอซี USB BRIDGE ของ FTDI เบอร์ FT232R สามารถติดต่อกับคอมพิวเตอร์ PC ผ่านทาง PORT USB ได้โดยตรงทำให้บอร์ด ET-EASY168 STAMPเป็นบอร์ดทดลองใช้งานขนาดเล็กที่เพียบพร้อมไปด้วยวงจร



รูปที่ 3.12 ลักษณะของ ET-EASY168 STAMP

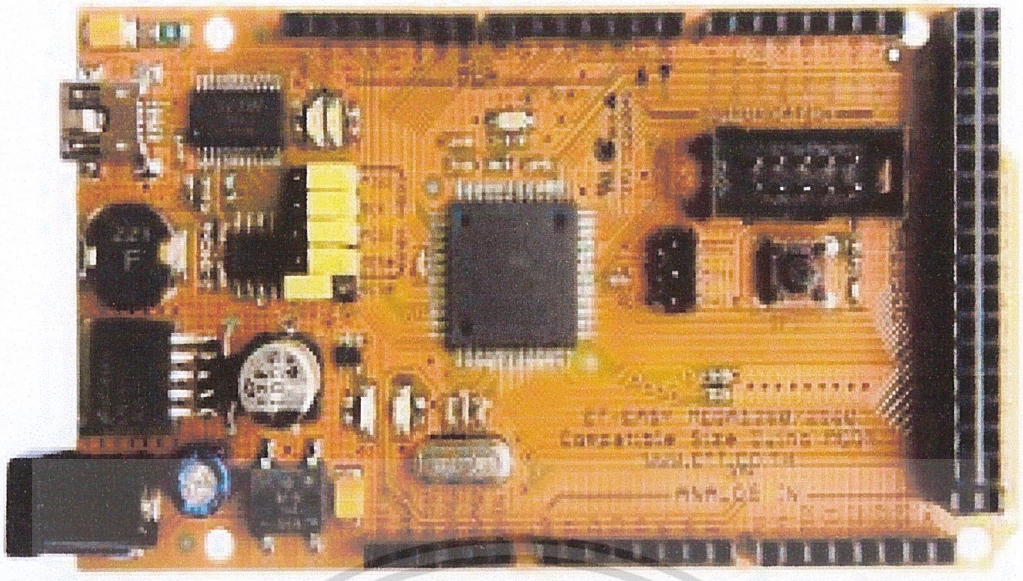
สาเหตุการเลือกใช้ ET-EASY168 STAMP

- เพราะตัว บอร์ดมีขนาดเล็ก ทำให้เราสามารถใช้งานในพื้นที่จำกัดได้
- ความสะดวกในการใช้งานเพราะเราสามารถเชื่อมต่อตัวบอร์ดเข้ากับเครื่องคอมพิวเตอร์ โดยผ่านทางสาย USB

3. Easy Mega1280

Easy Mega1280เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR ที่ได้รับการพัฒนาขึ้นมาจากรุ่น ET-EASY168 STAMP ซึ่งตัวบอร์ดมีความสามารถมากขึ้นสามารถรองรับการทำงานของระบบใหญ่ๆได้ ตัวบอร์ดได้เลือกใช้ ATMEGA1280 เป็น MCU ประจำบอร์ด และยังได้รวมเอา IC เบอร์ FT232RLของFTDI ทำให้สามารถติดต่อกับคอมพิวเตอร์PC ผ่านทาง PORT USB ได้โดยตรงทำให้ ตัวบอร์ด Easy Mega1280 ง่ายต่อการใช้งานและสะดวกในการโปรแกรมข้อมูล โดยลักษณะของ Et-easy Mega1280 เป็นดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

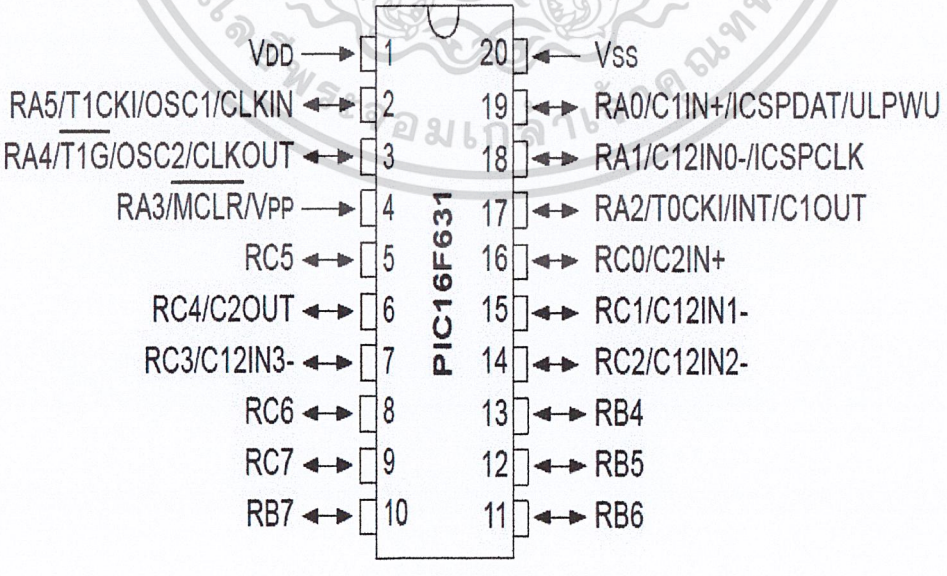


รูปที่ 3.13 ลักษณะของ Easy Mega1280

สาเหตุการเลือกใช้ Easy Mega1280

- เป็นบอร์ดที่มี พอร์ต ดิจิตอลและ อนาล็อกเป็นจำนวนมากเหมาะสำหรับระบบใหญ่
- ความสะดวกในการใช้งานเพราะเราสามารถเชื่อมต่อตัวบอร์ดเข้ากับ เครื่องคอมพิวเตอร์โดยผ่านทางสาย USB

4. PIC16F690



รูปที่ 3.14 ลักษณะของ PIC16F690

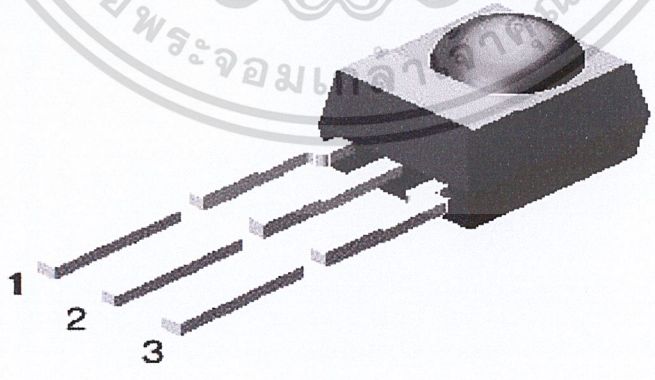
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไมโครคอนโทรลเลอร์ที่นิยมใช้โดยทั่วไปพื้นฐานการทำงานก็คือ ระบบดิจิตอลโดยค่าเอาท์พุตที่ได้จากไมโครคอนโทรลเลอร์ จะเป็น 0 กับ 1 แต่ก็สามารถนำมาประยุกต์เชื่อมต่อกับอุปกรณ์ภายนอกได้มากมาย

ลักษณะโดยทั่วไปจากรูปที่ 3.14 จากรูปขาของ PIC16F690 แต่ละขาจะมีหน้าที่แตกต่างกันไปซึ่งแยกออกเป็น PORT A , PORT B , PORT C โดยพื้นฐานแต่ละพอร์ตสามารถทำงานเป็นอินพุตและเอาท์พุต สำหรับไมโครคอนโทรลเลอร์ที่ใช้ต้องมีสัญญาณนาฬิกาภายในตัวเพื่อสร้างพัลซ์มาป้อนให้กับมอเตอร์เพื่อที่จะคุมให้มอเตอร์ทำงานตามที่เราต้องการซึ่งใน PIC16F690 มีสัญญาณนาฬิกาภายในตัวอยู่แล้วโดยที่หนึ่งไซเคิล(Clock Bus) ของซีพียูจะประกอบไปด้วย สัญญาณนาฬิกาภายนอกจำนวน 4 ไซเคิล คือ Q1,Q2,Q3 และ Q4 ดังนั้นความถี่ซีพียูประมวลผลต่อหนึ่งคำสั่งเท่ากับความถี่ของสัญญาณนาฬิกาภายนอกหารด้วย4หรือหากจะพิจารณาความเร็วของไมโครคอนโทรลเลอร์ตระกูล PIC สามารถประมวลผลต่อหนึ่งคำสั่งเท่ากับ 1/4 เท่าของความถี่ออสซิเลเตอร์ภายนอก

5. TSOP34836

เป็นอุปกรณ์ขนาดเล็กที่ใช้ในการรับสัญญาณอินฟราเรดโดยข้อมูลจะถูกส่งผ่านหลอดอินฟราเรด ซึ่งการส่งข้อมูลผ่านอินฟราเรดเราต้องทำการเลือกความถี่ช่วงสัญญาณ เพื่อที่จะทำการเลือกตัวรับได้ว่า จะให้รับสัญญาณอินฟราเรดในช่วงใด ซึ่ง TSOP34836 จะสามารถรับสัญญาณอินฟราเรดได้ในช่วง 38KHz โดยลักษณะของ TSOP34836 เป็นดังรูปที่ 3.15



รูปที่ 3.15 ลักษณะของ TSOP34836

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาเหตุการเลือกใช้ TSOP34836

- ความต้องการกระแสต่ำ
- มีความม่นยำในการส่งข้อมูลสูง
- ราคาถูก

3.4 การติดต่อสื่อสารของระบบ

Computer กับ Main Controller : จะใช้สาย USB ในการเชื่อมต่อ

Main Controller กับ สถานี : Main Controller จะควบคุมสถานีทั้ง 4 ด้วย AVR ATmega 1280 เขียนโปรแกรมด้วยภาษา C และใช้สายแพในการเชื่อมต่อข้อมูล

สถานี กับ รถ : สถานีและรถถูกควบคุมด้วย AVR ATmega8L PU ซึ่งระหว่างรถกับสถานี จะสื่อสารกันผ่าน Infrared sensor โดยมีตัวส่ง คือ หลอดอินฟาเรด และตัวรับ คือ โมดูลรับสำเร็จรูป (3ขา) จะส่งด้วยความถี่ 40 KHz

รถ กับ แขนกล : แขนกลจะถูกควบคุมด้วย AVR ATmega168 โดยโปรแกรม Arduino เขียนภาษา C ซึ่งแขนกับรถจะทำงานสัมพันธ์กัน โดยใช้การหน่วงเวลา(ในระบบ auto)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การทำงานของระบบ

การทำงานของระบบแบ่งการทำงานออกเป็น 2 แบบ คือ

1. โหมดควบคุมด้วยมือ (Manual mode)

ในการทำงานโหมดควบคุมด้วยมือ นั้นเราจะเป็นผู้สั่งการได้ว่า จะให้รถไปเริ่มต้นที่สถานีใด โดยที่เราจะเป็นผู้กำหนดได้เองถึงสถานีต้นทาง และ สถานีปลายทาง

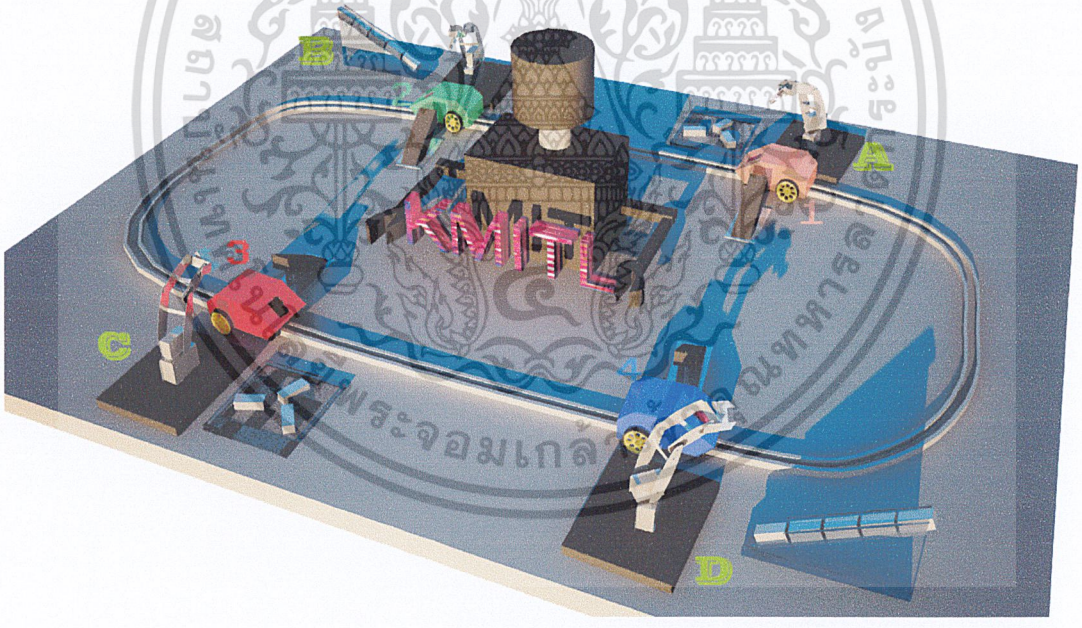
2. โหมดควบคุมอัตโนมัติ (Auto mode)

สำหรับการทำงานในโหมดควบคุมอัตโนมัติ นั้นจะมีลำดับการทำงานดังนี้

1). รถทั้ง 4 คันจะเข้ามาจอดประจำสถานี

2). แขนกลจะทำงานโดยแบ่งการทำงานออกเป็น 2 แบบ คือ แขนกลสถานี A กับ C จะนำสินค้าออกจากหลังรถ แขนกลสถานี B กับ D จะหยิบของมาวางไว้หลังรถ

3). เมื่อแขนกลทำงานเสร็จ รถทั้ง 4 คัน จะเคลื่อนที่ไปจอดสถานีต่อไป



รูปที่ 3.16 ภาพรวมการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ขั้นตอนการทำงาน

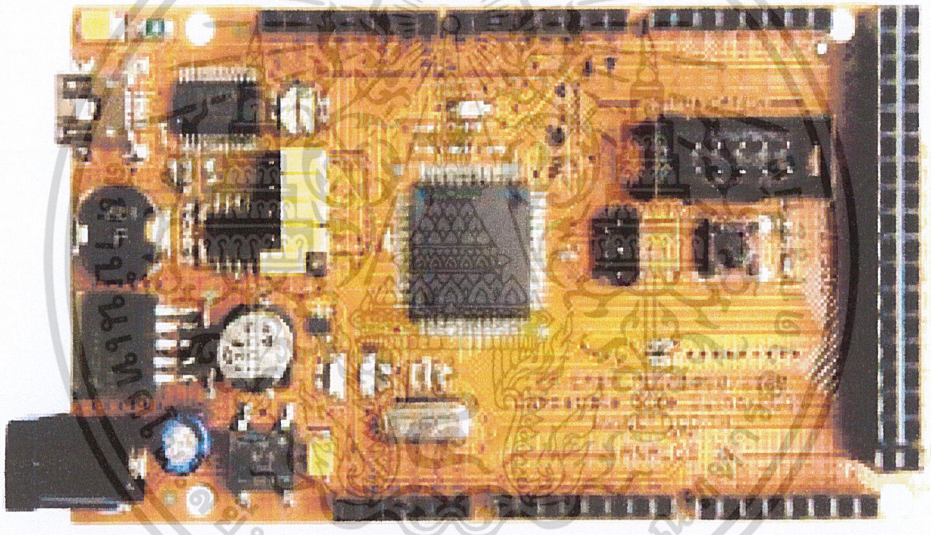
รูปที่ 3.17 ขั้นตอนการทำงานของระบบลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 โครงสร้างของระบบ

1. Main Controller ใช้ " Et-easy mega1280 "

ในบอร์ดรุ่น ET-EASY MEGA1280 (DUINO MEGA) ของบริษัทอีทีที ได้เลือกใช้ชิพไมโครคอนโทรลเลอร์ ATmega1280 ที่เป็นชิพตระกูล AVR รองรับการเขียนโปรแกรมภาษาซีของ Arduino ได้ทันทีมาเป็นหน่วยประมวลผลหลัก โดยชิพรุ่นนี้มีหน่วยความจำแฟลชสำหรับเก็บเขียนโปรแกรม 128 กิโลไบต์ มีหน่วยความจาแรม 8 กิโลไบต์ มี EEPROM อีก 4 กิโลไบต์สาหรับใช้เป็นที่เก็บข้อมูลถาวรได้เมื่อไม่มีไฟเลี้ยง และยังมีขานำเข้า/ส่งออก หรือไอ/โอ (I/O) สาหรับต่อใช้งานทั่วไป 86 ขา มี PWM (สำหรับควบคุมอัตราการหมุนของมอเตอร์) ที่กำหนดความละเอียดได้ระดับ 16 บิตให้ใช้งานถึง 12 ช่องสัญญาณ มีช่องสื่อสารแบบอนุกรม 4 พอร์ต และสามารถแปลงสัญญาณแอนาล็อกเป็นดิจิทัล (ADC) อีก 16 ช่องสัญญาณ ซึ่งจะเห็นว่ามีความสามารถพื้นฐานที่มากพอสำหรับงานควบคุมที่หลากหลาย



รูปที่ 3.18 ลักษณะของ Easy Mega1280

คุณสมบัติของบอร์ดEasy Mega1280

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ประสิทธิภาพสูงแต่ใช้พลังงานต่ ในตระกูล AVR®
- สถาปัตยกรรมแบบ RISC
  - มีชุดคาสั่ง 135 คาสั่ง และส่วนใหญ่คาสั่งเหล่านี้จะใช้เพียง 1 สัญญาณนาฬิกาในการประมวลผลคาสั่ง
  - มีเรจิสเตอร์สาหรับใช้งานทั่วไปขนาด 8 บิต จานวน 32 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

– ทางานได้สูงสุดที่ 16 ล้านคาสั่งต่อวินาที (MIPS) เมื่อใช้สัญญาณนาฬิกา 16 เมกะเฮิร์ซ (MHz)

• หน่วยความจำ

– หน่วยความจำแฟลชสาหรับโปรแกรมขนาด 128 กิโลไบต์ เขียน/ลบได้ 10,000 ครั้ง

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

– หน่วยความจาแรมชนิดเอสแรม (SRAM) ขนาด 8 กิโลไบต์

– เก็บข้อมูลได้กว่า 20 ปีที่อุณหภูมิ 85°C และกว่า 100 ปีที่อุณหภูมิ 25°C

• มีระบบโปรแกรมตัวเองอยู่ในตัวชิพ

• สามารถทาการอ่านขณะเขียนได้จริง โดยสามารถล็อกการทางานได้เพื่อความปลอดภัยของซอฟต์แวร์

