สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมตำแหน่งลูกบอลบนระนาบ

BALL PLATE CONTROL SYSTEM

T119177

นางสาวฉัตราภรณ์ ศรีพิศุทธิ์
นายณัฐชัย มาสกุลรัตน์
นายธีร์พิสุทธิ์ เตชะไกรศรี

เลขหมู่.....
เลขทะเบียน 119177
วัน,เดือน,ปี - 6 S.A. 2554
b.12366948
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BALL PLATE CONTROL SYSTEM

Ms. Shattraporn Sripusut

Mr. Nattachai Maskulrat

Mr. Teepisutt Techakaisri

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมตำแหน่งลูกบอลบนระนาบ

BALL PLATE CONTROL SYSTEM

ผู้จัดทำ นางสาวฉัตราภรณ์ ศรีพิศุทธิ์ 50010279

นายณัฐชัย มาสกุลรัตน์ 50010446

นายธีร์พิสุทธิ์ เตชะไกรศรี 50010701

.....อาจารย์ที่ปรึกษา

(รองศาสตราจารย์.ดร. ถาวร เบญจนราสุทธิ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมตำแหน่งลูกบอลบนระนาบ

โดย

นางสาวฉัตราภรณ์ ศรีพิศุทธิ์ 50010279

นายณัฐชัย มาสกุลรัตน์ 50010446

นายธีพิสุทธิ์ เตชะไกรศรี 50010701

อาจารย์ที่ปรึกษา

รองศาสตราจารย์.ดร. ถาวร เบญจนราสุทธิ์

ปีการศึกษา 2553

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอทฤษฎีและการออกแบบระบบควบคุมอัตโนมัติสำหรับควบคุมตำแหน่งลูกบอลบนระนาบ ซึ่งโครงสร้างของระบบประกอบด้วย โครงสร้างทางกลของแผ่นระนาบลูกบอล กล้องดิจิตอล คอมพิวเตอร์ ไมโครคอนโทรลเลอร์ และวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง โดยเป้าหมายที่ตั้งไว้คือ ควบคุมลูกบอลบนแผ่นระนาบให้สามารถไปยังตำแหน่งที่ต้องการและรักษาตำแหน่งของลูกบอลนั้นได้ โดยควบคุมผ่านมอเตอร์กระแสตรงที่ต่อกับแผ่นระนาบ

ในโครงงานนี้ใช้การควบคุมแบบวงปิด 2 วง ประกอบด้วยการควบคุมวงในซึ่งทำหน้าที่ควบคุมมุมของระนาบ ส่วนการควบคุมวงนอกจะทำหน้าที่ควบคุมตำแหน่งลูกบอล โดยการควบคุมวงในอาศัยตัวควบคุมพีดี และอาศัยตัวควบคุมพีไอดีในส่วนการควบคุมส่วนนอก ผลการทดลองพบว่าระบบควบคุมที่ออกแบบสามารถควบคุมลูกบอลบนแผ่นระนาบได้ให้สามารถเคลื่อนที่ไปตำแหน่งที่ต้องการและรักษาตำแหน่งของลูกบอลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BALL PLATE CONTROL SYSTEM

By

Ms.Chattraporn Sripisut 50010279

Mr. Nattachai Maskulrat 50010446

Mr.Teepisutt Techakaisri 50010701

Advisor

Assoc. Prof. Dr. Taworn Benjanarasuth

Academic Year 2010

ABSTRACT

This thesis presents theories and designs of a ball plate control system. The system consists of the ball and plate structure, digital camera, computer, microcontroller, and interfacing circuits. The goal is to control the ball to move to the specified position on the plate and maintains the ball at that position by controlling the dc motors connected to the plate.

In this project, the control scheme employs two feedback loops. The first control loop is used to control the angular position of the plate and the second loop is used to control the ball position. Both loops are controlled by PID controllers. The experimental results verify that the designed control system can successfully control the ball to move to the specified position and maintains the ball at that position.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ที่ปรึกษา รศ.ดร.ถาวร เบญจนราสุทธิ์ เป็นอย่างสูง ที่ได้กรุณาให้คำปรึกษาแนวทางในการแก้ปัญหา ความคิดริเริ่ม รวมถึงให้ความเอาใจใส่ดูแลสอบถามถึงความก้าวหน้าอย่างสม่ำเสมอ ทำให้คณะผู้จัดทำได้ทำงานอย่างมีประสิทธิผลยิ่งขึ้น

ขอขอบพระคุณ อาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทความรู้ทั้งทางทฤษฎีและทางปฏิบัติ แก่คณะผู้จัดทำ ทำให้เข้าใจและสามารถนำความรู้มาใช้ในปริญญานิพนธ์ฉบับนี้

ขอขอบคุณสาขาวิชาวิศวกรรมวัดและควบคุม ที่เอื้อเฟื้ออุปกรณ์ และเครื่องมือ ตลอดจนอำนวยความสะดวกให้แก่คณะผู้จัดทำจนปริญญานิพนธ์ฉบับนี้เสร็จสิ้น

ขอบคุณเพื่อน ๆ ทุก ๆ คนที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ และคอยถามไถ่ความคืบหน้าของโครงงานตลอดเวลา

และท้ายสุดขอขอบพระคุณ คุณพ่อ คุณแม่และครอบครัว ที่คอยให้การสนับสนุนและโอกาสแก่คณะผู้จัดทำได้เล่าเรียนจนถึงวันนี้ ตลอดจนเพื่อน ๆ พี่ ๆ และน้อง ๆ ในห้องวิจัยทุกคน ที่ให้คำปรึกษาแนะนำ และเป็นกำลังใจจนปริญญานิพนธ์ฉบับนี้เสร็จสิ้นสมบูรณ์

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำขอมอบแก่ผู้มีพระคุณทุกท่าน

ผู้จัดทำ

นางสาวฉัตราภรณ์ ศรีพิศุทธิ์

นายณัฐชัย มาสกุลรัตน์

นายธีร์พิสุทธิ์ เตชะไกรศรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของการจัดทำโครงงาน	1
1.2 เนื้อหาในปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	3
2.1 ทฤษฎีการประมวลผลภาพ	3
2.1.1 การกำหนดสี	3
2.1.2 ภาพดิจิตอล	4
2.1.3 การแปลงภาพขาวดำ	6
2.1.4 **สัญญาณข้อมูลภาพจากดิจิตอลวิดีโอ**	6
2.1.5 ข้อมูลภาพชนิดบิตแมป	6
2.2 ตัวควบคุม	7
2.2.1 ระบบควบคุมแบบพี	7
2.2.2 ระบบควบคุมแบบไอ	9
2.2.3 ระบบควบคุมแบบดี	10
2.2.4 ระบบควบคุมแบบพีไอ	11
2.2.5 ระบบควบคุมแบบพีดี	14
2.2.6 ระบบควบคุมแบบพีไอดี	14
2.3 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม	15
2.3.1 การสื่อสารแบบอะซิงโครนัส	15
2.3.2 การสื่อสารแบบซิงโครนัส	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.4 การควบคุมมอเตอร์ไฟฟ้ากระแสตรงด้วยสัญญาณพีดับบิวเอ็ม 18

2.5 หลักการทำงานของโพเท็นทิโอมิเตอร์ 19

2.6 แบบจำลองทางคณิตศาสตร์ของระบบควบคุมลูกบอลบนระนาบ 23

2.4 ไมโครคอนโทรลเลอร์ 18

บทที่ 3 โครงสร้างของระบบ 22

3.1 ส่วนประกอบของโครงสร้างทางกายภาพ 22

3.1.1 แผ่นระนาบอลูมิเนียม 23

3.1.2 โพเท็นทิโอมิเตอร์ 23

3.1.3 มอเตอร์ไฟฟ้ากระแสตรง 24

3.1.4 วงจรขับมอเตอร์ 24

3.1.5 กล้องดิจิตอลวีดีโอ 25

3.1.6 คอมพิวเตอร์ 25

3.1.7 ไมโครคอนโทรลเลอร์ 26

3.2 การออกแบบโปรแกรม 28

3.2.1 โปรแกรมส่วนคอมพิวเตอร์ 28

3.2.2 โปรแกรมส่วนไมโครคอนโทรลเลอร์ 29

บทที่ 4 ผลการทดลอง 33

4.1 การระบุตำแหน่งมุมของระนาบโดยเทียบกับค่าสัญญาณอนาล็อกเป็นดิจิตอล 33

4.2 การทดสอบขับมอเตอร์โดยตรง และการประมาณฟังก์ชั่นถ่ายโอน 33

4.3 การจำลองการควบคุมมุมแผ่นระนาบแบบวงปิด 36

4.4 การทดลองควบคุมมุมระนาบ 37

4.5 การระบุตำแหน่งของลูกบอลโดยเทียบพิกเซล 39

4.6 การทดลองควบคุมตำแหน่งลูกบอลบนแผ่นระนาบ 39

4.6.1 การทดลองปล่อยลูกบอลจริงครั้งที่ 1 40

4.6.2 การทดลองปล่อยลูกบอลจริงครั้งที่ 2 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 5 บทวิจารณ์และสรุปผล 44

5.1 สรุปผลการทดลอง 44

5.2 ปัญหาที่พบและแนวทางการแก้ไข 44

5.3 ข้อเสนอแนะ 44

ภาคผนวก ก โปรแกรมการประมวลผลภาพและโปรแกรมควบคุม 45

ก.1 โปรแกรมในส่วนคอมพิวเตอร์ 47

ก.1.1 โปรแกรมควบคุมตำแหน่งลูกบอล 47

ก.1.2 โปรแกรมการควบคุมมุมระนาบ 55

ก.2 โปรแกรมในส่วนไมโครคอนโทรลเลอร์ 70

ก.2.1 โปรแกรม ADC chennel 70

ก.2.3 โปรแกรมดีเลย์ 86

ก.2.3 โปรแกรม USART 86

ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์ 89

ข.1 เอกสารคู่มือการใช้งาน ATMEL32 89

ข.2 เอกสารคู่มือการใช้งาน L293D 91

ข.3 เอกสารคู่มือการใช้งาน MAX232 92

ข.4 เอกสารคู่มือการใช้งาน 7805 93

เอกสารอ้างอิง 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ **หน้า**

2.1 โมเดลอาร์จีบี 4
2.2 บล็อกไดอะแกรมของระบบควบคุมป้อนกลับแบบลบหนึ่งหน่วย 7
2.3 ผลตอบสนองของตัวควบคุมแบบพี 8
2.4 ผลตอบสนองของตัวควบคุมแบบไอ 10
2.5 ผลตอบสนองของตัวควบคุมแบบดี 10
2.6 ผลตอบสนองของตัวควบคุมแบบพีไอ 12
2.7 ผลตอบสนองตัวควบคุมแบบพีไอ 12
2.8 ผลตอบสนองของตัวควบคุมแบบพีดี 14
2.9 ผลตอบสนองของตัวควบคุมแบบพีไอดี 14
2.10 รูปแบบการสื่อสารแบบอะซิงโครนัส 16
2.11 ตัวอย่างการส่งข้อมูลขนาด 8 บิต แบบอนุกรม 17
2.12 สัญญาณพีดับบิวเอ็มมีความกว้างพัลส์เท่ากับ 10 เปอร์เซ็นต์ 18
2.13 สัญญาณพีดับบิวเอ็มมีความกว้างพัลส์เท่ากับ 50 เปอร์เซ็นต์ 18
2.14 สัญญาณพีดับบิวเอ็มมีความกว้างพัลส์เท่ากับ 90 เปอร์เซ็นต์ 18
2.15 โพเท็นทิโอมิเตอร์ 19
2.16 แนวการเคลื่อนที่ของลูกบอล 19
2.17 บล็อกไดอะแกรมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ 21
3.1 หลักการทำงานของระบบควบคุมตำแหน่งลูกบอลบนระนาบ 22
3.2 โครงสร้างทางกายภาพของระบบ 23
3.3 โพเท็นทิโอมิเตอร์ขนาด 10 กิโลโอห์ม 23
3.4 มอเตอร์ไฟฟ้ากระแสตรงติดตั้งกับแผ่นระนาบพร้อมต่อกับโพเท็นทิโอมิเตอร์ 24
3.5 วงจรขับมอเตอร์ 24
3.6 กล้องดิจิตอลวีดีโอ 25
3.7 ไมโครคอนโทรลเลอร์ Atmega32 16PU 26
3.8 วงจรไฟเลี้ยง 26
3.9 วงจรการใช้งานไมโครคอนโทรลเลอร์ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่ **หน้า**

3.10 วงจรแปลงสัญญาณ UART – RS232 28

3.11 โฟลว์ชาร์ตควบคุมตำแหน่งลูกบอลบนแผ่นระนาบในคอมพิวเตอร์ 30

3.12 โฟลว์ชาร์ตการหาตำแหน่งลูกบอลบนแผ่นระนาบในคอมพิวเตอร์ 31

3.13 โฟลว์ชาร์ตการควบคุมมุมของระนาบโดยไมโครคอนโทรลเลอร์ 32

4.1 ทดสอบระบบโดยจ่ายระดับแรงดันที่ 70% ของสัญญาณพีดับบิวเอ็มแกน X 34

4.2 ทดสอบระบบโดยจ่ายระดับแรงดันที่ 70% ของสัญญาณพีดับบิวเอ็มแกน Y 35

4.3 ผลการจำลองระบบควบคุมมุมระนาบแกน X 36

4.4 ผลการจำลองระบบควบคุมมุมระนาบแกน Y 37

4.5 ผลการทดลองควบคุมมุมระนาบแกน X 38

4.6 ผลการทดลองควบคุมมุมระนาบแกน Y 38

4.7 ผลการทดลองควบคุมตำแหน่งลูกบอลแกน X 39

4.8 ผลการทดลองควบคุมตำแหน่งลูกบอลแกน Y 40

4.9 ผลการทดลองควบคุมตำแหน่งลูกบอลโดยปล่อยลูกบอลครั้งที่ 1 41

4.10 ผลการทดลองควบคุมตำแหน่งลูกบอลโดยปล่อยลูกบอลครั้งที่ 2 43

ก.1 โปรแกรมอินเตอร์เฟสของวิชชวลเบสิก 45

ก.2 หน้าต่างการปรับพารามิเตอร์พีไอดีของระบบโดยรวม 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

4.1 ผลทดลองความสัมพันธ์ระหว่างค่าระดับสัญญาณกับมุมของระนาบ 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การศึกษาในสาขาวิชาวิศวกรรมแมคคาทรอนิกส์ เป็นการศึกษาและประยุกต์ทฤษฎีต่าง ๆ เพื่อการออกแบบในเชิงเครื่องกล และการควบคุมระบบเชิงกลให้มีเสถียรภาพ ด้วยการควบคุมทางอิเล็กทรอนิกส์และคอมพิวเตอร์ เพื่อให้ได้สมรรถนะตามความต้องการหรือให้เป็นตามที่กำหนดไว้ อาทิ ระบบควบคุมความเร็วและตำแหน่งการหมุนของมอเตอร์ไฟฟ้ากระแสตรง และระบบควบคุมตำแหน่งลูกบอลบนระนาบ

ในโครงงานนี้เป็นการนำความรู้ที่ได้ศึกษาในสาขาวิชาวิศวกรรมแมคคาทรอนิกส์ตลอด 4 ปีมาประยุกต์ใช้ให้เกิดเป็นรูปธรรมขึ้น โดยเลือกศึกษาการควบคุมในระนาบสองมิติแบบอัตโนมัติ ดังนั้นเพื่อให้เป็นไปตามวัตถุประสงค์ที่ต้องการ จึงจำเป็นต้องศึกษาถึงแบบจำลองระบบควบคุมด้วยสมการคณิตศาสตร์ การออกแบบตัวชดเชยต่าง ๆ อีกทั้งศึกษาในเรื่องวงจรและอุปกรณ์ทางอิเล็กทรอนิกส์ และโปรแกรมคอมพิวเตอร์ ตลอดจนการวัดและแปลงสัญญาณ ซึ่งต้องบูรณาการเรื่องที่ศึกษาทั้งหมดเหล่านี้เพื่อประยุกต์ใช้กับระบบควบคุมทางกายภาพจริง โดยเลือกศึกษาระบบควบคุมตำแหน่งลูกบอลบนระนาบ

ระบบควบคุมตำแหน่งลูกบอลบนระนาบ เป็นระบบที่ไม่มีเสถียรภาพวงเปิด จึงจำเป็นที่จะต้องมีการควบคุมแบบวงปิดเพื่อให้มีเสถียรภาพ ซึ่งในโครงงานนี้เลือกใช้การควบคุมแบบพีไอดี โดยเป้าหมายคือการควบคุมลูกบอลบนระนาบให้สามารถเคลื่อนไปยังตำแหน่งที่ต้องการและรักษาตำแหน่งของลูกบอล ณ ตำแหน่งที่ต้องการนั้นได้

รายละเอียดของบทนี้จะกล่าวถึงวัตถุประสงค์ของการจัดทำโครงงาน รวมถึงขั้นตอนการศึกษาและการจัดทำโครงงาน ตลอดจนรายละเอียดต่าง ๆ ที่จะกล่าวถึงในบทต่อไป

1.1 วัตถุประสงค์ของการจัดทำโครงงาน

1. ศึกษาทฤษฎีระบบควบคุมเพื่อสามารถออกแบบตัวควบคุมให้มีประสิทธิภาพ และสามารถนำไปใช้จริงได้

2. ศึกษาและทดลองการใช้งานอุปกรณ์ต่าง ๆ ซึ่งจำเป็นในระบบควบคุม ทั้งทางอิเล็กทรอนิกส์และทางกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ศึกษาและสร้างระบบควบคุมตำแหน่งลูกบอลบนระนาบ ซึ่งประกอบด้วยส่วนที่เป็นอุปกรณ์เชิงกลและเชิงอิเล็กทรอนิกส์ ตลอดจนการประยุกต์การประมวลผลภาพเพื่อใช้กล้องดิจิตอลเป็นอุปกรณ์วัดตำแหน่งของลูกบอลบนระนาบ ผ่านโปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นโดยใช้โปรแกรมวิชชวลเบสิค (Visual basic)

