สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำเลียง

COMPUTER INTEGRATED MANUFACTURING

CONVEYOR SYSTEM SECTION

T119327

นาย พรชัย พงศ์พิพัฒน์ภักดี

นาย สุทธิพจน์ ชาญสิริธนภัทร

นาย อภิสิทธิ์ ลีเลิศยุทธ์

เลขหมู่.....
เลขทะเบียน 119327
วัน,เดือน,ปี - 7 ส.ค. 2554

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTER INTEGRATED MANUFACTURING :

CONVEYOR SYSTEM SECTION

Mr. Pornchai Pongpipatpakdee

Mr. Suttipoj Chansiritanapat

Mr. Apisit Leeleryut

THIS THESIS IS SUBMITTED IN PARTAIL 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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำเลียง

COMPUTER INTEGRATED MANUFACTURING

: CONVEYOR SYSTEM SECTION

ผู้จัดทำ นาย พรชัย พงศ์พิพัฒน์ภักดี

นาย สุทธิพจน์ ชาญสิริธนภัทร

นาย อภิสิทธิ์ ลีเลิศยุทธ์

.....อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร.วันชัย ริ้วรุจา)

.....อาจารย์ที่ปรึกษา

(ดร.วรรณดี เพชรมณีล้ำค่า)

.....อาจารย์ที่ปรึกษา

(อาจารย์ เทพจิตร์ เชยโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การผลิตผสมผสานด้วยคอมพิวเตอร์ : ส่วนระบบลำเลียง

โดย

นายพรชัย พงศ์พิพัฒน์ภักดี รหัส 50011036

นายสุทธิพจน์ ชาญสิริธนภัทร รหัส 50011040

นายอภิสิทธิ์ ลีเลิศยุทธ์ รหัส 50011872

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร.วันชัย ริ้วรุจา

ดร.วรรณดี เพชรมณีล้ำค่า

อาจารย์เทพจิตร์ เชยโภคา

ปีการศึกษา 2553

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอทฤษฎีและระบบจำลองกระบวนการผลิตอัตโนมัติ โดยโครงสร้างของระบบประกอบไปด้วย ระบบลำเลียง แขนกล คอมพิวเตอร์ ไมโครคอนโทรลเลอร์ และวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง จุดมุ่งหมายของโครงงานนี้ คือการควบคุมระบบจำลองกระบวนการผลิตอัตโนมัติ แสดงผลและทำงานตามคำสั่งที่ได้ป้อนเข้าไป

ขั้นตอนการดำเนินการ เริ่มจากการออกแบบโครงสร้างของระบบกระบวนการผลิตอัตโนมัติ ศึกษาการทำงานของไมโครคอนโทรลเลอร์แต่ละประเภท และเลือกหาแบบที่เหมาะสม ออกแบบวงจรอิเล็กทรอนิกส์ที่สอดคล้อง ออกแบบตัวพาหนะที่ใช้ในการลำเลียง การทำงานใช้ไอซี PIC ไมโครคอนโทรลเลอร์ในการควบคุมทั้งหมด ซึ่งใช้ภาษาซี และ Visual Basic เพื่อที่จะรับค่าป้อนเข้ามา แล้วกำหนดให้ตัวพาหนะและแขนกลทำงานตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTER INTEGRATED MANUFACTURING

: CONVEYOR SYSTEM SECTION

By

Mr. Pornchai Pongpipatpakdee 50011036

Mr.Suttipoj Chansiritanapat 50011040

Mr.Apisit Leeleryut 50011872

Advisors

Assoc.Prof.Dr.Vanchai Riewruja

Dr. Wandee Petchmaneelumka

Mr. Thepjit Cheypoca

Academic Year 2010

ABSTRACT

This thesis presents theory and simulation of automation process. The structure of the system includes conveyor system, robot arm, computer, microcontroller and electronic circuits.The aim of this project is to control the simulated automation manufacturing system and operate under the commend.

Procedure of process starts from design the structure of process automation systems. Each type of microcontroller is studied and selected for the proper form. Electronic circuits and vehicle for transportation are designed, PIC microcontroller is used for control all of system with C language and Visual Basic programs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถที่จะสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับการช่วยเหลือเป็นอย่างดีจาก รองศาสตราจารย์ ดร.วันชัย ริ้วรุจา ดร.วรรณดี เพชรมณีล้ำค่า อาจารย์เทพจิตร์ เชยโภคา ที่ได้ให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งคอยถามถึงความคืบหน้าอยู่ตลอดเวลา และความช่วยเหลืออื่นๆ ที่เป็นประโยชน์ต่อโครงงาน ผู้จัดทำรู้สึกซาบซึ้งและกราบขอบพระคุณอย่างสูง

ขอบคุณเพื่อนๆพี่ๆ ที่ให้ความช่วยเหลือในส่วนของการเขียนโปรแกรม ซึ่งเป็นส่วนที่สำคัญของปริญญานิพนธ์นี้

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่สำคัญตลอดการทำงานรวมถึงสนับสนุนงบประมาณช่วยเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงงานนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นาย พรชัย พงศ์พิพัฒน์ภักดี

นาย สุทธิพจน์ ชาญสิริธนภัทร

นาย อภิสิทธิ์ ลีเลิศยุทธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VIII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน	1
1.4 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ภาพรวมของระบบจำลองกระบวนผลิตอัตโนมัติ	3
2.1 ส่วนลำเลียง	3
2.1.1 รถลำเลียงสินค้า	3
2.1.2 สถานี	4
2.2 แขนกล	5
2.3 หุ่นแมลง 6 ขา	6
2.4 โปรแกรมควบคุมการทำงาน	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
บทที่ 3 ทฤษฎีและความรู้ที่เกี่ยวข้อง	8
3.1 มอเตอร์ไฟฟ้ากระแสตรง	8
3.1.1 คุณสมบัติของมอเตอร์กระแสตรง	8
3.1.2 โมเดลคณิตศาสตร์ของมอเตอร์กระแสตรง	9
3.2 ตัวรับส่งสัญญาณอินฟราเรด (Infrared)	16
3.2.1 TSOP34836	17
3.2.1.1 คุณสมบัติของ TSOP34836	18
3.2.1.2 โครงสร้างการทำงานและวงจรภายในของ TSOP34836	18
3.2.1.3 รายละเอียดของ TSOP34836	19
3.3 ไมโครคอนโทรลเลอร์ PIC16F690	19
3.3.1 แรงดันในการทำงาน	20
3.3.2 สัญญาณนาฬิกา	20
3.3.3 คุณสมบัติของ PIC16F690	21
3.3.4 โครงสร้างการทำงานภายในของ PIC16F690	22
3.3.5 รายละเอียดของขา PIC16F690	23
3.3.6 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F690	24
3.3.7 การจัดสรรหน่วยความจำข้อมูล RAM ของ PIC16F690	25
3.3.8 การทำงานในโหมด PWM	27
3.3.8.1 หลักการสร้างสัญญาณ PWM	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
3.3.8.2 การกำหนดคาบเวลาของสัญญาณ PWM	27
3. 4 ไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 (ARDIUNO MEGA)	28
3.4.1 คุณสมบัติของ ET-EASY MEGA1280 (ARDIUNO MEGA)	28
3.5 การออกแบบ	30
3.5.1 กล้องดิจิตอลวีดีโอ	31
3.5.2 คอมพิวเตอร์	32
3.6 การศึกษาค้นคว้า	32
3.6.1 การวิจัยหรือศึกษาจากเอกสารต่างๆที่เกี่ยวข้อง	32
3.6.2 การวิจัยทดลอง	33
3.7 หลักการทำงานของระบบ	33
3.7.1 หลักการการทำงาน ส่วน Vehicle	33
3.7.2 หลักการทำงาน ส่วน Main Controller	34
3.7.3 หลักการทำงานส่วน Station	36
บทที่ 4 การทดลองและผลการทดลอง	37
4.1 การทดลองการเคลื่อนที่ของรถ	37
4.2 การทดลองการเคลื่อนที่ของรถ	42
บทที่ 5 บทวิจารณ์และสรุป	43
5.1 สรุป	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง **หน้า**

5.2 ปัญหาที่พบและแนวทางแก้ไข 43

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา 44

ภาคผนวก ก ส่วนโปรแกรมภาษาซีที่ใช้ในโครงงาน 45

ภาคผนวก ข โปสเตอร์ระบบลำเลียง 94

เอกสารอ้างอิง 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ หน้า

2.1 องค์ประกอบรวมของระบบ 3

2.2 ลักษณะของ Vehicle 4

2.3 ลักษณะของ Station 4

2.4 ลักษณะของ แขนกล 5

2.5 ลักษณะของ หุ่นยนต์แมลง 6

2.6 ลักษณะของ โปรแกรมVisualBasic 7

3.1 วงจรภายในของมอเตอร์กระแสตรง 9

3.2 อินพุตและเอาต์พุตของโมเดลทางคณิตศาสตร์ของมอเตอร์ 10

3.3 โมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น 10

3.4 แรงบิดต่างๆที่เกิดขึ้นต่อโหลดของมอเตอร์ 12

3.5 บล็อกไดอะแกรมของดีซีมอเตอร์โมเดล 13

3.6 แบบจำลองของ TSOP34836 และลักษณะของขาการใช้งาน 17

3.7 โครงสร้างการทำงานภายในของ TSOP34836 18

3.8 วงจรภายในของ TSOP34836 18

3.9 การจัดขาของ PIC16F690 21

3.10 โครงสร้างการทำงานภายใน PIC16F690 22

3.11 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F690 24

3.12 การจัดการหน่วยความจำข้อมูล RAM และตำแหน่งรีจิสเตอร์ ของ PIC16F690 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่	หน้า
3.13 บอร์ด ET-EASY MEGA1280 (ARDIUNO MEGA)	28
3.14 สถาปัตยกรรมภายในของ AVR	30
3.15 แผนผังรวมของระบบจำลองกระบวนการผลิตอัตโนมัติ	31
3.16 วงจรภายในส่วนของVehicle	33
3.17 วงจรภายในส่วนของVehicle	34
3.18 วงจรภายในตัวบอร์ด Et-easy mega128	35
3.19 แผนภาพแสดงการทำงานของส่วน Main Controller	35
3.20 ภาพรวมของวงจรส่วนStation	36
3.21 ภาพวงจรภายในส่วนของ Station	36
4.1 การ com port และการ เชื่อมต่อ	37
4.2 การจับภาพ	38
4.3 โหมดการทำงาน	38
4.4 รายละเอียดในโหมด Manual	39
4.5 Mode Auto	40
4.6 สถานีการทำงาน	41
4.7 ภาพรวมของระบบ	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 รายละเอียดของ TSOP 34836 19

3.2 รายละเอียดของขา PIC16F690 23

3.3 ค่า RP0 และค่า RP1 ในการเลือก Bank ของ PIC16F690 26

4.1 การเคลื่อนที่ของรถ 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

จากการที่ในปัจจุบันกระบวนการผลิตของโรงงานอุตสาหกรรมส่วนใหญ่ได้มีการนำเอาระบบควบคุมแบบอัตโนมัติเข้ามาใช้งานเพิ่มมากขึ้น โดยมีการนำเอาเทคโนโลยีสมัยใหม่เข้ามาทดแทนแรงงานมนุษย์มากขึ้น เพื่อลดต้นทุนในการผลิต เพิ่มประสิทธิภาพ ความแม่นยำ และความรวดเร็วในกระบวนการผลิต ทำให้ตระหนักถึงความสำคัญของระบบควบคุมแบบอัตโนมัติ จึงมีความสนใจศึกษาระบบการลำเลียงและแขนกลในโรงงานอุตสาหกรรม และได้ทำการจำลองระบบนี้ขึ้น เพื่อใช้ในการศึกษาทดลอง และ ทำความเข้าใจในกระบวนการ เพื่อพัฒนาขีดความสามารถการในทำงานของระบบควบคุมอัตโนมัติ ให้ตอบสนองต่อความต้องการของโรงงานในอนาคตได้

สำหรับปริญญานิพนธ์นี้ ทางคณะผู้จัดทำได้เลือกทำการศึกษาและสร้างชิ้นงานเกี่ยวกับระบบจำลองกระบวนการผลิตอัตโนมัติ โดยได้จำลองกระบวนการควบคุม ซึ่งถือเป็นตัวอย่างของกระบวนการลำเลียงอัตโนมัติ

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. เพื่อศึกษาทฤษฎีการใช้ไมโครคอนโทรลเลอร์

2. เพื่อศึกษาอุปกรณ์ต่างๆที่จะนำมาสร้างให้เป็นระบบจำลองกระบวนการผลิตอัตโนมัติ และทดลองอุปกรณ์ต่างๆที่จะนำมาใช้งานจริง

1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน

1. ศึกษาส่วนประกอบทั้งหมดที่เกี่ยวข้องในการทดลองทั้งหมดในการทำระบบจำลองกระบวนการผลิตอัตโนมัติ ค้นคว้าข้อมูลและศึกษาการทำงานทางกลและทางอิเล็กทรอนิกส์ ควบคู่กับการศึกษาการทำงานของอุปกรณ์ต่างๆที่ควบคุมแบบอัตโนมัติ และศึกษาการเขียนโปรแกรม

2. จัดหาวัสดุและส่วนประกอบต่างๆทั้งทางกลและทางอิเล็กทรอนิกส์ที่จะนำมาใช้ประกอบขึ้นเป็นระบบ ได้แก่ ไมโครคอนโทรลเลอร์,มอเตอร์ไฟฟ้ากระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทดสอบโปรแกรมควบคุมการทำงานของอุปกรณ์

4. ประกอบเป็นโครงสร้างแบบจำลองระบบจำลองกระบวนการผลิตอัตโนมัติ และทดสอบการทำงานของอุปกรณ์ทั้งหมด

5. เขียนโปรแกรมเพื่อควบคุมการทำงานของอุปกรณ์ที่ประกอบขึ้นมาให้สามารถควบคุมได้ทั้งหมดพร้อมทดสอบการทำงานของโปรแกรม

6. ทดลองและบันทึกผลการทดลอง รวมถึงสรุปผลการทำงานทั้งหมด

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ ขั้นตอนการศึกษา และการจัดทำโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์แต่ละบท

บทที่ 2 ภาพรวมและส่วนประกอบของระบบการผลิตผสมผสานด้วยคอมพิวเตอร์

บทที่ 3 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึง การศึกษาค้นคว้า อธิบายส่วนประกอบและหลักการเลือกอุปกรณ์ ขั้นตอนในการสร้าง และหลักการทำงานส่วนต่างๆ

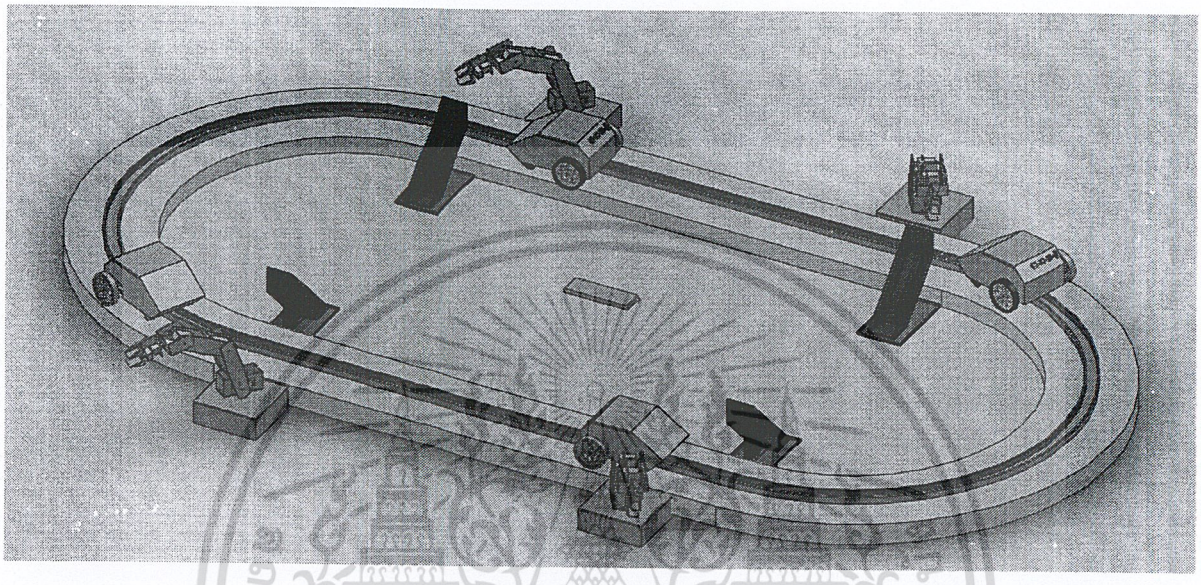
บทที่ 4 การทดลอง กล่าวถึง การทดลองการทำงานของระบบ

บทที่ 5 บทวิจารณ์และสรุป กล่าวถึงบทสรุปของโครงงาน ปัญหาที่เกิดขึ้น แนวทางในการแก้ไข และแนวทางในการปรับปรุงโครงงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ภาพรวมของระบบในโครงงาน



รูปที่ 2.1 องค์ประกอบรวมของระบบ

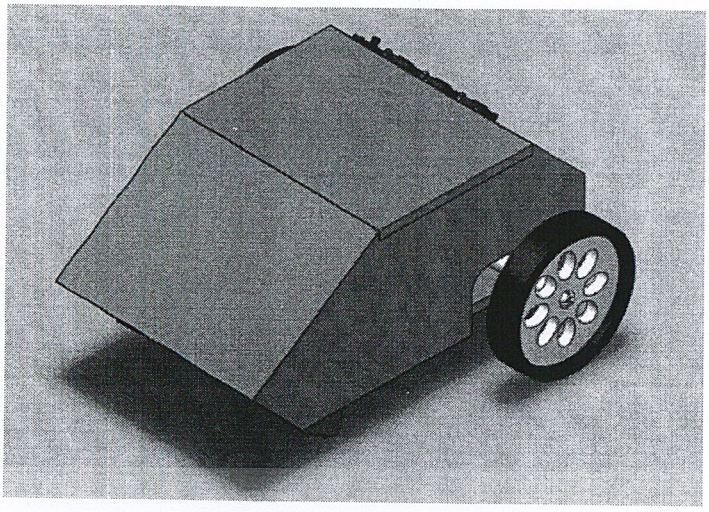
สำหรับระบบลำเลียงนั้นเราได้แบ่งส่วนประกอบต่างๆออกเป็นส่วนย่อยๆ ตามรูปที่ 2.1 ซึ่งในแต่ละส่วนจะมีหน้าที่ หลักการทำงาน และการสื่อสารที่แตกต่างกันโดยเราสามารถแบ่งส่วนประกอบต่างๆของระบบลำเลียง ออกได้เป็น 4 ส่วน ดังนี้

2.1 ส่วนของระบบลำเลียง

2.1.1 ส่วนรถลำเลียง

ในส่วนของ รถลำเลียงนั้นนอกจากจะมีวงจรขับ เพื่อขับให้รถวิ่ง และ วงจรตรวจจับ(Sensor) ทำหน้าที่ตรวจจับ เพื่อสั่งให้รถหยุดแล้ว ส่วนรถลำเลียงยังมีวงจร รับและส่งสัญญาณ อินฟราเรด (Infrared) เพิ่มเติมมาอีกหนึ่งวงจรด้วย ลักษณะของ รถลำเลียงเป็นดังรูปที่ 2.2