• มีการเชื่อมประสานกับ JTAG (IEEE std. 1149.1 compliant)

• คุณสมบัติการเชื่อมต่อกับอุปกรณ์ภายนอก

– มีตัวตั้งเวลาและตัวนับขนาด 8 บิต จานวน 2 ตัว ที่สามารถแยกโหมดการทางานจากกันได้ 2 โหมด คือ Prescalar และ Compare

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

– มีตัวนับแบบเวลาจริง (Real Time Counter) ที่แยกวงจรกำหนดความถี่ได้

– มี PWM จานวน 12 ช่องสัญญาณที่สามารถกำหนดความละเอียดได้ 16 บิต

– มีตัวปรับผลการเปรียบเทียบของเอาต์พุต

–มีตัวแปลงสัญญาณแอนาล็อกให้เป็นดิจิทัลขนาด 10 บิต จานวน 16 ช่องสัญญาณ

– มีพอร์ตสื่อสารอนุกรมที่สามารถกำหนดอัตราการรับ/ส่งได้จานวน 4 พอร์ต

–เชื่อมประสานอนุกรมแบบ SPI ได้ทั้งการเป็นมาสเตอร์และสเลฟ (Master/Slave)

– มีการเชื่อมประสานแบบอนุกรมด้วยสายสัญญาณ 2 เส้นแบบ ส่งข้อมูลแบบเรียงไบต์ (Byte Oriented)

– มีตัวตั้งเวลาแบบวอตช์ด็อกที่สามารถกำหนดการทางานได้โดยสามารถแยกสัญญาณนาฬิกาได้จากตัวชิพ

–มีตัวแปรียบเทียบสัญญาณแบบอนาล็อกอยู่ในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

– มีการรองรับการขัดจังหวะและการเวก-อัพ (Wake-up) เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับขาของชิพ

• คุณสมบัติพิเศษ

– มีระบบเริ่มระบบเมื่อมีการรีเซ็ตและมีระบบตรวจจับการเกิดบราวน์เอาท์ (Brown-out) ที่สามารถกำหนดการทางานได้

– มีตัวตรวจหาความเที่ยงตรงของออสซิเลเตอร์อยู่ในตัว (Internal Calibrated Oscillator)

– มีแหล่งการขัดจังหวะทั้งภายในและภายนอก (External and Internal Interrupt Sources)

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

• ไอ/โอ และตัวถัง

– มีขาของไอ/โอที่สามารถกำหนดการทางานได้ 86 ขา

– ตัวถังแบบ TQFP ชนิด 100 ขา

• ช่วงอุณหภูมิที่ชิพทางานได้ -40°C ถึง 85°C

• การใช้พลังงาน

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

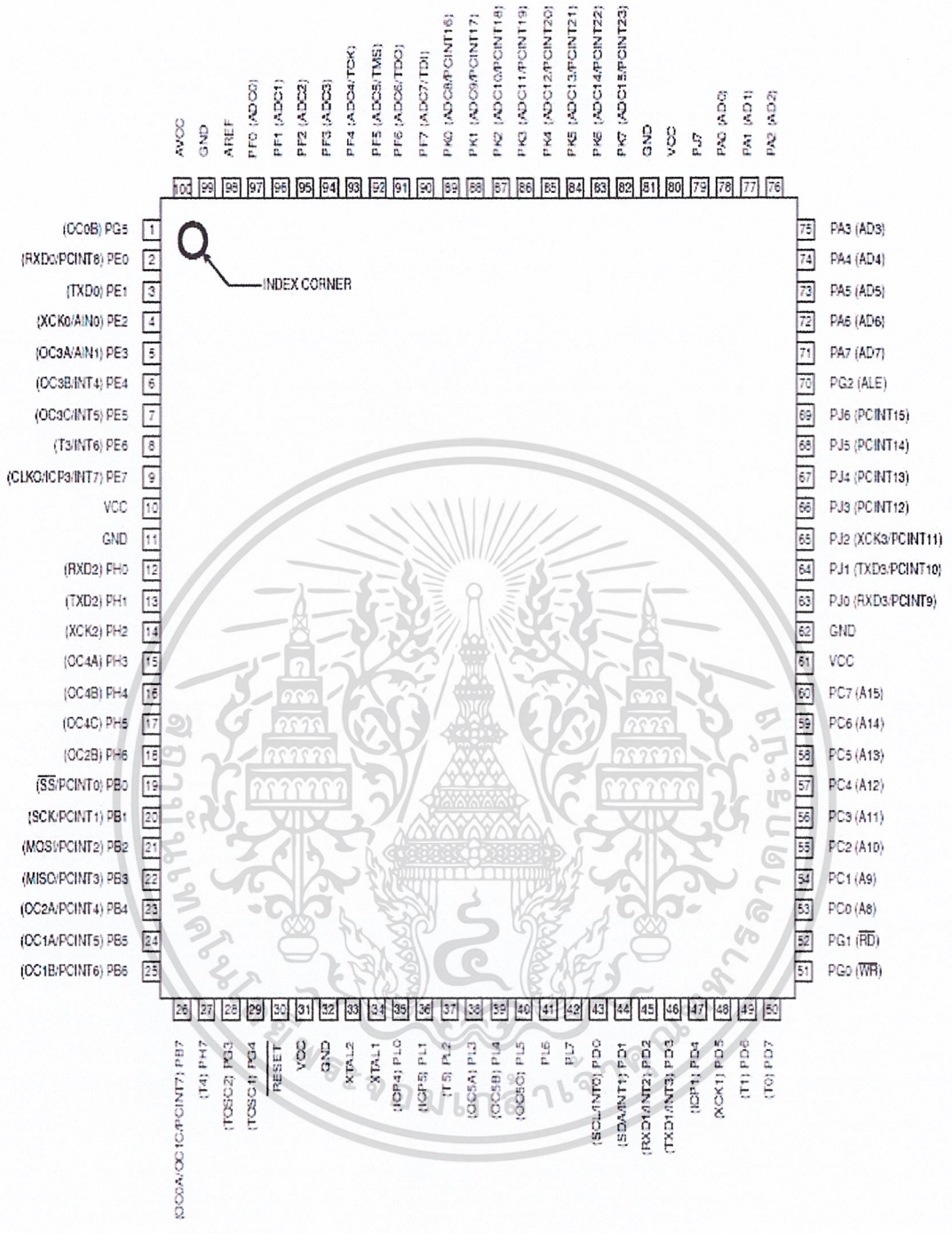
– โหมดเพาเวอร์ดาวน์ (Power-down) ต้องการกระแสเพียง 0.1 μA ที่แรงดัน 1.8V

สาเหตุที่เลือกใช้ Easy Mega1280

-เป็นบอร์ดที่มีพอร์ตดิจิตอลและอนาล็อกเป็นจำนวนมากเหมาะสำหรับระบบใหญ่

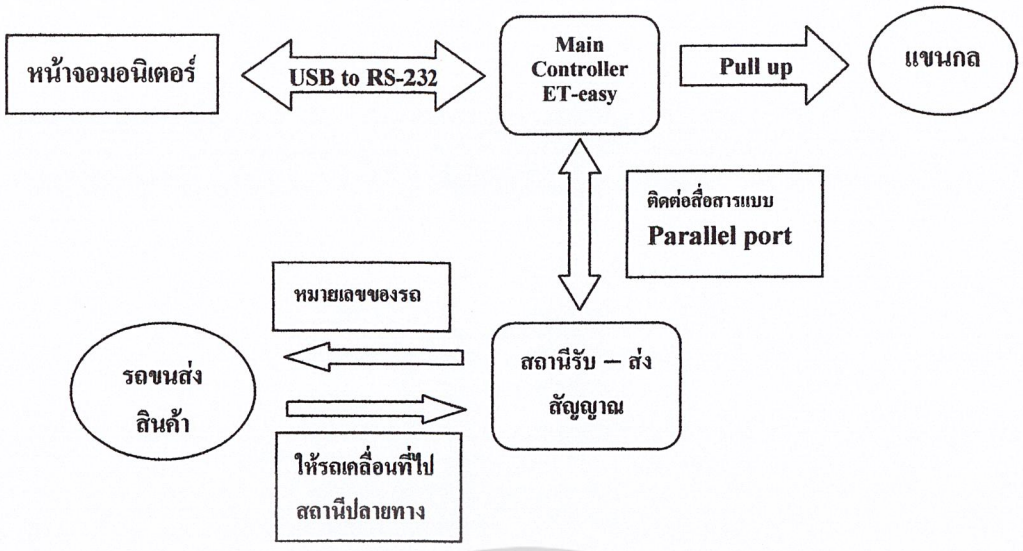
-ความสะดวกในการใช้งานเพราะเราสามารถเชื่อมต่อตัวบอร์ดเข้ากับ เครื่องคอมพิวเตอร์โดยผ่านทางสาย USB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



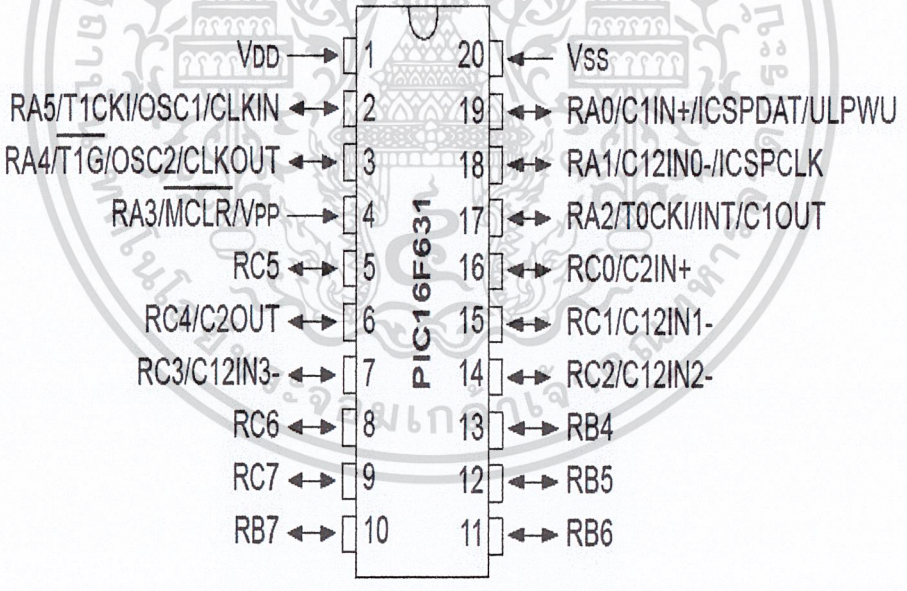
รูปที่ 3.19 การจัดเรียงขาของ AT mega 1280

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 แผนภาพแสดงการทำงานของส่วน Main Controller

2. รถลำเลียงใช้ “ PIC16F690 ”



รูปที่ 3.21 ลักษณะของ PIC16F690

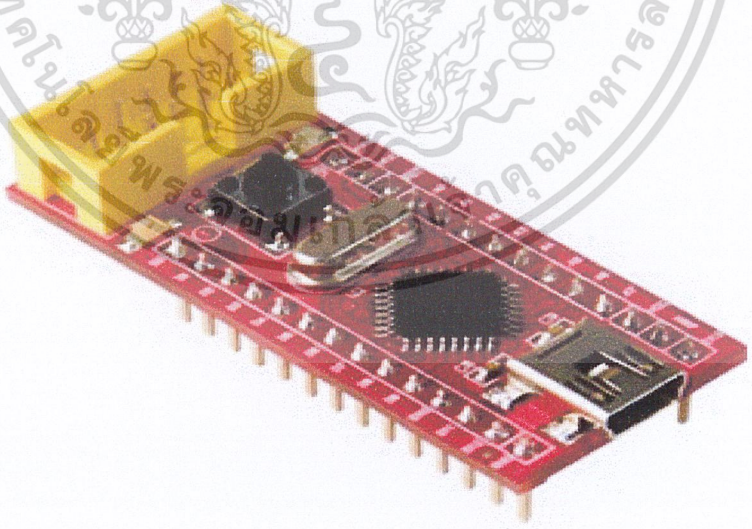
เป็นไมโครคอนโทรลเลอร์ที่นิยมใช้โดยทั่วไปพื้นฐานการทำงานก็คือ ระบบดิจิตอลโดยค่าเอาท์พุตที่ได้จากไมโครคอนโทรลเลอร์ จะเป็น 0 กับ 1 แต่ก็สามารถนำมาประยุกต์เชื่อมต่อกับอุปกรณ์ภายนอกได้มากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะโดยทั่วไปจากรูปที่ 3.20 จากรูปขาของ PIC16F690 แต่ละขาจะมีหน้าที่แตกต่างกันไปซึ่งแยกออกเป็น PORT A , PORT B , PORT C โดยพื้นฐานแต่ละพอร์ตสามารถทำงานเป็นอินพุตและเอาท์พุต สำหรับไมโครคอนโทรลเลอร์ที่ใช้ต้องมีสัญญาณนาฬิกาภายในตัวเพื่อสร้างพัลซ์มาป้อนให้กับมอเตอร์เพื่อที่จะคุมให้มอเตอร์ทำงานตามที่เราต้องการซึ่งใน PIC16F690 มีสัญญาณนาฬิกาภายในตัวอยู่แล้วโดยที่หนึ่งไซเคิล (Clock Bus) ของซีพียูจะประกอบไปด้วยสัญญาณนาฬิกาภายนอกจำนวน 4 ไซเคิลคือ Q1 ,Q2 , Q3 และ Q4 ดังนั้นความถี่ซีพียูประมวลผลต่อหนึ่งคำสั่งเท่ากับความถี่ของสัญญาณนาฬิกาภายนอกหารด้วย4หรือหากจะพิจารณาความเร็วของไมโครคอนโทรลเลอร์ตระกูลPIC สามารถประมวลผลต่อหนึ่งคำสั่งเท่ากับ 1/4 เท่าของความถี่ออสซิเลเตอร์ภายนอก

3. แขนกลใช้ “ ET - EASY168 STAMP ”

ET-EASY168 STAMP เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR8 ขนาดเล็กและได้ทำการเลือกใช้ IC เบอร์ ATMEGA168เป็น MCU ประจำบอร์ด นอกจากนี้แล้วภายในตัวบอร์ดยังได้รวมเอาไอซี USB BRIDGE ของ FTDI เบอร์ FT232R สามารถติดต่อกับคอมพิวเตอร์ PC ผ่านทาง PORT USB ได้โดยตรงทำให้บอร์ด ET-EASY168 STAMP เป็นบอร์ดทดลองใช้งานขนาดเล็กที่เพียบพร้อมไปด้วยวงจรพื้นฐานที่จำเป็นต่อการใช้งานในการควบคุมระบบโดยลักษณะของ Et-easy Mini168 เป็นดังรูป



รูปที่ 3.22 ลักษณะของ ET-EASY168 STAMP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของบอร์ด ET-EASY168 STAMP

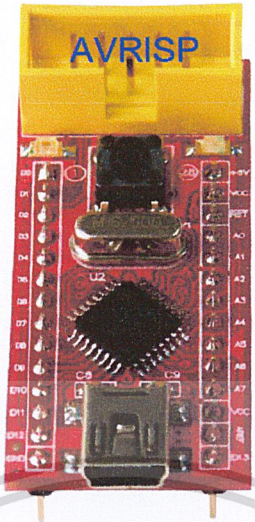
- เลือกใช้ MCU ตระกูล AVR8 เบอร์ ATMEGA168 ของ ATMEL ใช้งาน RUN ความถี่ 16.00 MHz
- หน่วยความจำ FLASH 16 KBYTE, SRAM 1 KBYTE, EEPROM 512 BYTE
- มี GPIO ใช้งาน 22 BIT เป็น DIGITAL จำนวน 14 BIT, และ A TO D ขนาด 10 BIT จำนวน 8 BITPOWER SUPPLY ต่อใช้งาน 5VDC โดยใช้ได้ทั้งกับ 5VDC จาก PORT USB และจากแหล่งจ่าย 5VDCภายนอก พร้อม LED POWER แสดงสถานะ
- มีวงจร EXTERNAL RESET แบบ RC RESET และ SW RESET
- ขั้วต่อใช้งานวางตัวบน PIN HEADER ระยะห่าง 2.54 mm. ขนาด 28 PIN ระยะห่าง 600 MIL ง่ายต่อการนำไปต่อประยุกต์ใช้งาน และต่อทดลองบน PROJECT BOARD
- ขนาดบอร์ด 2 X 5 CM. ขนาดบอร์ดประมาณเท่ากับไอซี 28 PIN
- ขั้วต่อ USB MINI และไอซี USB BRIGE ของ FDTI เบอร์ FT232R บนบอร์ด
- ขั้วต่อ AVR ISP แบบ IDE 10 PIN สำหรับต่อใช้งาน DOWNLOAD ให้กับ MCU ในบอร์ด ในกรณีไม่ต้องการ DOWNLOAD ผ่านทาง PORT USB

สาเหตุการเลือกใช้ ET-EASY168 STAMP

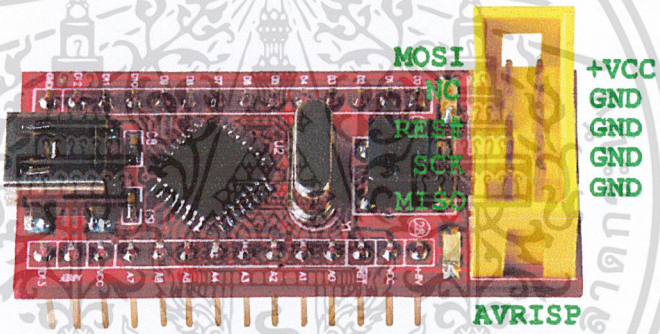
- เพราะตัว บอร์ดมีขนาดเล็ก ทำให้เราสามารถใช้งานในพื้นที่จำกัดได้
-ความสะดวกในการใช้งานเพราะเราสามารถเชื่อมต่อตัวบอร์ดเข้ากับเครื่องคอมพิวเตอร์ โดยผ่านทางสาย USB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AVR	Arduino	Pin	ET-EASY168 STAMP	Pin	Arduino	AVR
PD0	Digital-0	1		28	+5V(+Vin)	+5V(+Vin)
PD1	Digital-1	2		27	+VCC(+5V)	+VCC(+5V)
PD2	Digital-2	3		26	RESET#	RESET(PC6)
PD3	Digital-3	4		25	Analog-0	PC0/ADC0
PD4	Digital-4	5		24	Analog-1	PC1/ADC1
PD5	Digital-5	6		23	Analog-2	PC2/ADC2
PD6	Digital-6	7		22	Analog-3	PC3/ADC3
PD7	Digital-7	8		21	Analog-4	PC4/ADC4
PB0	Digital-8	9		20	Analog-5	PC5/ADC5
PB1	Digital-9	10		19	Analog-6	ADC6
PB2	Digital-10	11		18	Analog-7	ADC7
PB3	Digital-11	12		17	+VCC(+5V)	+VCC(+5V)
PB4	Digital-12	13		16	+AREF	+AREF
GND	GND	14		15	Digital-13	PB5