1.2 เนื้อหาในปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ประกอบด้วย บทที่ 1 บทนำ กล่าวนำถึงวัตถุประสงค์ ขั้นตอนการศึกษา การจัดทำโครงงาน พร้อมทั้งรายละเอียดของแต่ละบทในปริญญานิพนธ์ บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้องในการจัดทำโครงงาน เช่น ทฤษฎีการประมวลผลภาพ ตัวควบคุม และการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ บทที่ 3 หลักการออกแบบ นำเสนออุปกรณ์ที่ใช้ในระบบ โครงสร้างทางกายภาพของระบบ บทที่ 4 การทดลอง และบทที่ 5 บทวิจารณ์และสรุป ซึ่งจะสรุปผลการดำเนินงานในโครงงานทั้งหมด ปัญหาที่เผชิญ และแนวทางการปรับปรุงแก้ไขรวมไปถึงการพัฒนาโครงงานนี้ต่อไป

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและความรู้ที่เกี่ยวข้อง

บทนี้กล่าวถึงทฤษฎีที่จำเป็นต่อการใช้งานในโครงงานระบบควบคุมลูกบอลบนระนาบ ประกอบด้วยทฤษฎีต่าง ๆ ที่เกี่ยวข้อง ได้แก่ การประมวลผลภาพ ตัวควบคุมในกลุ่มพีไอดี การควบคุมมอเตอร์ด้วยสัญญาณพีดับบิวเอ็ม (Pulse width modulation) และการจำลองทางคณิตศาสตร์ของลูกบอลบนระนาบ

2.1 ทฤษฎีการประมวลผลภาพ

ความรู้เกี่ยวกับการประมวลผลภาพ เป็นส่วนสำคัญที่ต้องศึกษาเพื่อให้สามารถระบุตำแหน่งลูกบอลบนระนาบได้ โดยต้องทำความใจเกี่ยวกับหลักการของการประมวลผลภาพเบื้องต้น ซึ่งประกอบไปด้วยหัวข้อต่าง ๆ ดังนี้

2.1.1 การกำหนดสี

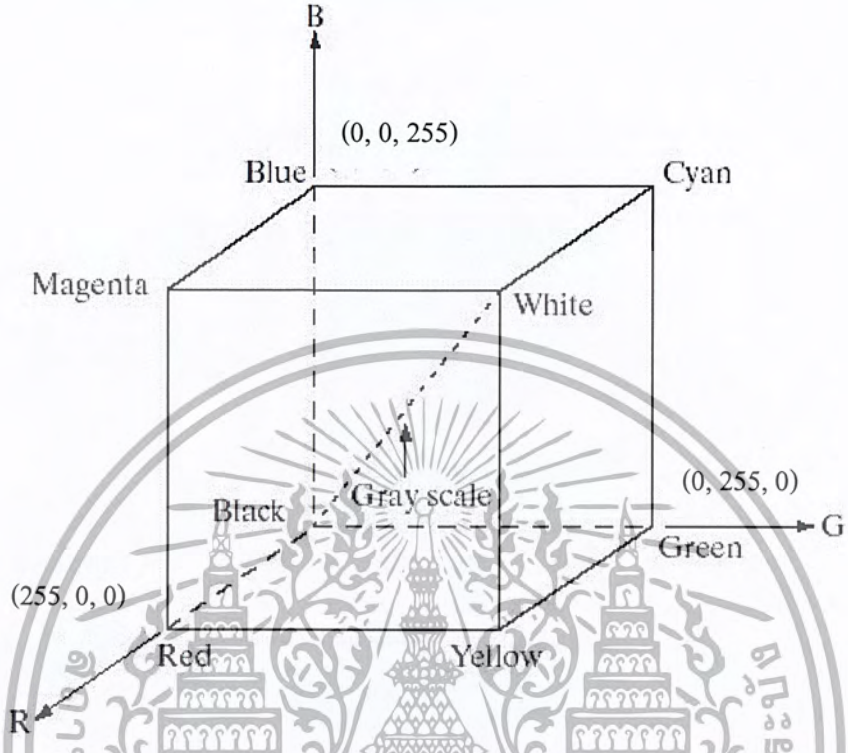
ในคอมพิวเตอร์ สีทุกสีจะนำเสนอด้วยสี 3 สีพื้นฐาน คือ สีแดง (Red) สีเขียว (Green) และสี น้ำเงิน (Blue) หรือเรียกว่า อาร์จีบี (RGB) ซึ่งสามารถสังเคราะห์สีอื่น ๆ ได้เกือบทุกสี โดยการผสมสีพื้นฐานทั้งสามในสัดส่วนต่าง ๆ กัน โดยการผสมขึ้นอยู่กับความเข้มขององค์ประกอบอาร์จีบี ซึ่งเรียกว่าโมเดลอาร์จีบี (RGB model)

สีแต่ละสีนำเสนอโดยสัดส่วนของสี 3 สีพื้นฐานข้างต้นในลักษณะลำดับ (แดง, เขียว, น้ำเงิน) โดยแต่ละสีนิยมนำเสนอด้วยเลขจำนวนเต็มขนาด 1 ไบต์ เท่ากับ 8 บิต เมื่อค่าที่น้อยที่สุดคือค่า 0 สอดคล้องกับการไม่มีสีพื้นตามสีนั้นเลย และค่าที่มากที่สุดคือค่า 255 บ่งบอกถึงความเข้มสูงสุดของสีพื้นฐานสีนั้น อาทิเช่น สัดส่วน (0, 0, 0) สอดคล้องกับสีดำ เพราะไม่มีความเข้มของสีทั้งสามเลย ในขณะที่สัดส่วน (255, 255, 255) สอดคล้องกับสีขาว ส่วนสีอื่น ๆ ที่เกิดจากการรวมตัวกันมีได้มากมาย เช่น (255, 0, 0) จะเป็นสีแดงบริสุทธิ์ และ (255, 255, 0) สีเหลือง ดังนั้นการรวมกันที่เป็นไปได้ของสีพื้นฐานในกรณีนี้มีขนาด 24 บิต จึงสามารถนำเสนอสีได้ 256x256x256 หรือเท่ากับ 16,777,216 ระดับสี

กระบวนการของการสร้างสี ซึ่งมีสีพื้นฐาน 3 สีนี้ ตั้งอยู่บนพื้นฐานกล่องสี ในรูปที่ 2.1 โดยมุมของกล่องนี้ จะสอดคล้องกับสีต่าง ๆ ดังรูป ปริมาณของสีพื้นฐานทั้งสามสีใด ๆ จะมีค่าตามแกนทั้งสามแกนคืออาร์จีบีตามลำดับ สีที่เป็นส่วนกลับ (Complementary colors) สามารถคำนวณได้อย่างง่าย ๆ โดยการลบค่าของสีจาก 255 ตัวอย่างเช่นสี (0, 0, 255) เป็นสีน้ำเงินบริสุทธิ์ เป็นสีส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลับของสีเหลืองบริสุทธิ์ ที่สอดคล้องกับ (255, 255, 0) ซึ่งสีทั้งสองอยู่มุมตรงข้ามกันของกล่องสี ส่วนมุมอื่น ๆ ก็เช่นเดียวกัน ดังสามารถแสดงได้ในรูปที่ 2.1



รูปที่ 2.1 โมเดลอาร์จีบี

จากรูปที่ 2.1 เห็นได้ว่าในกรณีที่ระดับสีพื้นฐานทั้ง 3 สี มีระดับเท่ากัน จะให้สีเป็นระดับสีเทา (Gray scale) ดังนั้นในกรณีนำเสนอภาพระดับสีเทาจึงสามารถนำเสนอด้วยเลขจำนวนเต็มขนาดเพียง 1 ไบต์ ดังนั้นระดับของสีเทาในกรณีนี้จึงมีค่าตั้งแต่ 0 ถึง 255 โดย 0 จะแทนความมืดมากที่สุดและ 255 จะแทนความสว่างมากที่สุดของระดับสีเทานั่นเอง

2.1.2 ภาพดิจิตอล

ภาพดิจิตอลโดยส่วนมากเป็นภาพที่ถูกแปลงมาจากภาพแบบอนาลอกเพื่อให้อยู่ในรูปของตัวเลขซึ่งสามารถใช้คอมพิวเตอร์ในการประมวลผลได้ โดยภาพจะถูกแบ่งพื้นที่เป็นสี่เหลี่ยมเล็ก ๆ ที่เรียกว่าจุดภาพ หรือ พิกเซล (Pixel) และระบุตำแหน่งของพิกเซลโดยพิกัด (x,y) โดยในแต่ละพิกเซลจะระบุข้อมูลระดับสีหรือค่าอาร์จีบี สำหรับภาพสี และระดับความเข้มสำหรับระดับสีเทา โดยการแปลงข้อมูลแบบดิจิตอลสามารถทำได้ดังขั้นตอนต่อไปนี้

เมื่อนำภาพแบบอนาลอกที่ต้องการประมวลผลมาผ่านส่วนที่เรียกว่าดิจิไตเซอร์ (Digitizer) ซึ่งมีหน้าที่เปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล จะทำให้ภาพที่นำเสนอด้วยฟังก์ชั่นต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

f(x,y) กลายเป็นฟังก์ชั่นที่ไม่ต่อเนื่องทั้งระนาบของภาพ ซึ่งเรียกว่าการสุ่ม (Sampling) และผ่านการควอนไตซ์ (Quantization) ก็จะได้ข้อมูลที่เป็นดิจิตอล

สมมติว่าสัญญาณภาพต่อเนื่อง *f(x,y)* ได้ถูกแปลงให้อยู่ฟังก์ชั่นที่ไม่ต่อเนื่องซึ่งจะสามารถนำเสนอในรูปเมตริกซ์ และสมมติว่ามีขนาด NxM ได้ตามสมการ (2.1)

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,M-1) \\ \vdots & \ddots & \vdots \\ f(N-1,0) & \cdots & f(N-1,M-1) \end{bmatrix} \quad (2.1)$$

โดยฟังก์ชั่น *f(x,y)* ทางขวาของสมการจะเรียกว่า ภาพดิจิตอล และทุกสมาชิกของเมตริกซ์จะเรียกว่าพิกเซล จากกระบวนการสร้างภาพดิจิตอลข้างต้น เห็นได้ว่าสามารถทราบขนาดของความละเอียดของภาพเป็น NxM พิกเซล และขนาดของข้อมูลภาพดิจิตอลจะมีขนาดดังสมการ (2.2)

$$B = NxMxG \quad (2.2)$$

เมื่อ B คือ ขนาดของข้อมูลภาพที่เป็นดิจิตอล (บิต)

G คือ จำนวนบิตที่ใช้ในการแทนข้อมูลภาพ 1 พิกเซล สามารถหาได้จากสมการ (2.3)

$$L = 2^G \quad (2.3)$$

เมื่อ L คือจำนวนระดับของสีที่ต้องการใช้ในการเก็บภาพ ภาพมาตรฐานโดยทั่วไปที่นิยมใช้งานมีการใช้ระดับสีต่าง ๆ ดังนี้

1. ภาพ 2 ระดับ *(L=2)* คือมีเพียงแต่จุดขาวกับจุดดำเท่านั้นโดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต *(G=1)*
2. ภาพ 16 ระดับ *(L=16)* คือในแต่ละจุดภาพจะมีขนาดของข้อมูล 4 บิต *(G=4)* สามารถทำให้แสดงได้ 16 ระดับความเข้ม หรือ16 ระดับสี ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือภาพขาวดำ
2. ภาพ 256 ระดับ *(L=256)* คือในแต่ละจุดภาพจะมีขนาดของข้อมูลขนาด 8 บิต *(G=8)* ซึ่งสามารถทำให้แสดงได้ 256 ระดับความเข้มหรือระดับสี ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือขาวดำ
3. ภาพทรูคัลเลอร์ (True color) ในแต่ละจุดภาพจะมีขนาดของข้อมูล 24 บิต *(G=24)* สามารถทำให้แสดงภาพได้เหมือนจริงเพราะสามารถแสดงสีได้ถึง 16,777,216 ระดับสี *(L=16,777,216)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การแปลงภาพเป็นภาพขาวดำ

การแปลงภาพเป็นภาพขาวดำ เป็นการประมวลผลภาพเชิงตัวเลขอย่างง่ายที่มีประโยชน์ใช้งานหลากหลาย โดยเป็นการประมวลผลภาพจากภาพหลายระดับสีให้เป็นภาพขาวดำหรือภาพ 2 ระดับ การแปลงสามารถอธิบายได้ดังสมการ (2.4)

$$F(x,y) = \begin{cases} 1, & f(x,y) > T \\ 0, & f(x,y) \leq T \end{cases} \tag{2.4}$$

โดยที่ $f(x,y)$ คือ ภาพดิจิตอลเริ่มต้น

T คือ ค่าเทรชโฮล (Threshold)

$F(x,y)$ คือ ภาพดิจิตอลขาวดำ หรือภาพดิจิตอล 2 ระดับ

ค่าเทรชโฮล T คือค่าสีระดับสีเทาที่นำมาใช้เป็นค่ามาตรฐานในการเปรียบเทียบ วิธีการหาค่าเทรชโฮลโดยง่ายวิธีหนึ่ง คือการทดลองปรับค่าเพื่อให้ได้ภาพขาวดำตรงกับความต้องการ

2.1.4 สัญญาณข้อมูลภาพจากดิจิตอลวิดีโอ

การส่งสัญญาณข้อมูลจากภาพวิดีโอมีลักษณะการส่งที่เป็นลำดับภาพเดี่ยวหรือเฟรม (Frame) ที่ฉายต่อเนื่องกัน เช่นระบบวีดีโอเอ็นทีเอสซี (Ntsc) จะส่งด้วยอัตราเร็ว 30 เฟรมต่อวินาที โดยดิจิตอลวีดีโอแต่ละเฟรมจะเป็นข้อมูลดิจิตอลในลักษณะเมตริกซ์ดังที่กล่าวไว้ในสมการ (2.1)

2.1.5 ข้อมูลภาพชนิดบิตแมป

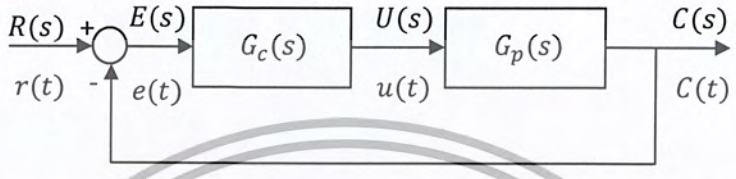
รูปแบบของไฟล์ข้อมูลภาพชนิดบิตแมป (Bitmap) เป็นรูปแบบไฟล์มาตรฐานสำหรับภาพกราฟฟิกส์บนวินโดว์ซึ่งจะใช้ในการตัดต่อ หรือทำสำเนาภาพต่าง ๆ ลงบนคลิปบอร์ด (Clipboard) เมื่อเวลาจัดเก็บไฟล์ที่มีนามสกุลบีเอ็มพี (Bmp) โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมปประกอบด้วยกัน 3 ส่วน คือ

1. ข้อมูลเฮดเดอร์ (Header) คือ ข้อมูลที่อยู่บริเวณส่วนหัวของไฟล์ ซึ่งประกอบด้วยข้อมูลที่บอกรายละเอียดต่าง ๆ ของภาพ เช่น ความกว้าง ความยาวของภาพ จำนวนสี จำนวนบิต และ ความละเอียดของภาพ
2. ข้อมูลจานสี (Palette) เป็นค่าแม่สีอาร์จีบี ของภาพ
3. ข้อมูลภาพ เป็นส่วนเก็บข้อมูลสีของแต่ละพิกเซลของภาพโดยข้อมูลแรกจะเป็นค่าสีของพิกเซลที่อยู่แถวล่างสุดที่ตำแหน่งซ้ายสุด ข้อมูลลำดับต่อไปจะเรียงจากทางขวาจากแถวล่างจนถึงแถวบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ตัวควบคุม

ตัวควบคุม เป็นส่วนประกอบหนึ่งในระบบควบคุมแบบป้อนกลับ ซึ่งมีสัญญาณความผิดพลาดเป็นอินพุต และมีเอาต์พุตเป็นอินพุตของอุปกรณ์ขับสำหรับระบบที่ต้องการควบคุม บล็อกไดอะแกรมของระบบควบคุมป้อนกลับแบบลบหนึ่งหน่วยซึ่งนิยมใช้งานโดยทั่วไป แสดงดังรูปที่ 2.2



รูปที่ 2.2 บล็อกไดอะแกรมของระบบควบคุมป้อนกลับแบบลบหนึ่งหน่วย

ในหัวข้อนี้จะกล่าวถึง ตัวควบคุมพื้นฐาน ซึ่งความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของตัวควบคุมที่นิยมใช้งานมี 3 รูปแบบ คือ

1. การควบคุมพี (P: Proportional control)
2. การควบคุมไอ (I: Integral control)
3. การควบคุมดี (D: Derivative control)

2.2.1 ระบบควบคุมแบบพี

ในระบบควบคุมพี เอาต์พุตของตัวควบคุม u(t) จะเป็นสัดส่วนกับอินพุตของตัวควบคุม และถ้ากำหนดสัญญาณให้ตัวควบคุมเป็น ค่าความผิดพลาด e(t) ซึ่งเป็นฟังก์ชันของเวลาจะได้

$$u(t) = K_p e(t) \tag{2.5}$$

เมื่อ K_p เป็นค่าคงที่ เรียกว่าอัตราขยายแบบสัดส่วน (Proportional gain) จะพบว่าเอาต์พุตที่ออกจากตัวควบคุมแบบพี เป็นสัดส่วนโดยตรงกับขนาดของความผิดพลาดในขณะที่กำลังพิจารณา ทำให้ฟังก์ชันถ่ายโอนของตัวควบคุม $G_c(s)$ มีค่าเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \tag{2.6}$$