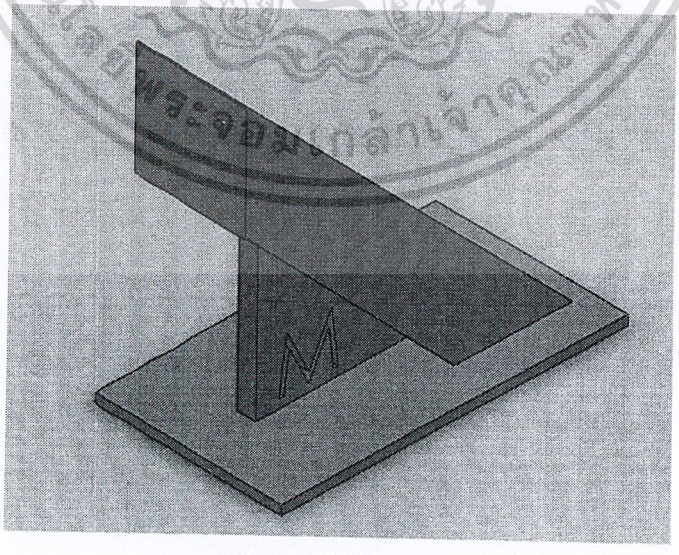
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ลักษณะของ รถลำเลียง

2.1.2 ส่วนสถานี

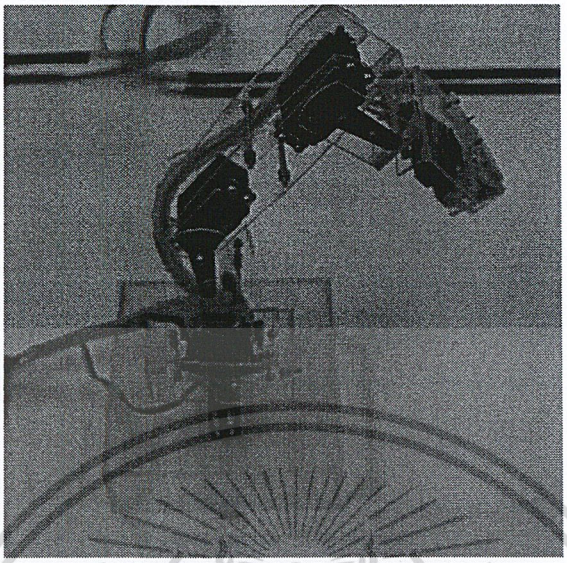
ในส่วนของ สถานีนั้น นอกจากจะมีวงจร รับและส่งสัญญาณอินฟราเรด ซึ่งจะเหมือนกับส่วนของรถลำเลียงแล้ว ยังมีวงจรควบคุมการทำงานของสถานี อีก1วงจรด้วย โดยลักษณะสถานีเป็นดังรูปที่ 2.3



รูปที่ 2.3 ลักษณะของสถานี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ส่วนของระบบแขนกล



รูปที่ 2.4 ลักษณะของแขนกล

ในส่วนของแขนกลจะทำหน้าที่หยิบจับชิ้นงาน เพื่อมาบรรทุกลงในรถ และมีหน้าที่หยิบชิ้นงานออกจากหลังรถ โดยที่ลักษณะของแขนกลจะเป็นดังรูปที่ 2.4

ส่วนประกอบของแขนกล จะแบ่งออกเป็น 4 ส่วนหลักๆ คือ

1 ส่วนควบคุมมีหน้าที่ควบคุมการทำงานของแขนกลทั้งระบบซึ่งในส่วนนี้จะรับค่าจากปุ่ม

2 ส่วนประมวลผล ทำหน้าที่ในการประมวลผลคำสั่งที่ได้จากส่วนควบคุมจากSTAMP 16 แล้วทำการประมวลผลค่าที่ได้มา จากนั้นทำการสั่งการไปยังเซอร์โวมอเตอร์ให้ทำงาน

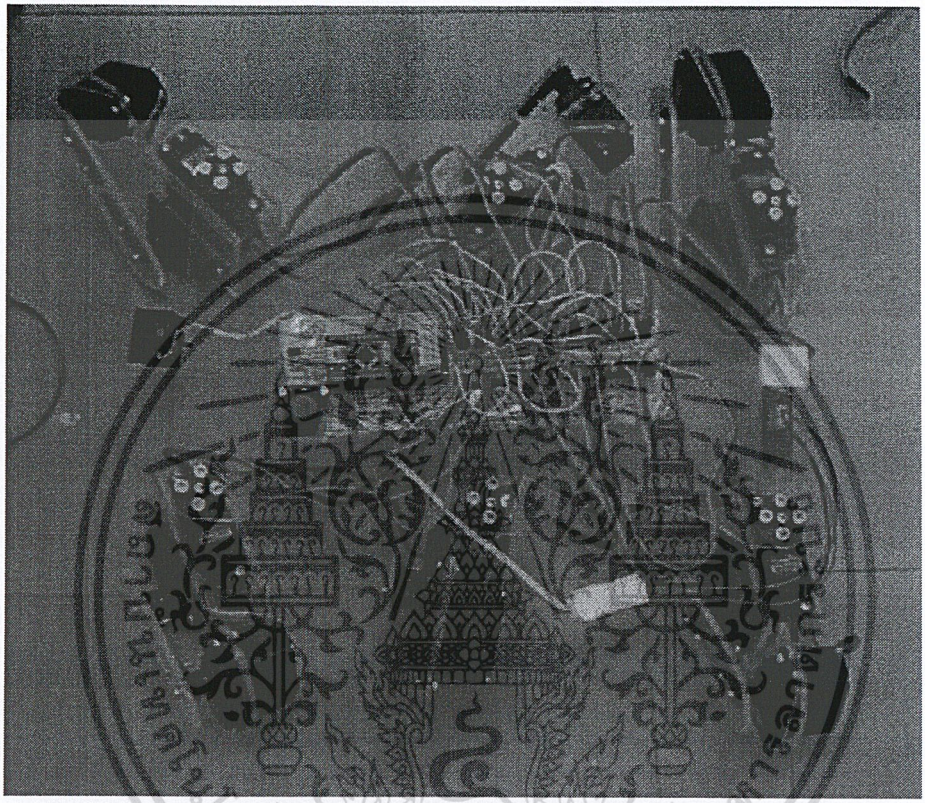
3 ในส่วนของเซอร์โวมอเตอร์นั้นเป็นส่วนที่ขับเคลื่อนตัวแขนกลให้เคลื่อนไหวไปยังตำแหน่งที่ต้องการซึ่งจะถูกติดตั้งยังข้อต่อของตัวแขนกล

4 ส่วนที่ประกอบเป็นแขนกล จะแบ่งออกเป็น 3 ส่วน คือ คลิปเปอร์ แขน และฐาน ทุกชิ้นส่วนทำจากแผ่นอะคริลิคซึ่งออกแบบจากโปรแกรมSolidworksแล้วกัดชิ้นงานด้วยเครื่องจักร (Computer Numerical Control , CNC) และนำมาประกอบเข้ากับเซอร์โวมอเตอร์เป็นแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หุ่นยนต์แมลง

เป็นการควบคุมแบบอัตโนมัติ ซึ่งจะรับชิ้นงานจากแขนกลซึ่งหยิบมาจากรถขนส่งภายในระบบลำเลียงสินค้าเพื่อจะนำชิ้นงานมาเก็บไว้ที่คลังสินค้า (Store) หรือ สถานีต่อไปในระบบ และสามารถควบคุมด้วยบอร์ดควบคุม เมื่อต้องการใช้กับการขนส่งแบบอื่น ๆ ที่ต้องการตามความเหมาะสมของงานนั้น ๆ โดยที่ลักษณะของแขนกลจะเป็นดังรูปที่ 2.5

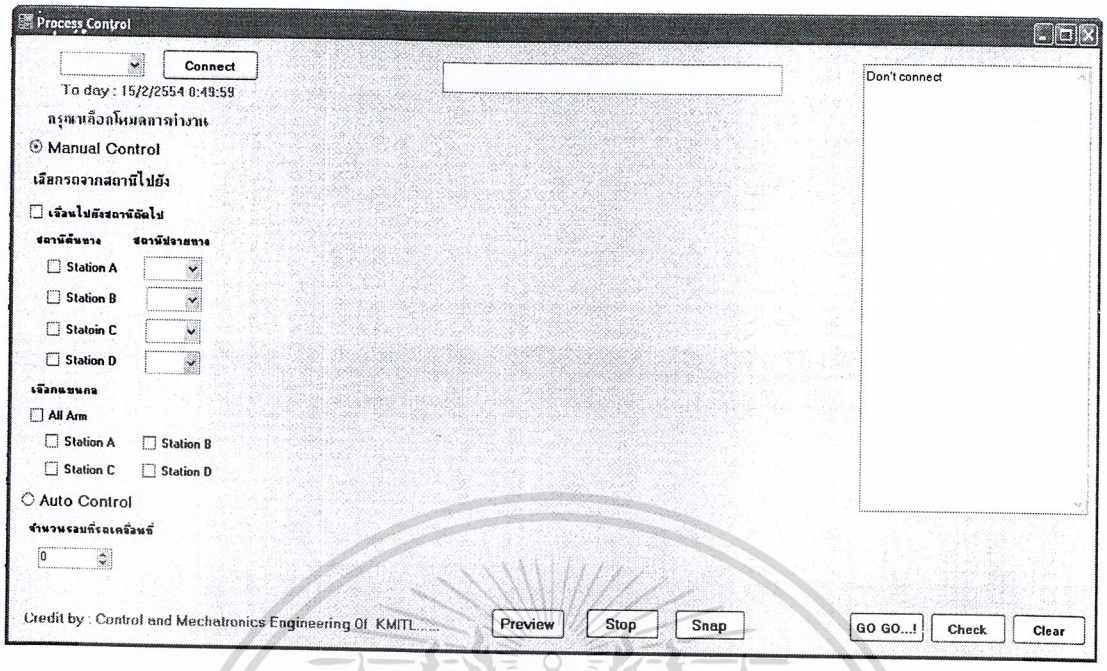


รูปที่ 2.5 ลักษณะของหุ่นยนต์แมลง

2.4 โปรแกรมควบคุมการทำงาน

โปรแกรมที่ใช้ในการควบคุมระบบคือโปรแกรม VisualBasic ซึ่งเป็นโปรแกรมที่ทำหน้าที่ตรวจสอบและสั่งการทำงานบนคอมพิวเตอร์ โปรแกรม VisualBasic มีลักษณะดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ลักษณะของโปรแกรม VisualBasic

เราแบ่งการทำงานของระบบ เป็น 2 โหมดคือโหมดควบคุมด้วยมือ(Manual mode) และโหมดควบคุมอัตโนมัติ (Automatic mode)

โหมดควบคุมด้วยมือ

ในการทำงานในโหมดควบคุมด้วยมือ นั้นเราจะเป็นผู้สั่งการทำงานของระบบผ่านโปรแกรมดังรูปที่ 2.6 โดยที่เราสามารถกำหนดได้ว่า จะให้รถที่สถานีที่เราต้องการไปจอดในสถานีที่เรากำหนด โดยเราต้องทำการเลือกสถานีต้นทาง และสถานีปลายทาง

โหมดควบคุมอัตโนมัติ

สำหรับการทำงานใน โหมดควบคุมอัตโนมัติ นั้น จะมีลำดับการทำงานดังนี้

1) รถทั้ง4คันจะเข้ามาจอดประจำแต่ละสถานี
2) แขนกลจะทำงานโดยแบ่งการทำงานออกเป็น2 แบบคือ แขนกลสถานี A กับ C จะหยิบของออกจากหลังรถ แขนกลสถานี B กับ D จะหยิบของมาวางไว้หลังรถ
3) เมื่อแขนกลทำงานเสร็จ รถทั้ง4คันจะเคลื่อนที่ไปจอดที่สถานีถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ทฤษฎีและความรู้ที่เกี่ยวข้อง

จากที่ได้กล่าวมาในบทที่ 1 แล้วว่า ก่อนที่จะทำการสร้างชิ้น โครงงานตลอดจนรายละเอียดต่างๆจำเป็นอย่างยิ่งที่จะต้องมีการศึกษาองค์ประกอบต่างๆที่จำเป็นของระบบควบคุมให้เข้าใจอย่างถ่องแท้เสียก่อน เริ่มจากมอเตอร์และเซนเซอร์ซึ่งในปัจจุบันมีหลากหลายประเภท ซึ่งขึ้นอยู่กับการใช้งาน แล้วเลือกมอเตอร์ (Motor) และเซนเซอร์ (Sensor) ที่จะนำมาควบคุมให้เหมาะสมการควบคุมระบบอัตโนมัตินั้นต้องรู้ทฤษฎีระบบควบคุมแบบต่างๆ รวมถึงการปรับแต่งค่าตัวควบคุม จากนั้นจะเป็นในส่วนของการทำงานของไมโครคอนโทรลเลอร์ โดยในที่นี้เลือกใช้ PIC16F690 และ AVR ATMEGA1280 โดยมีรายละเอียด ดังนี้

3.1 มอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรงที่สำคัญ มีองค์ประกอบที่สำคัญอยู่ 4 ส่วน คือ สนามแม่เหล็กกระแสอาร์เมเจอร์ซี่คอมมิวเตเตอร์และแปรงถ่าน

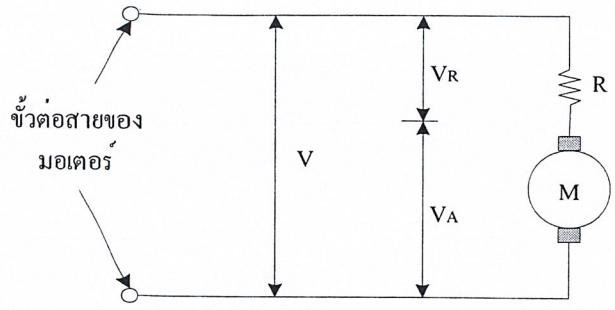
- สเตเตอร์ (Stator) คือส่วนที่อยู่กับที่ทำหน้าที่สร้างสนามแม่เหล็กเมื่อมีการป้อนแรงดันไฟฟ้ากระแสตรงให้กับขดลวด (Main winding) หรือ (Filed winding)

- โรเตอร์ (Rotor) คือส่วนที่หมุนทำจากแผ่นเหล็กบางๆมาประกบกันแล้วเซาะร่องสลอตและพันด้วยขดลวดซึ่งเรียกว่าขดลวดอาแมเจอร์ (Armature winding) ที่ปลายด้านหนึ่งมีชุดซี่คอมมิวเตเตอร์ซึ่งเป็นทองแดงวางอยู่และมีแปรงถ่าน

3.1.1 คุณสมบัติของมอเตอร์กระแสตรง

ในการอธิบายคุณสมบัติของมอเตอร์กระแสตรงให้ละเอียดนั้นต้องพิจารณาแรงดันที่ป้อนและความต้านทานของโรเตอร์ด้วย วงจรภายในของมอเตอร์เขียนได้ดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 วงจรภายในของมอเตอร์กระแสตรง

โดยสมมติให้ทุ่นโรเตอร์ไม่มีความต้านทานอยู่เลย ต่ออนุกรมกับความต้านทานซึ่งในที่นี้ก็คือความต้านทานของขดลวดนั่นเองแรงดันที่ขั้วต่อสายของมอเตอร์ก็คือผลบวกระหว่างแรงดันที่ทุ่นโรเตอร์ (V_A) และ แรงดันตกคร่อมความต้านทานขดลวด (V_R)

แรงดัน V_A ถูกเรียกว่าแรงเคลื่อนเหนี่ยวนำป้อนกลับ (Back EMF) ซึ่งเกิดขึ้นในโรเตอร์ขณะที่หมุนแรงดันที่เกิดขึ้นนี้เป็นไปตากกฎของการเหนี่ยวนำแม่เหล็กไฟฟ้าจากการเคลื่อนที่ของตัวนำในสนามแม่เหล็กสัมพันธ์กับแรงเคลื่อนเหนี่ยวนำแม่เหล็กและความเร็วในการเคลื่อนที่ของตัวนำ แรงดันที่เกิดขึ้นจะมีขั้วตรงกันข้ามกับแรงดันที่ป้อนให้กับมอเตอร์และแปรผันตรงกับความเร็วในการหมุนผลบวกของแรงดันที่ทุ่นโรเตอร์ (V_A) และแรงดันตกคร่อมขดลวด (V_R) ต้องเท่ากับแรงดันที่ป้อนให้กับมอเตอร์ (V)

$$V = V_A + V_R \qquad 3.1$$

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

$$I = V_R / R \qquad 3.2$$

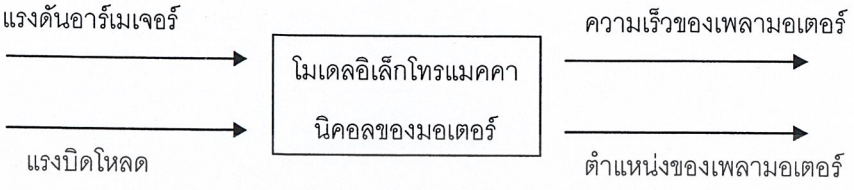
เมื่อมอเตอร์เริ่มหมุนจะมีความเร็ว และ V_A เพิ่มขึ้นเป็นเส้นตรงตามความเร็ว V_R ซึ่งมีค่าเท่ากับความแตกต่างระหว่าง V_A และ V จะเริ่มลดลงกระแส I ก็จะเริ่มลดลงเช่นกันขณะที่มอเตอร์ยังมีความเร่งอยู่ ความเร็วจะเพิ่มขึ้นแรงบิดจะลดลงจนกว่าจะถึงจุดซึ่งแรงบิดของมอเตอร์รับภาระโหลดได้สมดุลพอดีขณะที่มอเตอร์ไม่มีโหลดและหมุนอย่างอิสระจะมีเพียงค่าความฝืดของแบริ่งและแรงต้านอากาศทำให้ V_A เกือบเท่ากับค่า V

3.1.2 โมเดลคณิตศาสตร์ของมอเตอร์กระแสตรง

ดีซีมอเตอร์ที่ใช้ร่วมกับดีซีแอมพลิไฟล์ทั้งในระบบการบังคับตำแหน่งและการบังคับความเร็วมักจะได้รับการประยุกต์ใช้เป็นส่วนประกอบสร้างกำลังงานในระบบการนำร่องและระบบบังคับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

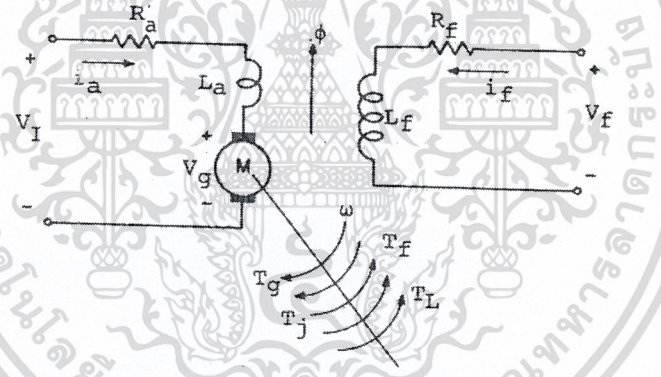
ต่างๆและเนื่องจากวิทยาการเกี่ยวกับสารแม่เหล็กและการขยายด้วยโซลิดสเตททำให้ดีซีมอเตอร์แบบแม่เหล็กถาวรได้รับความนิยมใช้เป็นส่วนประกอบการขับเคลื่อนในระบบการบังคับแบบวงปิดต่างๆมากขึ้นการออกแบบและการชดเชยระบบดังกล่าวได้อย่างเหมาะสมจะต้องใช้โมเดลทางคณิตศาสตร์ของส่วนประกอบทั้งหมดในระบบ



รูปที่ 3.2 อินพุตและเอาต์พุตของโมเดลทางคณิตศาสตร์ของมอเตอร์

3.1.3 โมเดลอิเล็กโทรแมคคานิคอล

ส่วนสำคัญของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้นมีโมเดลดังแสดงในรูป 3.3



รูปที่ 3.3 โมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น

R_a: ความต้านทานของอาร์เมเจอร์

L_a: อินดัคแต้นซ์ของอาร์เมเจอร์

V_g: โวลท์เต็จกำเนิดในอาร์เมเจอร์(โวลท์เต็จย้อนกลับ)