ตารางที่ 3.1 การจัดสรรขาสัญญาณของบอร์ด ET-EASY168 STAMP



รูปที่ 3.23 ลักษณะของบอร์ด ET-EASY168 STAMP

4. DC Motor

เป็นอุปกรณ์ที่สามารถเปลี่ยนพลังงาน จากไฟฟ้าให้กลายเป็น พลังงานทางกล ซึ่งเราต้องทำการป้อนพลังงานไฟฟ้าเข้าไปในตัวมอเตอร์ จากนั้นจะเกิดสนามแม่เหล็กเหนี่ยวนำทำให้เกิดการหมุนของมอเตอร์ออกมาเป็นพลังงานกล โดยเราเลือกใช้ มอเตอร์กระแสตรง 12V ในการขับเคลื่อนระบบโดยลักษณะของ DC Motor

5. Infrared Sensor

ในการติดต่อสื่อสารระหว่างรถกับสถานีจะเลือกใช้ Infrared sensor แทนการเลือกใช้สัญญาณคลื่นวิทยุ เนื่องจากสัญญาณคลื่นวิทยุเป็นสัญญาณที่แผ่กระจาย ทำให้ยากแก่การตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

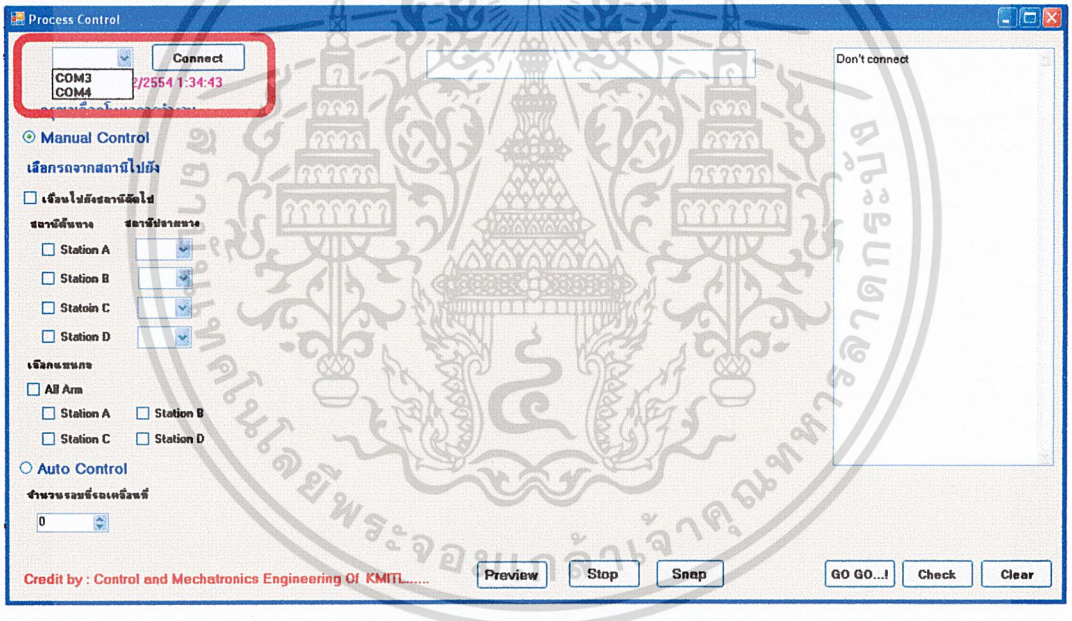
บทที่ 4
การทดลอง

ในบทนี้จะกล่าวถึงการทดลองและผลการทดลองของโปรแกรมและการทำงานของระบบลำเลียง เมื่อมีการสั่งการจากโปรแกรมโดยมีรายละเอียดของการทดลองดังนี้

4.1 การทดลองโปรแกรม และผลที่ได้จากการทำงาน

ในส่วนนี้เป็นการศึกษาเฉพาะส่วนโปรแกรม Visual basic 2008 และผลที่ได้จากการสั่งจากโปรแกรม ซึ่งผลการทดลองมีดังนี้

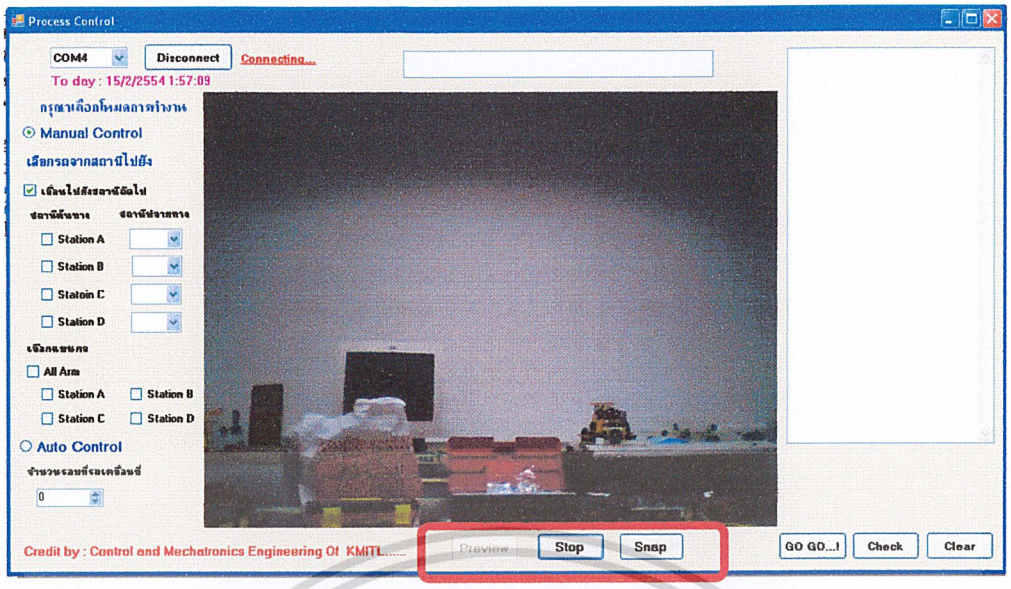
หลังจากทำการรันโปรแกรมจาก Visual basic 2008



รูปที่ 4.1 การเลือก port เพื่อ connect

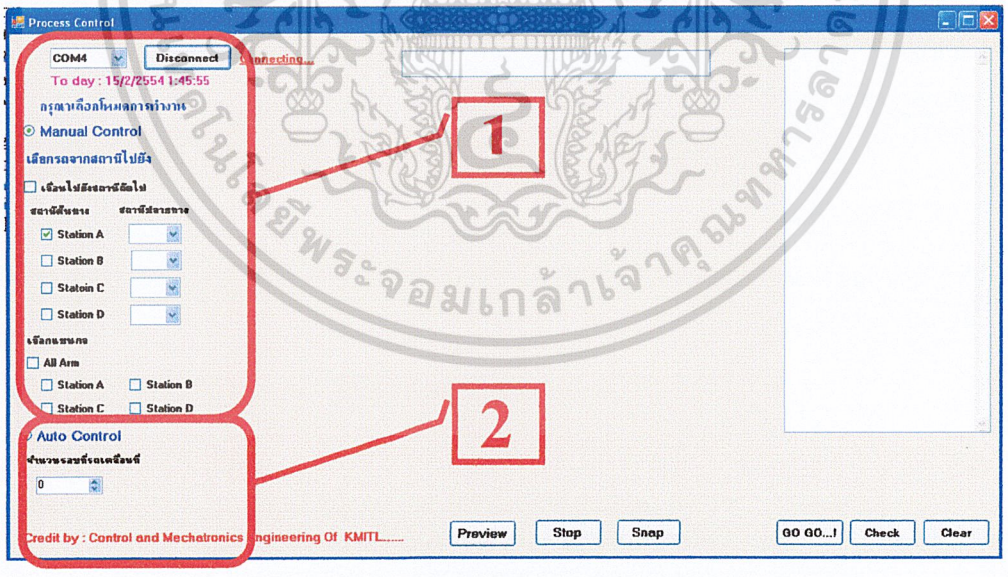
ทำการเลือก com port และกดปุ่ม Connect เพื่อทำการเชื่อต่อระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การดูการทำงานจริงผ่านกล้อง webcam

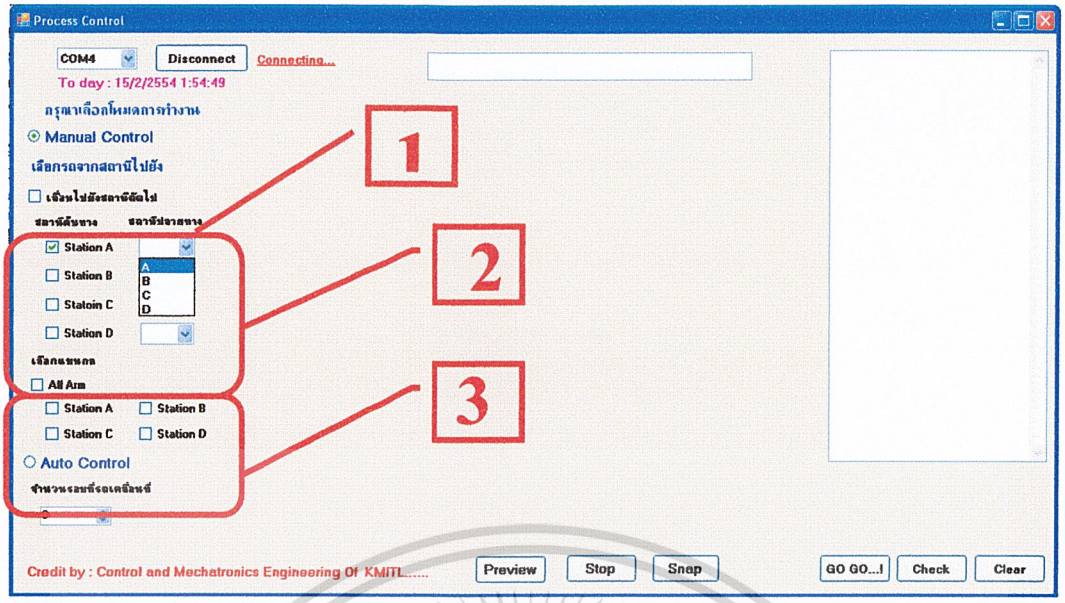
เราสามารถดูการทำงานจริงของระบบโดยผ่านทางกล้อง webcam โดยกดปุ่ม Preview และเราสามารถทำการ save รูปภาพในขณะนั้นได้โดยการกดปุ่ม Snap และทำการปิดกล้องได้โดยการกดปุ่ม Stop



รูปที่ 4.3 การเลือกโหมดการทำงาน

เมื่อเราเข้ามายังตัวโปรแกรมแล้วเราสามารถเลือกการทำงานของระบบลำเลียงได้ 2 แบบ คือ โหมดควบคุมด้วยมือ และ โหมดควบคุมอัตโนมัติ

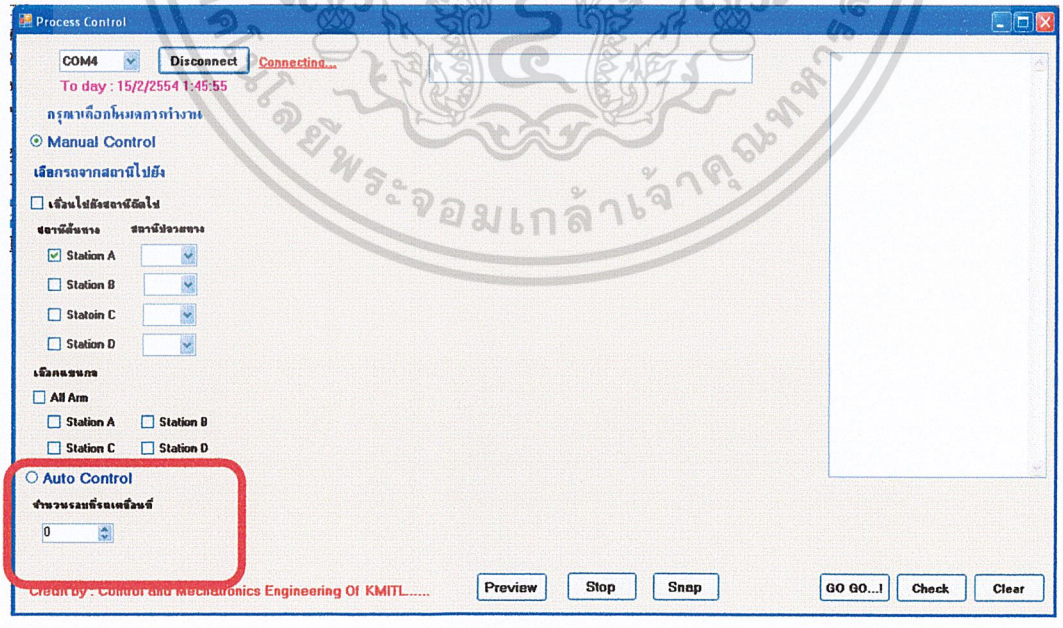
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การเลือกโหมดควบคุมด้วยมือ

ในโหมดควบคุมด้วยมือ นั้น เราอาจแบบเป็น 3 ส่วนหลักๆ คือ

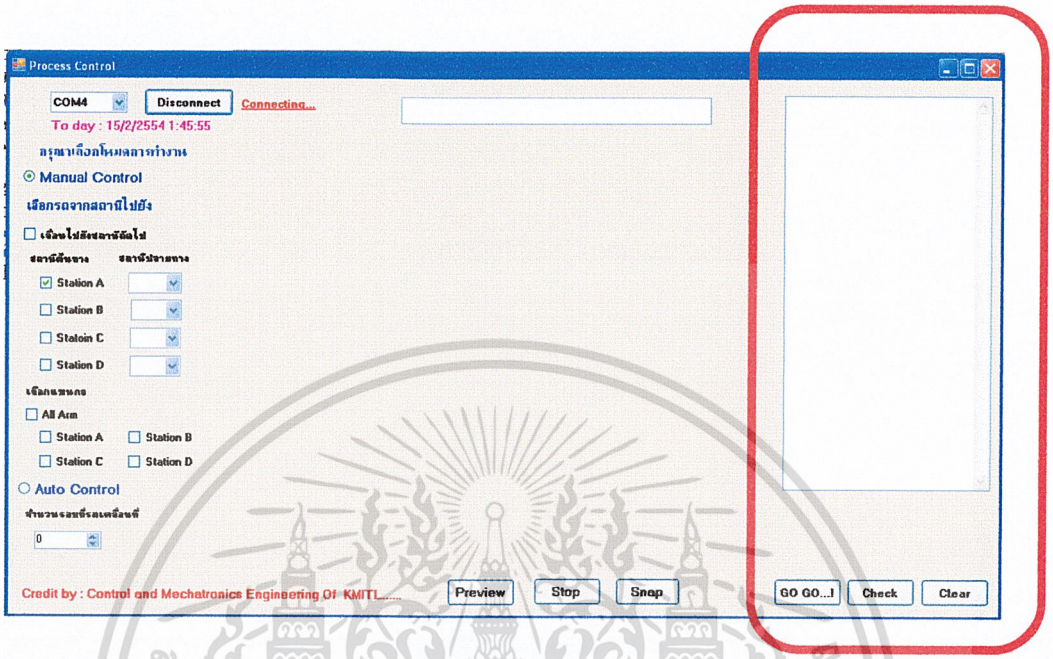
1. เป็นการสั่งการให้รถเคลื่อนที่ไปยังสถานีถัดไป โดยไม่มีการวิ่งข้ามสถานี
2. เป็นการสั่งการให้รถเคลื่อนที่ไปยังสถานีที่ต้องการได้
3. เป็นกานสั่งการให้แขนกลในสถานีต่างให้ทำงาน



รูปที่ 4.5 การเลือกโหมดอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในโหมดควบคุมอัตโนมัตินั้น เราสามารถกำหนดจำนวนรอบการทำงานได้ ว่าต้องการให้ระบบของเราทำงานได้กี่รอบ โดยเมื่อรถเคลื่อนที่มาถึงยังสถานี แขนกลจะทำการหยิบจับสิ่งของเองโดยอัตโนมัติ ไม่จำเป็นต้องรอคำสั่ง



รูปที่ 4.6 การตรวจสอบสถานะการทำงาน

และเราสามารถตรวจสอบสถานะการทำงาน ทั้งระบบได้ด้วยการกดปุ่ม Check ไม่ว่าจะเป็นการตรวจสอบว่าสถานีไหนมีรถคันใดจอดอยู่บ้าง และเราสามารถทำการลบข้อมูลได้โดยการกดปุ่ม Clear

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากโครงงานนี้คือโครงงานระบบลำเลียงสินค้าที่มีความสำคัญอย่างมากกับกระบวนการผลิตในอุสาหกรรม ซึ่งระบบลำเลียงสินค้าในธุรกิจการผลิตก็เปรียบเสมือนระบบประสาทของร่างกายมนุษย์

ซึ่งเป็นการจำลองการลำเลียงสินค้าด้วยรถขนส่งและแขนกลแบบอัตโนมัติโดยมีการสั่งงานจากคอมพิวเตอร์ มีการวางระบบโดยนำทฤษฎีและความรู้ทางด้านอิเล็กทรอนิกส์มาประยุกต์ใช้ในการสร้างวงจรควบคุม ในส่วนของชิ้นงานได้นำอุปกรณ์ต่างๆมาใช้ในการควบคุม เช่น เซอร์โวมอเตอร์ ไมโครคอนโทลเลอร์ เป็นต้น มาทำงานรวมกับวงจรอิเล็กทรอนิกส์ และโปรแกรมเพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพ

ในการทำงานของระบบลำเลียงสินค้านั้นเมื่อเราทำการเขียนโปรแกรมและทำการสั่งให้ระบบทำงานจากคอมพิวเตอร์ ระบบจะทำงานตามที่เรากำหนด คือ รถจะวิ่งออกจากสถานีใด แล้วไปจอดที่สถานีใด ที่เหลืออีก สามคัน จะวิ่งไปจอดถัดไปอีกหนึ่งสถานี อย่างเช่น ถ้าเราต้องการให้รถคันที่หนึ่ง วิ่งไปยังสถานีที่สี่ ดังนั้น รถคันที่สองจะวิ่งไปจอดที่สถานีที่หนึ่ง รถคันที่สามจะวิ่งไปจอดที่สถานีที่สอง และรถคันที่สี่จะวิ่งไปจอดสถานีที่สาม จะเป็นอย่างนี้ไปเรื่อยๆ ไม่ว่าเราจะทำการเลือกรถคันที่เท่าไร สถานีต้นทางหรือปลายทางใด รถคันที่เหลือจะวิ่งไปจอดตามสถานีตามที่กล่าวดังข้างต้น

ในปัจจุบันมีการแข่งขันด้านอุตสาหกรรมค่อนข้างสูงและในอนาคตเป็นไปได้มากที่แบบจำลองนี้จะถูกพัฒนานำไปประยุกต์ใช้งานจริงในโรงงานอุตสาหกรรมเพื่อลดปริมาณแรงงานคน ลดเวลาในการผลิตสินค้า มีความปลอดภัยสูง ส่งผลให้ได้ผลผลิตที่มีประสิทธิภาพและสามารถผลิตสินค้าได้มากขึ้นอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาที่พบและแนวทางแก้ไข

1.เนื่องจากม้วนทองแดงที่อยู่ใต้ท้องรถไม่สัมผัสกับราง สาเหตุมาจากต้องทำการทดลองหลายครั้ง ทำให้ทองแดงไม่เรียบส่งผลให้รถหยุดวิ่งในบางจังหวะ ทำให้ต้อง Reset ระบบเพื่อเริ่มทำงานใหม่ดังนั้นแก้ไขโดยการเปลี่ยนทองแดงใหม่

2.แผ่นไม้ที่ใช้รองรางไฟฟ้าเกิดการโก่งตัว ทำให้ล้อของรถไม่สัมผัสกับพื้นทำให้เกิดอาการล้อฟรีเกิดขึ้นแก้ไขปัญหาเฉพาะหน้าโดยการติดเทปกาวสองหน้าไว้ที่แผ่นไม้และพื้น

3.พื้นรางมีสิ่งสกปรกติดอยู่ทำให้ Sensor ใต้ท้องรถ ตรวจจับว่าเป็นสีดำจึงทำให้รถจอดก่อนถึงสถานีจริงแก้ไขโดยการติดเทปกาวสีขาวที่พื้นราง

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

โครงการนี้เป็นโครงการแบบจำลองระบบลำเลียงสินค้าซึ่งในอนาคตสามารถนำมาพัฒนาปรับใช้ได้ในโรงงานอุตสาหกรรมโดยมีแนวทางในการพัฒนาระบบลำเลียงนี้คือ พัฒนาในส่วนของเซนเซอร์ให้สามารถทำงานได้จริง ในสถานที่จริง ตำแน่งการใช้งานจริงได้อย่างมีประสิทธิภาพ พัฒนาในส่วนของโปรแกรมเพื่อสามารถรองรับการใช้งานจริงในสถานที่จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

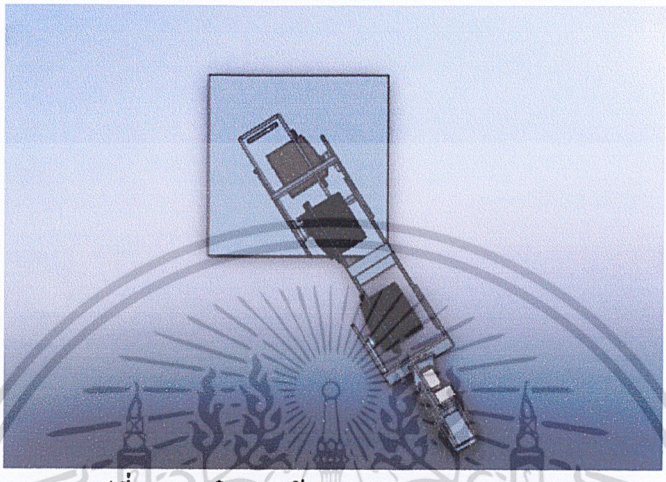
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

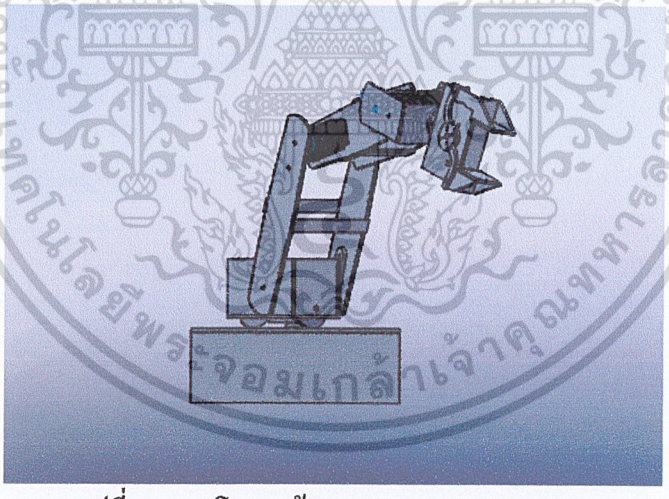
ภาคผนวก ก
โครงสร้างของหุ่นยนต์

ก.1 แขนกล

โครงสร้างแขนกล

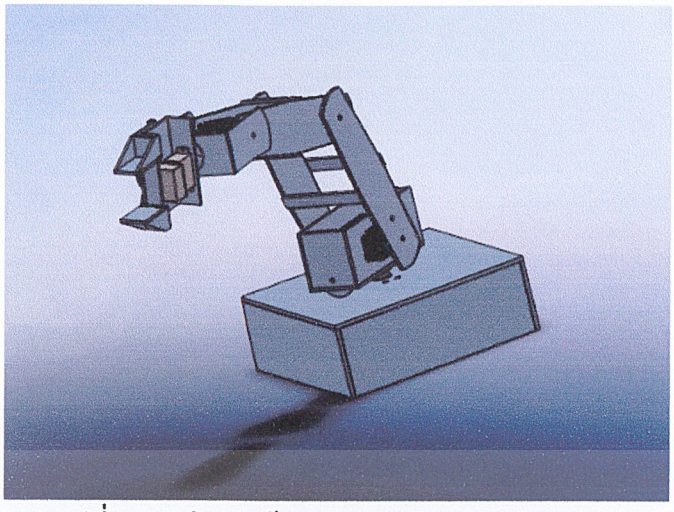


รูปที่ ก.1.1 โครงสร้างแบบ Top ของแขนกล



รูปที่ ก.1.2 โครงสร้างแบบ Font ของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1.3 โครงสร้างแบบ Isometric ของแขนกล

ส่วนประกอบของแขนกล จะแบ่งออกเป็น 4 ส่วนหลัก ๆ คือ

1. ส่วนควบคุม ทำหน้าที่ควบคุมการทำงานของแขนกลทั้งระบบ ซึ่งในส่วนนี้จะรับค่าจากปุ่ม

2. ส่วนประมวลผล ทำหน้าที่ในการประมวลผลคำสั่งที่ได้จากส่วนควบคุมจากSTAMP 168 แล้วทำการประมวลผลค่าที่ได้มา จากนั้นทำการสั่งการไปยังเซอร์โวมอเตอร์ให้ทำงาน

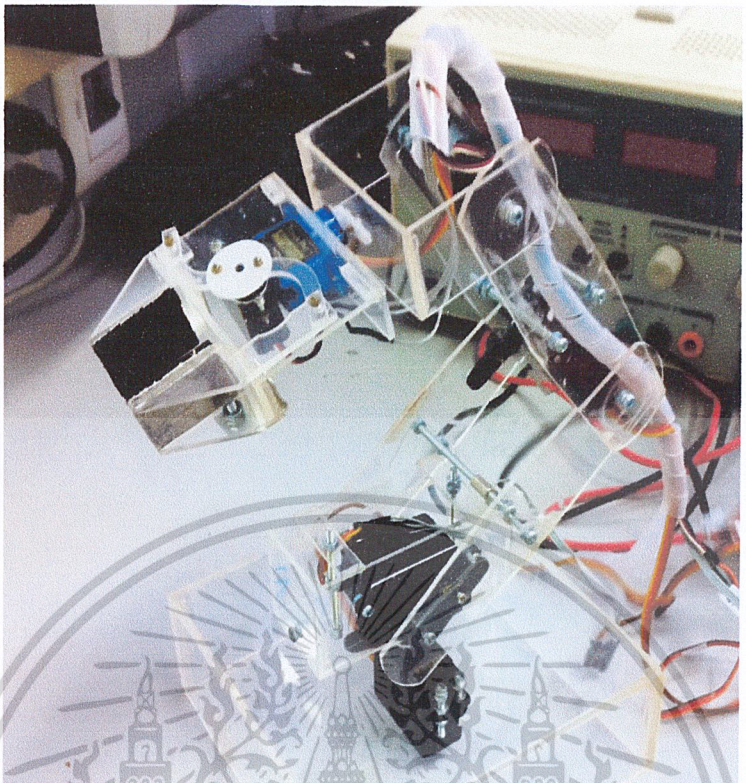
3. ส่วนเซอร์โวมอเตอร์ เป็นส่วนที่ขับเคลื่อนตัวแขนกลให้เคลื่อนไหวไปยังตำแหน่งที่ต้องการ ซึ่งจะถูกติดตั้งยังข้อต่อของตัวแขนกล

4. ส่วนที่ประกอบเป็นแขนกลจะแบ่งออกเป็น 3 ส่วน คือ คลิ๊ปเปอร์ แขน และฐาน ทุกชิ้นส่วนทำจากแผ่นอคิลิค ซึ่งออกแบบจากโปรแกรม Solidworks แล้วกัดชิ้นงานด้วยเครื่อง CNC และนำมาประกอบเข้ากับเซอร์โวมอเตอร์เป็นแขนกล

หลักการทำงาน

ส่วนประกอบในโครงสร้างของแขนกลนั้นถูกสร้างขึ้นจากแผ่นอคิลิคที่ตัดเป็นชิ้น ตามที่ออกแบบไว้ และในการเคลื่อนที่ของแขนกลใช้เซอร์โวมอเตอร์ จำนวน 6 ตัว ซึ่งจะมีเซอร์โวมอเตอร์อยู่ในแต่ละข้อของแขนกล ซึ่งจะทำให้การเคลื่อนไหวของแขนกลสามารถเคลื่อนไหวได้คล้ายกับแขนมนุษย์ และสามารถนำไปประยุกต์ใช้ในการหยิบจับสิ่งของได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

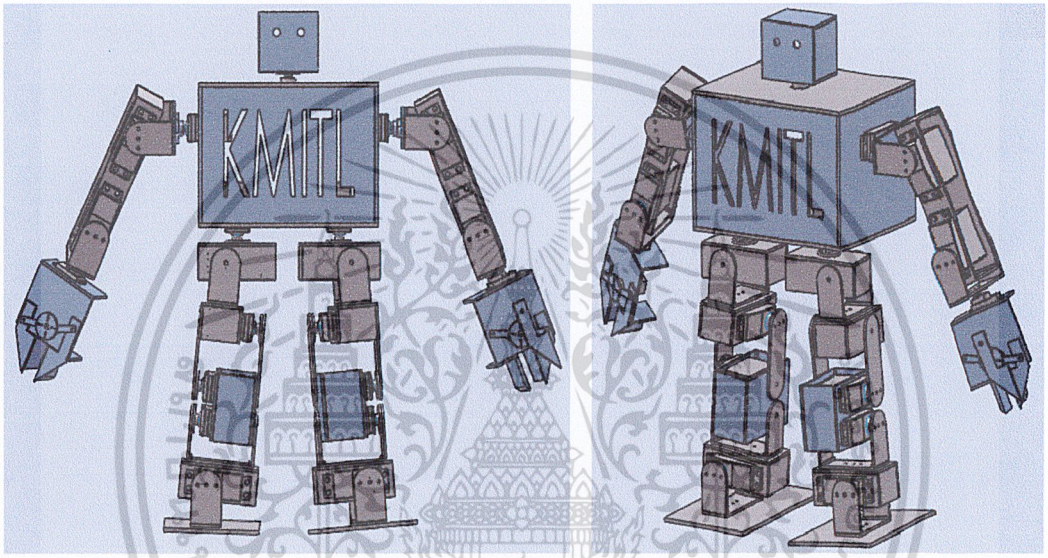


รูปที่ ก.1.4 แขนกลที่ประกอบเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 หุ่นยนต์ 2 ขา

วัสดุที่ใช้คือ อะคลิลิค เนื่องจากอะคลิลิคเป็นวัสดุที่มีความเหนียว (toughness) ความโปร่งใส(transparent) สามารถขึ้นรูปได้ง่ายโดยมีคุณสมบัติที่น่าสนใจและได้เลือกนำมาใช้ทำคือ ทนต่อจุดเดือด จุดหลอเหลวสูง สามรถทนต่อแรงกระแทก(impact strength) สูง แต่อาจเกิดรอยขูดขีดได้ง่ายและไม่ทนทานต่อตัวทำละลายหลายชนิด เนื่องจากมีเนื้อพลาสติกอ่อน รวมทั้งยังมีความทนทานต่อสภาพแวดล้อมดีอีกด้วย



รูปที่ ก.2.1 โครงสร้างหุ่นยนต์ 2 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Servo motor ที่ใช้ในโครงงาน

Servo motor ที่ใช้ในโครงงานนี้คือ MG945 S3003 และ Digital servo E-sky EK2-0508 โดย servo motor แต่ละตัวมีรายละเอียดดังนี้

- **Tower Pro MG945 Digital Servo**

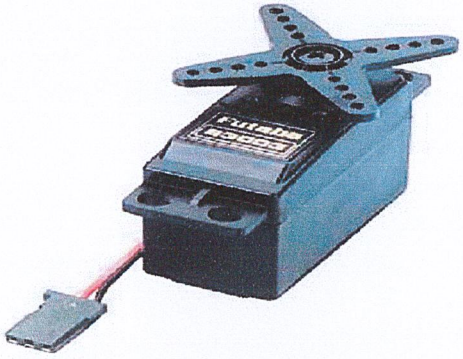
รูปที่ ก.2.2 เซอร์โวมอเตอร์ รุ่น MG945

Detailed Specifications

Operation speed:	0.25Sec/60degrees
Stall Torque:	12KG/CM
Voltage Range:	4.8v – 7.2v
Dimension:	40.7mm x 19.7mm x 42.9mm
Weight:	55g

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Servo motor S3003**



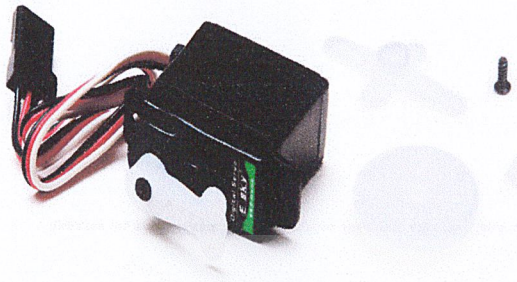
รูปที่ ก.2.3 Servo motor รุ่น S3003

Detailed Specifications

Control System:	+Pulse Width Control 1520usec Neutral
Required Pulse:	3-5 Volt Peak to Peak Square Wave
Operating Voltage:	4.8 – 6.0 Volts
Operating Temperature Range:	-20 to +60 Degree C
Operating Speed (4.8V):	0.23 sec/60 degrees at no load
Operating Speed (6.0V):	0.19 sec/60 degrees at no load
Stall l Torque (4.8V):	44 oz/in. (3.2 kg.cm)
Stall Torque (6.0V):	56.8 oz/in. (4.1 kg.cm)
Operating Angle:	45 Deg. One side pulse traveling 400 usec360
Modifiable:	Yes
Current Drain (4.8V):	7.2mA/idle
Current Drain (6.0V):	8mA/idle
Direction:	Counter Clockwise Pulse Traveling 1520-1900 usec
Motor Type:	3 Pole Ferrite
Potentiometer Drive:	Indirect Drive
Bearing Type:	Plastic Bearing
Gear Type:	All Nylon Gears
Connector Wire Length:	12”
Dimensions:	1.6” x 0.8” x 1.4” (41 x 20 x 36mm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Digital servo motor EK 2-0508**

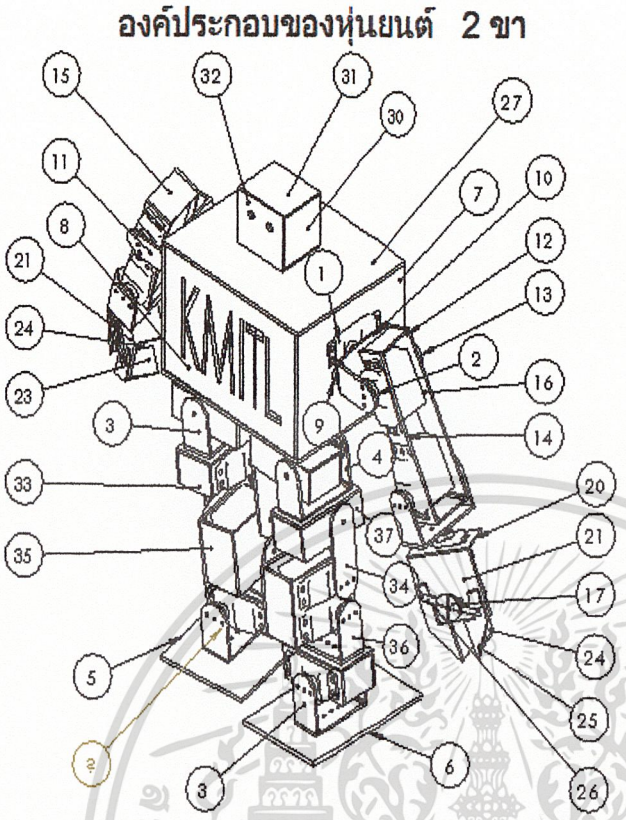


รูปที่ ก.2.4 servo motor EK 2-0508

Detailed Specifications

Standard Voltage:	DC5V +/- 1V
PWM Input Range:	Pulse Cycle 20+-2ms, Positive Pulse 1~2ms
Speed:	60 degrees / 0.1s
Torgue:	> 1KG/cm (Vcc=5V)
Max Corner:	> 150 degrees
Operation Temperature:	-20 degrees ~ 70 degrees
Weight:	7.5g
Dimensions:	22.8 * 11.5 * 20.8mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