ดังนั้นการควบคุมด้วยตัวควบคุมแบบนี้ก็จะเป็นเพียงการขยายสัญญาณความผิดพลาดเท่านั้น การที่ได้สัญญาณความผิดพลาดขนาดใหญ่ที่เวลาหนึ่ง ทำให้เกิดเอาต์พุตที่มีขนาดใหญ่จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กว่าค่าค่าหนึ่งก็ได้ ซึ่งการกำหนดเอาต์พุตของตัวควบคุมแบบพี ช่วงที่มีการกำหนดสัดส่วนเรียกว่าพร็อพพอร์ชั่นนอลแบน (Proportional band)

การกำหนดพร็อพพอร์ชั่นนอลแบนนี้ จะช่วยให้สัญญาณเอาต์พุตมีค่าจำกัด ไม่ไปสู่ค่าอนันต์ทั้งทางด้านบวก และทางด้านลบ และเมื่อตัวควบคุมมีเอาต์พุตสูงสุดที่จะเป็นไปได้ค่าหนึ่ง ก็นิยมที่จะกำหนดเอาต์พุตค่าใดๆ เป็นร้อยละของค่าสูงสุดที่จะเป็นไปได้ ดังนั้นการเปลี่ยนแปลงค่าเอาต์พุตของตัวควบคุม 100% ก็หมายถึงว่าเอาต์พุตจะเปลี่ยนค่าจากต่ำสุดที่เป็นไปได้ ไปสู่ค่าสูงสุดที่เป็นไปได้ ซึ่งจะได้ว่า

$$K_p = \frac{100}{Proportional\ Band} \tag{2.7}$$

เนื่องจากเอาต์พุตของตัวควบคุมจะเป็นสัดส่วนกับอินพุต ดังนั้น ถ้าหากอินพุตมีลักษณะเป็นสัญญาณระดับ (Step) เอาต์พุตที่ได้ก็จะมีลักษณะเป็นสัญญาณระดับ เช่นกัน โดยลักษณะของกราฟแสดงอินพุตและเอาต์พุตจะมีสัดส่วนที่แน่นอนค่าหนึ่ง

รูปที่ 2.3 ผลตอบสนองของตัวควบคุมแบบพี

ในทางปฏิบัติตัวควบคุมแบบพีนี้ มีลักษณะเหมือนกับเครื่องขยายสัญญาณรูปแบบหนึ่ง ซึ่งอาจจะเป็นในลักษณะของอุปกรณ์ไฟฟ้า หรืออาจเป็นเครื่องขยายสัญญาณเชิงกล และได้ฟังก์ชันถ่ายโอนระบบเปิดเป็นสมการ (2.8)

$$G_o(s) = K_p G_p(s) \tag{2.8}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $G_p(s)$ เป็นฟังก์ชั่นถ่ายโอนของระบบที่ต้องการควบคุม

ข้อเสียของระบบควบคุมแบบพี คือไม่มีการเพิ่มอินทิเกรตหรือ 1/s ในส่วนฟอร์เวิร์ดพาร์ท (Forward path) ซึ่งหมายความว่าถ้าระบบเป็นชนิด (Type) 0 ตัวควบคุมจะไม่เปลี่ยนแปลงชนิด ทำให้ระบบยังคงเป็นชนิดเดิม เกิดความผิดพลาดที่สภาพคงตัว (Steady state error) และมีฟังก์ชันถ่ายโอนเป็น

$$G(s) = \frac{K_p G_p(s)}{1+K_p G_p(s)} \tag{2.9}$$

ดังนั้นสมการคุณลักษณะจึงเป็น $(1 + K_p G_p(s))$ และมีรากเปลี่ยนไปตามค่าของ K_p

2.2.2 ระบบควบคุมแบบไอ

ในระบบควบคุมแบบไอ เอาต์พุตของตัวควบคุมจะเป็นสัดส่วนกับการอินทิเกรตสัญญาณของค่าผิดพลาดเทียบเวลา ดังสมการ (2.10)

$$u(t) = K_i \int_0^t e(t)dt \tag{2.10}$$

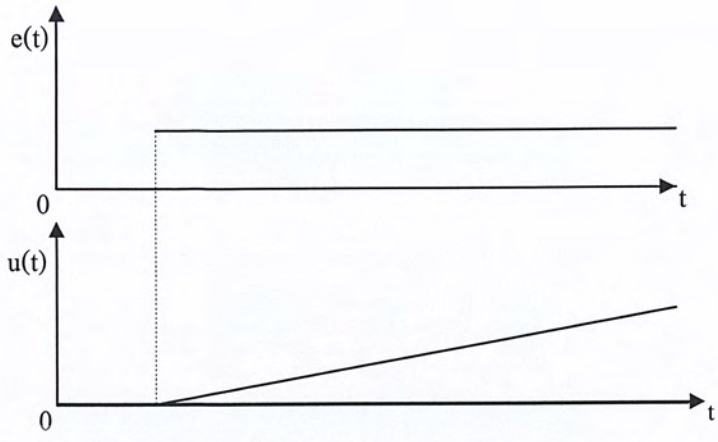
เมื่อ K_I เป็นค่าคงที่เรียกว่า อัตราขยายแบบอินทิเกรต (Integral Gain) ซึ่งจะมีหน่วยเป็น sec^{-1}รูปที่ 2.4 แสดงลักษณะการตอบสนองของตัวควบคุมแบบไอ เมื่อได้รับสัญญาณอินพุตแบบระดับ ค่า U(t) ณ เวลา t ใดๆ หมายถึงพื้นที่ใต้กราฟของสัญญาณอินพุต จากเวลา 0 ถึง t คูณด้วยอัตราขยายอินทิเกรต ดังนั้นเนื่องจากเริ่มการมีสัญญาณอินพุตแบบระดับ เอาต์พุตที่ออกจากตัวควบคุมจึงมีค่ามากขึ้นเรื่อยๆด้วยอัตราคงที่ ในลักษณะรูปสัญญาณความชัน (Ramp) จึงได้ฟังก์ชันถ่ายโอนตัวควบคุมแบบไอเป็นดังสมการที่ (2.11)

$$G_c(s) = \frac{U(s)}{E(s)} = \frac{K_i}{S} \tag{2.11}$$

ทำให้ระบบควบคุมแบบไอที่แสดงในรูปที่ 2.5 มีฟังก์ชันถ่ายโอนฟอร์เวิร์ด เป็น $\frac{K_i}{s} G_p(s)$ และมีฟังก์ชันถ่ายโอนวงปิดดังสมการ (2.12)

$$G(s) = \frac{\frac{K_i}{s} G_p(s)}{1+\frac{K_i}{s} G_p(s)} \tag{2.12}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ผลตอบสนองของตัวควบคุมแบบไอ

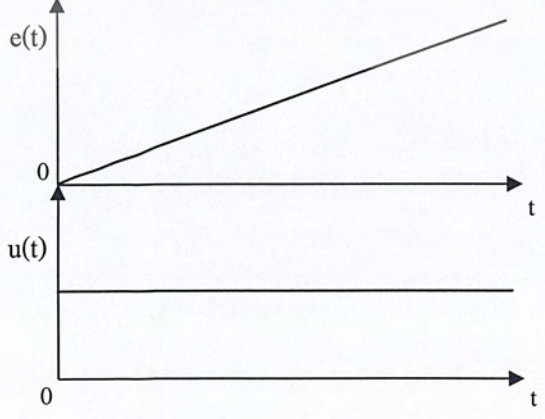
ดังนั้นสามารถพิจารณาถึงข้อได้เปรียบของการควบคุมแบบไอได้จากฟังก์ชันถ่ายโอน ซึ่งจะเห็นได้ว่าระบบควบคุมแบบไอ จะเพิ่มชนิดของระบบขึ้นมา 1 ระดับ ซึ่งทำให้ค่าความผิดพลาดที่สภาพคงตัวเป็นศูนย์เทียบต่อสัญญาณอินพุตแบบระดับ อย่างไรก็ตามการเพิ่มโพลที่ s = 0 ในระบบวงเปิดและไม่มีการเพิ่มซีโร่ให้ระบบควบคุม ทำให้ความแตกต่างระหว่างโพล (n) และซีโร่ (m) เพิ่มขึ้นอีก 1 ซึ่งมีผลให้มุมของอะซิมโทต (Asymptote angles) ของเส้นทางรากลดลง และจุดตัดเคลื่อนไปทางขวาของระนาบ s (s-plane) มากขึ้น มีผลทำให้เสถียรภาพของระบบลดลง

2.2.3 ระบบควบคุมแบบดี

ระบบควบคุมแบบดีนี้ เอาต์พุตของตัวควบคุมจะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงอินพุต ความผิดพลาดเทียบต่อเวลา นั่นคือ

$$u(t) = K_d \frac{de(t)}{dt} \tag{2.13}$$

เมื่อ K_d คือ อัตราขยายอนุพันธ์ (Derivative gain) และมีหน่วยเป็นวินาที



รูปที่ 2.5 ผลตอบสนองของตัวควบคุมแบบดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 แสดงผลตอบสนองของระบบควบคุมแบบดี เมื่อสัญญาณอินพุตเป็นสัญญาณแบบความชัน เห็นว่าตัวควบคุมแบบดีจะให้สัญญาณส่งออกจากตัวควบคุม มีค่ามากเมื่อค่าอินพุตความผิดพลาดมีแนวโน้มเพิ่มขึ้นในอัตราที่สูงก่อนที่ความผิดพลาดจะเกิดขึ้นจริงๆ

อย่างไรก็ตาม หากความผิดพลาดมีค่าคงที่ก็จะไม่มีการสะสมค่าความผิดพลาด แม้ว่าค่าความผิดพลาดจะมีมากก็ตาม ทำให้การควบคุมแบบดีนี้ ไม่เปลี่ยนแปลงต่อค่าความผิดพลาดที่คงที่หรือมีการเปลี่ยนแปลงอย่างช้าๆ ดังนั้นการควบคุมแบบนี้จึงไม่นิยมใช้ตามลำพัง แต่มักใช้ร่วมกับการควบคุมแบบอื่น โดยที่ฟังก์ชันถ่ายโอนตัวควบคุมแบบดี จะเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_d \tag{2.14}$$

ดังนั้น สำหรับระบบควบคุมแบบดี จะได้ฟังก์ชันถ่ายโอนระบบวงปิดเป็น

$$G(s) = \frac{K_d s G_p(s)}{1 + K_d s G_p(s)} \tag{2.15}$$

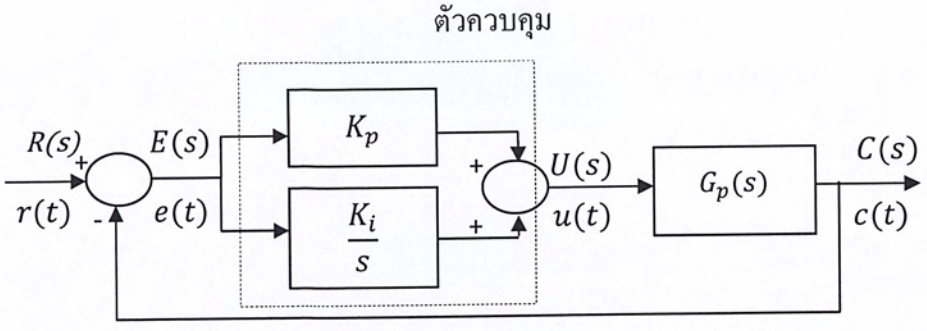
ถ้าหากว่าระบบเป็นแบบชนิด 1 หรือสูงกว่า การควบคุมแบบดีจะลดเทอม s ของพหุนามในส่วนของฟังก์ชันถ่ายโอนของฟอร์เวิร์ดพาร์ทลง ซึ่งจะไปลดชนิดของระบบลง 1 อย่างไรก็ตาม ดังที่ได้กล่าวไว้ก่อนหน้านี้แล้วว่า การควบคุมแบบดีนี้จะไม่ใช้เพียงลำพังแต่จะใช้ร่วมกับตัวควบคุมแบบอื่น เพราะเมื่อใช้การควบคุมแบบดี ทำให้เพิ่มความเร็วในการตอบสนองของระบบต่อความผิดพลาดที่เกิดขึ้น

ในทางปฏิบัติการนำการควบคุมดีไปใช้นั้นค่อนข้างจะลำบาก ดังนั้น ในทางปฏิบัติทั่วไปจะเป็นการประมาณการควบคุมดี โดยใช้ตัวชดเชยแบบมุมนำ (Lead compensator)

2.2.4 ระบบควบคุมแบบพีไอ

การที่ระบบควบคุมมีความเสถียรสัมพัทธ์ลดลง เมื่อใช้การควบคุมแบบไอ สามารถที่จะแก้ไขได้ในระดับหนึ่งโดยการใช้การควบคุมแบบพีร่วมกับแบบไอ ซึ่งลักษณะการควบคุมเป็นไปตามรูปที่ 2.6

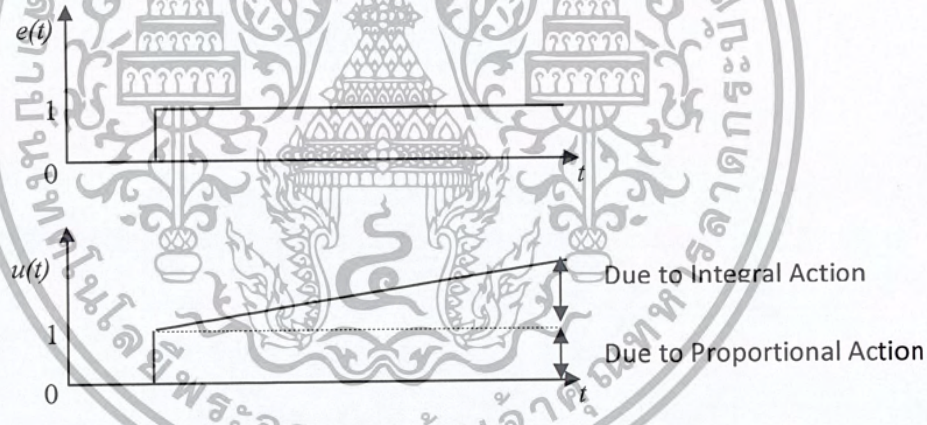
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 บล็อกไดอะแกรมตัวควบคุมแบบพีไอ

สำหรับระบบควบคุมดังกล่าวมีมีเอาต์พุตของตัวควบคุมเป็น

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt \tag{2.16}$$



รูปที่ 2.7 ผลตอบสนองตัวควบคุมแบบพีไอ

จากรูปที่ 2.7 แสดงเอาต์พุตของตัวควบคุมที่ได้รับเมื่อมีสัญญาณอินพุตเป็นแบบระดับ จากสมการ (2.15) ฟังก์ชันถ่ายโอนของตัวควบคุมแบบพีไอ เป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s}$$

$$= \frac{sK_p + K_i}{s}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= K_p \frac{(s+\frac{K_i}{K_p})}{s} \tag{2.17}$$

นิยามค่าเวลาคงตัวอินทิกรัล(Integral time constant) เป็น

$$T_i = \frac{K_p}{K_i} \tag{2.18}$$

ดังนั้นจะได้

$$G_c(s) = \frac{K_p[s+(1/T_i)]}{s} \tag{2.19}$$

และจะทำให้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$G_o(s) = G_c(s)G_p(s) \tag{2.20}$$

$$= \frac{K_p[s + (1/T_i)G_p(s)]}{s}$$

จะเห็นว่ามีซีโร่ที่ $s = -1/T_i$ และโพลที่ $s = 0$ เพิ่มให้ฟังก์ชันถ่ายโอนระบบเมื่อใช้การควบคุมแบบพีไอ การเพิ่มตัวประกอบ s เข้าถับพหุนามส่วนถ่ายโอน วงเปิดเป็นการเพิ่ม ชนิดระบบขึ้นไป 1 จึงทำให้ระบบควบคุมแบบพีไอนี้ไม่มีความผิดพลาดที่สภาพคงตัว สำหรับอินพุตแบบระดับ นอกจากนั้นการเพิ่มซีโร่ให้ระบบไปพร้อมๆกันก็ทำให้ความแตกต่างระหว่างโพล (n) กับซีโร่ (m) มีค่าคงที่ ดังนั้นมุมของเส้นอะซิมโทดสำหรับเส้นทางรากมีค่าคงเดิม แต่จุดตัดของเส้นอะซิมโทดบนแกนจริง (Centroid) จะเคลื่อนเข้าหาจุดกำเนิดมากขึ้นเนื่องจาก

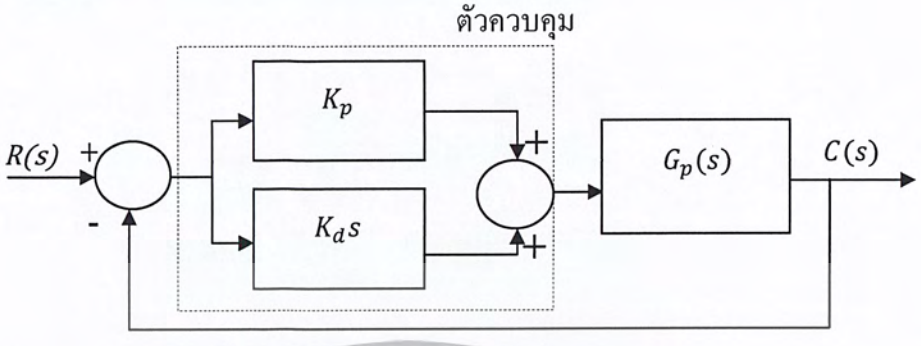
จุดตัดของเส้นอะซิมโทดบนแกนจริง = (ผลรวมของโพล – ผลรวมของซีโร่) / (n-m)

ดังนั้นจุดตัดของเส้นอะซิมโทดจะเปลี่ยนไปเท่ากับ $-(\frac{1}{T_i})/(n-m)$ ส่งผลให้ความเสถียรภาพของระบบลดลง อย่างไรก็ตามการลดลงของความเสถียรสัมพัทธ์นี้จะน้อยกว่าการที่ใช้การควบคุมแบบไอเพียงอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 ระบบควบคุมแบบพีดี