R_f: ความต้านทานของฟิลด์

L_f: อินดัคแต้นซ์ของฟิลด์

ϕ: ช่องว่างอากาศของเส้นแรงสนามแม่เหล็ก

ω: ความเร็วของเพลาอาร์เมเจอร์

T_g: แรงบิดที่พัฒนาขึ้นในมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

T_f: แรงบิดเสียดทานของมอเตอร์

T_J: แรงเฉื่อยของมอเตอร์

T_L: แรงบิดโหลดบนเพลาของมอเตอร์

ขั้นแรกเราจะหาสมการพื้นฐานโมเดลของดีซีมอเตอร์ได้จากลูฟของอาร์เมเจอร์

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

เทอมของโวลท์เตจ $V_g(t)$ คือโวลท์เตจย้อนกลับของมอเตอร์ซึ่งเกิดขึ้นเมื่อเส้นลวดตัวนำของอาร์มาเจอร์หมุนตัดเส้นแรงแม่เหล็กซึ่งเกิดขึ้นในกระแสของฟิลด์ (i_f) ตามกฎของฟาราเดย์ลูปของเส้นลวดตัวนำหมุนในฟิลด์แม่เหล็กคงที่จะมีการเหนี่ยวนำโวลท์เตจขึ้นในขดลวดนั้น

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

เมื่อ $\lambda(t)$ คือเส้นแรงแม่เหล็กที่ลิงเกจ (Linkages) ไปยังขดลวดและ t คือเวลาในการหมุนของคอมมิวเตเตอร์ของมอเตอร์ การควบคุมวงจรของแต่ละส่วนของตัวนำในโรเตอร์จะเกิดโวลท์เตจขึ้นในส่วนของตัวนำนั้นเมื่อ $\frac{d\lambda(t)}{dt}$ จะเป็นสัดส่วนต่อเส้นแรงแม่เหล็กในช่องว่างอากาศและความเร็วเชิงมุม $\omega(t)$ เราจะได้ว่า

$$V_g(t) = K\phi(t)\omega(t) \qquad 3.5$$

สมมติให้กระแสของฟิลด์มีค่าคงที่และไม่คิดถึงส่วนการเปลี่ยนแปลงในเส้นแรงฟิลด์เนื่องจากอาร์เมเจอร์รีแอคชั่นเส้นแรงฟิลด์ก็จะมีค่าคงที่ดังนั้นสมการ (3.5) ก็จะเป็น

$$V_g(t) = K_e\omega(t) \qquad 3.6$$

เมื่อเราสมมติให้เส้นแรงของฟิลมีค่าคงที่ แรงบิดของแม่เหล็กไฟฟ้าซึ่งเกิดขึ้นแก่โรเตอร์ของมอเตอร์จะเป็นสัดส่วนกับกระแสของอาร์มาเจอร์

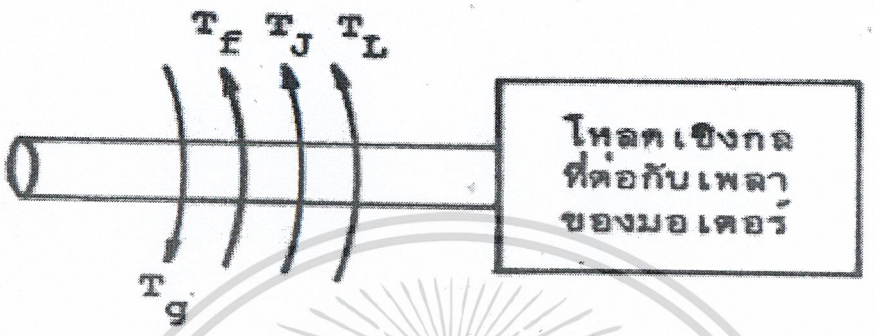
$$T_g(t) = K_t i_a(t) \qquad 3.7$$

เมื่อ K_t คือ ค่าคงที่ของแรงบิดของมอเตอร์ กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์คือผลคูณของแรงบิดที่เกิดขึ้นและความเร็วเชิงมุม

$$P_g(t) = T_g(t)\,\omega(t) \qquad 3.8$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์ทั้งหมดนี้จะจ่ายไปยังโหลดที่ต่ออยู่กับเพลาของมอเตอร์แต่กำลังงานนี้บางส่วนจะสูญเสียไปในมอเตอร์ การสูญเสียจากแรงเสียดทานหมายถึงความหน่วงเนื่องจากลมที่มีต่อโรเตอร์แรงเสียดทานตัวรองรับโรเตอร์ กระแสที่ไหลวนในเหล็กของโรเตอร์และฮีสเทริซิส โดยแรงบิดต่างๆแสดงดังนี้



รูปที่ 3.4 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

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

เมื่อ J คือผลรวมของโมเมนต์แรงเฉื่อยของโรเตอร์และโหลดที่ต่ออยู่ที่เพลาของมอเตอร์

ผลรวมของแรงบิดเสียดทานที่ประกอบกันขึ้นที่เพลาของมอเตอร์ ซึ่งเป็นลิเนียร์ฟังก์ชันกับความเร็วเชิงมุมของโรเตอร์เรียกว่าส่วนประกอบของวิสคอสฟิคชั่นและมักจะอยู่ในเทอมที่แยกออกจากฟิคชั่นอื่นๆ ซึ่งแสดงได้ด้วยสมการต่อไปนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ B คือสัมประสิทธิ์ของวิสคอสฟิคชั่นของมอเตอร์และโหลดที่ต่ออยู่กับเพลาของมอเตอร์ $T_f(t)$ คือผลรวมของฟริคชั่นของโหลดและมอเตอร์ทั้งหมด มีแรงต้านของลมและการสูญเสียกำลังในเหล็กของเพลามอเตอร์ยกเว้นวิสคอสฟิคชั่น

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

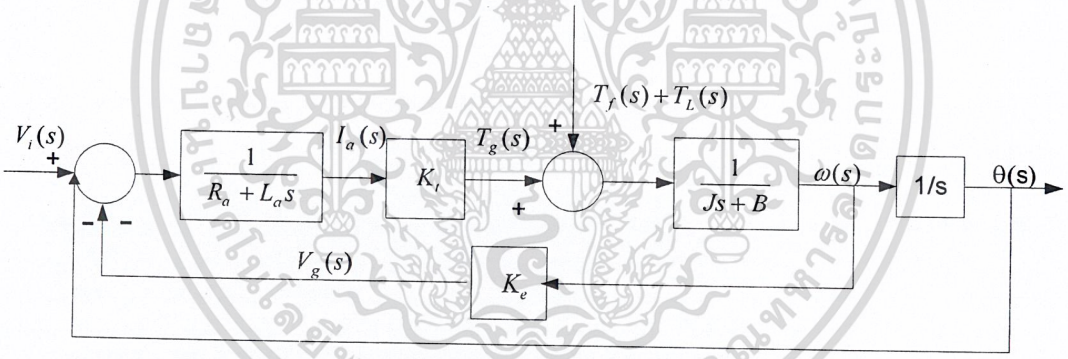
$$V_i(s) - V_g(s) = (R_a + sL_a)\, I_a(s) \qquad 3.11$$

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

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

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

สามารถเขียนเป็นบล็อกไดอะแกรมที่แสดงสมการพื้นฐานเหล่านี้ได้ดังนี้



รูปที่ 3.5 บล็อกไดอะแกรมของดีซีมอเตอร์โมเดล

ข้อสังเกต

สมมติว่าแรงดันที่ป้อนให้กับวงจรอาร์มาเจอร์ของมอเตอร์มีค่าคงที่ดังนั้นมอเตอร์จะหมุนด้วยความเร็วคงที่คือทำงานอยู่ที่สภาวะสงบนิ่งด้วยโหลดที่คงที่ กำลังงานเชิงกลที่เกิดขึ้นจะได้

$$P_g = T_g \omega = K_t\, I_a\, \omega \qquad 3.15$$

เมื่อทุกเทอมในสมการสุดท้ายมีค่าคงที่เนื่องจากมอเตอร์ทำงานอยู่ที่สภาวะสงบนิ่งกำลังไฟฟ้าที่ถูกดูดกลืนโดยอาร์เมเจอร์ต้องเท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P = V_g\, I_a = K_e\, \omega\, I_a \qquad 3.16$$

ดังนั้นเราจะได้ว่ากำลังงานเชิงกลที่เกิดขึ้นต้องเท่ากับกำลังงานไฟฟ้าที่ถูกดูดกลืนในโรเตอร์คือ สรุปได้ว่า $K_e = K_t$

ทรานสเฟอร์ฟังก์ชั่นของดีซีมอเตอร์ บล็อกไดอะแกรมของรูปที่ 3.5 แสดงถึงระบบที่มีสองอินพุท และมีเอาต์พุทเป็นความเร็วเชิงมุม $\omega(s)$ และการเคลื่อนที่แบบเชิงมุม $\omega(s)$ จากรูปที่ 3.5 ความเร็วเอาท์พุทของระบบเขียนได้เป็น

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

เมื่อ

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

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

$G_1(s)$ คือทรานสเฟอร์ฟังก์ชันระหว่างโวลท์เตจและความเร็ว

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

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

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

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

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ถ้าอินดัคแต้นท์ของอาร์มาเจอร์มีค่าน้อย ไทม์คอนสแต้นท์ทางไฟฟ้าสามารถตัดทิ้งได้และได้สมการเป็น

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

$$= \frac{K_m}{\tau s+1} \qquad 3.22$$

เมื่อ $$\tau = \frac{R_aJ}{R_aB+K_tK_e}$$

ในสมการ (3.22) ค่าคงที่ K_m อาจเรียกได้ว่าเป็นค่าคงที่ของมอเตอร์

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

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

ซึ่งจากสมการที่ (3.25) เมื่อให้ค่าของ ว่า T_f และ T_L มีค่าเป็นศูนย์จะทำให้ค่าทรานสเฟอร์ฟังก์ชั่นมีค่าดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเดลคณิตศาสตร์ในการควบคุมตำแหน่งของมอเตอร์จาก transfer function โมเดลคณิตศาสตร์ของดีซีมอเตอร์ ซึ่งจะได้ model ของการควบคุมความเร็วของมอเตอร์ในรูปของสมการอันดับหนึ่ง เป็นดังนี้ คือ

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

จากสมการข้างต้นดังกล่าวนั้นจะเห็นถึงความสัมพันธ์ระหว่างความเร็วเชิงมุม (output) และ ค่าแรงดันที่ป้อน (input) และในการควบคุมตำแหน่งจะมีการผ่านตัว Integrator (1/s) ทำให้ได้เอาท์พุต คือ มุมการหมุนของมอเตอร์ ซึ่งจากรูปที่ 3.5 และ transfer function ที่ได้เมื่อทำการผ่าน Integrator (1/s) เข้าไปจะทำให้ได้ค่าของ output เป็นมุมในการเคลื่อนที่ของมอเตอร์ ซึ่งเขียนในรูปของสมการได้ดังนี้

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

3.2 ตัวรับส่งสัญญาณอินฟราเรด

สิ่งที่สำคัญในการรับส่งสัญญาณอินฟราเรดคือรังสีอินฟราเรด อินฟราเรด (Infrared) คือ คลื่นแม่เหล็กไฟฟ้าที่มีความถี่อยู่ในช่วง 1011 – 1014 เฮิรตซ์ หรือความยาวคลื่น 10.3 – 10.6 เมตร หรือเรียกอีกอย่างหนึ่งว่า คลื่นความถี่สั้น (Millimeter waves)ซึ่งจะมีย่านความถี่คาบเกี่ยวกับย่านความถี่ของคลื่นไมโครเวฟอยู่บ้างวัตถุร้อนจะแผ่รังสีอินฟราเรดที่มีความยาวคลื่นสั้นกว่า 10.4 เมตร ออกมา เราสามารถอธิบายรังสีอินฟราเรดง่ายๆ ได้ดังนี้

- เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความถี่อยู่ในระหว่างแสงที่ตามองเห็น

- ลำแสงอินฟราเรดเดินทางเป็นเส้นตรง ไม่สามารถผ่านวัตถุทึบแสงและสามารถสะท้อนแสงในวัสดุผิวเรียบได้เหมือนกับแสงทั่วไป

- ใช้มากในการสื่อสารระยะใกล้ เช่น รีโมทคอนโทรลของเครื่องรับโทรทัศน์

- ปัจจุบันถูกพัฒนาใช้ในการสื่อสารไร้สาย สำหรับเครือข่ายเฉพาะบริเวณ

- เมื่อใช้ในการสื่อสารข้อมูลในเครือข่ายสามารถส่งสัญญาณได้ในระยะ 30-80 ฟุต หรือ 10-30 เมตร

- เป็นสื่อที่มีช่องสัญญาณกว้างพอประมาณในระดับสูงเมื่อเปรียบเทียบกับสายยูทีพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

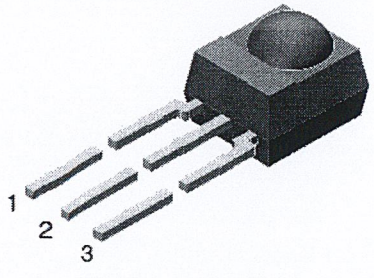
- อุปกรณ์คอมพิวเตอร์ที่มีช่องสื่อสารอินฟราเรด เรียกว่า IrDa (Infrared Data Association) สามารถสั่งงานระยะไกลได้ประมาณ1 – 5 เมตร

- เป็นระบบสำหรับควบคุมการทำงานของอุปกรณ์ต่างๆ จากระยะไกล โดยรังสีอินฟราเรดจะเป็นตัวนำคำสั่งจากเครื่องควบคุมไปยังเครื่องรับ

คุณสมบัติของรังสีอินฟราเรดคือ เป็นคลื่นสั้นทางเดินของแสงเป็นแนวตรง ราคาถูกเพราะง่ายต่อการผลิต มีความปลอดภัยต่อการดักสัญญาณ และไม่สามารถทะลุผ่านวัตถุ ทำให้สามารถติดตั้งอินฟราเรดในห้องทำงานติดกันได้ โดยในงานที่เราใช้นั้นเราได้ใช้เซนเซอร์ตรวจจับแสงอินฟราเรด เพราะแสงอินฟราเรด คือแสงที่มีความยาวคลื่นต่ำกว่าแสงสีแดงลงไป ดังนั้นจึงไม่สามารถมองเห็นได้ด้วยสายตา ของมนุษย์ ซึ่งคุณสมบัตินี้เอง จึงทำให้ เซ็นเซอร์ชนิดที่ใช้แสงอินฟราเรด เป็นที่นิยมนำมาใช้กันมาก โดยจะอาศัยหลักการของการสะท้อนของแสง กล่าวคือ ใช้อุปกรณ์ส่งแสง เป็นแหล่งกำเนิด ปล่อยแสงออกไป และเมื่อแสงกระทบกับวัตถุด้านหน้า มันก็จะสะท้อนแสงกลับมาเข้าที่ตัวรับแสง ส่วนอัตราของการสะท้อนกลับนั้น ขึ้นอยู่กับสี และสภาพความมัน ของวัตถุที่สะท้อน เช่น สีดำ จะมีอัตราการสะท้อนกลับ น้อยกว่าสีขาว , หรือสภาพพื้นผิวที่มีความราบเรียบเป็นมันวาว จะสามารถสะท้อนแสงได้ดีกว่า พื้นผิวที่มีลักษณะด้าน และขรุขระ เป็นต้น สวนในงานของเราเราได้ใช้ตัวรับส่งสัญญาณอินฟราเรดสามขา มีข้อมูลดังต่อไปนี้

3.2.1 TSOP34836

TSOP34836 เป็นอุปกรณ์ตัวรับส่งของบริษัท VISHAY เป็น ตัวควบคุมระยะไกลของ IR Receiver Modules มีขาใช้งานทั้งหมด 3 ขา ขาที่ 1 มีหน้าที่เป็นเอาต์พุต(OUT) ขาที่ 2 เป็นสายดิน (GND) และขาที่ 3 เป็นไฟเลี้ยง(V_s) ดังรูปที่ 3.6



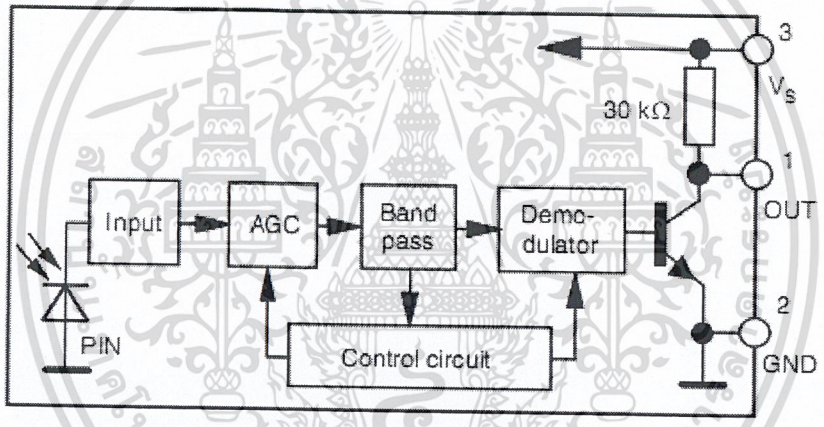
รูปที่ 3.6 แบบจำลองของ TSOP34836 และลักษณะของขาการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
119327

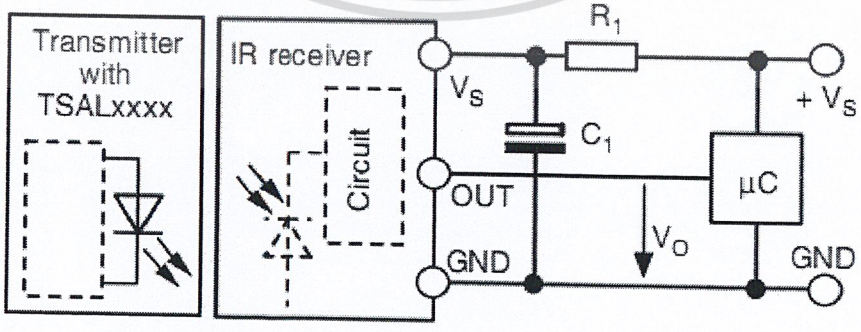
3.2.1.1 คุณสมบัติของ TSOP34836

- ใช้กระแสไฟฟ้าน้อยมาก
- มีตัวตรวจจับ Photo และ Preamplifier ในตัวเดียวกัน
- ภายในมีการกรองความถี่ PCM
- มีการป้องกันค่า EMI
- มีปรังปรุงค่าต่อการรบกวนของแสง
- ไม่มีผลต่อ คลื่นสัญญาณของแรงดัน(supply voltage ripple) และสัญญาณรบกวน(noise)
- ชิ้นส่วนเป็นไปตาม RoHS 2002/95/EC และ WEEE2002/96/EC
- ใช้สัญญาณความถี่เท่ากับ 36 kHz

3.2.1.2 โครงสร้างการทำงานและวงจรภายในของ TSOP34836



รูปที่ 3.7 โครงสร้างการทำงานภายในของ TSOP34836



รูปที่ 3.8 วงจรภายในของ TSOP34836

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8สามารถอธิบาย R_1 และ C_1 มีการใช้งานเพื่อป้องกันการ EOS โดยส่วน ประกอบควรจะอยู่ในช่วงของ 33 Ω <R1 <1 kΩ, C1> 0.1 μF

3.2.1.3 รายละเอียดของ TSOP34836

ตารางที่ 3.1 รายละเอียดของ TSOP 34836

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Supply voltage (pin 3)		VS	- 0.3 to + 6.0	V
Supply current (pin 3)		IS	3	mA
Output voltage (pin 1)		V_O	- 0.3 to (V_S + 0.3)	V
Output current (pin 1)		I_O	5	mA
Junction temperature		T_j	100	°C
Storage temperature range		T_{stg}	- 25 to + 85	°C
Operating temperature range		T_{amb}	- 25 to + 85	°C
Power consumption	Tamb $\leq$ 85 °C	P_{tot}	10	mW
Soldering temperature	t $\leq$ 10 s, 1 mm from case	T_{sd}	260	°C