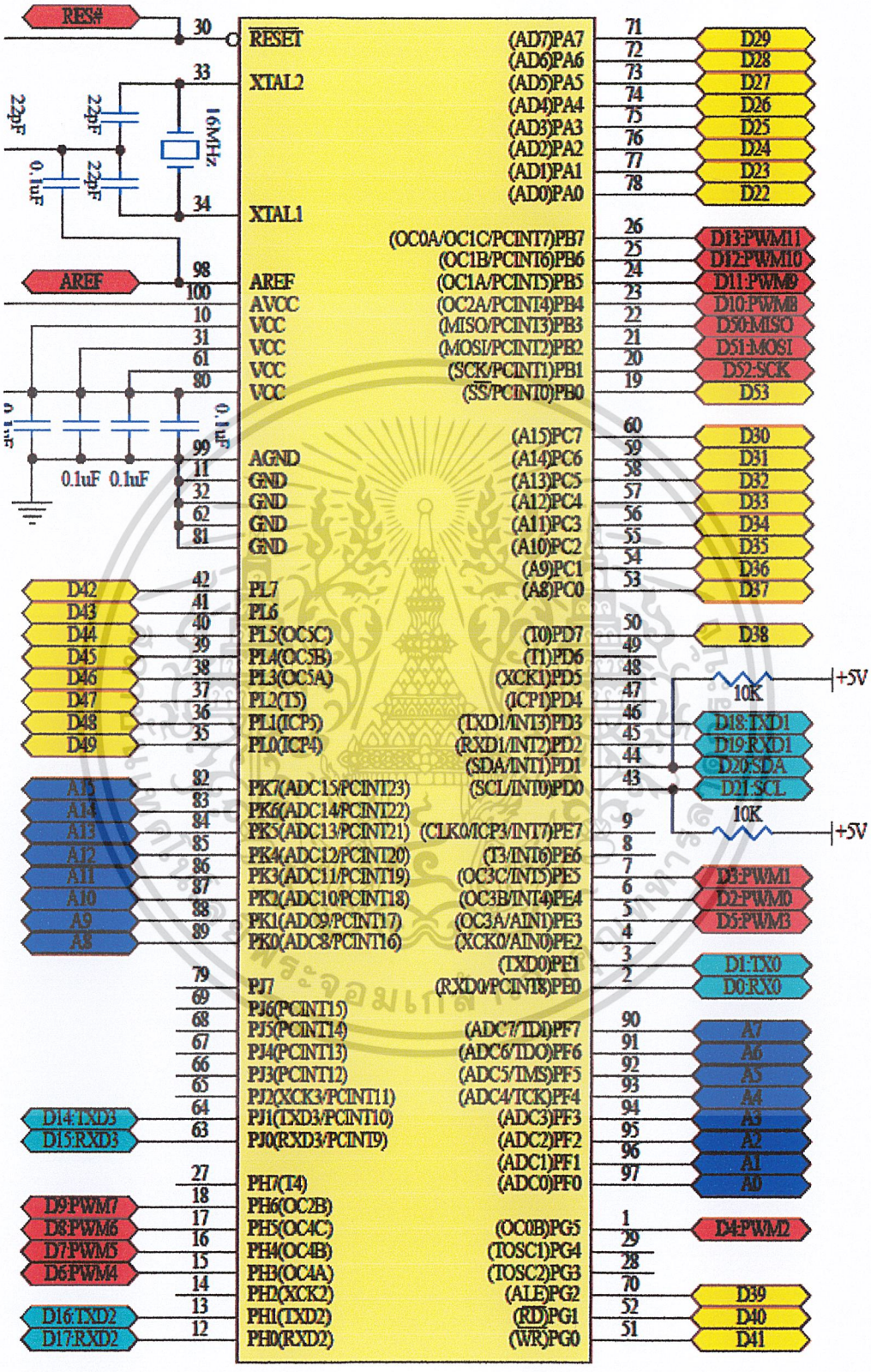


ITEM NO.	PART NUMBER	QTY.
1	BodyDown	1
2	Cupling	6
3	ServoCup	2
4	ServoCup2	2
5	Foot	1
6	Foot2	1
7	Body1	2
8	Body2	1
9	Cupling2	2
10	sholder1	2
11	SupportUp2	4
12	Support1	12
13	SupportDown2	4
14	Sholder2	2
15	SupportSide1	4
16	sholder3	2
17	Culping	5
18	Hand1	2
19	ServoE-sky	5
20	Hand2	2
21	Hand3	2
22	Grip1	2
23	Grip2	2
24	Grip3	4
25	Grip6	2
26	Grip7	4
27	BodyUp	1
๏	ServoS3003	8
29	Head1	1
30	head2	2
31	head3	1
32	Head4	1
33	ServoCup3	1
34	Calf1	4
35	KneeBox	2
36	KneeBox2	2
37	ServoCup4	1

รูปที่ ก.2.5 องค์ประกอบของหุ่นยนต์ 2 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

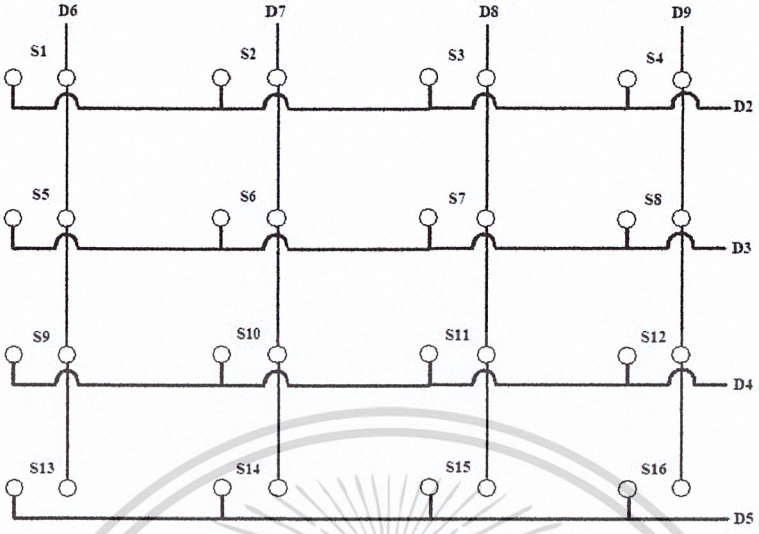
ในการควบคุมหุ่นยนต์ 2 ขา ได้ใช้ Microcontroller AVR MEGA 1280 ในการควบคุมการทำงาน



รูปที่ ก.2.6 วงจรของหุ่นยนต์ 2 ขา

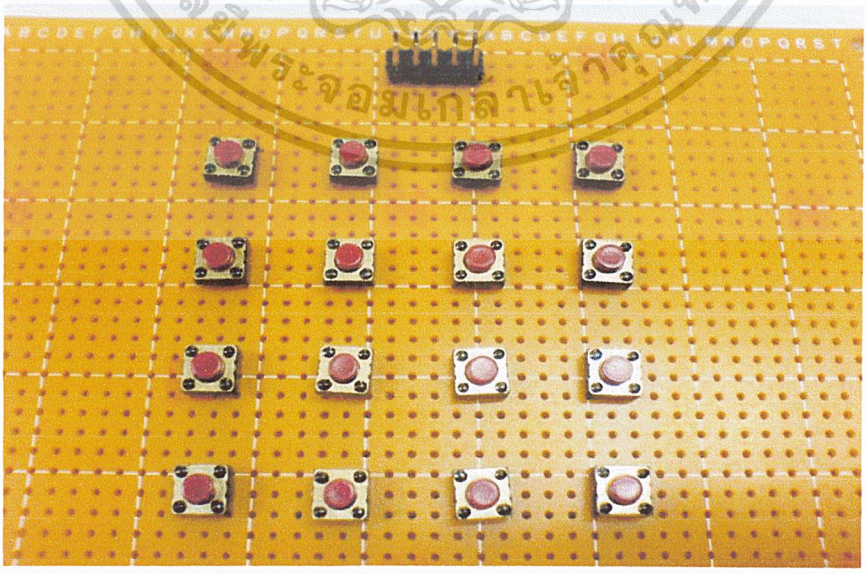
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสวิตช์ควบคุม



รูปที่ ก.2.7 วงจรสวิตช์ควบคุม

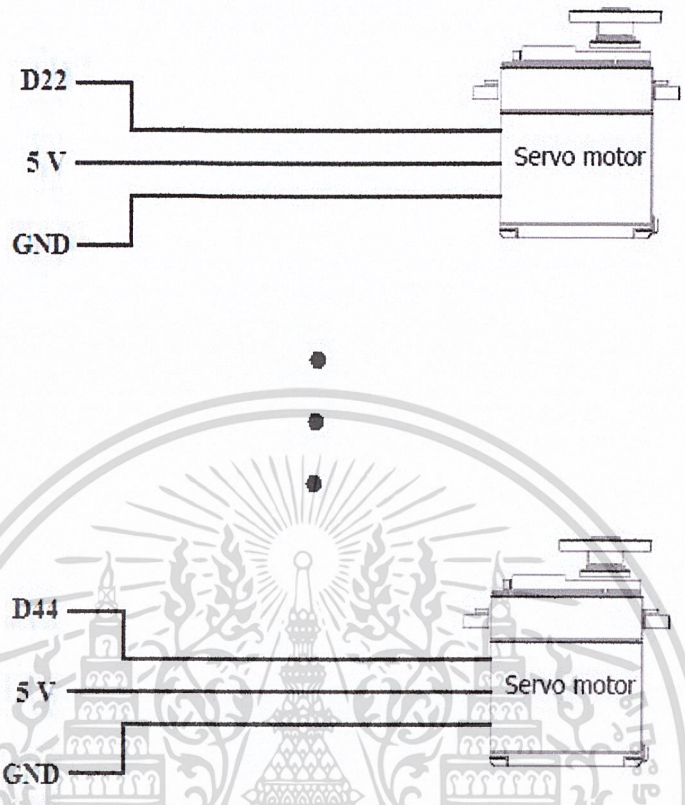
ในการควบคุมหุ่นยนต์ 2 ขา ในเบื้องต้น ในการจัดท่าหุ่นยนต์ ได้ทำการประยุกต์ใช้วงจรควบคุมการจัดท่าของหุ่นยนต์แขนกลมาใช้ในการจัดท่าการเดินของหุ่นยนต์ 2 ขา โดยใช้ Digital Pin D2-D9 ซึ่งแต่ละ Pin จะส่งสัญญาณ Logic 1 ออกมา เช่น สวิตช์ S1 จะมี Pin D2 กับ D6 เชื่อมกันอยู่ เมื่อกดปุ่ม S1 Logic จาก D2 และ D9 จะเชื่อมถึงกัน ทำให้ Microcontroller รู้ว่ามีการกดปุ่ม S1 ตามที่เขียนโปรแกรมไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.2.8 บอร์ดควบคุมเซอร์โวมอเตอร์

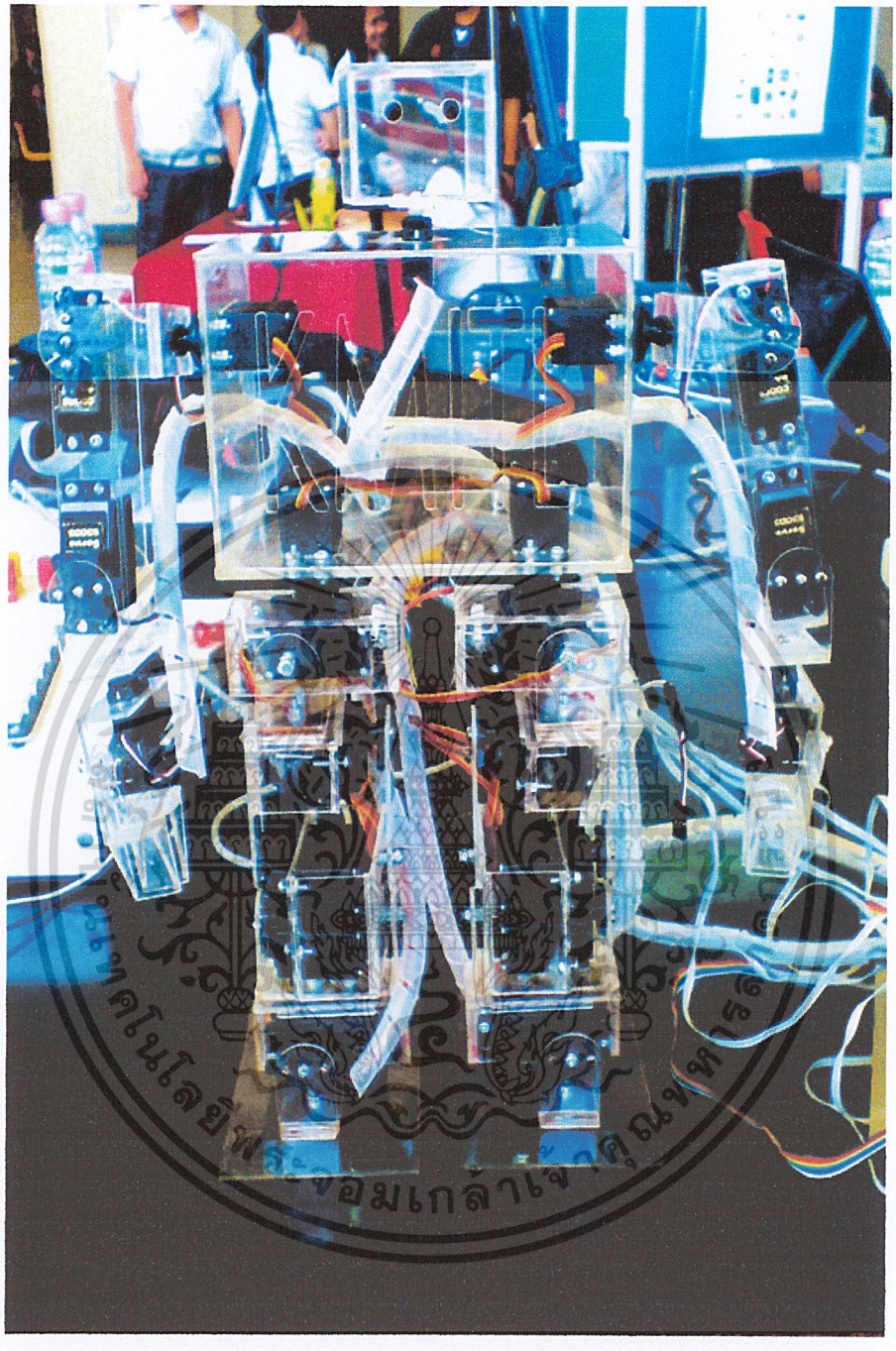
การเชื่อมต่อ servo motor กับ pin ของ AVR MEGA 1280



รูปที่ ก.2.9 แสดงการเชื่อมต่อ servo motor กับ pin ของ AVR MEGA 1280

หุ่นยนต์ 2 ขา ประกอบด้วย Servo Motor จำนวน 23 ตัว ควบคุม Servo Motor โดยเชื่อมต่อสายสัญญาณกับ Pin D22 – D44 ที่กำหนดให้ปล่อยค่า Duty Cycle Pulse ตามที่เขียนโปรแกรมไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

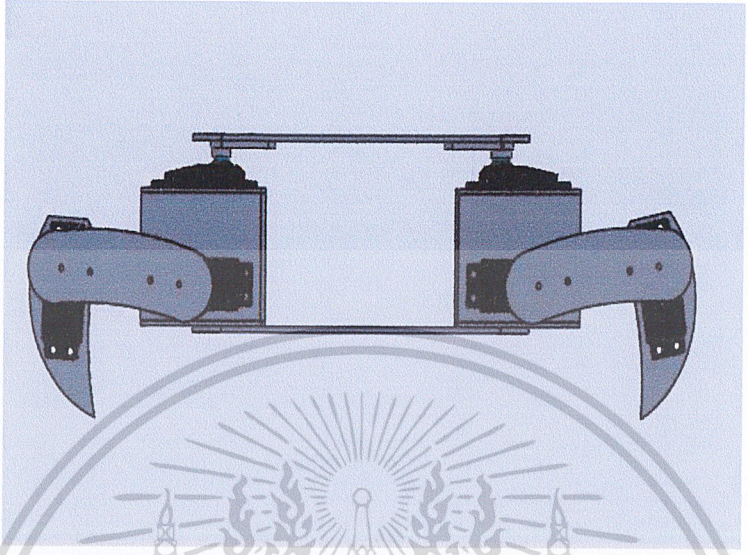


รูปที่ ก.2.10 หุ่นยนต์ 2 ขา ที่ประกอบเสร็จสมบูรณ์

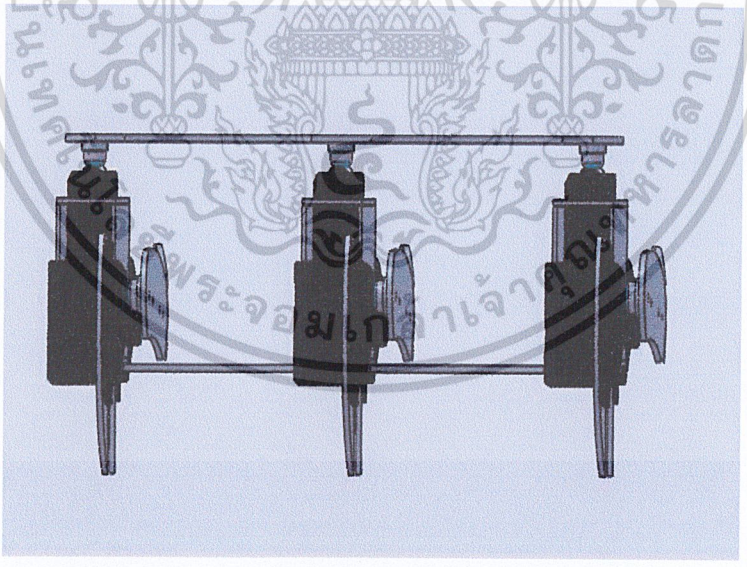
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.3 หุ่นยนต์แมลง 6 ขา