ถ้าการควบคุมแบบดีใช้ร่วมกับการควบคุมแบบพี ดังที่แสดงในรูป 2.8 ได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น



รูปที่ 2.8 บล็อกไดอะแกรมตัวควบคุมแบบพีดี

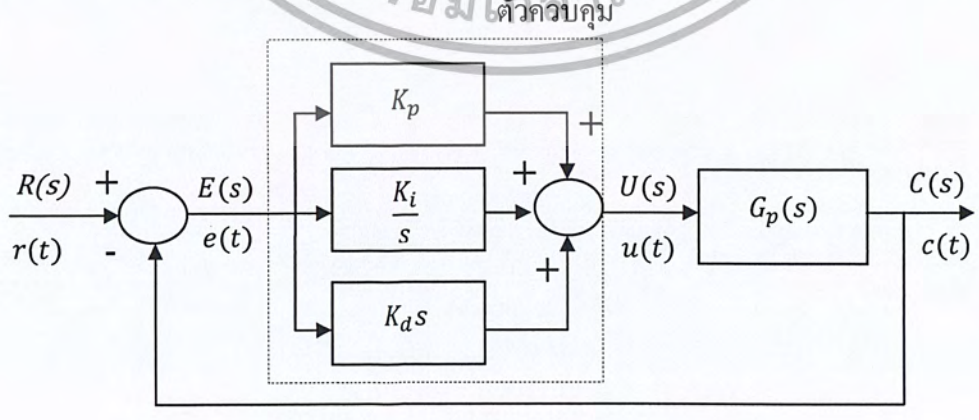
$$G_o(s) = \left(K_p + K_d s\right) G_p(s) \tag{2.21}$$

$$G_o(s) = (1 + T_d s) G_p(s)$$

เมื่อ $T_d = \frac{K_p}{K_d}$ คือค่าเวลาคงตัวอนุพันธ์ (Derivative time constant) ซึ่งในการควบคุมแบบนี้ จะมีซีโร่เพิ่มขึ้นที่ $s = -T_d$ และจะเห็นว่าไม่มีการเปลี่ยนแปลงชนิดของระบบ ทำให้ไม่มีการเปลี่ยนแปลงค่าความผิดพลาดที่สภาวะคงตัว

2.2.6 ระบบควบคุมแบบพีไอดี

การควบคุมโดยใช้แบบพีรวมกับแบบไอและแบบดี หรือที่เรียกว่า การควบคุมแบบ 3 เทอม (Three-term control) ระบบจะมีลักษณะตามรูป 2.11 เมื่อรับอินพุตเป็นความผิดพลาด e(t) ทำให้เอาต์พุตของตัวควบคุมเป็นดังสมการ (2.23)



รูปที่ 2.9 บล็อกไดอะแกรมตัวควบคุมแบบพีไอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันถ่ายโอนของระบบควบคุมจะเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \tag{2.22}$$

ซึ่งสามารถจัดรูปได้เป็น

$$G_c(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \tag{2.23}$$

ฟังก์ชันถ่ายโอนระบบแบบเปิดของระบบควบคุมพีไอดี ที่แสดงในรูป 2.9 จะเป็น

$$G_o(s) = G_c(s)G_p(s) = K_p(1 + \frac{1}{T_i s} + T_d s)$$

$$G_o(s) = \frac{K_p(T_i T_d s^2 + T_i s + 1)}{T_i s} G_p(s) \tag{2.24}$$

ดังนั้นการควบคุมแบบพีไอดี จะเพิ่มจำนวนซีโรให้กับระบบเท่ากับ 2 ตัว และเพิ่มโพล 1 ตัว และทำให้ชนิดระบบเพิ่มขึ้น 1

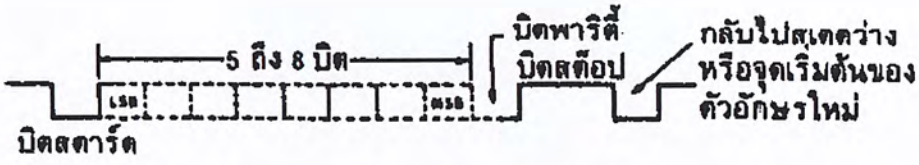
2.3 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม

การสื่อสารแบบอนุกรมแบ่งออกเป็น 2 แบบ คือ แบบซิงโครนัสและการสื่อสารแบบอะซิงโครนัสและการสื่อสารแบบซิงโครนัส

2.3.1 การสื่อสารแบบอะซิงโครนัส

การสื่อสารแบบนี้ประกอบด้วยบิตเริ่มต้นหรือบิตเริ่มต้น (Start bit) และบิตสิ้นสุดหรือบิตหยุด (Stop bit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 รูปแบบการสื่อสารแบบอะซิงโครนัส

ขณะที่สถานะของการส่งเป็นแบบว่าง หรือ ไอเดิล (Idle) คือยังไม่มีสัญญาณที่ส่งออกมา แต่จะมีสัญญาณ หรือมีแรงดันตลอดเวลาเพื่อความแน่ใจว่าฝ่ายรับยังติดต่ออยู่กับฝ่ายส่งฝ่ายส่งจะเริ่มส่งข้อมูลบอกจุด เริ่มต้น สัญญาณของอะซิงโครนัสจะเป็น "0" หนึ่งช่วงสัญญาณนาฬิกา บิตนี้เรียกว่าบิตสตาร์ต ข้อมูล 1 ตัวอักษรที่ตามหลังบิตสตาร์ตจะมีขนาดตั้งแต่ 5 บิต จนถึง8 บิต โดยอักขระนี้ส่วนมากจะนิยมใช้ รหัสแอสกี (ASCII code)

แรกเริ่มที่เดียวของการส่งข้อมูล จะส่งข้อมูลจะส่งรหัสโบดอต (Baudot code) ซึ่งใช้ 5 บิต ในการแทนอักขระ 1 ตัว ส่วนที่ตามหลังข้อมูลก็จะเป็นบิตพาริตี้ ซึ่งจะอาจจะใช้หรือไม่ใช้ก็ได้บิตพาริตี้จะทำหน้าที่เป็นตัวตรวจสอบ ความถูกต้องของสัญญาณที่ได้รับ บิตพาริตี้อาจจะเป็นแบบคู่ (Even) หรือแบบคี่ (Odd) ก็ได้ หมายความว่า ถ้าหากเป็นพาริตี้คู่ จำนวนบิตที่เป็น "1" ในช่วงบิตข้อมูลกับบิตพาริตี้รวมกันแล้วต้องเป็นเลขคู่ผู้ส่งข้อมูล จะทำหน้าที่ตรวจสอบข้อมูลแล้วใส่บิตพาริตี้เอง

ฝ่ายรับ เมื่อรับสัญญาณแล้วก็จะต้องตรวจสอบดูว่าเป็นจริงดังสถานการณ์ที่ตั้งไว้หรือไม่ หากผิดพลาด ก็หมายความ ว่า สัญญาณที่รับนั้นผิดพลาดไปจากสถานีที่ส่งออกมาทั้งนี้ทั้งนั้นจะต้องผิดเป็นจำนวนคี่ เท่านั้นคือ ผิดไป 1 บิต 3 บิต หรือ 5 บิตพร้อมกัน จึงจะตรวจสอบได้ว่าผิดเป็นจำนวนคู่ผลรวมของ จำนวนบิตที่เป็น "1" ก็ยังเป็นคู่อยู่ดี

ทั้งนี้ทั้งนั้นไม่ได้หมายความว่า พาริตี้จะตรวจสอบการผิดพลาดเป็นจำนวนคู่ได้ความจริงแล้ว สามารถตรวจสอบความผิดพลาด ได้เหมือนพาริตี้คู่แต่แทนที่จะตรวจสอบดูว่าสัญญาณ ที่รับเข้ามามีจำนวนคู่ ก็ตรวจสอบดูว่ามีจำนวนคี่หรือเปล่า อย่างไรก็ตาม โอกาสที่จะผิดพลาดเป็น 2,4,6 หรือ 8 บิตพร้อมกันมีน้อยมาก

ย้อนกลับมาดูสัญญาณอะซิงโครนัสใหม่ หลังจากบิตพาริตี้แล้วจะต้องมีบิตหยุด (Bit stop) ซึ่งเป็น "1" ความกว้าง ของบิตหยุดอาจจะเป็น 1,1.5 หรือ 2 พัลส์ของสัญญาณนาฬิกา ซึ่งแล้วแต่ผู้รับและผู้ส่งจะตกลงใช้กันเอง

การเริ่มใช้พอร์ตอนุกรมจึงจำเป็นต้องตั้งค่าต่างๆสำหรับการสื่อสาร ซึ่งมีดังต่อไปนี้ คือ

1.ความเร็วของการส่ง

2.ความยาวของรหัส 1 อักขระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.บิตตรวจสอบ (Bit check)

4.จำนวนบิตหยุด

ซึ่งโครงงานนี้เลือกใช้การสื่อสารแบบอะซิงโครนัสแบบไปกลับ ด้วยความเร็ว 76,800 บิตต่อวินาที ซึ่งเพียงพอต่อการใช้งานเชื่อมต่อข้อมูลระหว่างคอนโทรลเลอร์กับคอมพิวเตอร์

2.3.2 การสื่อสารข้อมูลแบบซิงโครนัส

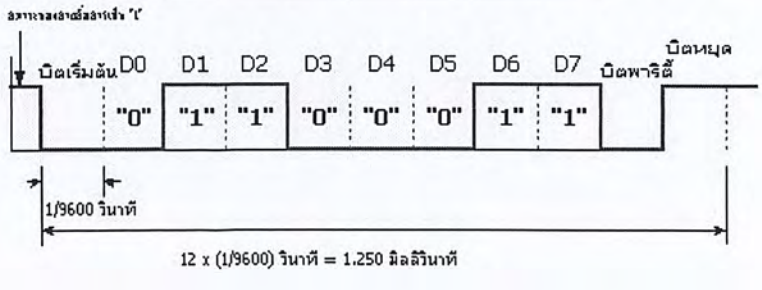
ข้อแตกต่างระหว่างวงจรการส่งข้อมูลอนุกรมแบบซิงโครนัสและแบบอะซิงโครนัสก็คือความต่อเนื่องของข้อมูลที่ส่งในแบบซิงโครนัส ข้อมูลที่ส่งออกมาเป็นแบบต่อเนื่องไม่มีบิตหยุดหรือบิตหยุด หรือแม้กระทั่งบิตพาริตี้รูปแบบที่ใช้ในการส่งข้อมูลแบบซิงโครนัสจึงแตกต่างไปจากการส่งข้อมูล แบบอะซิงโครนัส เช่น รูปแบบของบริษัทไอบีเอ็ม ใช้รูปแบบไบซิงก์ (Binary synchronous transmission)

การซิงโครไนซ์จะทำในระดับอักขระซึ่งหมายความว่าอักขระแต่ละตัวมีขอบเขตที่แน่นอนแต่ละอักขระไม่มีบิตเริ่ม หรือบิตหยุดเหมือนอะซิงโครนัส การซิงโครนัสจะกระทำที่จุดเริ่มต้นของการส่งข้อมูล สถานีส่งจะส่งสัญญาณที่เรียกว่า ตัวอักษรนำ (Leading pad character) ไปยังสถานีรับก่อนที่จะเริ่มส่งข้อมูล ตัวอักษรนำจะประกอบด้วย "0" และ "1" สลับกัน เพื่อให้สถานีรับจัดสัญญาณนาฬิกาให้ตรงกันก่อนส่ง ข้อมูลก็จะมีการส่งอักขระที่เรียกว่า Synchronous ตามหลังตัวอักษรนำออกมาสถานีส่งจำเป็นจะต้องบอกความยาว ของข้อมูลมาในกลุ่มนี้ และต้องบอกเครื่องหมายที่เป็นตัวบอกจุดเริ่มต้นของข้อมูลด้วย

ในการส่งข้อมูลแบบอนุกรมมีสิ่งที่สำคัญที่จะต้องพิจารณาคือ ความเร็วของข้อมูลในการส่งซึ่งเราเรียกว่า อัตราบิต(Bit rate) ตามที่กล่าวมา และกรณีที่ให้อัตราการเปลี่ยนแปลงของสัญญาณ 1 ครั้งต่อข้อมูล 1 บิต จะได้อัตราบิตเท่ากับอัตราโบด์

อัตราโบด์ที่ใช้ในการส่งข้อมูลทั่วไปคือ 110,150,300,1200,2400,4800 และ 9600 สมมติว่า ถ้าต้องการส่งข้อมูลด้วยอัตราโบด์ 9600 โบด์ ข้อมูลจะได้รับการส่งออกไปดังรูป 2.11

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	1	1	0



รูปที่ 2.11 ตัวอย่างการส่งข้อมูลขนาด 8 บิต แบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
119177

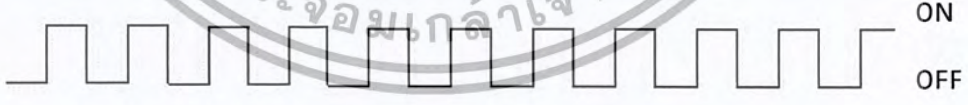
แสดงให้เห็นช่วงเวลาของการส่ง ซึ่งในช่วงเวลาของแต่ละบิตจะมีขนาดช่วงเวลาเท่ากับ 1/2400 เท่ากับ 0.000416 วินาที หรือ 416 ไมโครวินาที ดังนั้นถ้าต้องการส่งข้อมูลที่มีขนาด 8 บิต ก็จะใช้เวลาทั้งสิ้น 8x416 ไมโครวินาที หรือ เท่ากับ 3,2328 ไมโครวินาที ซึ่งเมื่อเทียบกับการส่งข้อมูลแบบขนาน จะใช้เวลาน้อยกว่า 1 ไมโครวินาที ด้วยเหตุนี้จึงมีการนำไปใช้งานที่ต้องการความเร็วสูง เช่น การส่งข้อมูลระหว่างคอมพิวเตอร์ เป็นต้น

2.4 การควบคุมมอเตอร์ไฟฟ้ากระแสตรงด้วยสัญญาณพีดับบิวเอ็ม

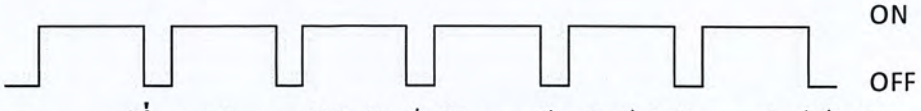
การออกแบบระบบควบคุมที่จะต้องกำหนดตำแหน่ง จะต้องอาศัยสัญญาณอ้างอิงในการกำหนดตำแหน่งที่ต้องการกับสัญญาณการป้อนกลับจากอุปกรณ์วัดตำแหน่ง ความแตกต่างระหว่างสัญญาณที่กำหนดกับสัญญาณป้อนกลับจากอุปกรณ์ที่วัดตำแหน่งในขณะนั้นก็คือสัญญาณคาดเคลื่อนซึ่งสัญญาณที่ได้ทั้งสองสัญญาณนี้ก็จะถูกป้อนเข้าตัวควบคุม หลังจากนั้นตัวควบคุมก็จะทำหน้าที่ประมวลผลแล้วส่งสัญญาณควบคุมออกไปขับมอเตอร์ให้หมุนไปยังตำแหน่งที่ต้องการได้ ซึ่งตำแหน่งที่ต้องการมีค่าเท่ากับตำแหน่งที่วัดได้จากการอ่านตำแหน่งของอุปกรณ์ป้อนกลับจนทำให้ผลต่างของสัญญาณมีค่าเท่ากับศูนย์ จะทำให้ไม่มีสัญญาณออกมากจากตัวควบคุมส่งผลให้มอเตอร์หยุดทำงาน ซึ่งโครงการนี้จึงเลือกใช้การควบคุมมอเตอร์กระแสตรงด้วยพีดับบิวเอ็ม

การทำงานของสัญญาณพีดับบิวเอ็มที่ได้จากไมโครคอนโทรเลอร์จะใช้ในการควบคุมความเร็วของมอเตอร์กระแสตรง ซึ่งสามารถแสดงได้ดังรูป 2.12, 2.13 และรูป 2.14

รูปที่ 2.12 สัญญาณพีดับบิวเอ็มมีความกว้างพัลส์เท่ากับ 10 เปอร์เซ็นต์



รูปที่ 2.13 สัญญาณพีดับบิวเอ็มมีความกว้างพัลส์เท่ากับ 50 เปอร์เซ็นต์



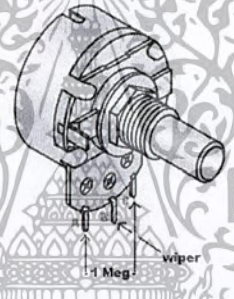
รูปที่ 2.14 สัญญาณพีดับบิวเอ็มมีความกว้างพัลส์เท่ากับ 90 เปอร์เซ็นต์

สัญญาณมีรอบการทำงานเป็น 10, 50 และ 90 เปอร์เซ็นต์ ตามลำดับซึ่งสอดคล้องกับการขับมอเตอร์ด้วยระดับแรงดันไฟฟ้ากระแสตรงเช่นกัน ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 หลักการทำงานของโพเท็นทิโอมิเตอร์