3.3 ไมโครคอนโทรลเลอร์ PIC16F690

ในปัจจุบันไมโครคอนโทรลเลอร์ตระกูล PIC มีการพัฒนาและผลิตออกมาหลายเบอร์ โดยทั่วไปจะแบ่งออกเป็น 3 แบบคือ

1. สถาปัตยกรรมแบบ 12-Bit Core (Base-Line) เป็นกลุ่มของไมโครคอนโทรลเลอร์ที่มีขนาดเล็กมีโครงสร้างของคำสั่งเพียง 12 bit และค่อนข้างมีข้อจำกัดในการใช้งาน เนื่องจากมีหน่วยความจำ RAM และ STACK ค่อนข้างจำกัด
2. สถาปัตยกรรมแบบ 14-Bit Core(Mid-Range) เป็นกลุ่มของไมโครคอนโทรลเลอร์ที่มีขนาดกลางมีโครงสร้างคำสั่ง 14 bit มีทั้งแบบที่โปรแกรมครั้งเดียว (OTP : One Time Programmable) และแบบเฟลช (Flash Memory)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สถาปัตยกรรมแบบ 16-Bit Core(High-End) เป็นกลุ่มของไมโครคอนโทรลเลอร์ที่อยู่ในกลุ่มระดับสูงซึ่งได้มีการพัฒนาโครงสร้างสถาปัตยกรรมทั้งในเรื่องหน่วยความจำ ความเร็วและคุณสมบัติอื่นๆ ที่เหนือกว่าสองกลุ่มที่ผ่านมามาก มีการจัดวางหน่วยความจำโปรแกรมอยู่ในเพจ (page) เดียวกันทำให้ไม่มีปัญหาเรื่องรอยต่อของหน่วยความจำ

นอกจากนี้แล้ว PIC ยังแบ่งออกเป็นประเภทของหน่วยความจำอีก โดยจะมีการจำแนกเป็น 3 ประเภท คือ

1. C เช่น PIC16CXXX โครงสร้างหน่วยความจำเป็น EPROM จัดอยู่ในจำพวกอุปกรณ์ OTP แต่สามารถลบได้ด้วยแสง UV
2. CR เช่น PIC16CRXXX คือโครงสร้างหน่วยความจำเป็น ROM จัดอยู่ในจำพวกอุปกรณ์ OTP ไม่สามารถลบได้
3. F เช่น PIC16FXXX คือโครงสร้างหน่วยความจำเป็น FLASH Memory สามารถทำการโปรแกรม ลบ แล้วโปรแกรมซ้ำได้หลายครั้ง

3.3.1 แรงดันในการทำงาน

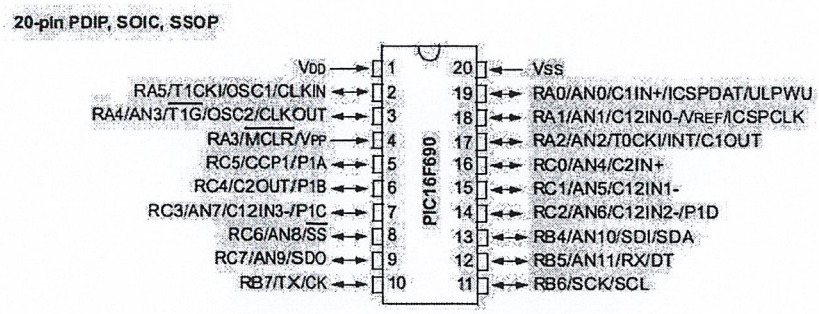
ช่วงแรงดันการทำงานของ PIC โดยปกติมาตรฐานแล้วจะอยู่ระหว่าง 4.5-6.0 V แต่จะมีบางเบอร์ที่ออกแบบมาให้สามารถทำงานได้ในช่วงแรงดันต่ำประมาณ 2.5-6 V ได้ซึ่งจะมีการระบุโค้ดให้ทราบ โดยมีการเพิ่มรหัสตัว L เข้าไปในเบอร์ของอุปกรณ์ เช่น PIC16LFXXX เป็นต้น

3.3.2 สัญญาณนาฬิกา

PIC จะใช้สัญญาณนาฬิกาโดยมองเป็นลักษณะของ วงรอบ (cycle) ซึ่งระบุเอาไว้ว่า 1 คำสั่งนั้นจะประกอบไปด้วย 1-2 วงรอบ โดยแต่ละวงรอบนั้นจะแบ่งเป็น 4 ส่วน คือ Q_1 , Q_2 , Q_3 และ Q_4 ด้วยเหตุนี้ความเร็วโดยรวมของ PIC จึงเท่ากับ ค่าความถี่ของสัญญาณนาฬิกา หารด้วย 4

$$1 \text{ cycle} = Q_1 + Q_2 + Q_3 + Q4 = \frac{XTAL}{4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 การจัดขาของ PIC16F690

3.3.3 คุณสมบัติของ PIC16F690

1. Controller Family / Series : PIC16F
2. Core Size : 8 bit
3. No. of I/O's : 18
4. Program Memory Size : 4 K words
5. EEPROM Memory Size : 256 Byte
6. RAM Memory Size : 256 Byte
7. CPU Speed : 20 MHz
8. Oscillator Type : External , Internal
9. No. of Timers : 3
10. No. of PWM Channels:4
11. Digital IC Case Style : PDIP
12. Supply Voltage Range:2V to 5.5 V
13. No. of Pins : 20
14. Operating Temperature Max : 125
15. Temperature Operating Min : -40 °C
16. Base Number : PIC 16 °F
17. Clock Frequency : 20 MHz
18. Device Marking : PIC16F690-E/P
19. Flash Memory Size : 7 KB
20. IC Generic Number : 16F690
21. IC Temperature Range : Extended
22. Interface : EUSART , I2C , SPI , SSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

23. Interface Type:EUSART, I2C, SPI

24. Logic Function Number : 16F690

25. No. of ADC Inputs : 12

26. No. of Bits : 8

27. No. of I/O's : 18

28. Number of bits in ADC : 10

29. Operating Temperature Range : -40°C to +125°C

30. Package / Case : PDIP

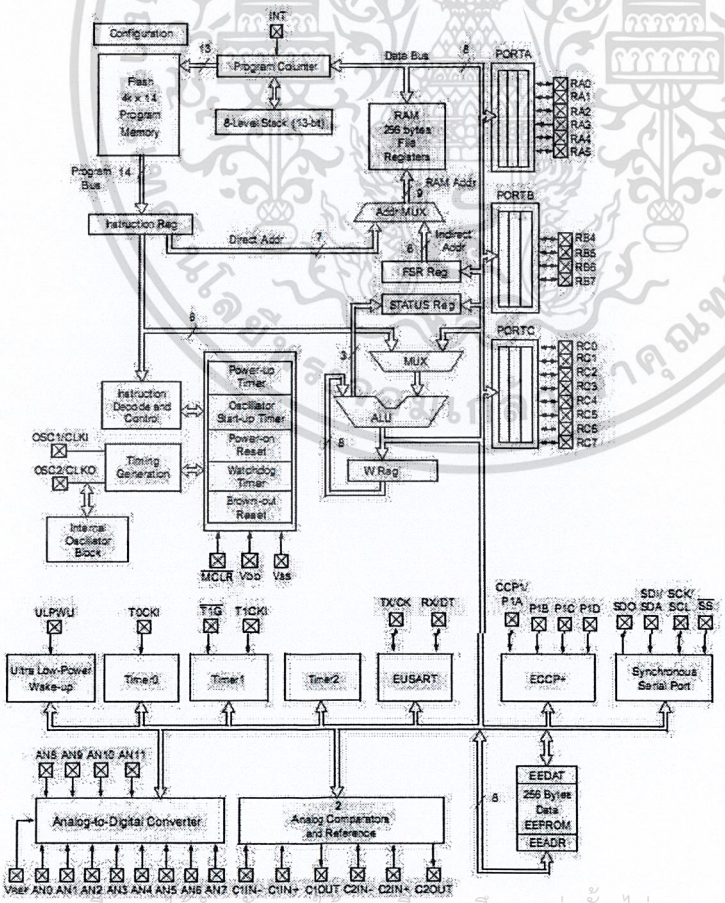
31. Peripherals : ADC , Comparator , PWM , Timer

32. Supply Voltage Max : 5.5V

33. Supply Voltage Min : 2V

34. Termination Type : Through Hole

3.3.4 โครงสร้างการทำงานภายในของ PIC16F690



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 โครงสร้างการทำงานภายใน PIC16F690

3.3.5 รายละเอียดของขา PIC16F690

ตารางที่ 3.2 รายละเอียดของขา PIC16F690

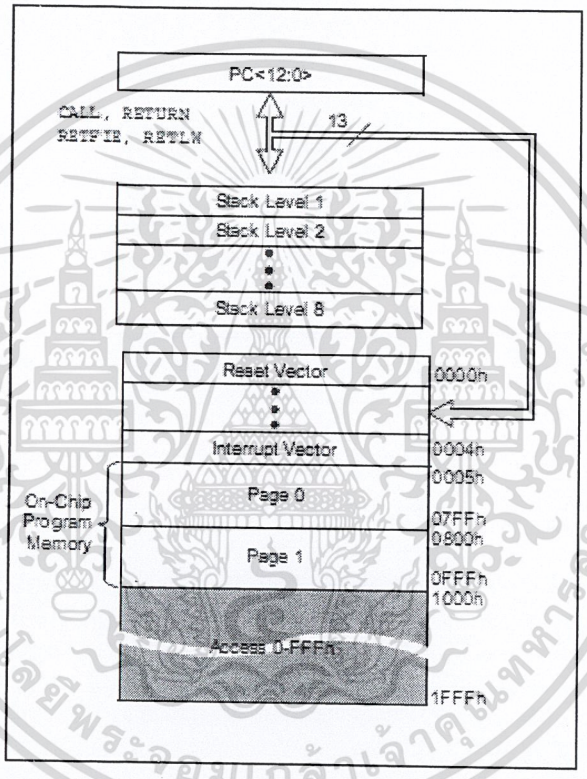
Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/ICSPDAT/ ULPWU	RA0	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN0	AN	—	A/D Channel 0 input.
	C1IN+	AN	—	Comparator C1 positive input.
	ICSPDAT	TTL	CMOS	ICSP™ Data I/O.
	ULPWU	AN	—	Ultra Low-Power Wake-up input.
RA1/AN1/C12IN0-/VREF/ICSPCLK	RA1	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN1	AN	—	A/D Channel 1 input.
	C12IN0-	AN	—	Comparator C1 or C2 negative input.
	VREF	AN	—	External Voltage Reference for A/D.
	ICSPCLK	ST	—	ICSP™ clock.
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN2	AN	—	A/D Channel 2 input.
	T0CKI	ST	—	Timer0 clock input.
	INT	ST	—	External interrupt.
	C1OUT	—	CMOS	Comparator C1 output.
RA3/$\overline{MCLR}$/VPP	RA3	TTL	—	General purpose input. Individually controlled interrupt-on-change.
	$\overline{MCLR}$	ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming voltage.
RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT	RA4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN3	AN	—	A/D Channel 3 input.
	$\overline{T1G}$	ST	—	Timer1 gate input.
	OSC2	—	XTAL	Crystal/Resonator.
	CLKOUT	—	CMOS	Fosc/4 output.
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	T1CKI	ST	—	Timer1 clock input.
	OSC1	XTAL	—	Crystal/Resonator.
	CLKIN	ST	—	External clock input/RC oscillator connection.
RB4/AN10/SDI/SDA	RB4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN10	AN	—	A/D Channel 10 input.
	SDI	ST	—	SPI data input.
	SDA	ST	OD	I^2C™ data input/output.
RB5/AN11/RX/DT	RB5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN11	AN	—	A/D Channel 11 input.
	RX	ST	—	EUSART asynchronous input.
	DT	ST	CMOS	EUSART synchronous data.

Legend: AN = Analog input or output; CMOS = CMOS compatible input or output; OD = Open Drain
TTL = TTL compatible input; ST = Schmitt Trigger input with CMOS levels
HV = High Voltage; XTAL = Crystal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.6 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F690

PIC16F690 จะมีขนาดของหน่วยความจำโปรแกรม ซึ่งสามารถอ้างอิงได้ถึง 8K byte โดย PIC16F690 จะมีขนาดหน่วยความจำเท่ากับ 8K x 14 bits ซึ่งตำแหน่ง Reset Vector จะอยู่ที่ 0000h และ Interrupt Vector จะอยู่ที่ 004h PIC จะแบ่งหน่วยความจำโปรแกรมออกเป็นหน้า ซึ่งแต่ละหน้าก็จะมีขนาด 2Kbytes ซึ่งคำสั่ง CALL และ GOTO สามารถสั่งให้ Program Counter กระโดดไปมาได้ในช่วงของ Page เท่านั้น แต่ถ้าเมื่อต้องการกระโดดจาก Page หนึ่งไปยังอีก Page หนึ่ง ดังนั้นจึงต้องไปควบคุม PCALTH



รูปที่ 3.11 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F690

การเลือกหน้า ของหน่วยโปรแกรม จะต้องเลือกในรีจิสเตอร์ PCALTH โดยการระบุตำแหน่งที่บิต 3 และบิต 4 เมื่อใช้คำสั่ง CALL ไปที่ Routine หนึ่งแล้ว แล้วจะใช้คำสั่ง RETURN ในการกลับไปที่ตำแหน่งเดิมการ RETURN กลับนั้นไม่จำเป็นต้องสั่ง PCALTH ให้ชี้ไปยังหน้า ก่อนหน้านี้ที่จะเรียก CALL เพราะค่า Address ดังกล่าวจะถูกเก็บไว้ใน STACK อยู่แล้วแต่สำหรับคำสั่ง GOTO เวลาข้ามหน้า จะต้องสั่งให้ PCALTH ชี้ไปยังหน้าที่ต้องการจะไปทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.7 การจัดสรรหน่วยความจำข้อมูล RAM ของ PIC16F690

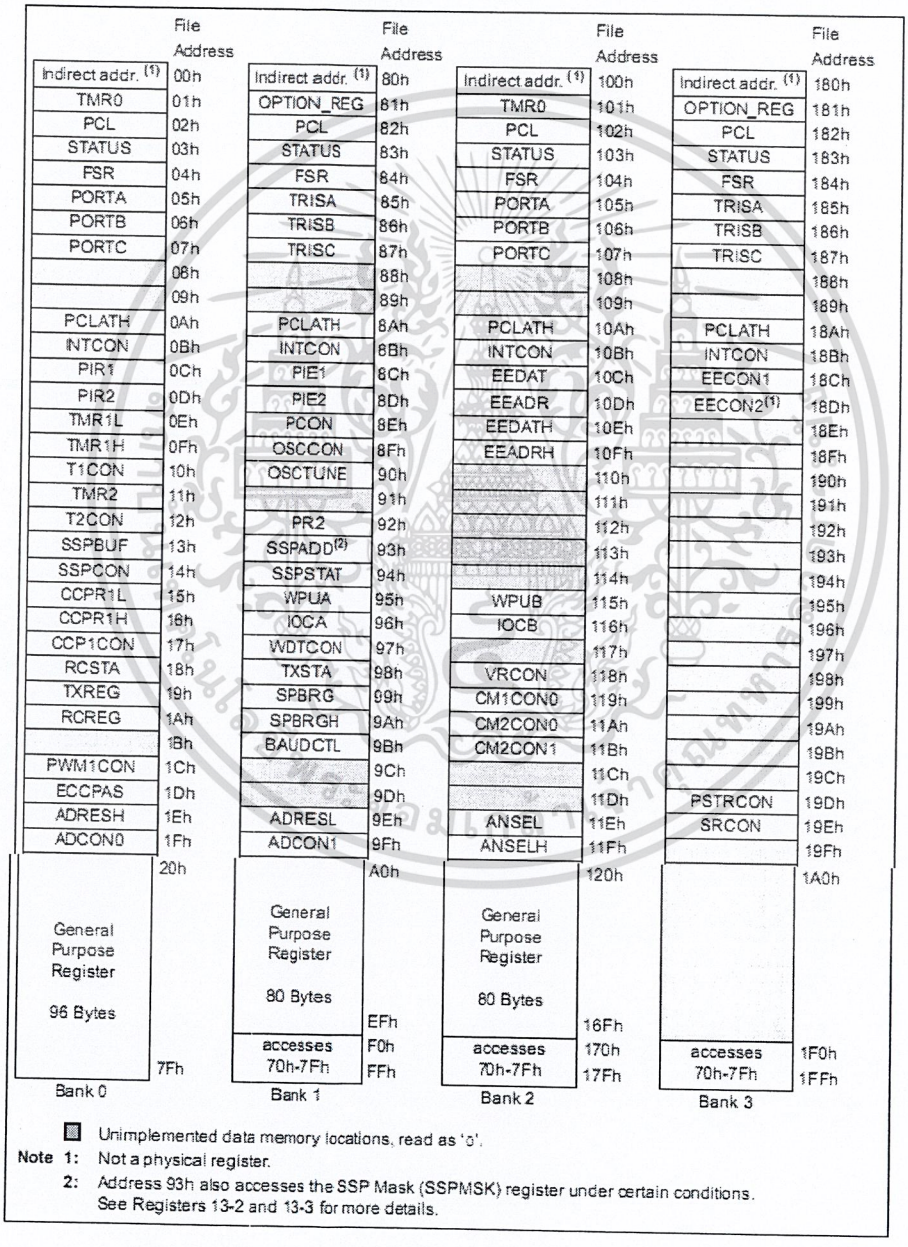
PIC16F690 มีหน่วยความจำข้อมูล RAM สำหรับใช้วานทั่วไป 368 bytes และมีรีจิสเตอร์ไฟล์ 8 bit 57 ตัว ในแต่ละ bank มีขนาดสูงสุด 128 bytes แต่มีการใช้งานได้จริงในแต่ละ bank ต่างกัน โดยในแต่ละ bank มีการจัดสรรพื้นที่ดังนี้

- Bank 0 มีช่วง Address 0x00 – 0x7F
- Address 0x00 – 0x1F เป็นพื้นที่ รีจิสเตอร์ไฟล์
- Address 0x20 – 0x7F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 96 bytes
- Bank 1 มีช่วง Address 0x80 – 0xFF
- Address 0x80 – 0x9F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบาง Address ไม่ใช้งาน
- Address 0xA0 – 0xEF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 bytes
- Address 0xF0 – 0xFF บรรจุข้อมูลเหมือนใน Address 0x70 – 0x7F ใน Bank 0 เพื่อช่วยให้สามารถใช้ข้อมูลจาก Address 0x70 – 0x7F ได้ง่ายขึ้นโดยไม่ต้องเปลี่ยน Bank
- Bank 2 มีช่วง Address 0x100 – 0x17F
- Address 0x100 – 0x10F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบาง Address ไม่ใช้งาน
- Address 0x110 – 0x11F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 bytes
- Address 0x120 – 0x16F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 bytes
- Address 0x170 – 0x17F บรรจุข้อมูลเหมือนใน Address 0x70 – 0x7F ใน Bank 0 เพื่อช่วยให้สามารถใช้ข้อมูลจาก Address 0x70 – 0x7F ได้ง่ายขึ้นโดยไม่ต้องเปลี่ยน Bank
- Bank 3 มีช่วง Address 0x180 – 0x1FF
- Address 0x180 – 0x18F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบาง Address ไม่ใช้งาน
- Address 0x190 – 0x19F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 bytes
- Address 0x1A0 – 0x1EF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 bytes
- Address 0x1F0 – 0x1FF บรรจุข้อมูลเหมือนใน Address 0x70 – 0x7F ใน Bank 0 เพื่อช่วยให้สามารถใช้ข้อมูลจาก Address 0x70 – 0x7F ได้ง่ายขึ้นโดยไม่ต้องเปลี่ยน Bank การเลือก Bank ของหน่วยความจำข้อมูล RAM จะต้องเลือกในรีจิสเตอร์ STATUS โดยระบุตำแหน่งที่บิต 5 และบิตที่ 6 (RP0 , RP1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ค่า RP0 และค่า RP1 ในการเลือก Bank ของ PIC16F690