โครงสร้างหุ่นยนต์แมลงหกขา

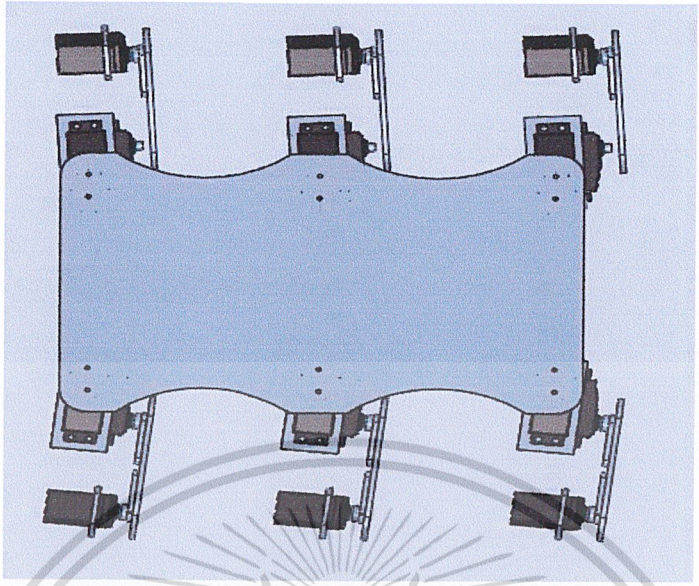


รูปที่ ก.3.1 โครงสร้างหุ่นยนต์แมลงหกขา ด้านหน้า

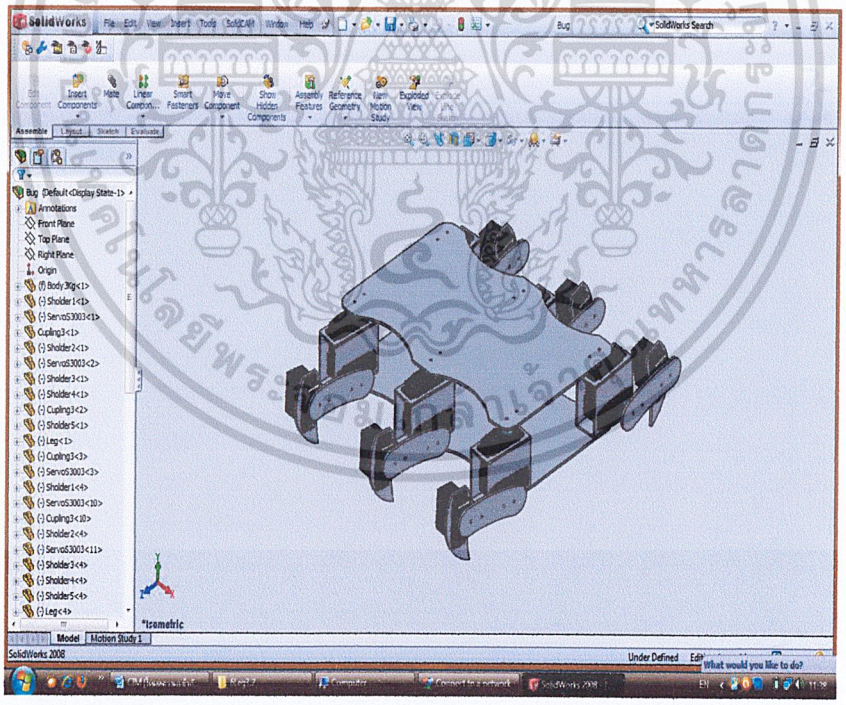


รูปที่ ก.3.2 โครงสร้างหุ่นยนต์แมลงหกขา ด้านข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



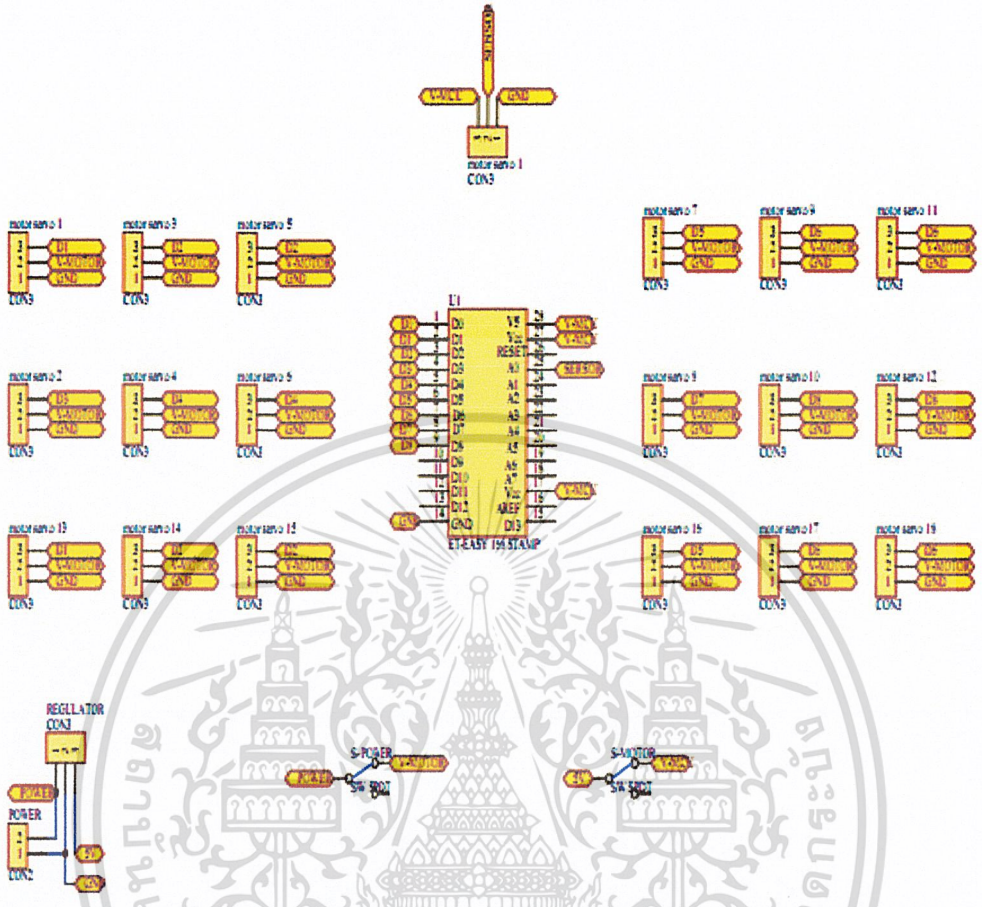
รูปที่ ก.3.3 โครงสร้างหุ่นยนต์แมลงหกขา ด้านข้าง



รูปที่ ก.3.4 โครงสร้างหุ่นยนต์แมลงหกขา มุม Isometric

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

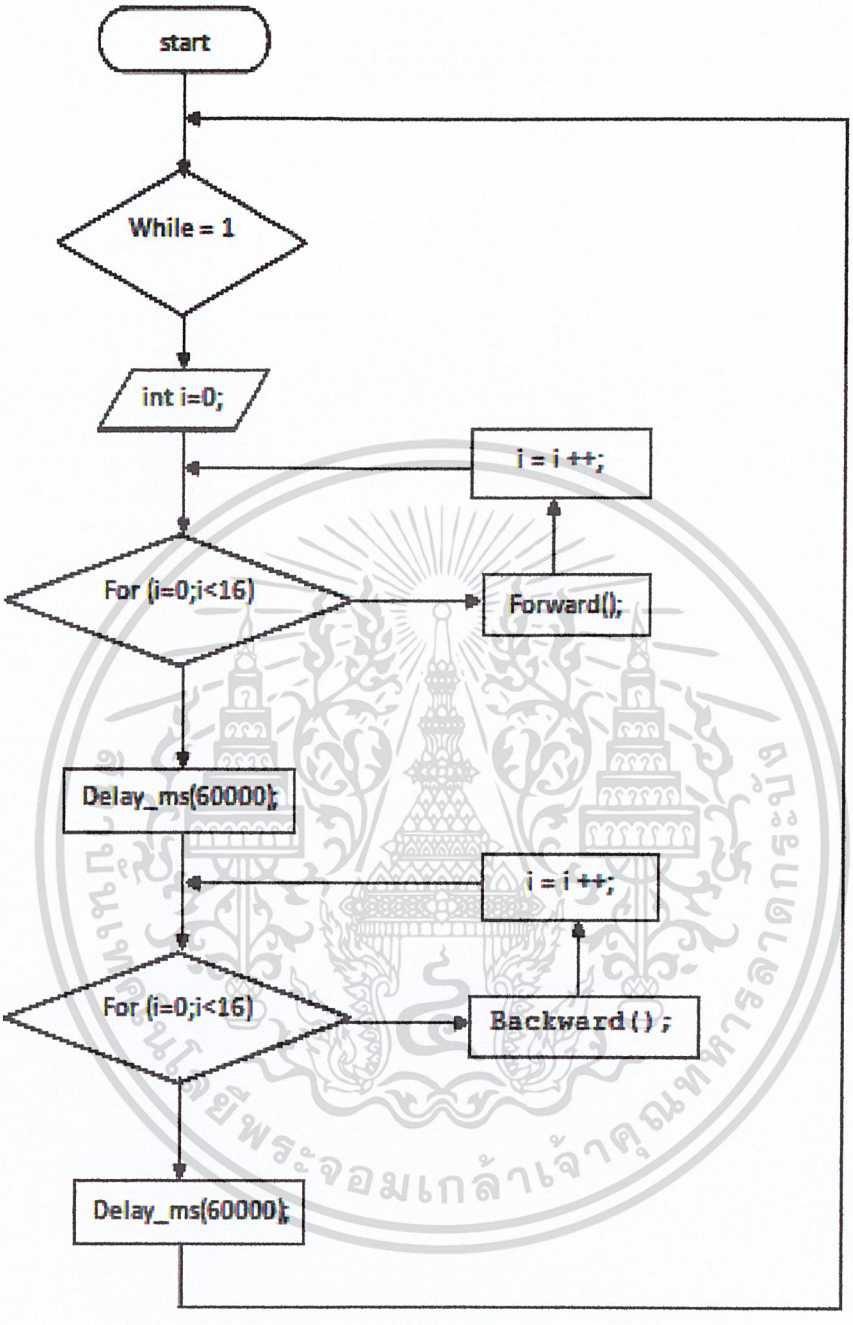
วงจรที่ใช้ในการควบคุมหุ่นยนต์แมลง 6 ขา



รูปที่ ก.3.5 วงจรที่ใช้ในการควบคุมหุ่นยนต์แมลงหกขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart ของโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์แมลง 6 ขา

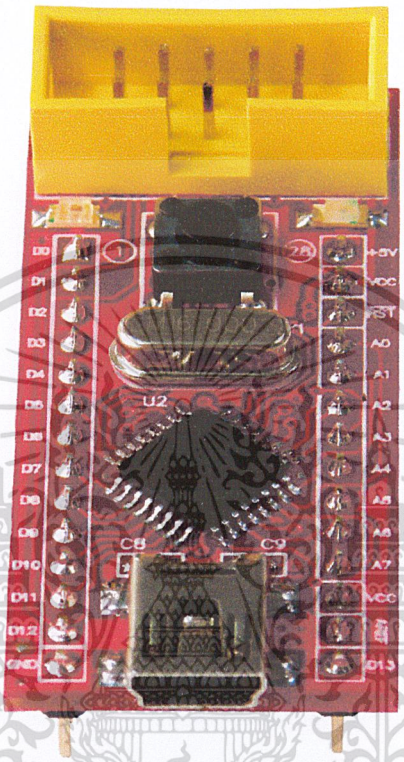


รูปที่ ก.3.6 Flowchart ของโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์แมลงหกขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข.1 ET-EASY168 STAMP



รูปที่ ข.1.1 ET-EASY168 STAMP

ET-EASY168 STAMP เป็นบอร์ดไมโครคอนโทรเลอร์ในตระกูล AVR8 ขนาดเล็กจิ๋ว โดยมีขนาดของบอร์ดเพียง 2cm x 5cm เท่านั้น ซึ่งขนาดบอร์ด ประมาณเท่ากับตัวถังของไอซี 28 DIP 300 โดยเลือกใช้ไมโครคอนโทรเลอร์ตระกูล AVR8 เบอร์ ATmega168 ของ ATMEL เป็น MCU ประจำบอร์ด โดยเลือกใช้MCU ที่มีรูปร่างตัวถังแบบ 32 TQFP พร้อมวงจรรอบนอกที่จำเป็นอย่าง Oscillator และ Reset รวมไว้ด้วยภายในบอร์ด นอกจากนี้แล้วภายในตัวบอร์ดยังได้รวมเอาไอซี USB Bridge ของ FTDI เบอร์ FT232R เพื่อใช้ติดต่อสื่อสารแบบอนุกรมด้วย RS232 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB ได้โดยตรง ทำให้บอร์ด ET-EASY168 STAMP เป็นบอร์ดทดลองขนาดเล็กที่เพียบพร้อมไปด้วยวงจรพื้นฐานที่จำเป็นต่อการใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR8 อย่างแท้จริง เพียงแต่เสียบสาย USB จากพอร์ตUSB ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้ว USB ของบอร์ด ET-EASY168 STAMP ก็สามารถทำการเขียนโปรแกรม และ Download Code ให้กับ MCU เพื่อทำการทดลองได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของบอร์ด

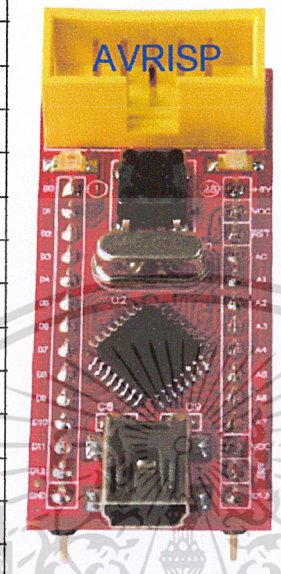
- เลือกใช้ MCU ตระกูล AVR8 เบอร์ ATmega168 ของ ATMEL Run ความถี่ 16.00 MHz
    - มีหน่วยความจำ Flash สำหรับเขียนโปรแกรม 16KByte ถ้าใช้การพัฒนาโปรแกรมผ่านระบบ AVRISP หรือ 14Kbyte เมื่อใช้การพัฒนาโปรแกรมผ่านระบบ Boot Loader RS232
    - มี SRAM ใช้งานขนาด 1KByte และ EEPROM ใช้งานขนาด 512 Byte
    - มี GPIO ใช้งานจำนวน 22 บิต
        - Digital GPIO จำนวน 14 บิต
        - Analog Input (ADC) ขนาดความละเอียด 10บิต จำนวน 8 ช่อง
- ใช้งานกับแรงดันไฟตรงขนาด +5VDC โดยใช้ได้ทั้งกับแหล่งจ่าย +5VDC/500mA จากพอร์ต USBและจากแหล่งจ่าย +5VDC จากภายนอกได้ด้วย พร้อม LED Power แสดงสถานะของแหล่งจ่าย
- มีวงจร External Reset แบบ RC Reset และ Switch Reset พร้อมภายในบอร์ด
- ขั้วต่อใช้งานวางตัวบน Pin Header ระยะห่าง 2.54mm(100mil) ขนาด 28 Pin (ด้านละ 14Pin)ระยะห่าง 600mil(1.5cm) ง่ายต่อการนำไปต่อประยุกต์ใช้งาน และ ขยายวงจร I/O สามารถใช้กับProject Board และ PCB เอนกประสงค์ได้โดยง่าย
- มีขั้วต่อ USB สำหรับเชื่อมต่อสื่อสารกับคอมพิวเตอร์ PC ผ่าน USB Bridge ของ FTDI ในรูปแบบของการสื่อสารอนุกรม RS232 สำหรับใช้งานสื่อสารและ Download Code ให้กับ MCU ในบอร์ด
- มีขั้ว AVRISP แบบ IDE 10PIN สำหรับใช้ Download โปรแกรมให้กับ MCU ภายในบอร์ดในกรณีไม่ต้องการใช้การพัฒนาโปรแกรมผ่านทาง Boot Loader
- มี LED แสดงสถานะ โดยต่อกับ PB5 ของ AVR (Digital-13 ของ Arduino Project) สำหรับใช้เป็นอุปกรณ์ทดลองการทำงานอย่างง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

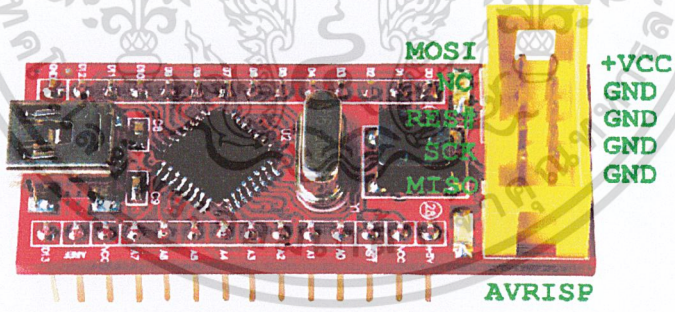
- **ET-EASY168 STAMP**

การจัดสรรขาสัญญาณของบอร์ด ET-EASY168 STAMP

AVR	Arduino	Pin	ET-EASY168 STAMP	Pin	Arduino	AVR
PD0	Digital-0	1		28	+5V(+Vin)	+5V(+Vin)
PD1	Digital-1	2		27	+VCC(+5V)	+VCC(+5V)
PD2	Digital-2	3		26	RESET#	RESET(PC6)
PD3	Digital-3	4		25	Analog-0	PC0/ADC0
PD4	Digital-4	5		24	Analog-1	PC1/ADC1
PD5	Digital-5	6		23	Analog-2	PC2/ADC2
PD6	Digital-6	7		22	Analog-3	PC3/ADC3
PD7	Digital-7	8		21	Analog-4	PC4/ADC4
PB0	Digital-8	9		20	Analog-5	PC5/ADC5
PB1	Digital-9	10		19	Analog-6	ADC6
PB2	Digital-10	11		18	Analog-7	ADC7
PB3	Digital-11	12		17	+VCC(+5V)	+VCC(+5V)
PB4	Digital-12	13		16	+AREF	+AREF
GND	GND	14		15	Digital-13	PB5



ตารางที่ ข.1 การจัดสรรขาสัญญาณของบอร์ด ET-EASY168 STAMP

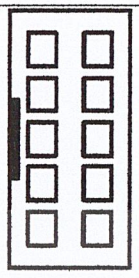


รูปที่ ข.1.3 ลักษณะของบอร์ด ET-EASY168 STAMP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ET-EASY168 STAMP

ตารางขาสัญญาณของ ET-EASY168 STAMP

AVR	Arduino	Pin	AVRISP	Pin	Arduino	AVR
PB3	Digital-11	MOSI		+VCC	+VCC	+VCC
-	-	NC		GND	GND	GND
RES#	RES#	RES#		GND	GND	GND
PB5	Digital-13	SCK		GND	GND	GND
PB4	Digital-12	MISO		GND	GND	GND

ตารางที่ ข.2 ตารางขาสัญญาณของ ET-EASY168 STAMP

หน้าที่ของขาสัญญาณในการใช้งานแบบ “Arduino Project”

• +5V(+Vin) เป็นขาสำหรับใช้เป็นจุดรับแรงดันขนาด +5VDC จากภายนอกเพื่อเป็นแหล่งจ่ายไฟเลี้ยงให้กับบอร์ด