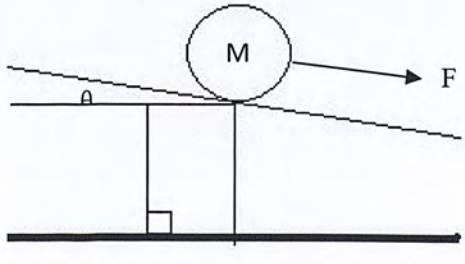
โพเท็นทิโอมิเตอร์(Potentiometer) เป็นตัวต้านทานทางไฟฟ้าชนิดหนึ่งที่มีสามขั้วและสามารถปรับค่าได้ โครงสร้างทั่วไปของโพเท็นทิโอมิเตอร์แสดงได้ดังรูปที่ 2.15 ประกอบด้วยหน้าสัมผัส (Sliding contact) ที่สามารถเลื่อนขึ้นเลื่อนลงผ่านตามความยาวของลวดความต้านทานได้ หน้าสัมผัสนี้จะถูกเชื่อมต่อกับแท่งวัดซึ่งต่ออยู่กับวัตถุที่ต้องการตรวจสอบระยะขจัด ทั่วไปแล้วรูปแบบของหน้าสัมผัสของโพเท็นทิโอมิเตอร์มักมีการออกแบบแตกต่างกันออกไปซึ่งขึ้นอยู่กับปัจจัยและเงื่อนไขในการนำไปประยุกต์ใช้งาน แต่ไม่ว่าจะมีรูปแบบแบบไหน วัสดุที่นำมาใช้ทำหน้าสัมผัสมักนิยมทำมาจากโลหะทองแดงผสมเป็นส่วนมาก ที่เป็นเช่นนี้เพราะว่าโลหะทองแดงผสมมีความยืดหยุ่นตัวสูง จึงทำให้นำขึ้นไปสู่รูปแบบทรงต่าง ๆ ได้ง่าย รวมทั้งยังมีคุณสมบัติในการนำไฟฟ้าได้ดีอีกด้วย ส่วนลวดความต้านทานของโพเท็นทิโอมอเตอร์นั้นนิยมทำมาจากลวดนิเกิลหรือลวดแพลทินัม นำมาพันรอบแกนที่เป็นฉนวนทางไฟฟ้า



รูปที่ 2.15 โพเท็นทิโอมิเตอร์

2.6 แบบจำลองทางคณิตศาสตร์ของระบบควบคุมลูกบอลบนระนาบ

ระบบควบคุมลูกบอลบนระนาบประกอบไปด้วยลูกบอล โดยระนาบสามารถเคลื่อนที่ได้ 2 ทิศทางคือ แนวตั้งและแนวนอน ซึ่งลูกบอลจะสามารถเคลื่อนที่ได้อย่างอิสระบนระนาบ โดยสามารถอธิบายการเคลื่อนที่ของลูกบอลโดยมองแยกเป็นที่ละแนวแกน ดังรูปที่ 2.16



รูปที่ 2.16 แนวการเคลื่อนที่ของลูกบอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกฎข้อที่ 2 ของนิวตัน ในสมการ (2.25)

$$F = ma \tag{2.25}$$

เมื่อแรง F เกิดจากแรงโน้มถ่วงของโลกที่กระทำต่อลูกบอลบนพื้นลาดเอียงทำมุม θ จากแนวแกน ดังรูปที่ 2.16 เมื่อลูกบอลเคลื่อนที่โดยไม่คิดแรงเสียดทาน สามารถอธิบายได้ดังนี้

$$F = mg\ sin\ \theta(t) \tag{2.26}$$

เมื่อ F คือ แรงที่กระทำต่อลูกบอลในขณะเคลื่อนที่ (นิวตัน)

M คือ มวลของลูกบอล (กิโลกรัม)

g คือ แรงโน้มถ่วงของโลกที่กระทำต่อลูกบอล (เมตรต่อวินาที)

จากสมการ (2.25) และ (2.26) จะได้ความสัมพันธ์เป็น

$$a = \mathrm{g}\sin\theta(\mathrm{t}) \tag{2.27}$$

และเมื่อลูกบอลเกิดการเคลื่อนที่ จะทำให้เกิดความเร่ง a ขึ้น ซึ่งความเร่งของลูกบอลนั้นจะมีความสัมพันธ์กับตำแหน่งของลูกบอล $Z(t)$ ในแนวแกนที่พิจารณา ดังสมการ (2.28)

$$a = \frac{d^2Z(t)}{dt^2} \tag{2.28}$$

จากสมการ (2.27) และ (2.28) จะได้ความสัมพันธ์ระหว่างตำแหน่งของลูกบอลและมุมของระนาบในแนวแกน $Z(t)$ ที่พิจารณา ดังสมการ (2.29)

$$\frac{d^2Z(t)}{dt^2} = \mathrm{g}\sin\theta(\mathrm{t}) \tag{2.29}$$

แต่เนื่องจากมุมที่ดำเนินการควบคุมนั้นมีค่าน้อยมาก จึงประมาณค่า $sin\theta(t)$ ได้เป็น $\theta(t)$

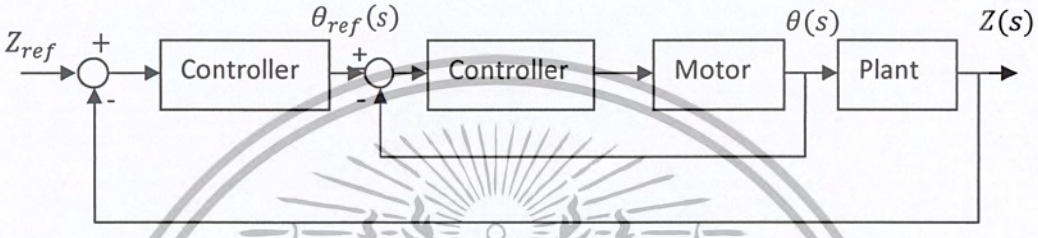
$$\frac{d^2Z(t)}{dt^2} = g\theta(\mathrm{t}) \tag{2.30}$$

เมื่อแปลงลาปลาซจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{Z(s)}{\theta(s)} = \frac{g}{s^2} \tag{2.31}$$

การควบคุมตำแหน่งลูกบอลบนระนาบแยกการควบคุมในสองแนวแกนอิสระจากกันในแนวแกน X และแนวแกน Y โดยในแต่ละแนวแกนอาศัยการควบคุมแบบ 2 วงปิด ดังรูปที่ 2.17 การควบคุมวงแรกเป็นการควบคุมมุมแผ่นระนาบ โดยมีมอเตอร์เป็นอุปกรณ์ทำให้มุมของระนาบเปลี่ยนแปลง ส่วนการควบคุมวงนอกเป็นการควบคุมตำแหน่งลูกบอล ซึ่งระบุตำแหน่งลูกบอลบนแผ่นระนาบได้โดยใช้กล้องดิจิตอลวีดิโอ และคำนวณมุมอ้างอิงที่แผ่นระนาบควรเป็น



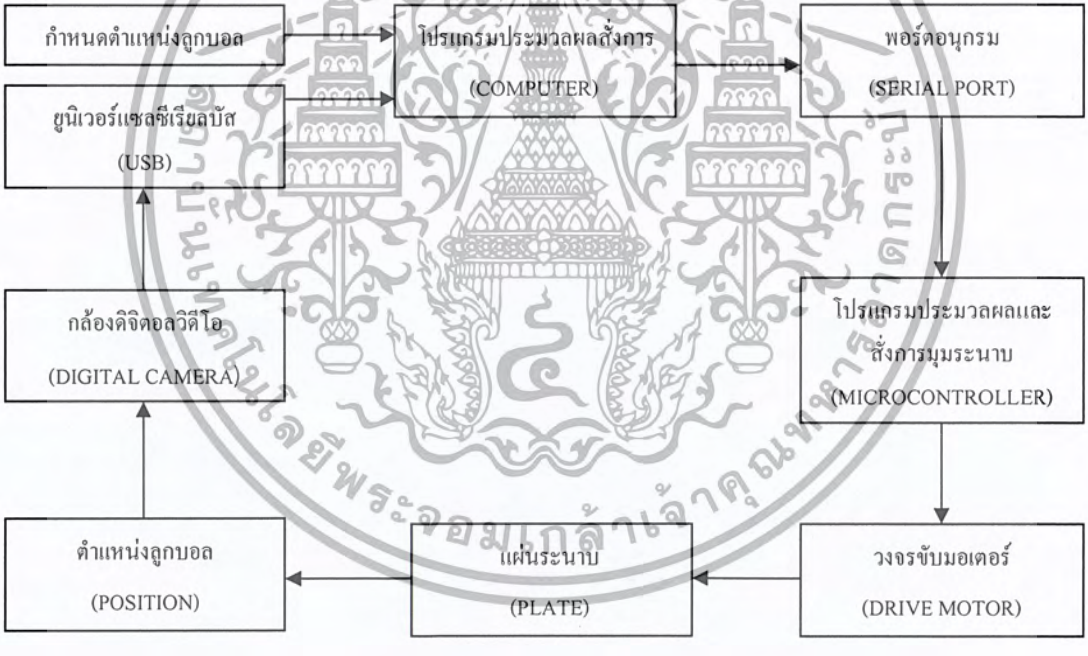
รูปที่ 2.17 บล็อกไดอะแกรมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างของระบบ

ในการออกแบบระบบควบคุมตำแหน่งลูกบอลบนระนาบนั้น จำเป็นจะต้องเข้าใจระบบโดยรวม แล้วจึงพิจารณาองค์ประกอบย่อย ๆ ทั้งส่วนโครงสร้างทางกายภาพและโปรแกรมที่ใช้ในการประมวลผลภาพและควบคุมลูกบอลบนระนาบ

ระบบการทำงานโดยรวมในการควบคุมลูกบอลบนแผ่นระนาบนี้สามารถสรุปได้ในรูปที่ 3.1 โดยเริ่มจากการกำหนดตำแหน่งของลูกบอลที่ต้องการไปยังคอมพิวเตอร์ เพื่อคำนวณหาสัญญาณควบคุมที่เหมาะสมในการควบคุมซึ่งจะถูกส่งผ่านพอร์ตอนุกรมไปยังไมโครคอนโทรลเลอร์ให้ควบคุมแผ่นระนาบ โดยผ่านวงจรขับมอเตอร์ซึ่งตำแหน่งลูกบอลจะถูกตรวจจับโดยกล้องดิจิตอลวิดีโอที่เชื่อมกับคอมพิวเตอร์เพื่อส่งข้อมูลผ่านยูนิเวอร์แซลซีเรียลบัสเป็นสัญญาณป้อนกลับในการประมวลผลต่อไป

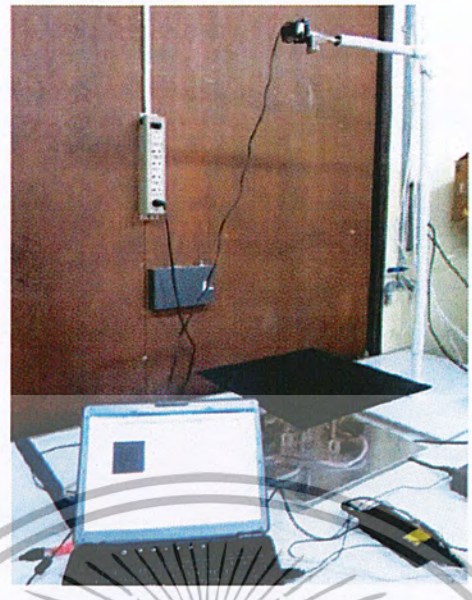


รูปที่ 3.1 หลักการทำงานของระบบควบคุมตำแหน่งลูกบอลบนระนาบ

3.1 ส่วนประกอบของโครงสร้างทางกายภาพ

โครงสร้างทางกายภาพของระบบแสดงดังรูปที่ 3.2 ซึ่งประกอบด้วยส่วนต่าง ๆ ได้แก่ แผ่นระนาบอลูมิเนียม โพเท็นทิโอมิเตอร์ มอเตอร์ วงจรขับมอเตอร์ กล้องดิจิตอล คอมพิวเตอร์ และไมโครคอนโทรเลอร์ โดยแต่ละส่วนมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



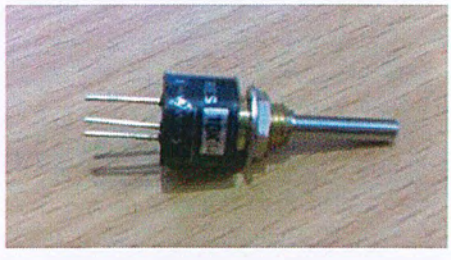
รูปที่ 3.2 โครงสร้างทางกายภาพของระบบ

3.1.1 แผ่นระนาบอลูมิเนียม

แผ่นระนาบอลูมิเนียม เป็นอุปกรณ์ที่ทำหน้าที่เป็นโครงสร้างหลักของระบบ โดยใช้แผ่นอลูมิเนียมแผ่นเรียบหนา 2 มิลลิเมตร ตัดเป็นแผ่นสี่เหลี่ยมจัตุรัสขนาด 400 x 400 มิลลิเมตร และมีแขนรองรับเพื่อให้สามารถปรับความเอียงทั้ง 2 แนวแกน และพ่นสีดำแบบไม่สะท้อนแสงให้สีตัดกับลูกบอลที่ใช้ซึ่งเป็นสีขาว เพื่อให้ง่ายต่อการประมวลผลภาพที่จะใช้งาน

3.1.2 โพเท็นทิโอมิเตอร์

โพเท็นทิโอมิเตอร์ เป็นตัวต้านที่สามารถปรับค่าได้จากการหมุนของโพเท็นทิโอมิเตอร์ ด้วยลักษณะนี้จึงสามารถนำมาวัดมุมเอียงของแผ่นระนาบโดยอาศัยกฎการแบ่งแรงดัน ซึ่งแรงดันที่เปลี่ยนแปลงนี้จะถูกส่งให้กับไมโครคอนโทรลเลอร์เพื่อประมวลผลและนำข้อมูลมาใช้ในส่วนของโปรแกรมควบคุมต่อไป โดยในโครงงานนี้เลือกใช้โพเท็นทิโอมิเตอร์ของบริษัท Japan Servo รุ่น VP12 ซึ่งมีค่าความต้านทานปรับค่าได้ในย่าน 0-10 กิโลโอห์ม แบบหมุน 1 รอบ ดังรูปที่ 3.3

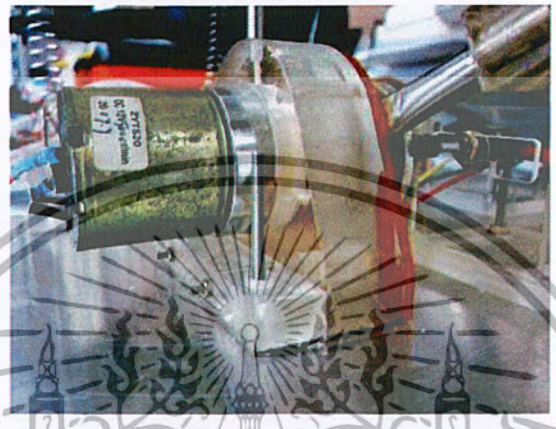


รูปที่ 3.3 โพเท็นทิโอมิเตอร์ขนาด 10 กิโลโอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 มอเตอร์ไฟฟ้ากระแสตรง

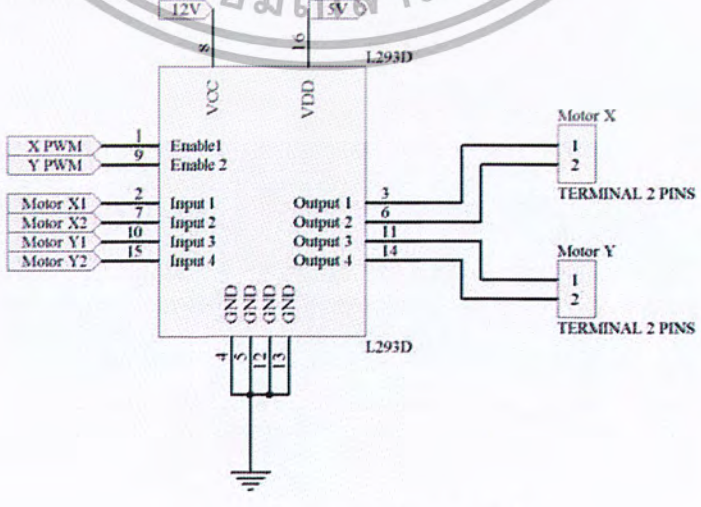
มอเตอร์ไฟฟ้ากระแสตรง เป็นอุปกรณ์ที่สำคัญอีกอุปกรณ์หนึ่ง ทำหน้าที่ในการควบคุมมุมของแผ่นระนาบ เพื่อกำหนดทิศทางของลูกบอล ซึ่งโครงงานนี้เลือกใช้มอเตอร์พิกัด 12 โวลต์ ความเร็วรอบ 5000 รอบต่อนาที โดยมีชุดทดทดความเร็วเหลือ 20 รอบต่อนาที แล้วจึงใช้ชุดเฟืองด้านนอกเพื่อให้โพเท็นทิโอมิเตอร์สามารถวัดมุมเอียงของแผ่นระนาบได้ ดังรูปที่ 3.4



รูปที่ 3.4 มอเตอร์ไฟฟ้ากระแสตรงติดตั้งกับแผ่นระนาบพร้อมต่อกับโพเท็นทิโอมิเตอร์

3.1.4 วงจรขับมอเตอร์

เนื่องจากการควบคุมแผ่นระนาบได้ใช้มอเตอร์ 2 ตัวในการควบคุม จึงเลือกใช้ไอซีเบอร์ L293D ซึ่งสามารถขับมอเตอร์แบบอิสระได้ 2 ตัว สามารถจ่ายแรงดันให้มอเตอร์สูงสุด 36 โวลต์ ซึ่งวงจรขับมอเตอร์สามารถจ่ายกระแสต่อเนื่องได้ถึง 600 มิลลิแอมป์ และทนกระแสสูงสุด 1.2 แอมป์ โดยสามารถสวิทช์สัญญาความถี่สูงสุดได้ถึง 5 กิโลเฮิรตซ์ จึงสามารถใช้ขับมอเตอร์ด้วยสัญญาณพัลส์วิทมอดดูเลชั่นได้ ในการใช้งานสามารถต่อวงจรได้ดังรูปที่ 3.5