RP1 : RP0	Bank
00	0
01	1
10	2
11	3



รูปที่ 3.12 การจัดการหน่วยความจำข้อมูล RAM และตำแหน่งรีจิสเตอร์ ของ PIC16F690

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.8 การทำงานในโหมด PWM

โมดูล CCPx จะทำสัญญาณการมอดูเลตความกว้างพัลส์(Pulse Width Modulation : PWM) ความละเอียด 10 บิตสัญญาณ PWM ที่สร้างขึ้นจะส่งออกทางขา RC2/CCP1หรือRC1/T1OSI/CCP2

3.3.8.1 หลักการสร้างสัญญาณ PWM

1. กำหนดค่าให้แก่รีจิสเตอร์ PR2 เพื่อกำหนดค่าเวลาของสัญญาณ PWM
2. กำหนดค่าดิวตี้ไซเกิล โดยเขียนข้อมูลลงในรีจิสเตอร์ CCPRxL ร่วมกับบิต 5-4 ของรีจิสเตอร์ CCPxCON
3. กำหนดให้ขาพอร์ท RC2/CCP1 หรือ RC1/T10SI/CCP2 เป็นเอาท์พุตเพื่อเป็น ทางออกของสัญญาณ PWM โดยการเคลียบิตที่ 2 หรือ 1 ของรีจิสเตอร์ TRISC
4. กำหนดค่าปรีสเกลเลอร์ของ TMR2 และเอนเนเบิลการทำงานของไทเมอร์ 2
5. กำหนดให้โมดูล CCPx ทำงานให้โหมด PWM

ทันทีที่โมดูล CCPx เริ่มทำงานค่าของ TMR2 จะเพิ่มขึ้นจนเท่ากับ PR2 ที่ขอพอร์ท RC2/CCP1 หรือ RC1/T1OSI/CCP2 จะเกิดลอจิก “1” และคงสถานะอยู่เช่นนั้นแล้วค่าของ TMR2 จะเคลียร์และเริ่มนับเพิ่มขึ้นหลังจากค่าดิวตี้ไซเกิล (Duty cycle) ที่กำหนดไว้ในรีจิสเตอร์ CCPRxL และ 2 บิตใน CCPxCON จะถูกถ่ายทอดไปยัง CCPxH และ 2 บิตในหน่วยความจำพิเศษเพื่อเปรียบเทียบกับค่าใน TMR2 และค่าปรีสเกลเลอร์ 2 บิตซึ่งเพิ่มค่าขึ้นเรื่อยๆจนกระทั้งเมื่อข้อมูลทั้งสองกลุ่มเท่ากันจะส่งสัญญาณไปทำให้ขาพอร์ต RC2/CCP1 หรือ RC1/T1OSI/CCP2 กลับมาเป็นลอจิก “0” และคงสถานะอยู่เช่นนั้นจนกระทั้งค่าของ TMR2 เท่ากับ PR2 อีกครั้งก็จะเกิดสัญญาณลอจิก “1” เป็นการเริ่มต้นรอบใหม่ของสัญญาณ และจะทำงานวนเช่นนี้จนกระทั่งมีการสั่งหยุดทำงาน (Disable)

ค่าดิวตี้ไซเกิลของสัญญาณ PWM ที่สร้างขึ้นนี้สามารถเปลี่ยนแปลงได้ตลอกเวลา จากการเปลี่ยนค่าที่ CCPRxL และ 2 บิต ใน CCPxCON แต่ค่าของคาบเวลาหรือความถี่จะเปลี่ยนแปลงไม่ได้จนกว่าจะหยุดการทำงาน แล้วกำหนดค่าของคาบเวลาลงในรีจิสเตอร์ PR2 ใหม่ ดังนั้นสัญญาณ PWM ที่สร้างขึ้นจึงสามารถกำหนดค่าดิวตี้ไซเกิลได้ตามความต้องการ

3.3.8.2 การกำหนดคาบเวลาของสัญญาณ PWM

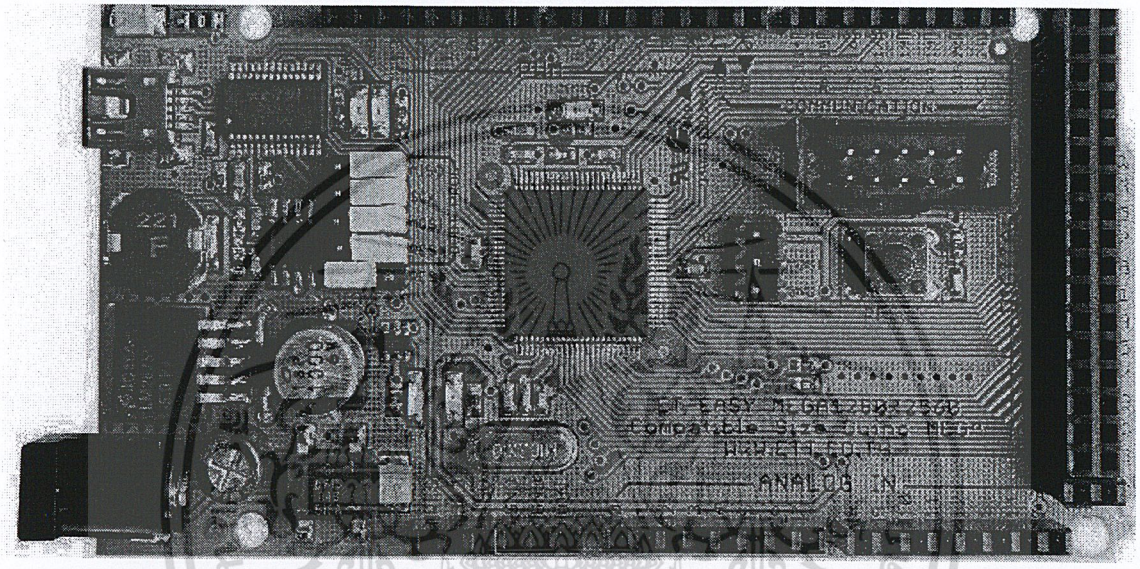
ทำได้โดยการเขียนข้อมูลลงในรีจิสเตอร์ PR2 (รีจิสเตอร์คาบเวลา) ของไทเมอร์ แล้วนำ ค่าของ PR2 มาคำนวณหาค่าคาบเวลาของสัญญาณ PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คาบเวลาของสัญญาณ PMW = (ค่าในรีจิสเตอร์ PR + 1) x 4 x T_{osc} x ค่าปรีสเกลเลอร์ของ TMR2 โดยที่ T_{osc} คือ คาบของสัญญาณนาฬิกาหลักมีหน่วยเป็นวินาที ค่ารีจีสเตอร์ทั้งหมด คำนวณในรูปของเลขฐานสิบ

ความถี่ของสัญญาณ PWM = 1/คาบเวลาของสัญญาณ PWM

3. 4 ไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 (ARDIUNO MEGA)



รูปที่ 3.13 บอร์ด ET-EASY MEGA1280 (ARDIUNO MEGA)

3.4.1 คุณสมบัติของ ET-EASY MEGA1280 (ARDIUNO MEGA)

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ประสิทธิภาพสูงแต่ใช้พลังงานต่ำ ในตระกูล AVR®
- สถาปัตยกรรมแบบ RISC
  - มีชุดคำสั่ง 135 คำสั่ง และส่วนใหญ่คำสั่งเหล่านี้จะใช้เพียง 1 สัญญาณนาฬิกาในการประมวลผลคำสั่ง
  - มีเรจิสเตอร์สาหรับใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว
- หน่วยความจำ
  - หน่วยความจำแฟลชสาหรับโปรแกรมขนาด 128 กิโลไบต์ เขียน/ลบได้ 10,000 ครั้ง
  - หน่วยความจำแบบ EEPROM ขนาด 4 กิโลไบต์ เขียน/ลบได้ 100,000 ครั้ง
  - หน่วยความจำแรมชนิดเอสแรม (SRAM) ขนาด 8 กิโลไบต์
- มีระบบโปรแกรมตัวเองอยู่ในตัวชิพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• สามารถทำการอ่านขณะเขียนได้จริง โดยสามารถล็อกการทำงานได้เพื่อความปลอดภัยของซอฟต์แวร์

• มีการเชื่อมประสานกับ JTAG (IEEE std. 1149.1 compliant)

• คุณสมบัติการเชื่อมต่อกับอุปกรณ์ภายนอก

– มีตัวตั้งเวลาและตัวนับขนาด 8 บิต จำนวน 2 ตัว ที่สามารถแยกโหมดการทำงานจากกันได้ 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 จำนวน 12 ช่องสัญญาณที่สามารถกำหนดความละเอียดได้ 16 บิต

– มีตัวปรับผลการเปรียบเทียบของเอาต์พุต

– มีตัวแปลงสัญญาณอนาล็อกให้เป็นดิจิตอลขนาด 10 บิต จำนวน 16 ช่องสัญญาณ

– มีพอร์ตสื่อสารอนุกรมที่สามารถกำหนดอัตราการรับ/ส่งได้จำนวน 4 พอร์ต

– เชื่อมประสานอนุกรมแบบ SPI ได้ทั้งการเป็นมาสเตอร์และสเลฟ (Master/Slave)

– มีการเชื่อมประสานแบบอนุกรมด้วยสายสัญญาณ 2 เส้นแบบ ส่งข้อมูลแบบเรียงไบต์ (Byte Oriented)

– มีตัวตั้งเวลาแบบวอตช์ด็อกที่สามารถกาหนดการทำงานได้โดยสามารถแยกสัญญาณนาฬิกาได้ จากตัวชิพ

– มีตัวเปรียบเทียบสัญญาณแบบแอนาล็อกอยู่ในตัว

– มีการรองรับการขัดจังหวะและการเวก-อัพ (Wake-up) เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับขาของชิพ

• คุณสมบัติพิเศษ

– มีระบบเริ่มระบบเมื่อมีการรีเซ็ตและมีระบบตรวจจับการเกิดบราวน์เอาท์ (Brown-out) ที่สามารถ กำหนดการทำงานได้

– มีตัวตรวจหาความเที่ยงตรงของออสซิเลเตอร์อยู่ในตัว (Internal Calibrated Oscillator)

– มีแหล่งการขัดจังหวะทั้งภายในและภายนอก (External and Internal Interrupt Sources)

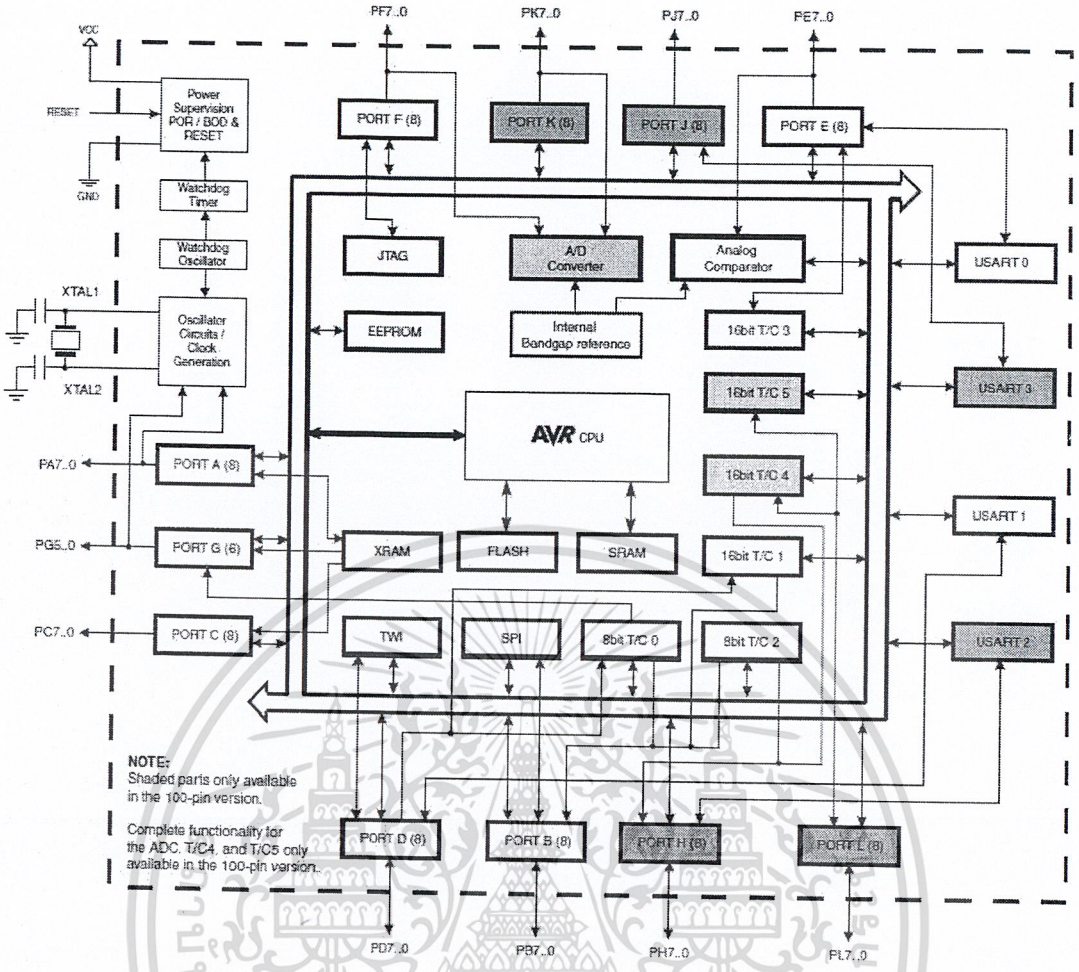
– มีโหนดการทำงานสลีป 6 แบบ คือ Idle, ADC Noise Reduction, Power-save, Power-down, Standby, และ Extended Standby

• การใช้พลังงาน

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



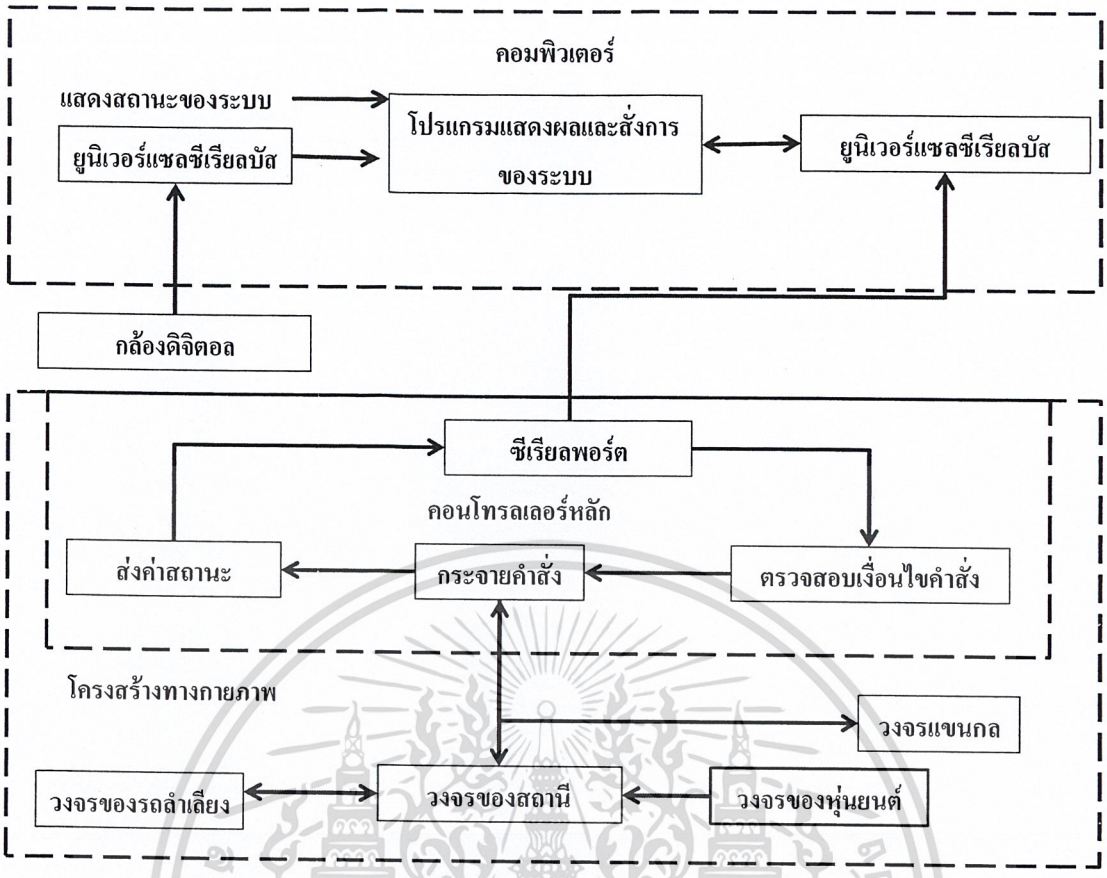
รูปที่ 3.14 สถาปัตยกรรมภายในของ AVR

3.5 การออกแบบ

การออกแบบระบบจำลองกระบวนการผลิตอัตโนมัติ ในเบื้องต้นนั้นต้องเข้าใจหลักการทำงานในภาพรวมของระบบกระบวนการก่อนว่ามีหลักการทำงานอย่างไร แล้วจึงพิจารณาส่วนย่อยต่างๆ อันประกอบไปด้วย มอนิเตอร์แสดงผลและสั่งการ ส่วนลำเลียง แขนกล หุ่นยนต์ 2 และหุ่นยนต์แมลง 6 ขา

หลักการทำงานในภาพรวมของระบบจำลองกระบวนการผลิตอัตโนมัติ สามารถแสดงดังแผนผังในรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แผนผังรวมของระบบจำลองกระบวนการผลิตอัตโนมัติ

จากรูปที่ 3.15 เห็นว่าระบบจำลองการผลิตอัตโนมัติประกอบด้วยส่วนหลักๆ ดังนี้คือ

- กล้องดิจิตอลวีดีโอ
- คอมพิวเตอร์
- โครสร้างทางกายภาพของระบบ
- วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง

3.5.1 กล้องดิจิตอลวีดีโอ

กล้องดิจิตอลวีดีโอเป็นอุปกรณ์ที่สำคัญสำหรับโครงงานนี้ เนื่องจากเป็นอุปกรณ์ที่ใช้สำหรับตรวจสอบสถานะของรถในระบบว่าตรงกับการแสดงผลของข้อมูล

สิ่งจำเป็นสำหรับการใช้งานกล้องดิจิตอลวีดีโอ

- เครื่องไมโครคอมพิวเตอร์
- โปรแกรมระบบปฏิบัติการไมโครซอฟท์วินโดว์
- พอร์ตยูนิเวอร์แซลซีเรียลบัส (Universal serial bus)
- โปรแกรมวิชชวลเบสิก (Visual Basic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• ต้องการไฟเลี้ยงจากคอมพิวเตอร์ 2 วัตต์

• ปรับโฟกัสได้ตั้งแต่ 1 นิ้ว ถึงอนันต์

3.5.2 คอมพิวเตอร์

คอมพิวเตอร์เป็นอุปกรณ์ที่ทำหน้าที่รับข้อมูลจากกล้องดิจิตอลวีดีโอผ่านทางยูนิเวอร์ แซลซี เรียลบัสเพื่อแสดงภาพที่ได้จากหน้างานจริงแล้วนำมาเปรียบเทียบกับข้อมูลที่ไดรับมาจากคอนโทรลเลอร์ที่ทำหน้าที่เก็บรวบรวมข้อมูล