• +VCC(+5V) เป็นขาแหล่งจ่ายไฟจุดเดียวกันกับที่ป้อนให้กับ +VCC ของ MCU ซึ่งจุดนี้จะรับแรงดันมาจาก 2 แหล่ง ด้วยกันคือ ขารับแรงดัน +5V(+Vin) จากขา 28 ของบอร์ด และ จากขา+VUSB(+5V) จากขั้ว USB ของบอร์ด โดยมี Diode ป้องกันการย้อนกลับของแรงดันไว้แล้ว

• +AREF เป็นขาสำหรับรับสัญญาณแรงดันอ้างอิง (Analog Reference) ให้กับวงจร Analog Input ในกรณีต้องการใช้แรงดันอ้างอิงจากภายนอก

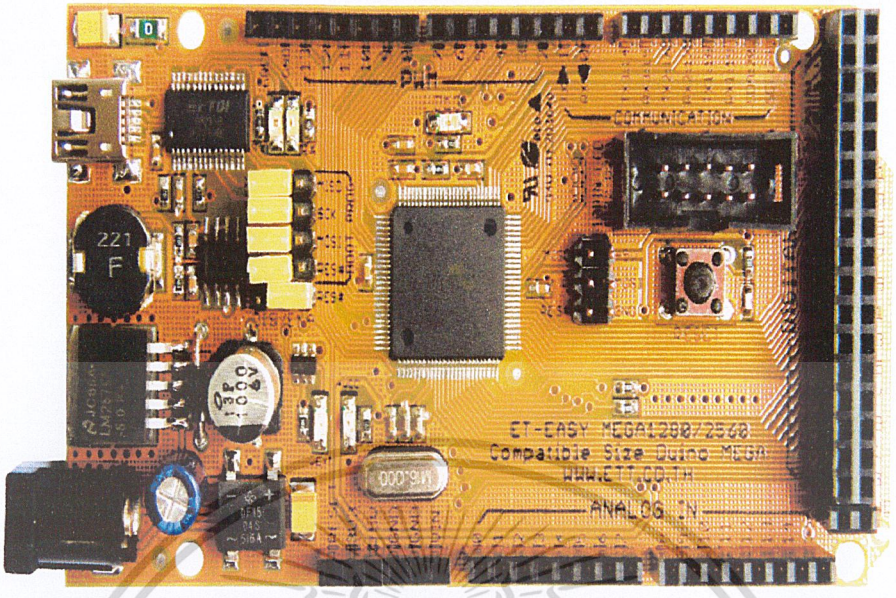
• RESET# เป็นขาสัญญาณ RESET ของ CPU ทำงานที่ Logic “0”

• Digital[0..13] เป็นขา I/O แบบ Digital สามารถใช้งานเชื่อมต่อกับสัญญาณ Logic TTL (5V) ต่างๆ

• Analog[0..7] เป็นขา Input แบบ Analog สามารถรับ Input แบบ Analog 0..+5V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 ET-EASY MEGA1280



รูปที่ ข.2.1 บอร์ด ET-EASY MEGA1280

จากการที่ Arduino ที่เป็นโครงการพัฒนาระบบ MCU ของ AVR แบบ Open Source ได้รับการแนะนำเผยแพร่ออกมาสู่สาธารณะ ซึ่งได้รับความนิยมกันอย่างแพร่หลายจากผู้คนทั่วโลก ภายในระยะเวลาอันรวดเร็ว ทางด้านของ Software ก็มีการพัฒนากันอย่างต่อเนื่อง ซึ่งในขณะนี้ (สิงหาคม 2552) โปรแกรมของ Arduino ได้รับการปรับปรุงเป็น Version "arduino-0016" แล้ว โดยทางด้าน Hardware เองก็ได้มีการพัฒนาปรับปรุงอย่างต่อเนื่องควบคู่กันไปด้วยเช่นเดียวกัน ซึ่งจากเดิมที่มีการพัฒนาโปรแกรมให้รองรับกับการใช้งานได้กับชิพ MCU รุ่นเล็ก 28 ขา อย่าง ATMEGA8,ATMEGA88/ATMEGA168/ATMEGA328 สำหรับเป็นจุดเริ่มต้นให้ผู้สนใจได้ใช้เป็นเครื่องมือและอุปกรณ์ในการศึกษาทดลองเรียนรู้ตลอดจนถึงนำไปประยุกต์ใช้งานกันนั้น มาถึงวันนี้ขนาดของทรัพยากรต่างๆรวมทั้งขนาดของหน่วยความจำ สำหรับเขียนโปรแกรมที่มีอยู่ในชิพ AVR รุ่นเล็กที่มีอยู่เริ่มไม่เพียงพอกับการประยุกต์ใช้งานในงานบางประเภทแล้ว ทาง Arduino เอง จึงได้ทำการพัฒนาให้ Arduino สามารถรองรับการใช้งานขนาดใหญ่ขึ้นอีก โดยปรับปรุงโปรแกรมให้ใช้กับชิพ AVR รุ่นใหญ่ขึ้น เพื่อให้มีจำนวน I/O ทั้ง Digital, Analog, PWM, UART และขนาดหน่วยความจำเพิ่มมากขึ้นกว่าเดิม แต่ก็ยังคงใช้แนวทางในการพัฒนาโปรแกรม แบบเดียวกันกับรุ่นเล็กทุกประการ โดยได้เลือกใช้ชิพ AVR เบอร์ ATMEGA1280 และออกแบบพัฒนา Hardware บอร์ดขึ้นมารองรับโดยใช้รหัสชื่อรุ่นว่า "Arduino Mega" ออกวางจำหน่าย และได้มีการเผยแพร่รายละเอียดทาง Hardware ต่าง ๆ ให้ผู้สนใจนำไปพัฒนาต่อยอดกันได้เอง แต่เนื่องจากชิพดังกล่าวมีโครงสร้างตัวถังเป็นแบบ SMD จึงทำให้เป็นอุปสรรคสำหรับผู้ใช้งานจำนวนไม่น้อยในการที่จะผลิตหรือสร้างบอร์ดขึ้นใช้งานเอง ทาง อีทีที จึงได้นำ ATMEGA1280 มาพัฒนาเป็นบอร์ด โดยให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

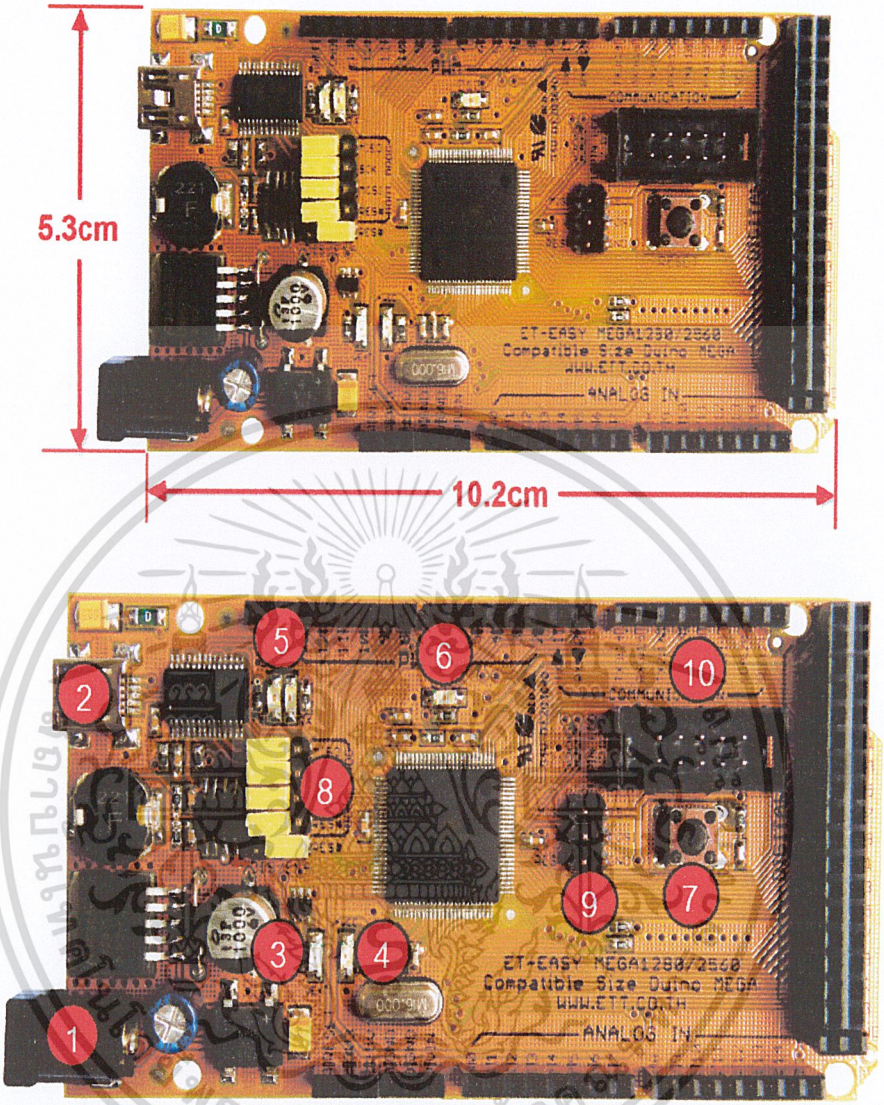
มีโครงสร้างการทำงานเช่นเดียวกันกับ Arduino Mega ขึ้นมา โดยใช้ชื่อว่า " ET-EASY MEGA1280 " โดยได้ออกแบบให้มีการจัดสรร Pin I/O ต่าง ๆ รวมทั้งขนาดให้ตรงตามมาตรฐานของบอร์ด " Arduino Mega " เพียงแต่ได้มีการปรับปรุงข้อจำกัดบางอย่างให้ดีขึ้นกว่า Arduino Mega รุ่นมาตรฐานเพื่อเพิ่มความสะดวกกับผู้ใช้งานมากยิ่งขึ้น

คุณสมบัติของบอร์ด

- ใช้ ATMEGA1280 เป็น MCU ประจำบอร์ด Run ความถี่ 16MHz จาก Crystal Oscillator
- 128KByte Flash(สงวนไว้ 4KByte สำหรับ Bootloader) / 8KByte SRAM / 4KByte EEPROM
- รองรับการพัฒนาโปรแกรมด้วยภาษา C++ ของ Arduino ตามแบบ Arduino Mega ได้ 100%
- ใช้ USB Bridge ของ FTDI เบอร์ FT232RL พร้อม Over Current Protection สำหรับติดต่อสื่อสารและ Download Code จากคอมพิวเตอร์ให้บอร์ด พร้อม Jumper สำหรับปรับใช้งานบอร์ดเป็นการProgram Bootloader ให้กับ MCU จากพอร์ต USB ในบอร์ดได้เอง โดยไม่ต้องใช้เครื่องโปรแกรมAVRISP จากภายนอก
- 54 Pin Digital I/O โดยมี 14Pin สามารถโปรแกรมหน้าที่เป็น PWM ได้
- 16 Pin Analog Input (ADC ขนาด 10 บิต 16 ช่อง)
- UART(Hardware Serial Port) แบบ TTL Logic
- ขนาดของ PCB บอร์ด และ ตำแหน่ง Pin Connector ต่างๆ ตรงกันกับ Arduino Mega ทั้งหมด ทำให้สามารถนำไปติดตั้งใช้งานร่วมกับบอร์ด Shield แบบต่างๆที่มีการผลิตขึ้นมาใช้งานร่วมกันกับบอร์ด Arduino Mega ได้ทั้งหมด โดยบอร์ดมีขนาด PCB Size 5.3cm x 10.2cm
- มีขั้ว Header 10Pin IDE ของ 8 บิต Digital I/O(D22...D29) สำหรับเชื่อมต่อกับ LCD หรือบอร์ดI/O แบบต่างๆ ของ อีทีที เพิ่มความสะดวกในการใช้งาน
- รองรับการใช้งานกับ External Supply ทั้งแบบ AC และ DC ขนาด 7-12V โดยเลือกใช้ Regulateแบบ Switching ขนาด 1A (LM2575-5V) ลดปัญหาเรื่องความร้อนเมื่อมีการใช้กระแสสูงๆ สามารถใช้แหล่งจ่ายจากพอร์ต USB ได้ในกรณีใช้กระแสไม่เกิน 500mA โดยมีวงจรเลือกแหล่งจ่ายอัตโนมัติ โดยจะตัดการใช้ไฟเลี้ยงจาก USB โดยอัตโนมัติ เมื่อมีการต่อแหล่งจ่ายจากภายนอกให้บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างบอร์ด ET-EASY MEGA1280



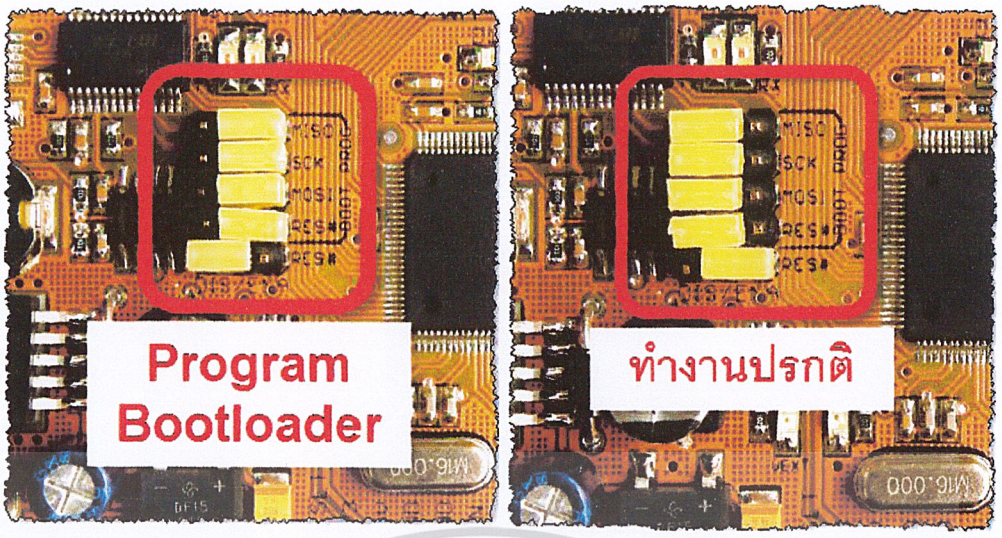
รูปที่ ข.2.2 โครงสร้างของบอร์ด ET-EASY MEGA1280 (Duino MeGA)

- หมายเลข 1 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงจากภายนอก สามารถใช้ได้กับแหล่งจ่ายทั้งแบบ AC และDC พร้อมวงจร Bridge Rectifier และ Regulate แบบ Switching ช่วยลดความร้อนของ IC Regulate เมื่อมีการดึงกระแสมากๆได้เป็นอย่างดี สามารถใช้กับแรงดัน Input 7-12V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

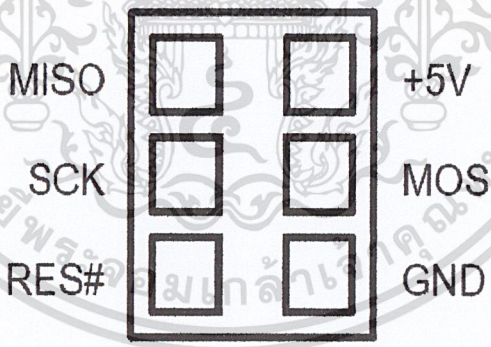
- หมายเลข 2 เป็นขั้วต่อ USB สำหรับติดต่อสื่อสารกับคอมพิวเตอร์ PC โดยใช้ FT232RL เป็นUSB Bridge ในการเชื่อมต่อระหว่างคอมพิวเตอร์ PC และ MCU ในบอร์ด และยังสามารถใช้ไฟจาก พอร์ต USB เป็นแหล่งจ่ายให้กับบอร์ดได้ด้วย โดยจะมี Poly Fuse ขนาด 500mA สำหรับป้องกันการดึงกระแสเกินจากพอร์ต USB ด้วย และที่พิเศษคือมีวงจรสำหรับตรวจสอบแหล่งจ่ายเพื่อสลับการใช้งานแหล่งจ่ายจาก USB ไปเป็น External Supply ได้เอง โดยอัตโนมัติ โดยเมื่อไม่ได้ต่อ External Supply บอร์ดจะใช้ไฟจากพอร์ต USB เป็นแหล่งจ่ายในการทำงาน แต่เมื่อมีการต่อ External Supply วงจรจะสลับไปใช้แหล่งจ่ายจาก External Supply เองโดยอัตโนมัติ
    - o LED +VCC ใช้แสดงสถานะเมื่อมีการจ่ายไฟให้กับบอร์ด
    - o LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟจาก External Supply
- หมายเลข 3 เป็น LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟเลี้ยงจาก External Supply
- หมายเลข 4 เป็น LED +VCC ใช้แสดงสถานะของแหล่งจ่ายไฟเลี้ยง (+VCC) ของบอร์ด โดยเมื่อบอร์ดใช้แหล่งจ่ายจาก External Supply จะแสดงสถานะโดยการให้ LED VEXT และ LED +VCC ติดสว่างพร้อมกันทั้งคู่ แต่ถ้าบอร์ดใช้แหล่งจ่ายจากพอร์ต USB จะแสดงสถานะโดยการให้ LED
  +VCC ติดสว่างเพียงดวงเดียว
- หมายเลข 5 เป็น LED แสดงสถานะของ RX และ TX ใช้สำหรับแสดงการรับส่งข้อมูลระหว่างบอร์ด ET-EASY MEGA1280 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB
- หมายเลข 6 เป็น LED D13 ใช้สำหรับทดแสดงการทำงานของ Bootloader และ ใช้ทดสอบการทำงานของบอร์ดจากการควบคุมของ Pin Digital-13 ทำงานด้วย Logic “1” และ หยุดทำงานด้วยLogic “0”
- หมายเลข 7 เป็นสวิตช์ Reset ใช้สำหรับสั่ง Reset การทำงานของบอร์ด
- หมายเลข 8 เป็นชุด Jumper สำหรับเลือก การ Program Bootloader ผ่าน USB Port และการใช้งานตามปรกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2.3 การทำงานของบอร์ด ET-EASY MEGA1280