รูปที่ 3.5 วงจรขับมอเตอร์

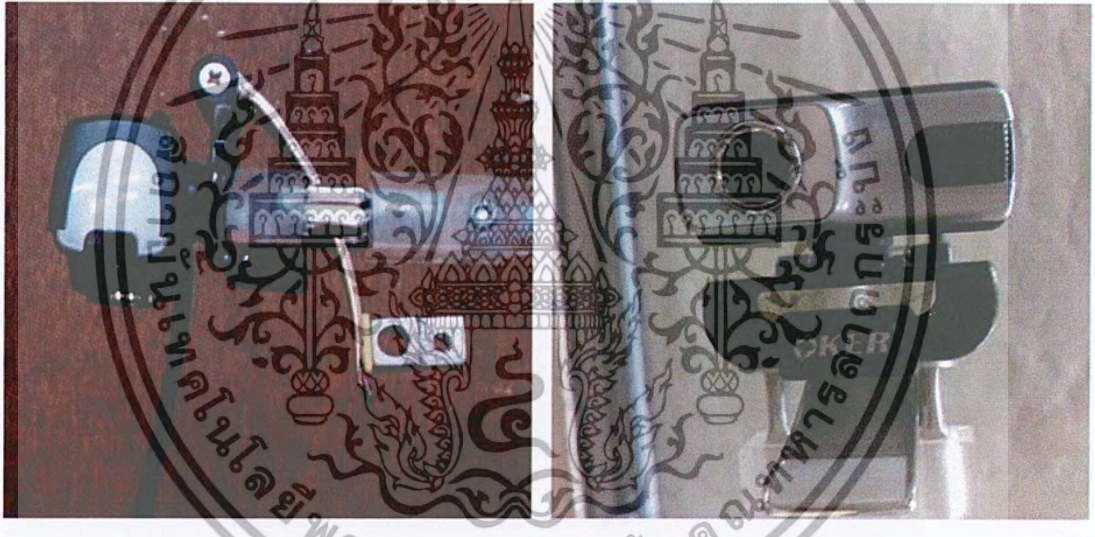
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 กล้องดิจิตอลวีดีโอ

กล้องดิจิตอลวีดีโอ เป็นอุปกรณ์สำคัญส่วนหนึ่งในโครงงานนี้ เนื่องจากใช้เป็นอุปกรณ์ที่ระบุตำแหน่งของลูกบอลบนระนาบ โดยส่งข้อมูลภาพให้แก่คอมพิวเตอร์ด้วยการเชื่อมต่อผ่านยูนิเวอร์แซลซีเรียลบัส

ในโครงงานนี้เลือกใช้กล้องดิจิตอลวีดีโอรุ่น OKER OE-052 ดังรูปที่ 3.6 ซึ่งมีคุณสมบัติดังนี้

- ความละเอียด : สูงสุด 2.0 ล้านพิกเซล
- อัตราการจับภาพ : สูงสุด 60 ภาพต่อวินาที
- อัตราสัญญาณรบกวนในการส่งข้อมูล : 48 เดซิเบล
- ระยะโฟกัส : 30 มิลลิเมตร
- เชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตยูนิเวอร์แซลซีเรียลบัส



(ก) ด้านข้าง (ข) ด้านหน้า

รูปที่ 3.6 กล้องดิจิตอลวีดีโอ

3.1.6 คอมพิวเตอร์

คอมพิวเตอร์เป็นอุปกรณ์หลักที่ทำหน้าที่รับข้อมูลจากกล้องดิจิตอลเพื่อประมวลผลและส่งข้อมูลผ่านพอร์ตอนุกรมให้กับไมโครคอนโทรลเลอร์ในการควบคุมตำแหน่งลูกบอลบนระนาบ

คอมพิวเตอร์ที่ใช้ในโครงงานนี้ จำเป็นต้องใช้คอมพิวเตอร์ที่มีความเร็วสูงเพียงพอในการประมวลผลของข้อมูลในทุกคาบการสุ่มสัญญาณที่มีค่าน้อย และเลือกใช้โปรแกรมวิชชวลเบสิกในการเขียนโปรแกรมบนคอมพิวเตอร์ในโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

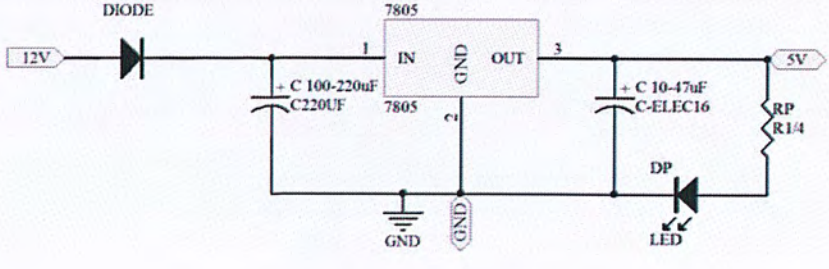
3.1.7 ไมโครคอนโทรลเลอร์

ในการควบคุมตำแหน่งของลูกบอลบนระนาบนี้ จะอาศัยไมโครคอนโทรลเลอร์ เป็นตัวควบคุมเพื่อให้แผ่นระนาบเอียงตามมุมอ้างอิงที่ต้องการ โดยไมโครคอนโทรลเลอร์จะรับค่าวัดมุมจริงจากโพเท็นทิโอมิเตอร์ และค่ารับมุมอ้างอิงจากคอมพิวเตอร์ แล้วประมวลผลเพื่อส่งสัญญาณแบบพัลวิตมอตดูเลชั่นไปขับมอเตอร์ไฟฟ้ากระแสตรง

ในโครงงานนี้เลือกใช้ไมโครคอนโทรลเลอร์ Atmega32 16PU ของบริษัท Atmel ลักษณะดังรูปที่ 3.7 เนื่องจากใช้งานง่าย ใช้ความถี่ของสัญญาณนาฬิกาได้สูงถึง 16 เมกะเฮิร์ทซ์ มีหน่วยความจำสำหรับเขียนโปรแกรมถึง 32 กิโลไบต์ อุปกรณ์ในการพัฒนามีราคาถูก และมีซอฟท์แวร์ในการพัฒนาโปรแกรมโดยสามารถดาวน์โหลดและใช้งานได้ฟรี ทำให้สะดวกต่อการประยุกต์ใช้งาน

รูปที่ 3.7 ไมโครคอนโทรลเลอร์ Atmega32 16PU

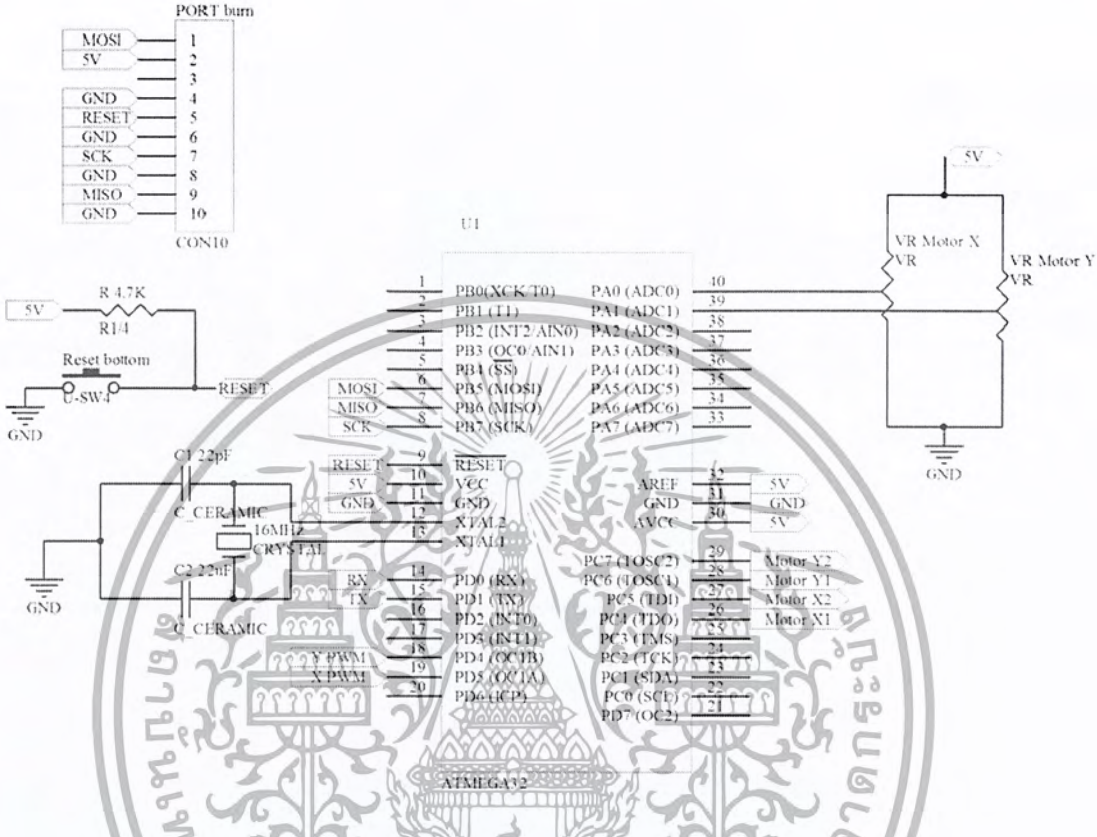
โดยไฟเลี้ยงที่ให้กับไมโครคอนโทรลเลอร์จะใช้ไอซีเบอร์ 7805 ซึ่งเป็นไอซีรักษาระดับแรงดัน สามารถควบคุมแรงดันในช่วง 5 – 24 โวลต์ เพียงพอต่อไฟเลี้ยงของไมโครคอนโทรลเลอร์ สามารถให้แรงดันไฟฟ้าคงที่ที่ 5 โวลต์ได้ ทนกระแสได้สูงสุดถึง 1.5 แอมป์



รูปที่ 3.8 วงจรไฟเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8 ซึ่งแสดงให้เห็นถึงวิธีการต่อวงจรรักษาระดับแรงดัน 7805 โดยใส่ตัวเก็บประจุเพื่อรักษาระดับสัญญาณเอาต์พุตในกรณีที่เกิดการแกว่งจากการกระชากไฟ ซึ่งวงจรเมื่อป้อนจะแปลงแรงดันอินพุต 12 โวลต์ ให้มีแรงดันเอาต์พุต 5 โวลต์ เพื่อส่งแรงดันไปยังวงจรอื่นต่อไป



รูปที่ 3.9 วงจรการใช้งานไมโครคอนโทรลเลอร์

จากรูปที่ 3.9 แสดงถึงการใช้งานไมโครคอนโทรลเลอร์ ซึ่งประกอบด้วยส่วนประกอบต่าง ๆ ดังนี้

ขา PB 5 -7 และขา Reset ใช้ในการเขียนโปรแกรมลงไมโครคอนโทรลเลอร์

ขา Reset ทำการต่อวงจรสวิตช์แบบพูลอัพ

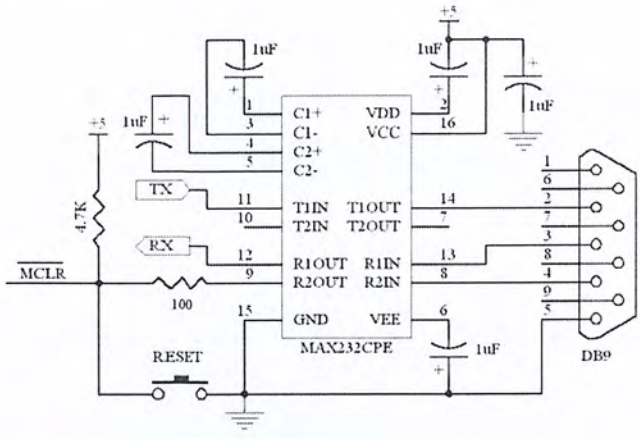
ขา PA 0 - 1 เป็นขาแปลงสัญญาณอนาล็อก/ดิจิตอลเพื่อใช้อ่านค่าระดับสัญญาณจากโพเท็นทิโอมิเตอร์

ขา PD 4 - 5 เป็นขาสัญญาณพัลส์วิตมอตดูเลชั่น โดยจะส่งสัญญาณพัลส์ไปยังวงจรขับมอเตอร์

ขา PC 4 – 7 เป็นขาสัญญาณลอจิก ต่อกับวงจรขับมอเตอร์ เพื่อควบคุมทิศทาง

ขา PD 0 – 1 เป็นขาสัญญาณการสื่อสารแบบ UART เพื่อใช้เชื่อมต่อการสื่อสารกับคอมพิวเตอร์แบบ RS232 โดยผ่านวงจรแปลงสัญญาณ ดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 วงจรแปลงสัญญาณ UART – RS232

3.2 การออกแบบโปรแกรม

3.2.1 โปรแกรมส่วนคอมพิวเตอร์

โปรแกรมในส่วนของคอมพิวเตอร์จะแบ่งออกเป็น 2 ส่วน โดยส่วนแรกเป็นการควบคุมตำแหน่งลูกบอลบนแผ่นระนาบ และส่วนที่สองเป็นโมดูลย่อยการหาตำแหน่งลูกบอลบนแผ่นระนาบ ซึ่งโปรแกรมในส่วนคอมพิวเตอร์จะทำงานทุก ๆ คาบการชักตัวอย่างที่ 50 มิลลิวินาที

การควบคุมตำแหน่งลูกบอลบนแผ่นระนาบ แสดงดังรูปที่ 3.11 เริ่มต้น คอมพิวเตอร์รับค่าตำแหน่งอ้างอิงของลูกบอลและโมดูลย่อยตรวจจับตำแหน่งลูกบอลบนแผ่นระนาบ จากนั้นคอมพิวเตอร์จะคำนวณค่าความผิดพลาดและอาศัยกฎการควบคุมแบบพีไอดีเพื่อคำนวณมุมอ้างอิงสำหรับสั่งงานแผ่นระนาบ ซึ่งมุมอ้างอิงที่ได้จากการคำนวณนั้นเกินย่านที่ตั้งไว้ ระบบจะเปลี่ยนมุมอ้างอิงให้เท่ากับย่านที่ตั้งโดยย่านที่ตั้งไว้ในที่นี้คือ ±5 องศา จากนั้นจึงส่งข้อมูลให้ไมโครคอนโทรลเลอร์เพื่อประมวลผลต่อไป โดยเก็บค่าที่จำเป็นสำหรับการทำงานแต่ละรอบทุก ๆ ครั้ง

โมดูลย่อยการหาตำแหน่งลูกบอล แสดงดังรูปที่ 3.12 โดยการตรวจจับภาพด้วยกล้องดิจิตอลวิดีโอนั้น เริ่มด้วยการเขียนโปรแกรมบันทึกภาพที่ได้จากกล้องให้เป็นไฟล์ภาพชนิดบิตแมป แล้วทำการเทรชโฮลภาพแต่ละพิกเซล ซึ่งโปรแกรมจะทำการตรวจสอบค่าสีทุกพิกเซลจากซ้ายไปขวาเริ่มจากด้านบน เมื่อมีการเปลี่ยนแปลงของค่าสีในพิกเซล โปรแกรมจะทำจุดเส้นขอบของวัตถุนั้นโดยทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะครบพิกเซลทั้งหมด จากนั้นนำจุดขอบของวัตถุมาจัดกลุ่มและคำนวณความสูง ความกว้าง และตำแหน่งของวัตถุโดยตีกรอบสี่เหลี่ยมของวัตถุที่มีพื้นที่มากที่สุดแล้วจึงส่งตำแหน่งวัตถุเข้าคอมพิวเตอร์ และมีการเก็บค่าที่จำเป็นสำหรับการทำงานทุก ๆ รอบเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

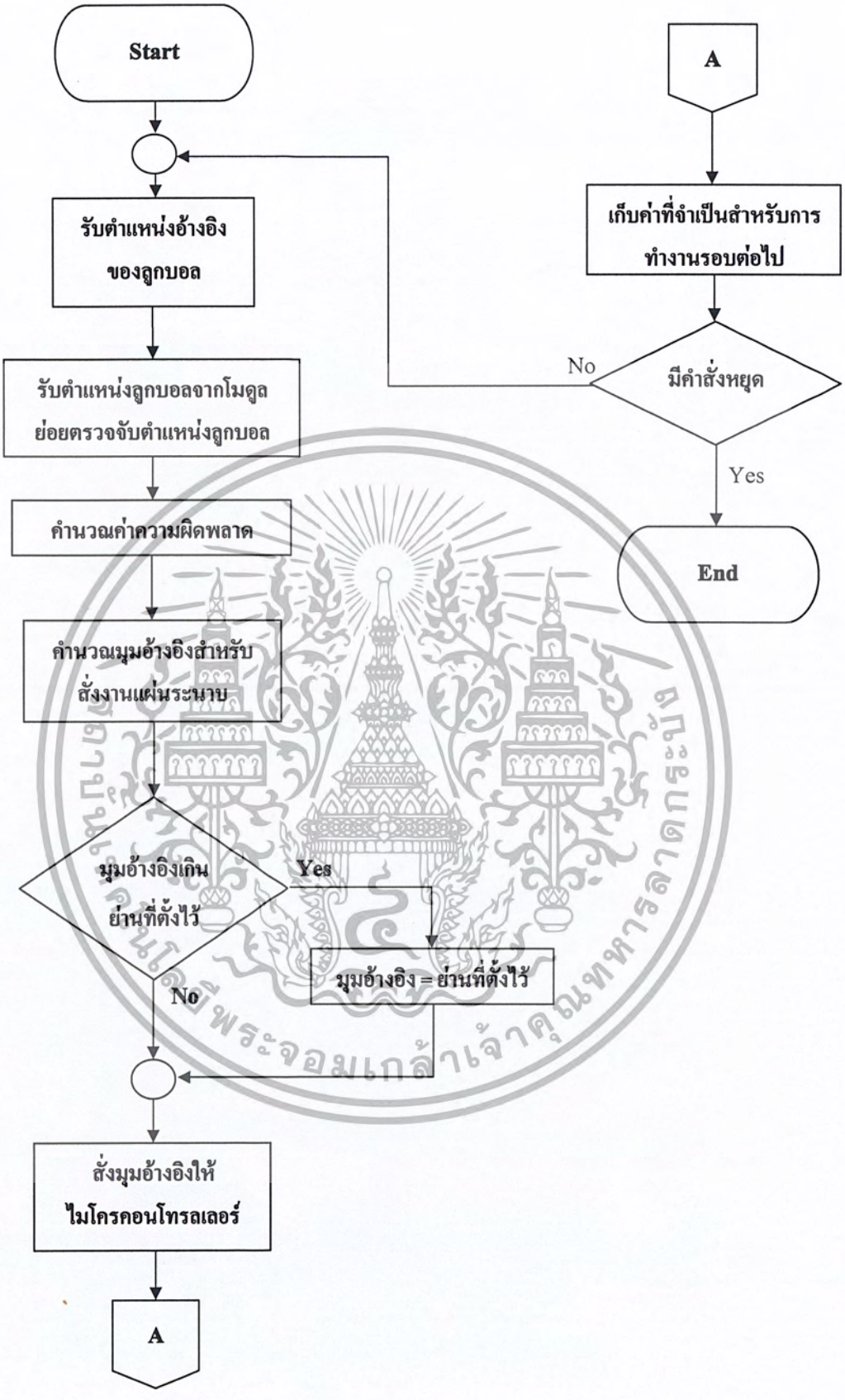
3.2.2 โปรแกรมส่วนไมโครคอนโทรลเลอร์

ในโปรแกรมในส่วนของไมโครคอนโทรลเลอร์ จะเป็นการเขียนโปรแกรมเพื่อควบคุมมุมของระนาบ โดยโปรแกรมจะทำงานทุก ๆ คาบการชักตัวอย่างที่ 0.11 มิลลิวินาที