นอกจากคอมพิวเตอร์ยังทำหน้าที่แสดงภาพจากกล้องดิจิตอลแล้ว ยังทำการแสดงของมูลของระบบจำลองที่คอนโทรลเลอร์หลักส่งค่าข้อมูลกับมาด้วย และยังทำหน้าที่สั่งการผลข้อมูลที่ได้กระบวนการทั้งหมดด้วยโดยให้คอนโทรลเลอร์หลักเป็นตัวกระจายคำสั่งที่ได้รับมาไปยังแต่ละหน่วยการทำงาน ทั้งสถานีและแขนกล

เนื่องจากการควบคุมในโครงงานนี้ใช้คอมพิวเตอร์เป็นอุปกรณ์ประมวลผลหลักดังนั้นการเขียนโปรแกรมจึงเป็นการดำเนินการบนระบบเวลาดีสครีต แทนการควบคุมแบบระบบเวลาต่อเนื่อง ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องมีความรัดกุม และทำงานได้รวดเร็ว ซึ่งคอมพิวเตอร์ที่ใช้จะต้องมีความเร็วพอที่จะทำให้คาบการสุ่มสัญญาณมีค่าน้อยเพียงพอ จนไม่มีผลต่อการควบคุมระบบทั้งหมดด้วย สำหรับโครงงานนี้เลือกใช้การเขียนโปรแกรมโดยภาษาวิชชวลซีเบสิก

3.6 การศึกษาค้นคว้า

สำหรับการศึกษาค้นคว้าเพื่อที่จะจำลองและสร้างระบบลำเลียงที่ใช้กันในโรงงานอุตสาหกรรมนั้นทางคณะผู้จัดทำได้ทำการศึกษาทั้งจากการวิจัยหรือศึกษาจากเอกสารต่างๆที่เกี่ยวข้องกับทุกๆส่วนประกอบในตัวระบบอีกทั้งยังได้ศึกษาจากการจำลองและการทำงานจริงควบคู่ไปเพื่อให้ได้มาซึ่งระบบที่มีเสถียรภาพมากที่สุดโดยการศึกษาค้นคว้ามีรายละเอียดดังนี้

3.6.1 การวิจัยหรือศึกษาจากเอกสารต่างๆที่เกี่ยวข้อง

ในการดำเนินการในส่วนนี้ได้มีการค้นคว้าศึกษาทั้งในหนังสือ เอกสารต่างๆ และยังรวมถึงการค้นคว้าตามเว็บไซต์ศึกษาในด้านหลักการทำงานของส่วนประกอบทั้งหมดของคอนโทรลเลอร์ และอุปกรณ์เซนเซอร์ ที่เราใช้ในระบบ คือ Et Easy Mega1280 , PIC16F690 , TSOP34836 , DC Motor, Infrared Sensorโดยที่เราได้ศึกษาเกี่ยวกับรูปแบบของการเขียนโปรแกรมเพื่อที่จะนำมาใช้ในการควบคุมส่วนต่างๆของระบบและในส่วนของวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้องกับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 การวิจัยทดลอง

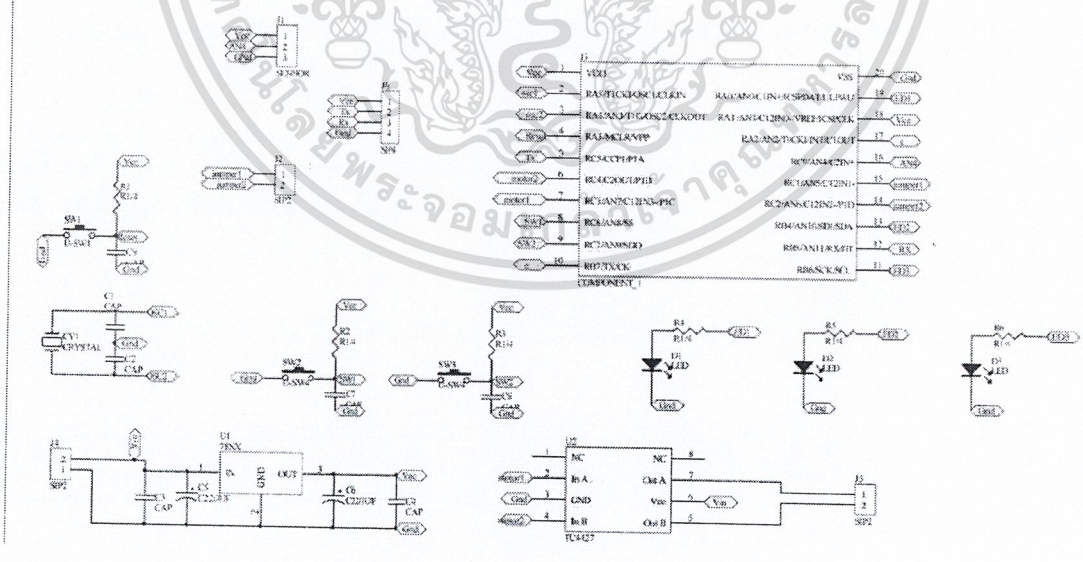
สำหรับการศึกษาดำเนินงานในส่วนนี้นั้นเป็นการเป็นการศึกษาควบคู่กับการทดลองปฏิบัติจริง ซึ่งการทดลองนั้นอาจมีความคลาดเคลื่อนดังนั้นจึงได้ทำการศึกษาในส่วนทฤษฎีควบคู่ไปกับส่วนปฏิบัติ เพื่อที่จะได้ลดอัตราการเกิดความผิดพลาดให้ได้มากที่สุด

3.7 หลักการทำงานของระบบ

สำหรับระบบลำเลียงนั้นเราได้แบ่งส่วนประกอบต่างๆออกเป็นส่วนย่อยๆได้3ส่วน ในแต่ละส่วนจะมี หลักการทำงาน และการสื่อสารที่แตกต่างกันโดยเราสามารถแบ่งหลักการทำงานต่างๆของระบบลำเลียง ออกได้เป็น3ส่วน ดังนี้

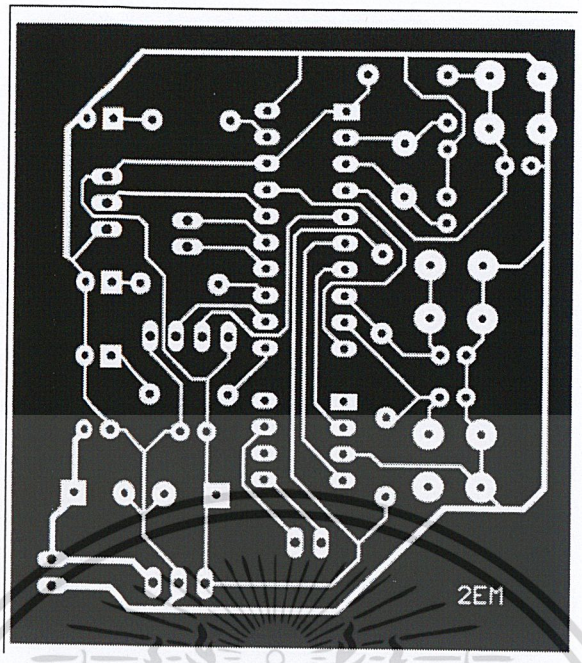
3.7.1 หลักการการทำงาน ส่วนรถลำเลียง

หลักการทำงานของส่วน รถลำเลียงนั้น เราใช้หลักการการส่งข้อมูลผ่านทางสัญญาณ Infrared โดยการทำงานจะเริ่มเมื่อ ตัวรถ นั้นเคลื่อนที่เข้ามาจอดที่ สถานีจากนั้น วงจรรับ-ส่ง จะเริ่มทำงาน โดยสิ่งที่ส่งไปจะเป็นหมายเลขประจำของตัวรถซึ่งส่วนที่ทำหน้าที่เป็นตัวรับสัญญาณที่ส่งไปคือส่วนสถานีนั่นเองโดยวงจรภายในส่วนของรถลำเลียงเป็นดังรูปที่3.16และลายวงจรของส่วน Vehicle เป็นดังรูปที่ 3.17



รูปที่ 3.16 วงจรภายในส่วนของรถลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

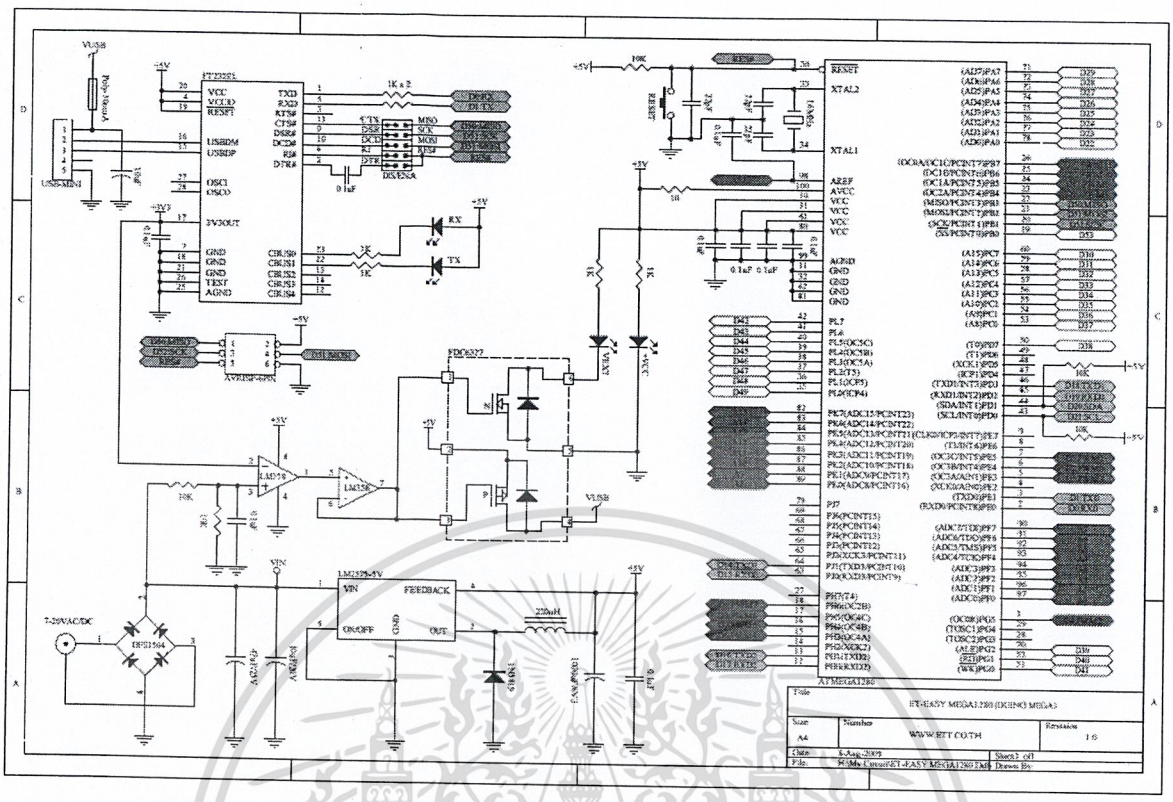


รูปที่3.17 วงจรภายในส่วนของVehicle

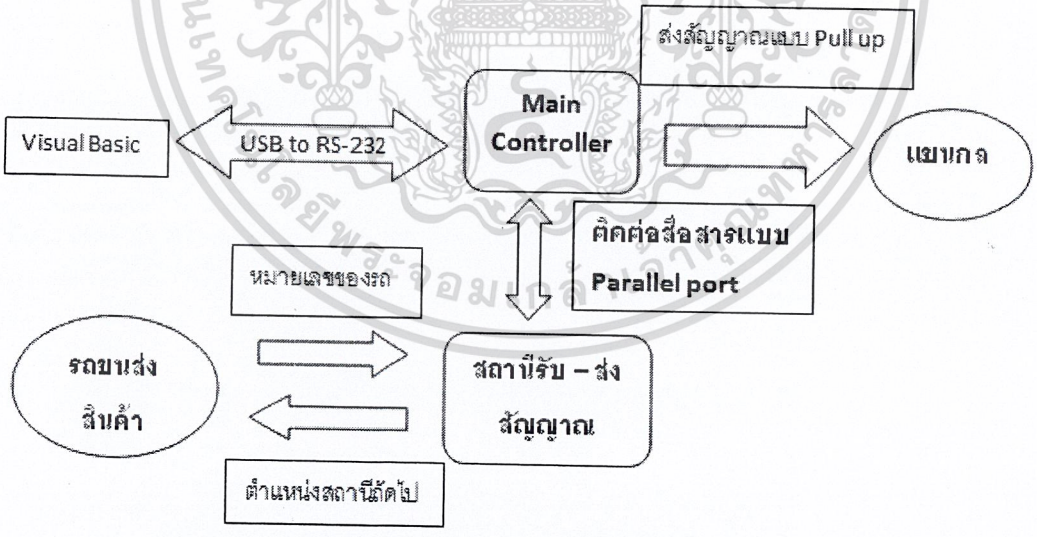
3.7.2 หลักการทำงาน ส่วนตัวควบคุมหลัก (Main Controller)

Et-easymega128 นั้นนอกจากจะทำหน้าที่เป็นเหมือนกลไกหลักในการตัดสินใจในการกระทำต่างๆของระบบแล้วตัว Controller ยังทำหน้าที่เป็นตัวเชื่อมสัญญาณต่างๆของระบบเข้าด้วยกันอีกด้วยเราสามารถยกตัวอย่างการทำงานเช่นเมื่อผู้ใช้งานต้องการตรวจสอบหมายเลขของรถที่เข้ามาจอดต่ละสถานีการทำงานจะเริ่มเมื่อผู้ใช้กดคำสั่งตรวจสอบหมายเลขรถสถานีแต่ละสถานีจะส่งสัญญาณมาทาง USB to Serial Port เพื่อแปลงสัญญาณจาก RS-232 เข้า USB Port จากนั้นบอร์ดจะติดต่อสื่อสารเข้าไปที่สถานีโดยส่งสัญญาณผ่านทาง Parallel Port เพื่อให้สถานีส่งหมายเลขรถที่จอดอยู่กลับมาเมื่อตัว Controller ได้หมายเลขรถมาแล้วจะทำการส่งค่ากลับไปหาผู้ใช้งานเพื่อให้ผู้ใช้ทราบถึงตำแหน่งรถแต่ละคันและเมื่อมีคำสั่งมาจากผู้ใช้ว่าต้องการให้รถลำเลียงเคลื่อนที่ไปยังสถานีปลายทางไหนตัว Controller จะรับสัญญาณจากการที่ผู้ใช้กดปุ่มสั่งการจากนั้นจะทำการส่งสัญญาณ ไปสั่งที่สถานีเพื่อให้สถานีส่งสัญญาณไปสั่งที่ตัวรถอีกทีหนึ่งให้รถเคลื่อนที่ไปสถานีปลายทางตามที่ได้รับสัญญาณที่ผู้ใช้สั่งมานอกจากนั้นตัว Controller เรายังทำหน้าที่ติดต่อสื่อสารกับตัว Controller ของส่วนแขนกลเพื่อควบคุมการทำงานของแขนกลเมื่อรถเคลื่อนที่เข้ามาจอดอีกด้วยโดยวงจรภายในตัวบอร์ดเป็นดังรูปที่ 3.18 และ ภาพแสดงหลักการทำงานของส่วน MainController เป็นดังรูป ที่ 3.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 วงจรภายในตัวบอร์ด Et-easy mega128

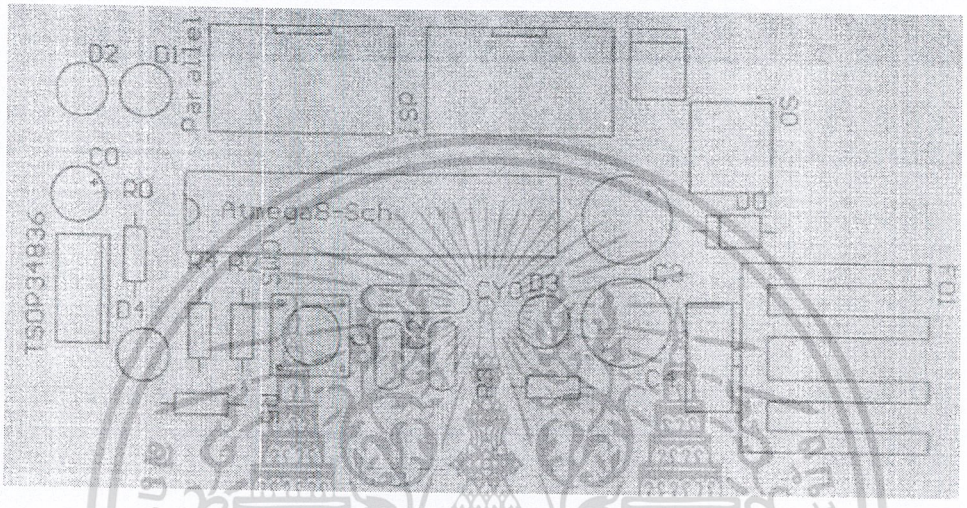


รูปที่ 3.19 แผนภาพแสดงการทำงานของส่วน Main Controller

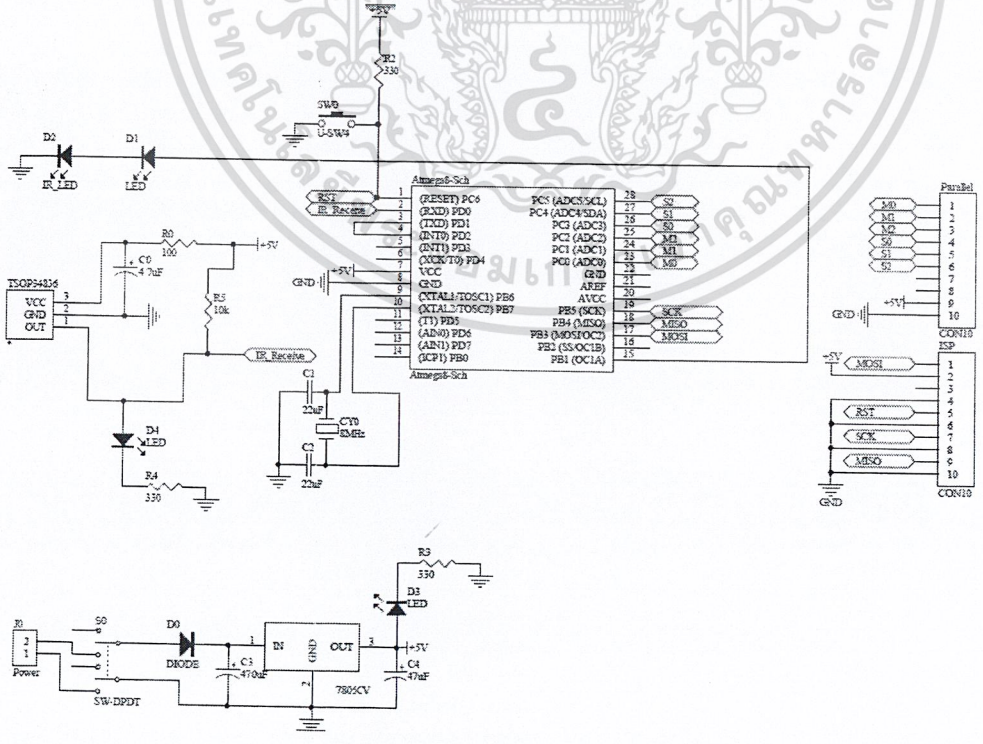
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.3 หลักการทำงานส่วนสถานี

หลักการทำงานของส่วนสถานีนั้น เราใช้หลักการการส่งข้อมูลผ่านทางสัญญาณ Infrared เช่นเดียวกับหลักการของส่วนรถลำเลียงโดยการทำงานจะเริ่มเมื่อรถเคลื่อนที่เข้ามาจอดและส่วนของสถานี ได้รับสัญญาณ Infrared ที่เป็นหมายเลขของรถแล้วจากนั้นสถานีจะทำการส่งหมายเลขของรถที่ได้รับมานั้นไปยังส่วน Main Controller เพื่อทำการประมวลผลและตัดสินใจโดยภาพรวมของวงจรส่วนสถานี เป็นดังรูปที่3.20และ วงจรภายในส่วนของสถานี เป็นดังรูปที่ 3.21