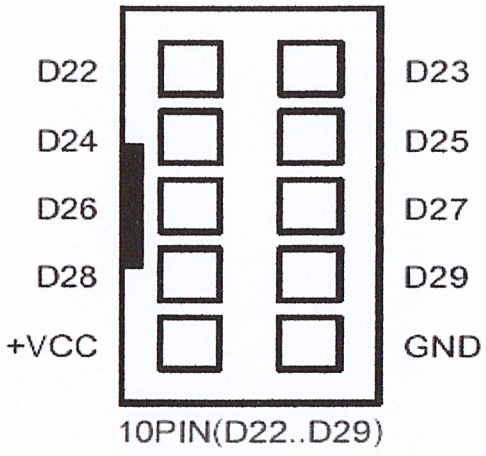
- หมายเลข 9 เป็นขั้วต่อ AVRISP ใช้สำหรับ Download Code ให้กับ MCU โดยขั้วต่อ AVRISPนี้จะสามารถใช้งานได้กับเครื่องโปรแกรมทุกรุ่นที่รองรับการใช้งานกับ ATMEGA1280 และใช้ขั้วต่อตรงตามมาตรฐาน AVRISP ดังรูป



รูปที่ ข.2.4 สัญญาณขาของ ET-EASY MEGA1280

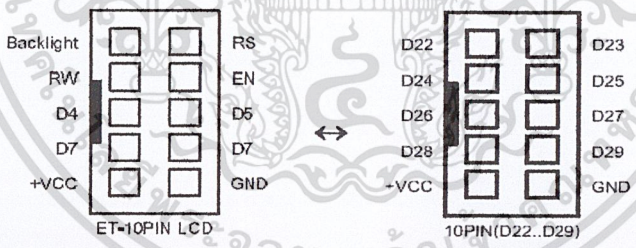
- หมายเลข 10 เป็นขั้วต่อสัญญาณจาก D[22..29] สำหรับเชื่อมต่อกับบอร์ด I/O ของ อีทีที รวมทั้งจอแสดงผล LCD โดยใช้ร่วมกับ 10PIN LCD หรือ ET-CONV SPI TO LCD

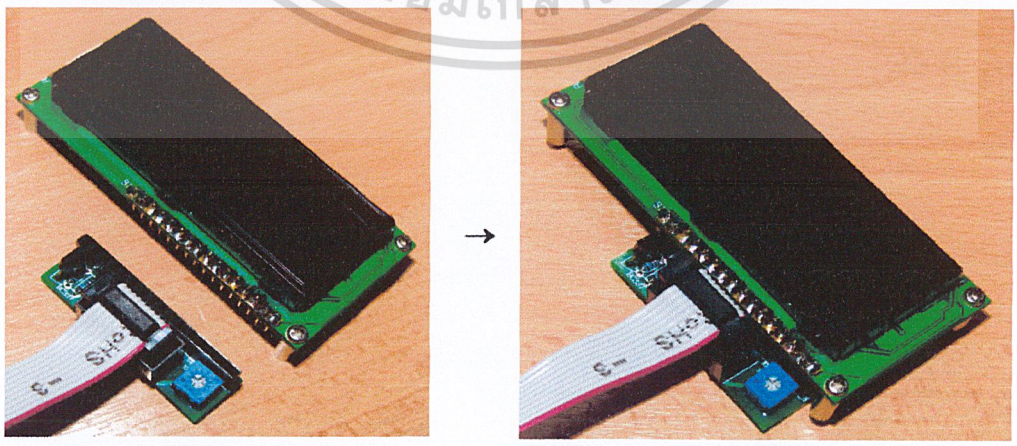
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2.5 สัญญาณขาของ ET-EASY MEGA1280

โดยในกรณีที่ต้องการนำขั้วต่อ 10PIN ไปเชื่อมต่อเพื่อใช้ควบคุมการแสดงผลของ LCD นั้น เพื่อความสะดวกในการใช้งาน ขอแนะนำให้จัดหาชุด ET-10PIN CLCD (ET-CONV 10 TO LCD) มาเป็นตัวกลางในการเชื่อมต่อระหว่าง ขั้ว IDE 10PIN ของบอร์ด ET-EASY MEGA1280 ดังตัวอย่างโดยเมื่อนำLCD มาติดตั้งเข้ากับชุด ET-10PIN CLCD เรียบร้อยแล้วจะทำให้สามารถเชื่อมต่อสายสัญญาณระหว่างขั้วET-10PIN CLCD กับ 10PIN ได้โดยง่าย โดยใช้การเชื่อมต่อผ่านทางสายแพร์ 10PIN ได้ทันทีดังตัวอย่าง



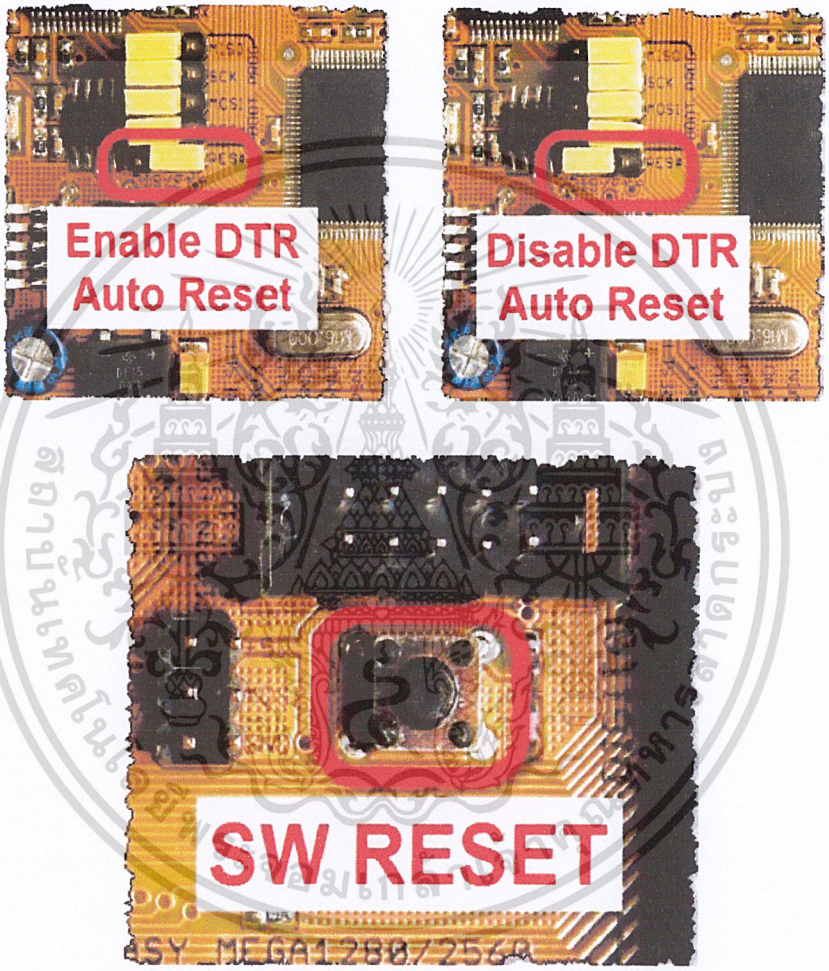


รูปที่ ข.2.6 การเชื่อมต่อของสายแพร์กับบอร์ด ET-EASY MEGA1280

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของสัญญาณต่าง ๆ ของบอร์ด ET-EASY MEGA1280

- RESET# เป็นสัญญาณ Input Logic Reset ของ MCU เมื่อเป็น Logic Low จะทำให้ MCU อยู่ในสภาวะรีเซ็ต เมื่อเป็น Logic High จะทำให้ MCU อยู่ในสภาวะทำงานตามปรกติ โดยสัญญาณRESET# นี้จะถูกควบคุมจาก 2 แหล่ง คือ จาก สวิตช์ RESET ภายในบอร์ด และจากสัญญาณDTR ของ FT232RL ถ้ามีการเลือก Enable Jumper ของ Auto Reset จาก DTR ไว้



รูปที่ ข.2.7 สัญญาณของบอร์ด ET-EASY MEGA1280

- +3V3 เป็นแหล่งจ่ายไฟขนาด +3.3V ที่ได้จากวงจร Regulate ภายในของ FT232RL สามารถจ่ายกระแสได้สูงสุด 50mA ซึ่งเมื่อต้องการนำแหล่งจ่าย +3.3V นี้ไปใช้งานเป็นแหล่งจ่ายให้กับอุปกรณ์ต่างๆ ต้องระมัดระวังเรื่องการดึงกระแสของโหลดด้วย ถ้าโหลดมีการดึงกระแสมากกว่า 50mA อาจทำให้ FT232RL เกิดความเสียหายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- +VIN เป็นไฟ DC ที่รับมาจาก Jack VIN(External Supply) แต่ผ่านการ Rectifier และ Filter เป็นDC แล้ว มีขนาดแรงดันเฉลี่ยตามขนาดแรงดันที่ป้อนให้กับบอร์ดทาง Jack VIN
- +5V เป็นจุดต่อแหล่งจ่ายไฟของบอร์ดออกไปใช้งาน ซึ่งมาจากแหล่งกำเนิด 2 แหล่ง คือ จากพอร์ตUSB และจาก External Supply ซึ่งถ้าต่อแหล่งจ่ายให้บอร์ดจาก External Supply ผ่านทาง JackVIN แหล่งจ่าย +5V นี้จะมาจาก Switching Regulate (LM2575-5V) สามารถจ่ายกระแสได้สูงสุดถึง 1A แต่ถ้าใช้แหล่งจ่ายจากพอร์ต USB แหล่งจ่าย +5V นี้จะมาจากพอร์ต USB โดยตรงโดยจะมีฟิวส์ แบบ Poly ขนาด 500mA ต่อป้องกันการดึงกระแสเกินเพื่อป้องกันความเสียหายของพอร์ตUSB โดยจะจ่ายกระแสได้สูงสุดไม่เกิน 500mA ขึ้นอยู่กับคุณสมบัติในการจ่ายกระแสของพอร์ตUSB และการ Configure ค่าให้กับ FT232RL ด้วย
- A0-A15 เป็นขาสัญญาณ Analog Input แบบ ADC มีขนาดความละเอียด 10บิต มี 16 Pin สามารถรับแรงดัน Analog Input ได้ 0-5VDC
- D0-D53 เป็นขาสัญญาณ Digital Input/Output แบบ TTL มีทั้งหมด 54 Pin สามารถใช้ทำหน้าที่เป็น Input หรือ Output ตามการกำหนดจากโปรแกรม โดยมีบาง Pin สามารถกำหนดหน้าที่ใช้งานเป็นฟังก์ชั่นพิเศษต่างๆเพิ่มเติมได้อีก
    - o D0-D1 ถูกสงวนไว้ใช้ทำหน้าที่เป็นพอร์ตสื่อสารอนุกรม RS232 (UART0) โดยได้ทำการเชื่อมต่อกับ USB Bridge ของ FT232RL เพื่อใช้ Upload Code ให้กับบอร์ด และยังสามารถใช้ทดลองติดต่อสื่อสารรับส่งข้อมูลระหว่างบอร์ดกับคอมพิวเตอร์ PC ได้ด้วย
    - o D2-D13 สามารถ โปรแกรมหน้าที่เป็น PWM ขนาด 8 บิต มี 14 Pin ได้
    - o D14 สามารถ โปรแกรมหน้าที่เป็น TX3 สำหรับ ส่งข้อมูลของ UART3 ได้ด้วย
    - o D15 สามารถ โปรแกรมหน้าที่เป็น RX3 สำหรับ รับข้อมูลให้กับ UART3ได้ด้วย
    - o D16 สามารถ โปรแกรมหน้าที่เป็น TX2 สำหรับ ส่งข้อมูลของ UART2 ได้ด้วย
    - o D17 สามารถ โปรแกรมหน้าที่เป็น RX2 สำหรับ รับข้อมูลให้กับ UART2ได้ด้วย
    - o D18 สามารถ โปรแกรมหน้าที่เป็น TX1 สำหรับ ส่งข้อมูลของ UART1 ได้ด้วย
    - o D19 สามารถ โปรแกรมหน้าที่เป็น RX1 สำหรับ รับข้อมูลให้กับ UART1ได้ด้วย
    - o D20 สามารถ โปรแกรมหน้าที่เป็น SDA ของ I2C Bus สำหรับใช้สื่อสารกับ I2C ได้ด้วย
    - o D21 สามารถ โปรแกรมหน้าที่เป็น SCL ของ I2C Bus สำหรับใช้สื่อสารกับ I2C ได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- AREF เป็นสัญญาณ Analog Reference จากภายนอกที่ต้องการป้อนให้กับ MCU ซึ่งตามปรกติแล้ว ATMEGA1280 สามารถโปรแกรมให้เลือกใช้แรงดันอ้างอิงจากภายในได้อยู่แล้วโดยสามารถเลือกเป็น 1.1V หรือ 2.56V หรือ AVCC(+5V) โดยไม่จำเป็นต้องป้อนแรงดันอ้างอิงจากภายนอกให้กับบอร์ดอีก แต่ถ้าต้องการแรงดันอ้างอิงที่มีความแตกต่างจากที่กล่าวมาแล้วก็สามารถป้อนเป็นแรงดันอ้างอิงจากภายนอกผ่านทางขา AREF นี้เข้าไปเองได้ระหว่าง 0-5V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โปรแกรมควบคุมส่วนต่างๆ

ค.1 โปรแกรมการควบคุมรถลำเลียง

```
/* 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                    rs2 3 2 ( baud=2 4 0 0 ,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.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#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.
#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.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งหมด ทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 */
#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;
  }
  Loopi++;
}
/**
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 * @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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
    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);
  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
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  /* 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);
}
 //* @brief Infinity Loop.
while(TRUE){
    /* Check Line 1 And Forward. *********/
```

เอกสารนี้จัดทำขึ้นสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // 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);
    }
    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;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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

ค.2 โปรแกรมควบคุมรถลำเลียงสินค้า

```
/* 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                    rs2 3 2 ( baud=2 4 0 0 ,parity=N,xmit=PIN_B7 ,rcv=PIN_B5 ,bits=8 ,
stream=AVR,timeout=KEYHIT_DELAY)

/* Define ------------------------------------------------------------*/
#define RobotNumber '4'         // 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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#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.
#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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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)){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น รวมถึงการนำไปอ้างอิงในวิทยานิพนธ์ต่างๆ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Output_Low(MotorDrive0);
    Output_Low(MotorDrive1);
  }
  else if(Loopi > ResMPWM){
    Loopi = 0;
  }
  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)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  /* 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);
  Setup_Comparator(NC_NC_NC_NC);
  Enable_Interrupts(INT_RTCC);
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    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);
}

/**
 * @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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  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();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Output_High(Led0);              // turn on running status led
      }

      /* Running Again ***************************************/
      do{
       Output_Low(Led0);              // turn on running status led
        MotorForward(MaxSpeed);
      }while(ANL0);
    }
    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){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  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

ค.3 โปรแกรมควบคุมMain 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){
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    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;
        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
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  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 ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**
 * @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';
  }
  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=");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    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]);
   }
   Serial.print("\n");
  }
  // Send to B
  else if((stackBu[0] == 'B')&&((stackBu[1]>=65)&&(stackBu[1]<=68))){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
  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';
  }
```

เอกสารนี้เผยแพร่เพื่อสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  else if(stackBu[1] == 'C'){
    keeping = '4';
  }
  else if(stackBu[1] == 'D'){
    keeping = '1';
  }
  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';
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    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

ค.4 โปรแกรมควบคุมสถานีรับ-ส่งสัญญาณ

```
/* 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.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 */
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)
{
  // 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);
}
/**
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 * @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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  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);
  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;
```

เอกสารนี้มีลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
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;
     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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        else{
          keep = parallelRead();
          if((keep==1)||(keep==2)||(keep==3)||(keep==4)){
            delay_ms(25);
            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;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      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

ค.5 โปรแกรม VB

เขียนเพื่อให้แสดงหน้าจอส่วนสั่งการและแสดงผล

```
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 สำหรับการแสดงภาพใน
แต่ละวิโดส์ว
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' 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

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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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")
        'b = InputBox("กรุณากรอกชื่อ", "ลงชื่อเข้าใช้งาน", "")
        'MessageBox.Show("สวัสดีครับคุณ " + b, "ยินดีต้อนรับ", MessageBoxButtons.OK)
        Label6.Text = " "
        Label5.Text = " "
    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'########## 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

'##############  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
        'Timer2.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")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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 ไม่ว่ากรณีใดๆทั้งสิ้น 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) 'ตัวแปรส่งค่าไปผ่านserialportเพื่อ
ส่งไปให้สถานีทำงาน
      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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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
   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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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"
              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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SRP.Write("A")
            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
          'MessageBox.Show("กรุณาเอาเครื่องหมายถูกออกก่อนทำการเลือก", "เลือกสั่งได้ทีละ
แบบนะ", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)

        ElseIf Ckb_stB.Checked = True Then
          Ckb_stB.CheckState = CheckState.Unchecked
          'MessageBox.Show("กรุณาเอาเครื่องหมายถูกออกก่อนทำการเลือก", "เลือกสั่งได้ทีละ
แบบนะ", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ElseIf Ckb_stC.Checked = True Then
            Ckb_stC.CheckState = CheckState.Unchecked
            'MessageBox.Show("กรุณาเอาเครื่องหมายถูกออกก่อนทำการเลือก", "เลือกสั่งได้ทีละ
แบบนะ", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)


        ElseIf Ckb_stD.Checked = True Then
            Ckb_stD.CheckState = CheckState.Unchecked
            'MessageBox.Show("กรุณาเอาเครื่องหมายถูกออกก่อนทำการเลือก", "เลือกสั่งได้ทีละ
แบบนะ", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)

        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
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
```

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If Rd_auto.Checked = True Then
            MessageBox.Show("กรุณากรอกตัวเลขเป็นตัวเลขครับ แล้วกดปุ่ม go", "กรอกจำนวน
รอบ", _
            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

'*****************************************************************************
************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

" ขั้นตอนสุดท้าย, การปิดแสดงภาพใน โปรแกรมของเรา,เลิกทำการติดต่อ device และ destroy (ทำลาย Object) ที่ เราสร้างขึ้นมาตอนต้นการเขียนโปรแกรม 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
      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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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
        bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)), Image)
        picCapture.Image = bmap
        ClosePreviewWindow()
        BtnSave.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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 ง

โปสเตอร์ระบบลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Wikipedia. "Computer-integrated manufacturing." [Online]. Available : http://en.wikipedia.org/wiki/Computer-integrated_manufacturing. 2010.

[2] Yoram Korem. Computer Control of Manufacturing Systems. McGraw Hill. Inc. 1983. 287pp.

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

[4] นิรุธ อำนวยศิลป์. คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0 ฉบับเพื่อการใช้งานจริง. กรุงเทพนคร : ซัคเซส มีเดีย.

[5] Webmaster. CNC."[Online].Available:http://pirun.ku.ac.th/~b4755376/link/CAD5.html.2553.

[6] Webmaster. "Computer-integrated manufacturing (CIM)." [Online]. Available : http://en.wikipedia.org/wiki/Computer-integrated_manufacturing.2553.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้