การควบคุมมุมระนาบ แสดงดังรูปที่ 3.13 เริ่มการทำงานโดยไมโครคอนโทรลเลอร์จะรับค่ามุมอ้างอิงจากคอมพิวเตอร์และมุมจริงจากโพเท็นทิโอมิเตอร์ มาคำนวณหาค่าความผิดพลาด ซึ่งค่าความผิดพลาดที่ได้จะนำไปคำนวณหาสัญญาณควบคุมมอเตอร์โดยอาศัยกฎการควบคุมแบบพีไอดีเช่นเดียวกัน ซึ่งในทางปฏิบัติจะบวกค่าชดเชยเพิ่มด้วยสำหรับลดผลแรงเสียดทาน สัญญาณควบคุมมอเตอร์จากตัวควบคุมจะถูกแปลงเป็นสัญญาณพัลส์วิตมอตดูเลชั่นที่ความถี่ 1 กิโลเฮิรตซ์ ซึ่งแบ่งเป็น 0-8,000 ระดับที่ระดับแรงดัน 0-12 โวลต์ และจะตรวจสอบทิศทางและแรงดันของสัญญาณ หากแรงดันมากกว่า 12 โวลต์ ให้เท่ากับ 12 โวลต์ แล้วส่งสัญญาณควบคุมไปขับมอเตอร์และเก็บค่าที่จำเป็นสำหรับการทำงานรอบต่อไป

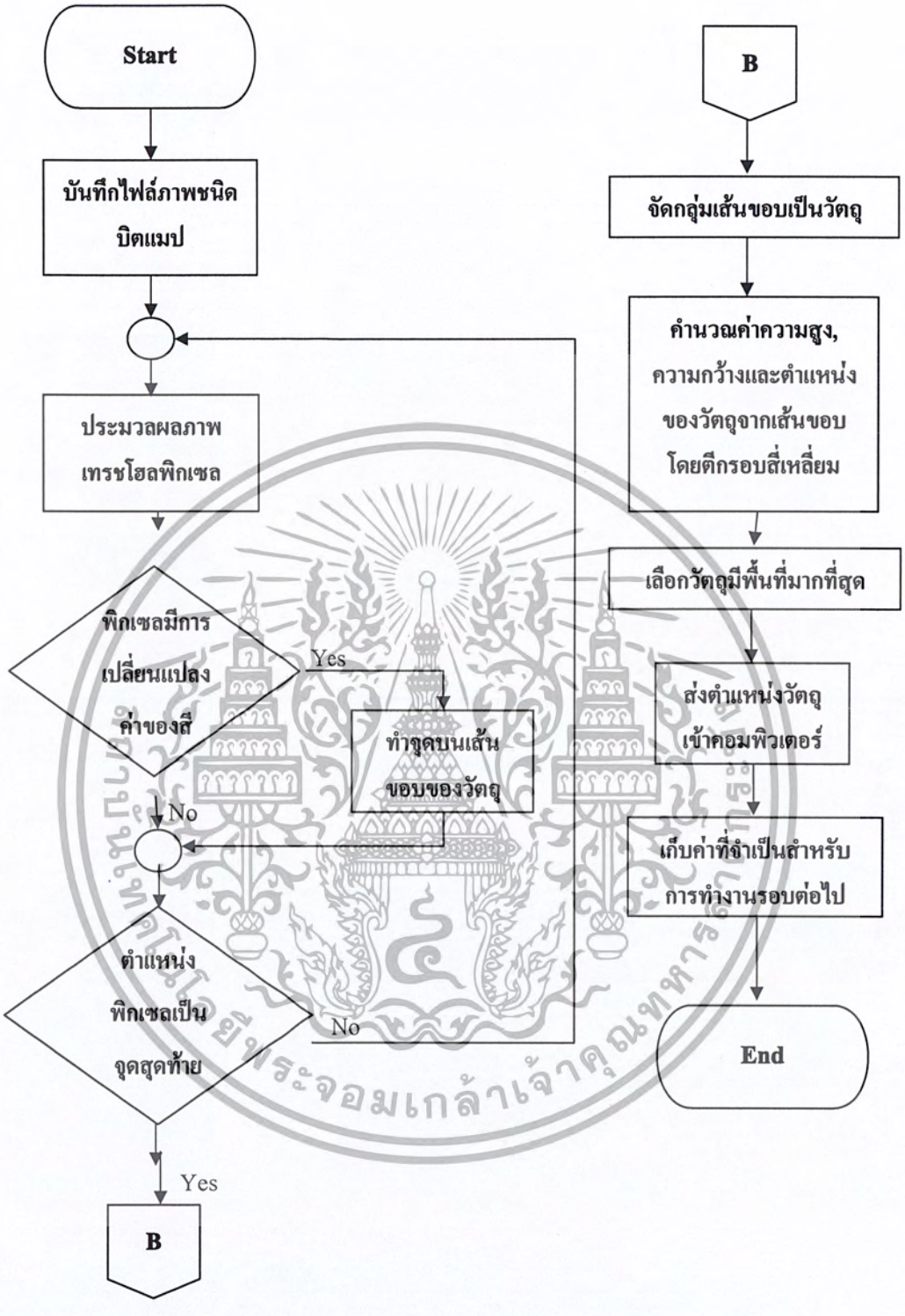
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



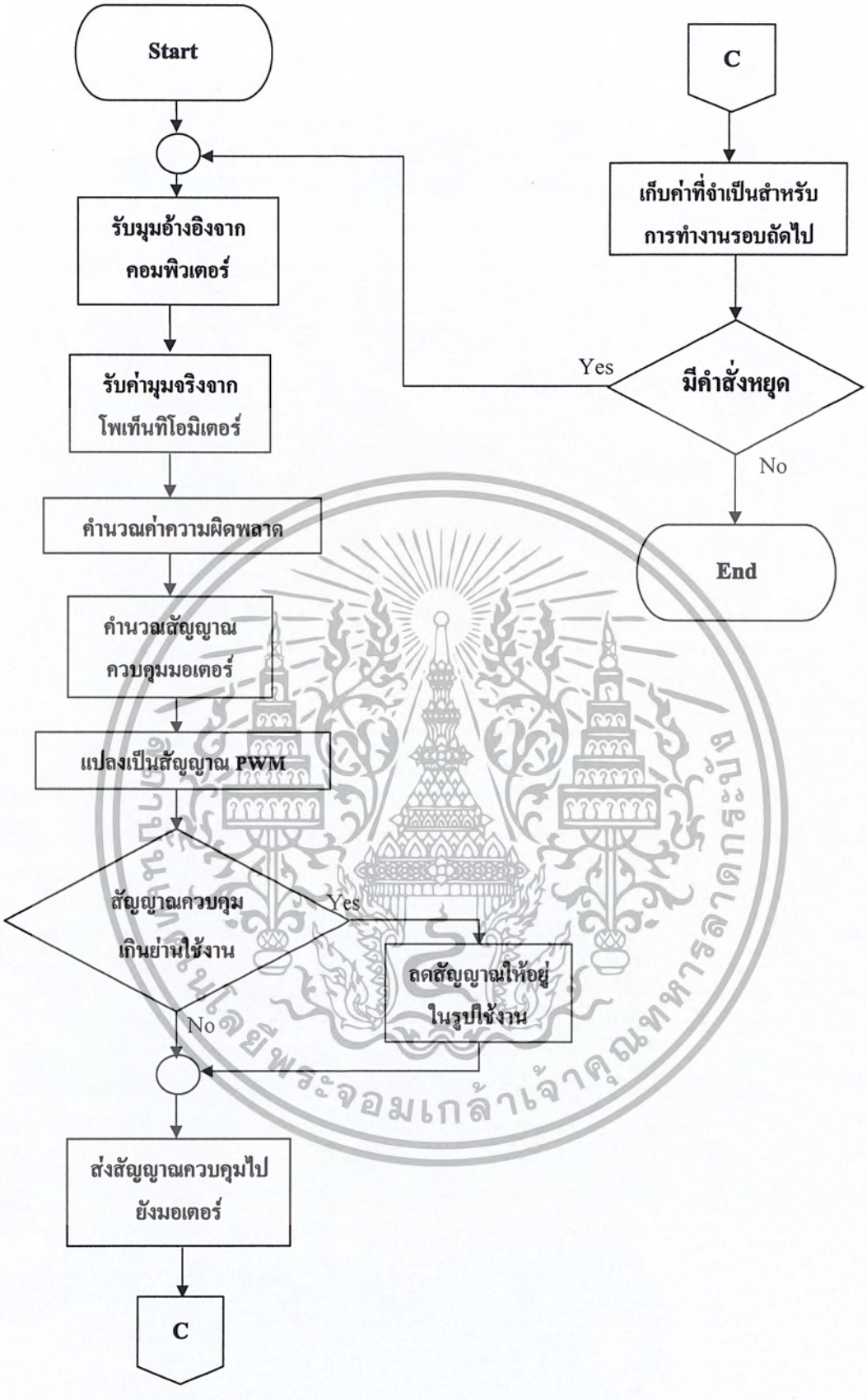
รูปที่ 3.11 โฟลว์ชาร์ตควบคุมตำแหน่งลูกบอลบนแผ่นระนาบในคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 โฟลว์ชาร์ตการหาตำแหน่งลูกบอลบนแผ่นระนาบในคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 โฟลว์ชาร์ตการควบคุมมุมของระนาบโดยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองต่าง ๆ ของโครงงานนี้ เริ่มจากการทดสอบการวัดตำแหน่งมุมของระนาบ การทดลองควบคุมตำแหน่งมอเตอร์เพื่อควบคุมตำแหน่งมุมระนาบ และการทดลองควบคุมตำแหน่งลูกบอล โดยมีรายละเอียดดังนี้

4.1 การระบุตำแหน่งมุมของระนาบโดยเทียบกับค่าสัญญาณอนาล็อกเป็นดิจิตอล

ในการระบุตำแหน่งตำแหน่งมุมระนาบโดยเปรียบเทียบสัญญาณดิจิตอล โดยเทียบกับระดับอ้างอิง และทำการวัดที่ 0, 7 และ -7 องศาอ่านค่าดิจิตอลจากตัวควบคุมทางพอร์ตอนุกรม สามารถสรุปได้ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองความสัมพันธ์ระหว่างค่าระดับสัญญาณกับมุมของระนาบ

แกน X		แกน Y	
มุม(องศา)	ระดับสัญญาณ	มุม(องศา)	ระดับสัญญาณ
7	789	7	181
0	530	0	579
-7	271	-7	846

จากการทดลองวัดระดับสัญญาณ สามารถนำค่าระดับสัญญาณมากำหนดเพื่อบ่งบอกองศาของมุมระนาบได้ และสามารถทำให้แผ่นระนาบทำมุมองศาไปตามที่ต้องการได้ แต่ในโครงงานนี้จะใช้มุมของระนาบเพียง -5 ถึง 5 เท่านั้น ซึ่งเผื่อไว้อีก 2 องศาเพื่อป้องกันอุปกรณ์ทางกลของระบบ

4.2 การทดสอบขับมอเตอร์โดยตรงและการประมาณฟังก์ชั่นถ่ายโอน

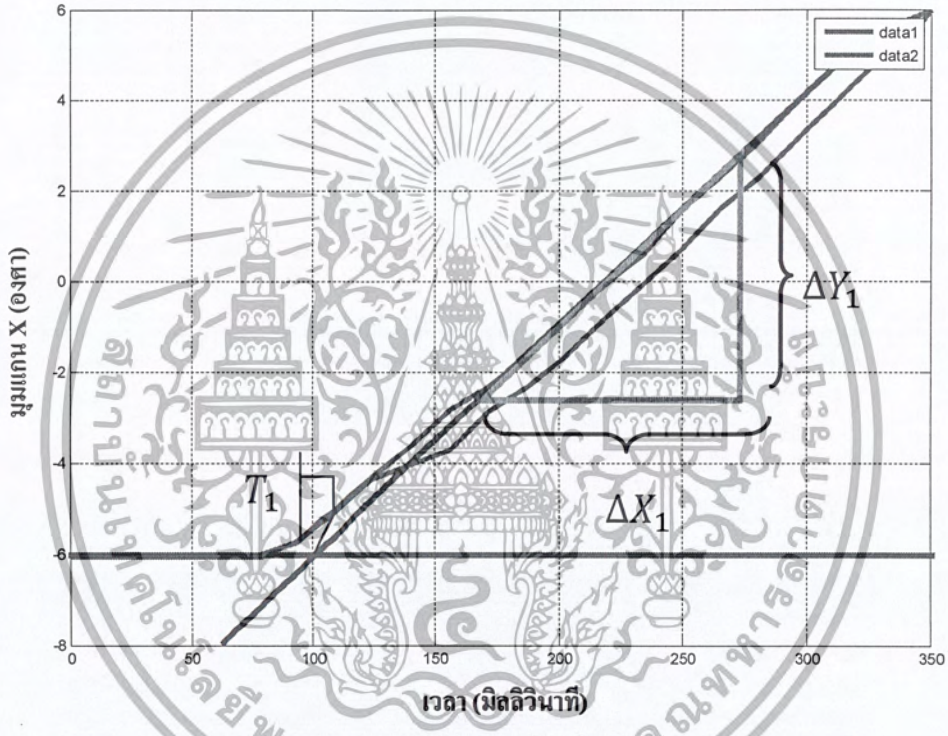
ความสัมพันธ์โดยประมาณของตำแหน่งมุมต่อแรงดันที่ขับมอเตอร์ อาจประมาณด้วยฟังก์ชั่นถ่ายโอนดังสมการ (4.1)

$$\frac{\theta(s)}{V(s)} = \frac{K}{s(Ts+1)} \tag{4.1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่าคงที่ K และค่าเวลาคงที่ T สามารถหาได้จากการทดลองดังนี้

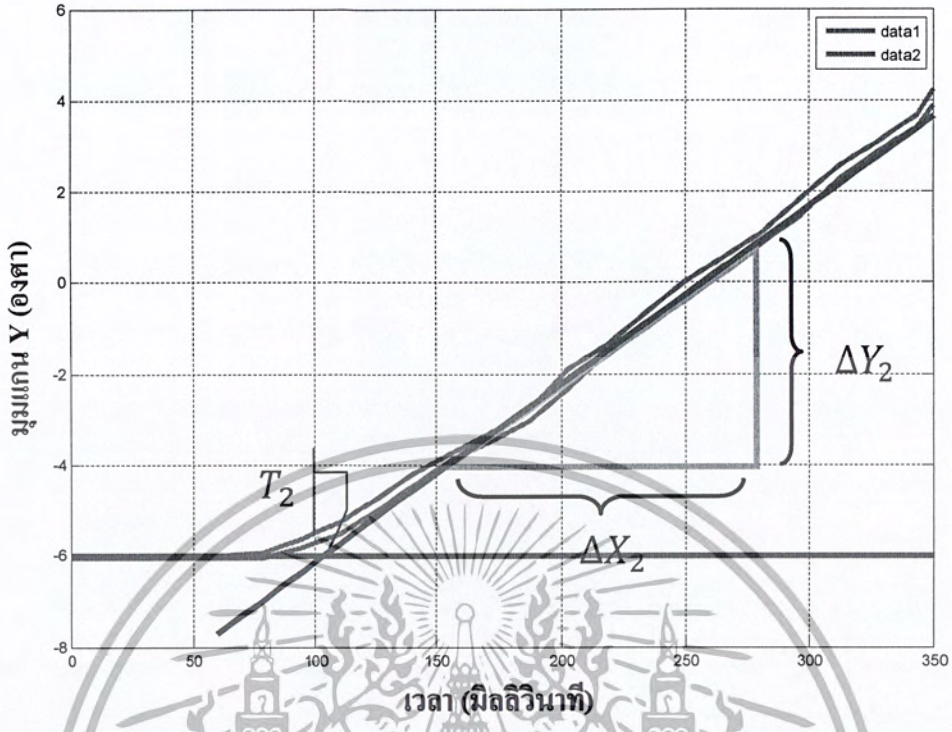
ทดสอบโดยจ่ายระดับแรงดันสัญญาณพีดับบิวเอ็มที่ความกว้างพัลส์ 70% ให้แก่มอเตอร์แกน X และแกน Y สามารถบันทึกตำแหน่งมุมที่เปลี่ยนไปได้ ดังรูปที่ 4.1 และ 4.2 ตามลำดับ โดยทำการทดสอบ 2 ครั้งในแต่ละแกน เพื่อใช้หาค่าเฉลี่ย