รูปที่ 3.20 ภาพรวมของวงจรส่วนสถานี



รูปที่ 3.21 วงจรภายในส่วนของสถานี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

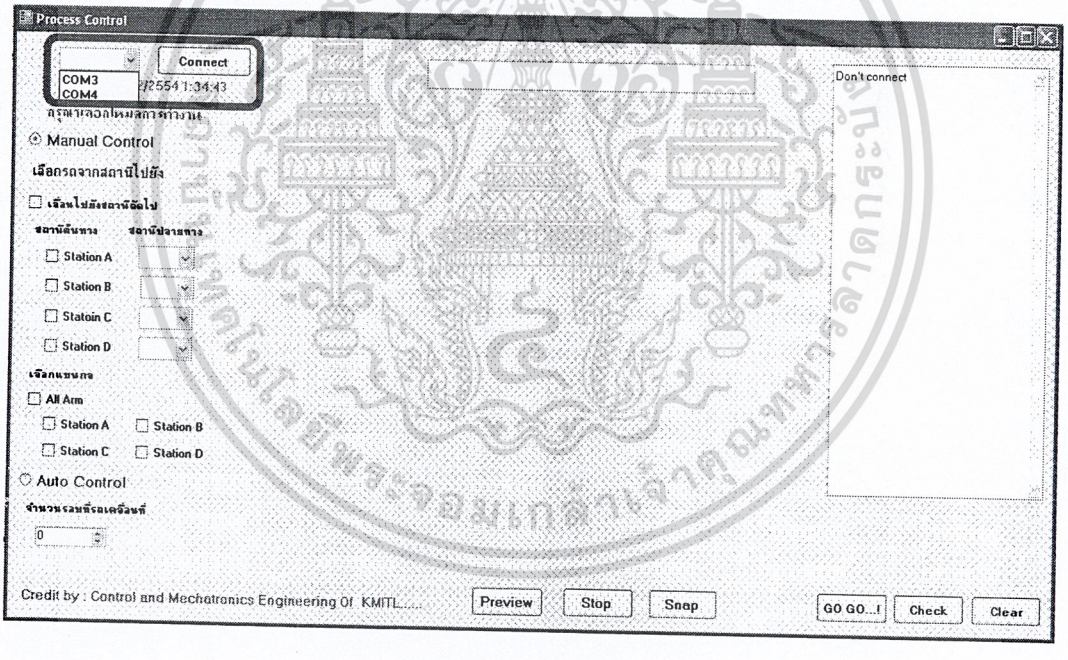
การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึง การทดลองโปรแกรมและผลที่ได้จากการทำงานการเคลื่อนที่ไปของตัวรถโดยการรับ-ส่งคำสั่งจากคอมพิวเตอร์โดยมีรายละเอียดของการทดลองดังนี้

4.1 การทดลองโปรแกรม และผลที่ได้จากการทำงาน

ในส่วนนี้ เป็นการศึกษาเฉพาะส่วนโปรแกรม Visual basic 2008 และ ผลที่ได้จากการสั่งจากโปรแกรม ซึ่งผลการทดลองมีดังนี้

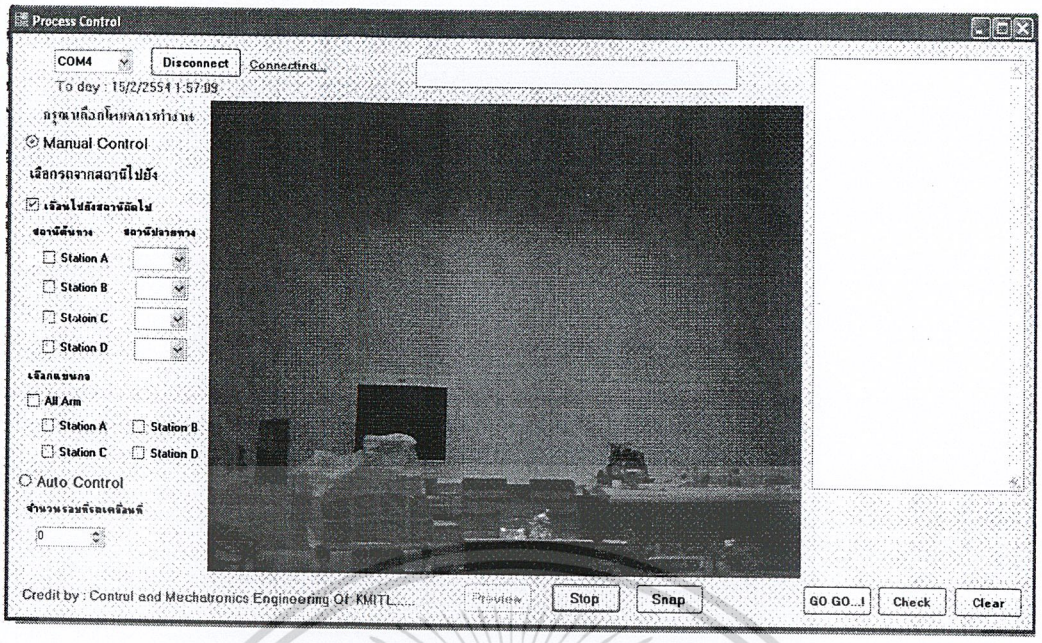
หลังจากทำการรันโปรแกรมจาก Visual basic 2008



รูปที่ 4.1 การ com port และการ เชื่อมต่อ

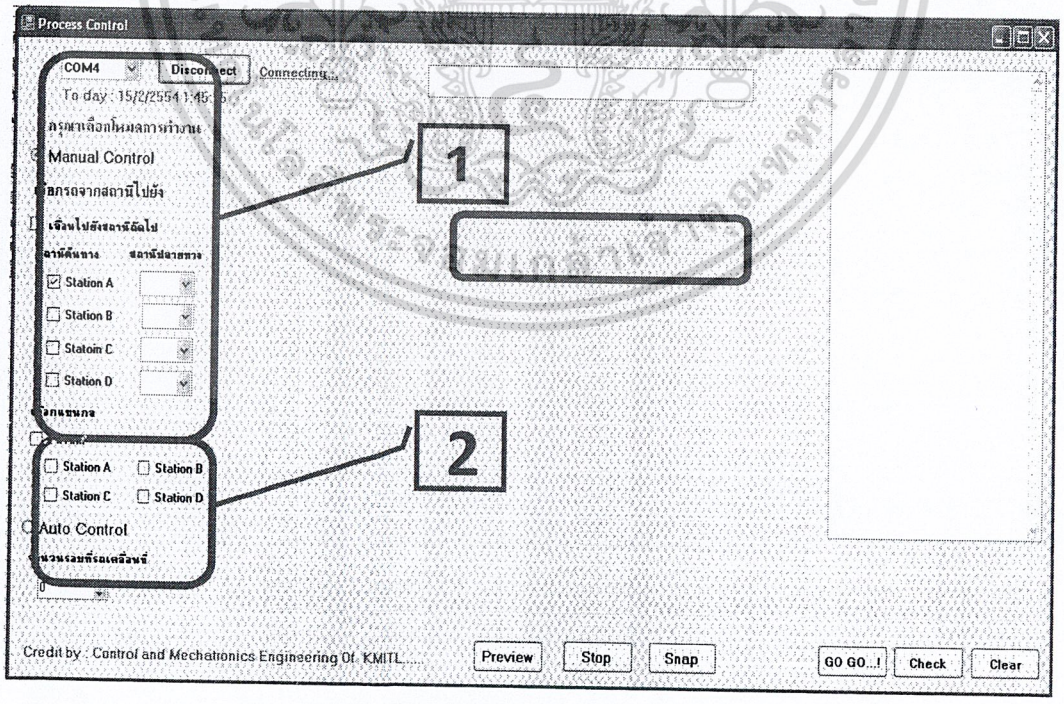
ทำการเลือก com port และกดปุ่ม Connect เพื่อทำการเชื่อมต่อระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การจับภาพผ่าน webcam

เราสามารถดูการทำงานจริงของระบบโดยผ่านทางกล้อง webcam โดยกดปุ่ม Preview และ เราสามารถทำการ save รูปภาพในขณะนั้นได้โดยการกดปุ่ม Snap และทำการปิดกล้องได้โดยการกดปุ่ม Stop



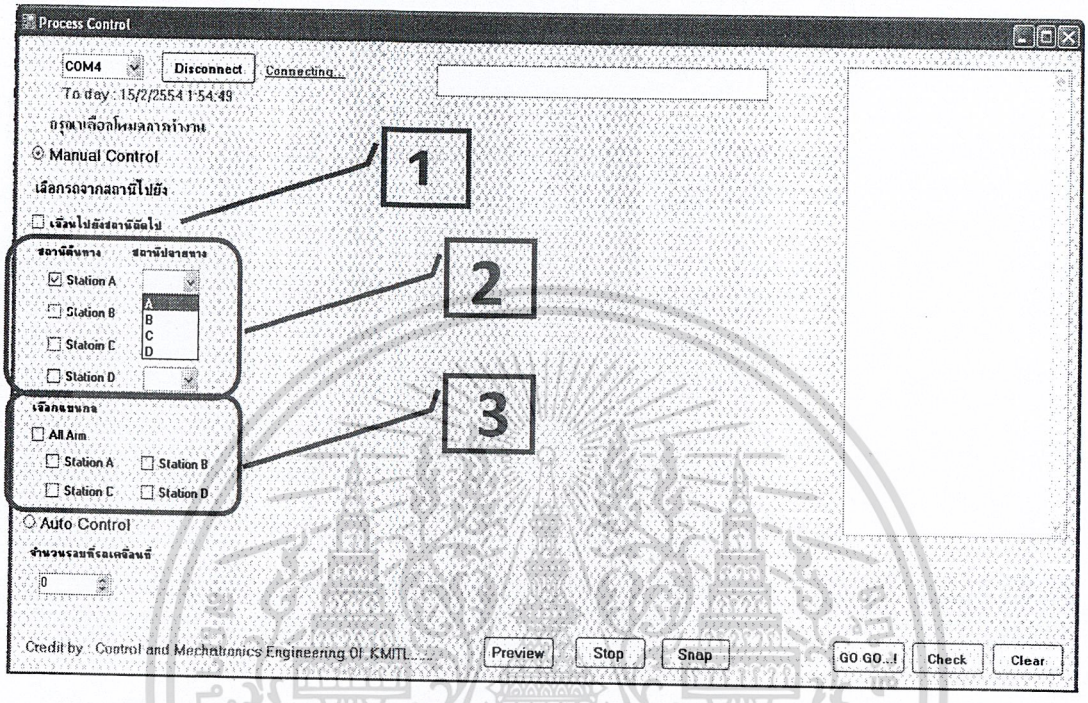
รูปที่ 4.3 โหมดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราเข้ามายังตัวโปรแกรมแล้วเราสามารถเลือกการทำงานของระบบลำเลียงได้ 2 แบบ คือ

1.แบบ Manual Control

2.แบบAuto Control



รูปที่ 4.4 รายละเอียดในโหมด Manual Comtrol

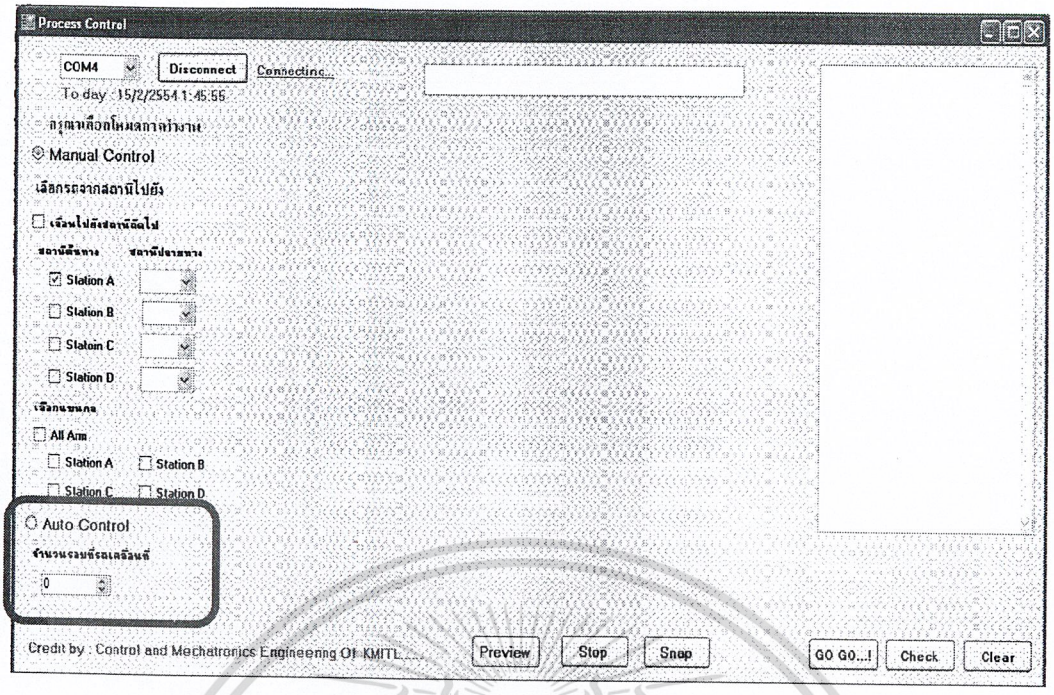
ใน Mode Manual Control นั้น เราอาจแบบเป็น 3 ส่วนหลักๆ คือ

1 เป็นการสั่งการให้รถเคลื่อนที่ไปยังสถานีถัดไป โดยไม่มีการวิ่งข้ามสถานี

2 เป็นการสั่งการให้รถเคลื่อนที่ไปยังสถานีที่ต้องการได้

3 เป็นการสั่งการให้แขนกลในสถานีต่างให้ทำงาน

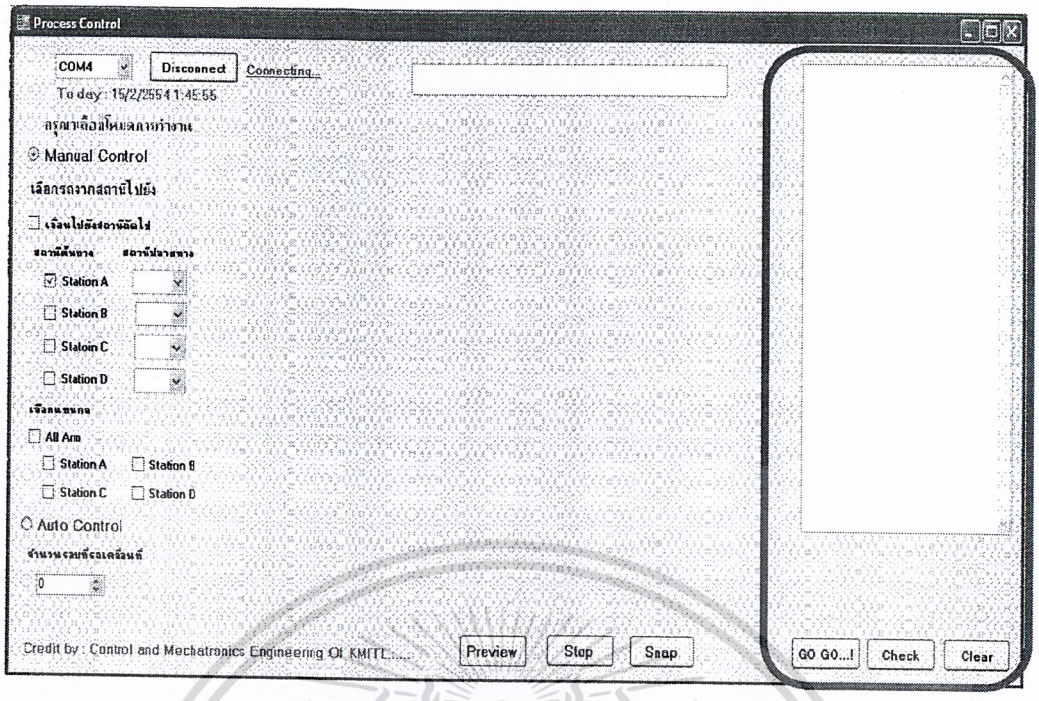
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การเลือก Mode Auto Control

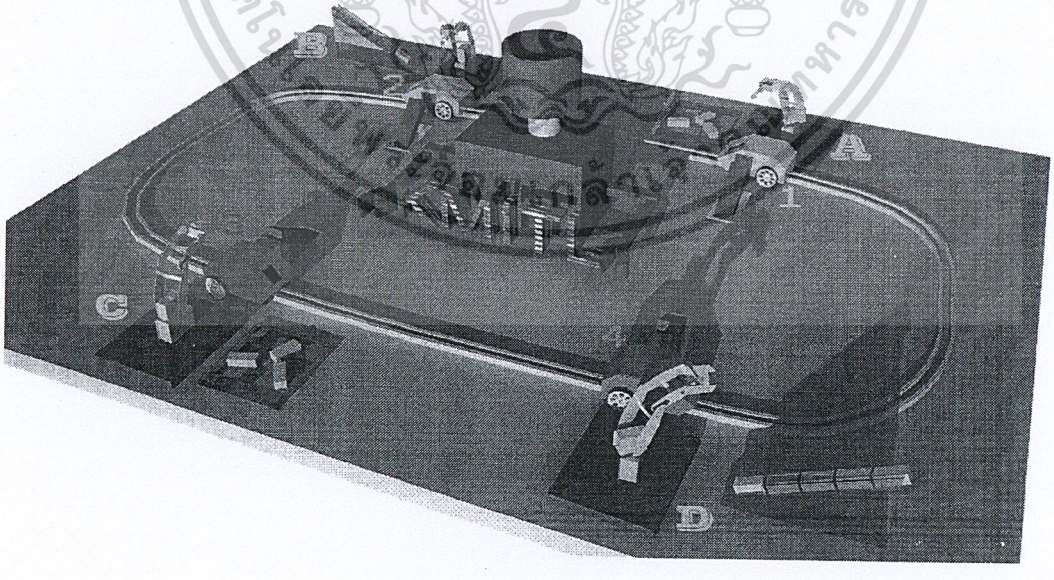
และใน Mode Auto Control นั้น เราสามารถกำหนดจำนวนรอบการทำงานได้ ว่าต้องการให้ระบบของเราทำงานได้กี่รอบ โดยเมื่อรถเคลื่อนที่มาถึงยังสถานี แขนกลจะทำการหยิบจับสิ่งของเองโดยอัตโนมัติ ไม่จำเป็นต้องรอคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 สถานีการทำงาน

และเราสามารถตรวจสอบสถานะการทำงาน ทั้งระบบได้ด้วยการกดปุ่ม Check ไม่ว่าจะเป็นการตรวจสอบว่าสถานีไหนมีรถคันใดจอดอยู่บ้าง และเราสามารถทำการลบข้อมูลได้โดยการกดปุ่ม Clear



รูปที่ 4.7 ภาพรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองการเคลื่อนที่ของรถ

ในส่วนนี้เป็นการศึกษาการเคลื่อนที่ของรถ โดยเริ่มจากการป้อนคำสั่งจากคอมพิวเตอร์ที่เขียนโปรแกรมควบคุมเข้าไปในตัวไมโครคอนโทรลเลอร์ ตัวไมโครคอนโทรลเลอร์ส่งสัญญาณไปในตัวสถานีเพื่อให้สถานีส่งสัญญาณสั่งตัวรถซึ่งเราจะสามารถกำหนดว่ารถที่อยู่ในสถานีใดๆ สามารถเคลื่อนที่ไปยังอีกสถานีได้ เพื่อนำมาปรับปรุงแก้ไขให้เป็นไปตามที่เขียนโปรแกรมควบคุมได้

ตารางที่ 4.1 ตำแหน่งการเคลื่อนที่ของรถ

คำสั่ง	ทิศการเคลื่อนที่			
A > B	A > B	B > C	C > D	D > A
A > C	A > C	B > D	C > A	D > B
A > D	A > D	B > A	C > B	D > C
A > A	A > A	B > B	C > C	D > D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุป

จากการทดลองเราสามารถสั่งการระบบลำเลียงทั้งหมดให้สามารถทำงานตามที่เราต้องการได้จากหน้าจอMonitorโดยการเชื่อมต่อส่วนController เข้ากับตัวโปรแกรม Visual Basic โดยที่เราสามารถกำหนดให้ระบบเป็นไปตามที่เราต้องการยกตัวอย่างเช่นเราสามารถกำหนดให้รถที่จอดอยู่เคลื่อนที่ไปในสถานีที่เราต้องการได้ (ในที่นี้ขอยกตัวอย่างจาก A ไป B)โดยการเคลื่อนที่ของรถนั้นจะเริ่มจากสถานี A จะเคลื่อนที่ไปยังสถานี B และรถจาก Bจะเคลื่อนที่ไป C และ C จะเคลื่อนที่ไป D และ Dจะเคลื่อนที่ไป A

5.2 ปัญหาที่พบและแนวทางแก้ไข

จากการศึกษาและทำโครงงานได้เกิดปัญหาคือ ทองแดงที่ใต้ฐานของรถไม่สัมผัสกับราง ทำให้รถหยุดวิ่งบางจังหวะ ทำให้ต้อง Reset ระบบเพื่อเริ่มทำงานใหม่ แผ่นไม้ที่ใช้ในการรอง รางไฟฟ้าเกิดการโก่งงอ ทำให้ล้อของรถไม่สัมผัสกับพื้นทำให้เกิดอาการ ล้อฟรีเกิดขึ้นพื้นรางมีสิ่งสกปรกติดอยู่ทำให้ Sensor ใต้ท้องรถ ตรวจจับว่าเป็นสีดำจึงทำให้รถจอดก่อนถึงสถานีจริง ซึ่งปัญหาส่วนใหญ่จะอยู่ในส่วนของตัวชิ้นงาน เมื่อมีการเคลื่อนย้าย หรือ มีการกระทบกระเทือนแม้เพียงเล็กน้อยก็ทำให้ตัวรถที่กระทบกระเทือนเกิดความเสียหายเปลี่ยนไปจากเดิม ได้ทำการแก้ไข คือ ตรวจสอบสภาพรถให้ปกติ แต่ปัญหาลักษณะเดิมก็ยังคงมีอยู่ และไม่ใช่แค่ส่วนตัวรถ ส่วนของราง หรือ สถานีก็มีการผิดพลาดเช่นกัน แนวทางแก้ไข คือตรวจสอบสภาพลดการกระทบกระเทือนของวงจร นำตัวไอซีPIC ไม่โครคอนโทรลเลอร์ มาเขียนโปรแกรมใส่ใหม่ ซึ่งก็สามารถแก้ปัญหาได้ในระดับหนึ่ง ปัญหานี้สันนิษฐานว่า เป็นความคลาดเคลื่อนเนื่องจากได้รับการกระทบกระเทือนจึงทำให้วงจรอิเล็กทรอนิกส์ควบคุมอยู่ข้างใน เกิดการคลาดเลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

ระบบลำเลียง นี้สามารถนำไปค้นคว้าพัฒนาต่อได้ ตัวแปรที่สำคัญในระบบลำเลียงอีกตัวหนึ่งคือสภาวะของระบบที่ต้องการลำเลียงที่ต้องการควบคุมจะถูกควบคุมโดยการสภาพของการลำเลียงทั้งสิ้นโดยส่งผลไปที่ตัวพาหนะเซนเซอร์ต่างๆในระบบดังนั้น ระบบลำเลียงนี้สามารถนำมาค้นคว้าพัฒนาต่อไปได้โดยการนำมาประยุกต์ให้เข้ากับสภาพของการลำเลียงและยังต้องเขียนโปรแกรมให้เหมาะสมกับตัวระบบลำเลียงให้ระบบทำหน้าที่ได้อย่างมีประสิทธิภาพ เพื่อที่จะสามารถนำไปใช้กับอุตสาหกรรมที่ต้องใช้การลำเลียงอาทิเช่นอุตสาหกรรมอาหารอุตสาหกรรมปิโตรเคมี หรือทางอุตสาหกรรมชิ้นส่วน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ส่วนโปรแกรมภาษาซีที่ใช้ในโครงงาน

ส่วน MainController

```
/* 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);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    readBu[1] = digitalRead(pinNum+1);
    readBu[2] = digitalRead(pinNum+2);
    return (readBu[0]*1 + readBu[1]*2 + readBu[2]*4);
}
/**
 * @brief Write Parallel
 */void writing (uint8_t pinNum,uint8_t wr){
 uint8_t wrBu[3];
 switch(wr){
  case 0: wrBu[0] = 0; wrBu[1] = 0; wrBu[2] = 0;
      break;
  case 1: wrBu[0] = 1; wrBu[1] = 0; wrBu[2] = 0;
    break;

case 2: wrBu[0] = 0; wrBu[1] = 1; wrBu[2] = 0;
      break;
  case 3: wrBu[0] = 1; wrBu[1] = 1; wrBu[2] = 0;
      break;
  case 4: wrBu[0] = 0; wrBu[1] = 0; wrBu[2] = 1;
      break;
  case 5: wrBu[0] = 1; wrBu[1] = 0; wrBu[2] = 1;
      break;
  case 6: wrBu[0] = 0; wrBu[1] = 1; wrBu[2] = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
    case 7: wrBu[0] = 1; wrBu[1] = 1; wrBu[2] = 1;
        break;
   }
   for(uint8_t i = 0; i<=2 ; i++){
    if(wrBu[i] == 1){
     digitalWrite(pinNum+i, HIGH);
    }
    else{
     digitalWrite(pinNum+i, LOW);
    }
   }
}
/**
 * @brief Parallel Control
 */
uint8_t parallelCtrl(uint8_t pin,uint8_t data){
 uint8_t readValue;
 writing(pin,0);       //close slave read
 digitalWrite(pin+6,LOW);
 delay(20);
 readValue = reading(pin+3);
 writing(pin,data);
 delay(20);            //delay for ready
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pinMode(30,OUTPUT);    //MOSI
pinMode(31,OUTPUT);    //MOSI
pinMode(32,INPUT);     //MISO
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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  pinMode(46,INPUT);     //MISO
  pinMode(47,INPUT);     //MISO
  pinMode(48,INPUT);     //MISO
  pinMode(49,OUTPUT);    //SCK
  digitalWrite(49,HIGH); //Clock High
  writing(43,0);
  delay(1000);

  // Go! Go! Go!
  Serial.println("command");
}
/**
 * @brief Super Loop
 */
void loop(){
  // Keyboard input
    unsigned char iBu = 0;
  do{
    while(Serial.available() <= 0);
    stackBu[iBu] = Serial.read();
    Serial.print(stackBu[iBu],BYTE);
  }while((stackBu[iBu] != ' ')&&(iBu++<=7));
  // Filter data : just pass '1' to '4'
  if((stackBu[1] >= 54)&&(stackBu[0] <= 60)){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  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=");
    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))){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 Serial.print("\n");
}
// Send to C
else if((stackBu[0] == 'C')&&((stackBu[1]>=65)&&(stackBu[1]<=68))){
 if(stackBu[1] == 'A'){
  keeping = '2';
 }
 else if(stackBu[1] == 'B'){
  keeping = '3';
 }
 else if(stackBu[1] == 'C'){
  keeping = '4';
 }
 else if(stackBu[1] == 'D'){
  keeping = '1';
 }
 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 ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Serial.print(flag, DEC);
      Serial.print(" go to station ");
      Serial.print(stackBu[1]);
     }
     Serial.print("\n");
    }
    // For Wrong Instruction
    else{
     Serial.println("Wrong Instruction");
    }
}
```

ส่วน Station

```
/* Includes ------------------------------------------------------*/
#include<avr/io.h>
#include<avr/interrupt.h>
#include<compat/deprecated.h>
#define F_CPU 8000000UL
#include<util/delay.h>

#define FOSC 8000000// Clock Speed
#define BAUD 2400
#define MYUBRR FOSC/16/BAUD-1

/* Define ------------------------------------------------------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define F_IR 38000          // Frequency of IR Receiver Module
#define TOP_PWM F_CPU/2/F_IR  // Maximum Value of Duty Cycle
#define data0 (PINC&0x01)
#define data1 (PINC&0x02)>>1
#define data2 (PINC&0b00000100)>>2

/* Private Variable ----------------------------------------------*/
unsigned char toggle = 1;       // Exti toggle
volatile unsigned char stTemp;          // stack temorary.
unsigned char dataTemp;          // data temporary.
volatile unsigned char dataStacked = '0';     // Data is stacked from serial
unsigned char dataBuffered;       // Data is bufferd from spi to serial
unsigned char keep = 0;
volatile char stFlag = 0;

/* Private Function ----------------------------------------------*/
/**
 * @brief Delay Millisecond.
 */
void delay_ms(unsigned int i){
  for(;i>0;i--)
    _delay_ms(1);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**
 * @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 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  sei();
}
/**
 * @brief Initialize PWM.
 */
void Init_PWM(){
  TCCR1A = 0xF2;
  TCCR1B = 0x11;
  ICR1 = TOP_PWM;
  OCR1A = TOP_PWM;
  OCR1B = TOP_PWM/2;
}
/**
 * @brief Initialize parallel
 */
void parallelSetup(void){
 // PC3 , PC4, PC5 -> MISO
 // PC0 , PC1, PC2 -> MOSI
 // PB2 -> SCK
 // MOSI
 cbi(DDRC,0);
 cbi(DDRC,1);
 cbi(DDRC,2);
 // SCK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break;
        case 7: sbi(PORTC,3);
            sbi(PORTC,4);
            sbi(PORTC,5);
            break;
    }
}

/**
 * @brief Main.
 */
int main(void){
  Init_PORT();
  Init_PWM();
  Init_Interrupt();
  Init_USART(MYUBRR);
  parallelSetup();
  /*while(1){
    if(stFlag == 1){
      delay_ms(50);
      transmit(dataStacked);
      stFlag = 0;
      dataStacked = '0';
      transmit(dataStacked);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
}*/
/* Start Process */
while(1){
  if((dataStacked<=47)&&(dataStacked>=55)){
    dataStacked = '0';
  }
  // Read when detect LOW
  if(!(PINB&0x04)){
      parallelWrite(dataStacked-48);
  }
  else{
    keep = parallelRead();
    if((keep==1)||(keep==2)||(keep==3)||(keep==4)){
      delay_ms(25);
      transmit(0x01);
      transmit(0x01);
      transmit('<');
      transmit(keep+48);

      delay_ms(500);
      transmit(0x01);
      transmit(0x01);
      transmit('<');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            transmit(keep+48);
            dataStacked = '0';
          }
        }

    }
    return 0;
}

/* Private Function -----------------------------------------*/
/**
 * @brief  Interrupt Service Routine of " USART1->RX ".
 *    Send Value From RX1[Infrared] to TX0[Computer].
 */
ISR(USART_RXC_vect){
    dataTemp = receive();
  if(dataTemp == '['){     // from car protocol
      stTemp = 1;
    stFlag = 0;
  }
  if((stTemp == 1)&&(dataTemp !='[')){
      stTemp = 0;
    stFlag = 1;
      dataStacked = dataTemp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      //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

ส่วน Vehicle

```
/* Include ------------------------------------------------------------*/

#include <16F690.h>


#device *=16

#device ICD=TRUE

#device adc=8


#FUSES NOWDT            //No Watch Dog Timer

#FUSES HS               //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)

#FUSES NOPROTECT        //Code not protected from reading

#FUSES NOBROWNOUT       //No brownout reset

#FUSES MCLR             //Master Clear pin enabled

#FUSES NOCPD            //No EE protection

#FUSES NOPUT            //No Power Up Timer

#FUSES IESO             //Internal External Switch Over mode enabled

#FUSES FCMEN            //Fail-safe clock monitor enabled


#define  KEYHIT_DELAY   500    // in milliseconds

#use delay(clock=20000000)

#use rs232(baud=2400,parity=N,xmit=PIN_B7,rcv=PIN_B5,bits=8, stream=AVR,timeout=KEYHIT_DELAY)


/* Define ------------------------------------------------------------*/

#define RobotNumber '4'         // RobotNumber 1 to 6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define Analog0 4                // AnalogChannel 4

#define MotorDrive0 PIN_C3       // MotorDrivePin

#define MotorDrive1 PIN_C4       // MotorDrivePin

#define MPWMPin PIN_C6           // Manual PWM Pin_C2

#define ResMPWM 100              // Maximum Resolution PWM

#define IrFreq 528               // Ir Receive Freq 38 kHz.

#define NcPin0 PIN_A0            // Empty Pin0

#define NcPin1 PIN_B4            // Empty Pin1

#define NcPin2 PIN_C6            // Empty Pin2

#define NcPin3 PIN_B6            // Empty Pin3

#define NcPin4 PIN_C7            // Empty Pin4

#define Button0 !Input(NcPin4)   // Input0 use to memo

#define Button1 !Input(NcPin2)   // Input1 use to run, skip

#define Led0  NcPin0             // Output0 use to run status

#define Led1  NcPin1             // Output1 use to memo status

#define Led2  NcPin3             // Output2 use to sense

#define MaxSpeed 100             // Maximum Motor Speed.

#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.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char Range;              // Error Range.
char Loopi;                       // Counter Loop.
unsigned char Duty;               // Set Duty Cycle Of Manual PWM.
unsigned char Mode;               // Select MotorMode.
unsigned char skipStation = 1;    // Value of skiping station


/* Private Function -----------------------------------------------------*/
void MotorForward(char);
void MotorStop(void);
void MotorBackward(char);
int stacker(void);
unsigned char receive(void);


/* All Interrupt Service Routine ----------------------------------------*/
/**
 * @ brief RTCC Interrupt Service Routine.
 *       Read Sensor.
 */
#int_RTCC
void  RTCC_isr(void)
{
 /* PWM Motor Control */
 if(Duty >= Loopi){
   if(Mode == 1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      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. ************************************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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

/* Memory Value Sensor ***********************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* 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);

  if(--skipStation == 0){          // Skip Check and Reduce
    /* Stop Motor. ***************************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Enable_Interrupts(INT_RTCC);
        delay_ms(25);
        Enable_Interrupts(INT_TIMER1);

        // receive skipStation
        skipStation = receive();

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

      /* Running Again ***************************************/
      do{
       Output_Low(Led0);          // turn on running status led
        MotorForward(MaxSpeed);
      }while(ANL0);
    }
    return 0;
}
/**
 * @brief Forward Motor.
 */
void MotorForward(char i){
  Mode = 1;
  Duty = i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/**
 * @brief Stop Motor.
 */
void MotorStop(){
  Mode = 2;
  Duty = 100;
}

/**
 * @brief Backward Motor.
 */
void MotorBackward(char i){
  Mode = 3;
  Duty = i;

}

unsigned char receive(void){
  unsigned char value;
  unsigned char value2 = 0;
  unsigned char keep = '1';
  do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    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

ส่วน Visual Basic

โปรแกรม VB

```
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 สำหรับการแสดงภาพในแตกละวิโดส์ว
    ' 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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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
  Cb_port.Sorted = True
  For Each name In portname '  เป็นการกำหนดค่าของport โดย portname คือ port IO
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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        '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")
            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 = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    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")
            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) 'ตัวแปรส่งค่าไปผ่านserialportเพื่อส่งไปให้สถานี
ทำงาน
        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")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End If
        If Ckb_arm1.Checked = True Then
          arm("3")
        End If
        If Ckb_arm2.Checked = True Then
          arm("4")
        End If
        If Ckb_arm3.Checked = True Then
          arm("1")
        End If
        If Ckb_arm4.Checked = True Then
          arm("2")
        End If
        If Ckb_stall.Checked = True Then
          output("A", "B") '5
          output("B", "C") '6
          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
If Ckb_stD.Checked = True Then
  Select Case Cb_d.Text
    Case "D"
      output("D", "D") '4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Case "A"
                    output("D", "A") '8
                Case "B"
                    output("D", "B") '12
                Case "C"
                    output("D", "C") '16
            End Select
            If Cb_d.Text = "" Then
                MessageBox.Show("ยังไม่ได้ทำการเลือกสถานีปลายทางของสถานี D ", "Select the final station",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
            End If
        End If
    ElseIf Rd_auto.Checked = True Then
        tim_robot_go.Interval = 60000
        Timer1.Interval = 1000
        tim_robot_go.Enabled = True
        Timer1.Enabled = True
    End If
  Else
    TXT_status.Text = "Don't connect"
  End If
End Sub
Private Sub tim_robot_go_Tick_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
tim_robot_go.Tick
  If Rd_auto.Checked = True Then
    If SRP.IsOpen = True Then
      If turnnum < NumericUpDown1.Value Then
        intnum = intnum + 1
        SRP.Write("A")
        SRP.Write("U")
        SRP.Write("T")
        SRP.Write(" ")
        If intnum Mod 4 = 0 Then
          turnnum = turnnum + 1
          intnum = 0
        End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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", "เลือกสั่งได้ทีละแบบนะ",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk)

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

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


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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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
            If Rd_auto.Checked = True Then
                MessageBox.Show("กรุณากรอกตัวเลขเป็นตัวเลขครับ แล้วกดปุ่ม go", "กรอกจำนวนรอบ", _
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
            End If
        Else
            TXT_status.Text = "Don't connect "
        End If
    End Sub
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
        If Rd_auto.Checked = True Then
            If SRP.IsOpen() = True Then
                i = i + 1
                If i = 60 Then
                    i = 0
                End If
                Label5.Text = " รอบที่ : " & turnnum.ToString & " รถเคลื่อนที่ไป " & intnum.ToString & " station"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Else
            i = 0
        End If
        Label6.Text = i.ToString
    End If
End Sub

'*****************************************************************************************
*

  " 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, _
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        480, picCapture.Handle.ToInt32, 0)
    ' ติดต่อกับ device
    If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then
      ' ตั้งค่า preview scale
      SendMessage(hHwnd, WM_CAP_SET_SCALE, True, 0)
      ' ตั้งค่า preview rate ในระดับ milliseconds
      SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, 66, 0)
      ' เริ่มต้นการแสดงภาพ จากงกล้อง
      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 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        btnStop.Enabled = False
        btnStart.Enabled = True
        If sfdImage.ShowDialog = DialogResult.OK Then
          bmap.Save(sfdImage.FileName, Imaging.ImageFormat.Bmp)
        End If
        OpenPreviewWindow()
          End If
    End Sub

    ' ขั้นตอนสุดท้าย, การปิดแสดงภาพใน โปรแกรมของเรา,เลิกทำการติดต่อ deviceและ destroy (ทำลาย Object)
ที่ เราสร้างขึ้นมาตอนต้นการเขียนโปรแกรม ที่ให้แสดงภาพ และปิดโปรแกรม
    Private Sub ClosePreviewWindow()
      ' เลิกติดต่อกับ device
      SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)
      ' ปิด window
      DestroyWindow(hHwnd)
    End Sub

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โปสเตอร์ระบบลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Webmaster. " S3003 FUTABA SERVO ."[Online]. Available : http://www.etteam.com/product/1602.html.2553.

[2] Webmaster. "(EK2-0508) Digital Servo."[Online]. Available : http://www.helipal.com/ek2-0508-digital-servo-8g-for-esky-helicopters.html.2553

[3] Webmaster. "TowerPro MG945 Digital Servo."[Online]. Available : http://chigift.en.alibaba.com/product/236283415200760159/TowerPro_MG945_Digital_Servo_toy_parts.html.2553.

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้