รูปที่ 4.1 ทดสอบระบบโดยจ่ายระดับแรงดันที่ 70% ของสัญญาณพีดับบิวเอ็มแกน X

จากผลทดสอบในรูปที่ 4.1 และ 4.2 สามารถหาค่าเวลาคงที่ T ได้โดยลากเส้นตรง 2 เส้น เส้นแรกที่ตำแหน่งมุมเริ่มต้นและเส้นที่ 2 ขนานผลตอบสนองสภาวะอยู่ตัว ซึ่งจะได้จุดตัดเส้นตรง คือ T และค่าความชันหาได้จาก ΔX และ ΔY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ทดสอบระบบโดยจ่ายระดับแรงดันที่ 70% ของสัญญาณพีดับบิวเอ็มแกน Y

ค่าความชันที่ได้สัมพันธ์กับค่าคงที่ K โดยขึ้นกับระดับแรงดันอินพุต อธิบายได้ดังสมการ (4.2)

$$KA = \frac{\Delta Y}{\Delta X} \tag{4.2}$$

เมื่อ $\frac{\Delta Y}{\Delta X}$ คือความชัน

A คือ ระดับแรงดันอินพุต

ดังนั้นสามารถประมาณฟังก์ชั่นถ่ายโอนระหว่างตำแหน่งมุมและแรงดันที่ขับมอเตอร์ได้ดังสมการ (4.3) และ (4.4) สำหรับมอเตอร์แกน X และแกน Y ตามลำดับ

$$\frac{\theta x(s)}{Vx(s)} = \frac{3.33}{s(0.052s+1)} \tag{4.3}$$

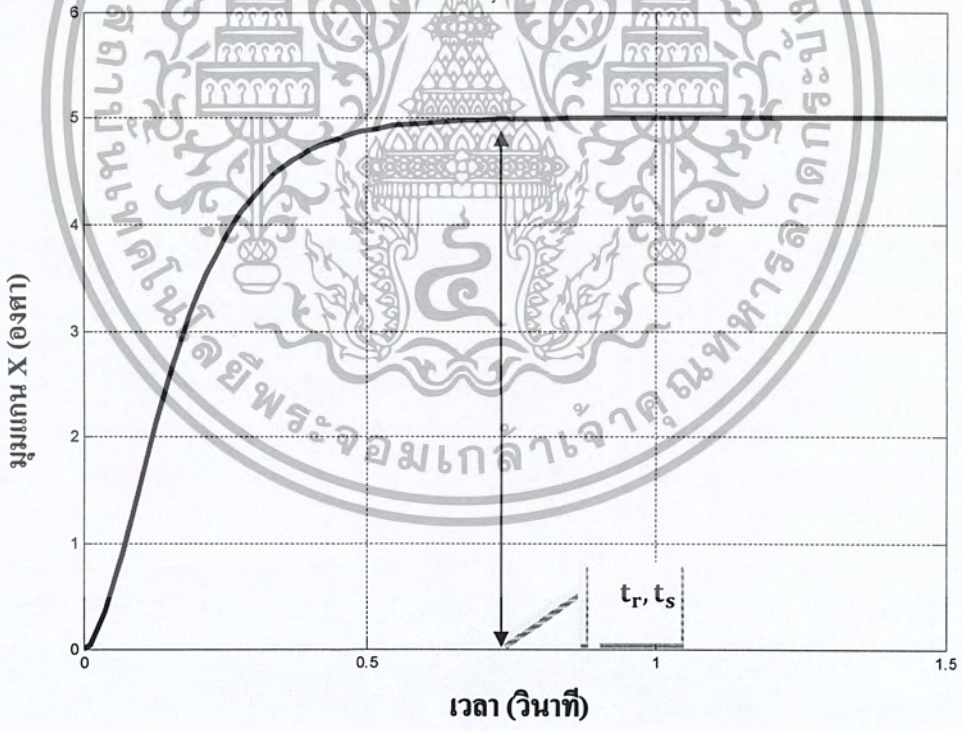
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\theta y(s)}{Vy(s)} = \frac{3.7037}{s(0.075s+1)} \quad (4.4)$$

4.3 การจำลองการควบคุมมุมแผ่นระนาบแบบวงปิด

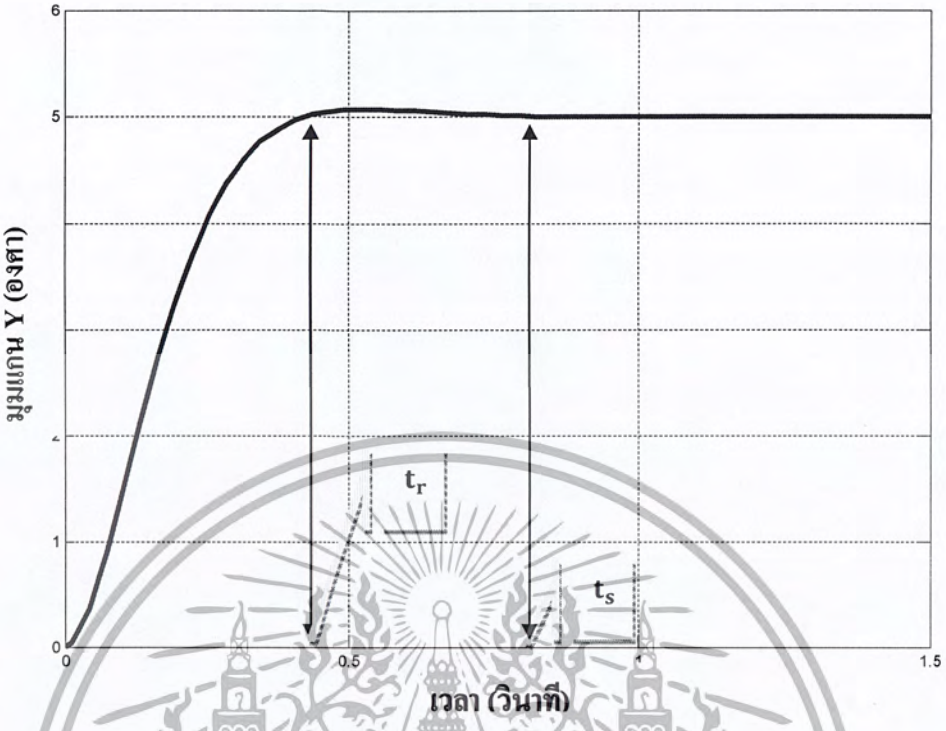
จากฟังก์ชั่นถ่ายโอนที่ได้จากสมการ (4.3) และ (4.4) นำมาออกแบบตัวควบคุม โดยในที่นี้ใช้ตัวควบคุมแบบพีดีทั้ง 2 แกน ซึ่งการจำลองเลือกใช้ $K_p = 2, K_d = 0.05$ เท่ากันทั้งในแกน X และแกน Y โดยได้ผลตอบสนองจากการจำลองดังรูปที่ 4.3 และ 4.4 ตามลำดับ

จากผลการจำลองการควบคุมมุมแผ่นระนาบจาก 0 ไปยัง 5 องศาทั้ง 2 แกนพบว่าในแนวแกน X ระบบมีค่า $t_r = 0.67s, t_s = 0.67s$ และแนวแกน Y ระบบมีค่า $t_r = 0.67s, t_s = 0.83s$ โดยผลตอบสนองของแกน Y จะมีช่วงของค่าพุ่งเกินเล็กน้อย ซึ่งจากเวลาตอบสนองที่ได้มีความเร็วเพียงพอต่อการควบคุมมุมแผ่นระนาบ จึงควรจะสามารถใช้เกนค่าดังกล่าวนี้ในระบบจริงได้



รูปที่ 4.3 ผลการจำลองระบบควบคุมมุมระนาบแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ผลการจำลองระบบควบคุมมุมระนาบแกน Y

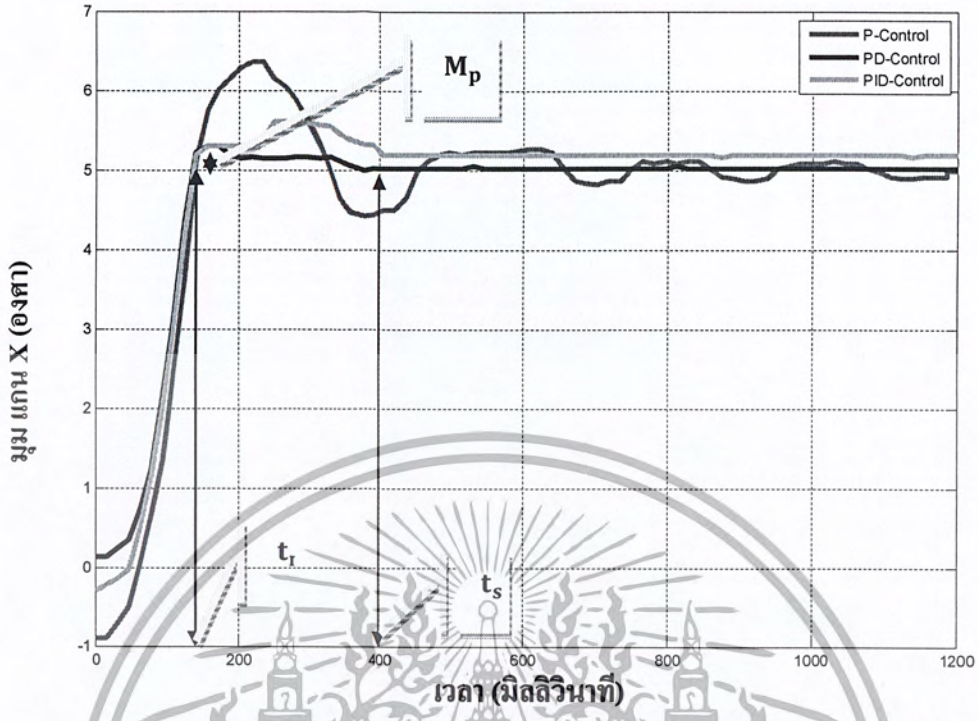
4.4 การทดลองควบคุมมุมระนาบ

ในการทดลองนี้จะนำเอาค่าเกนที่ได้จากการจำลองการควบคุมมุมระนาบแบบวงปิด มาทดลองใช้กับระบบจริง พร้อมทั้งทดลองปรับใช้ตัวควบคุมแบบพี และพีไอดี เพื่อเปรียบเทียบผล โดยแต่ละแกนจะควบคุมมุมแผ่นระนาบจากตำแหน่งเริ่มต้นที่ประมาณ 0 ไปยัง 5 องศาเช่นเดิม แกน X ตัวควบคุมพีใช้เกน $K_p = 2$ ตัวควบคุมพีดีใช้เกน $K_p = 2, K_d = 0.05$ และตัวควบคุมพีไอดีใช้เกน $K_p = 2, K_i = 1$ $K_d = 0.05$ และแกน Y ตัวควบคุมพีใช้เกน $K_p = 2$ ตัวควบคุมพีดีใช้เกน $K_p = 2, K_d = 0.05$ และตัวควบคุมพีไอดีใช้เกน $K_p = 2, K_i = 1$ $K_d = 0.05$ เช่นเดียวกัน

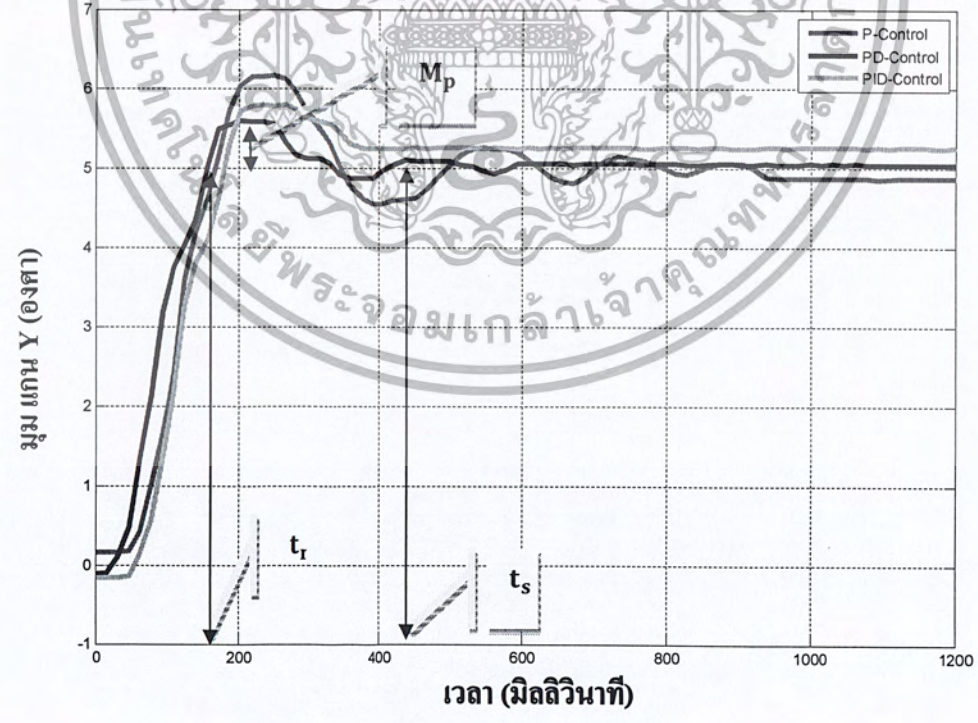
จากผลการทดลองในรูปที่ 4.5 พบว่าตัวควบคุมพีดีให้ผลตอบสนองเข้าสู่สภาวะคงตัวเร็วกว่าตัวควบคุมแบบอื่นโดยมีค่า $t_r = 1.29s, t_s = 3.87s, M_p = 6.7\%$ ซึ่งมีค่าต่างจากผลการจำลองอันเนื่องมาจากโครงสร้างทางกลต่าง ๆ ที่อาจมีการสูญเสียพลังงานในแต่ละจุด อย่างไรก็ตามผลตอบสนองที่ได้จากระบบจริงยังสอดคล้องกับผลการจำลอง

จากผลการทดลองในรูปที่ 4.6 ผลตอบสนองให้ผลใกล้เคียงกับผลการทดลองในรูปที่ 4.5 โดยผลของตัวควบคุมพีดีมีค่า $t_r = 1.5s, t_s = 6.0s, M_p = 10\%$ โดยมีช่วงประมาณ 2 วินาทีที่ยังสั่นเล็กน้อยก่อนถึงวินาทีที่ 6 ซึ่งสามารถเข้าสู่สภาวะคงตัวได้ใน 6 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ผลการทดลองควบคุมมุมระนาบแกน X



รูปที่ 4.6 ผลการทดลองควบคุมมุมระนาบแกน Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การระบุตำแหน่งของลูกบอลโดยเทียบพิกเซล

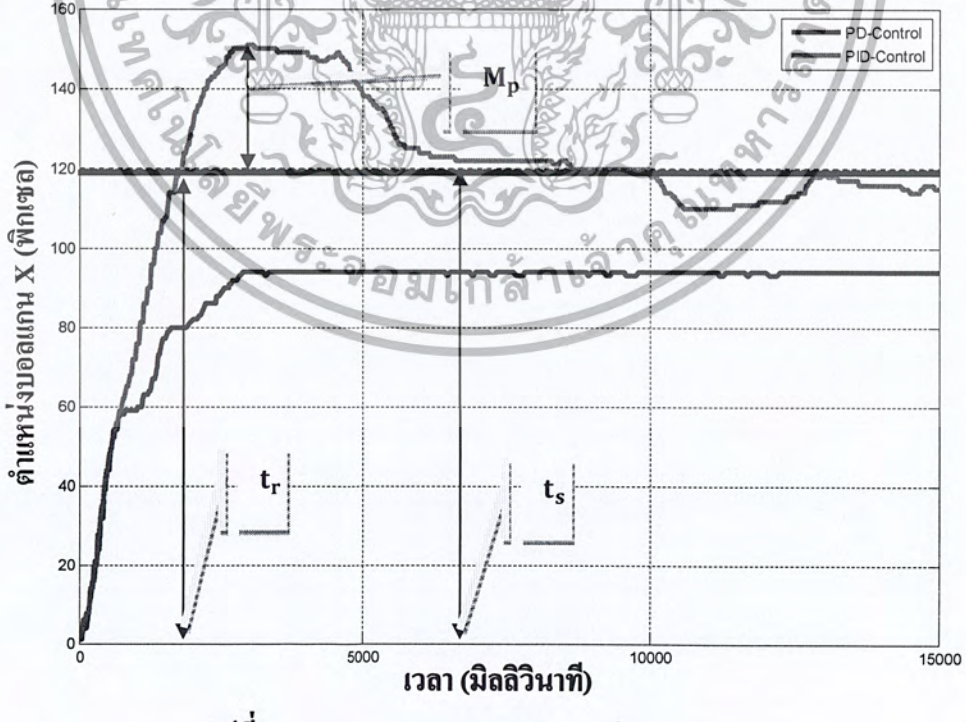
ในโครงงานนี้ได้ติดตั้งกล้องดิจิตอลวีดีโอสำหรับการตรวจจับลูกบอลบนแผ่นระนาบ ซึ่งสามารถอ่านค่าตำแหน่งได้จากคอมพิวเตอร์ จึงต้องทำการเปลี่ยนแปลงความกว้างเทียบพิกเซลโดยใช้ความกว้างของระนาบเท่ากับ 0.4 เมตร มีค่าเท่ากับ 240 พิกเซล โดยความสัมพันธ์ดังกล่าวสามารถเขียนได้โดยดังสมการ (4.5)

$$A = \frac{B \times 0.5}{240} \tag{4.5}$$

เมื่อ A คือ ตำแหน่งจริงลูกบอลบนระนาบมีหน่วยเป็น B เมตร นั่นก็คือ ตำแหน่งลูกบอลที่อ่านได้จากหน้าจอคอมพิวเตอร์มีหน่วยเป็นพิกเซล

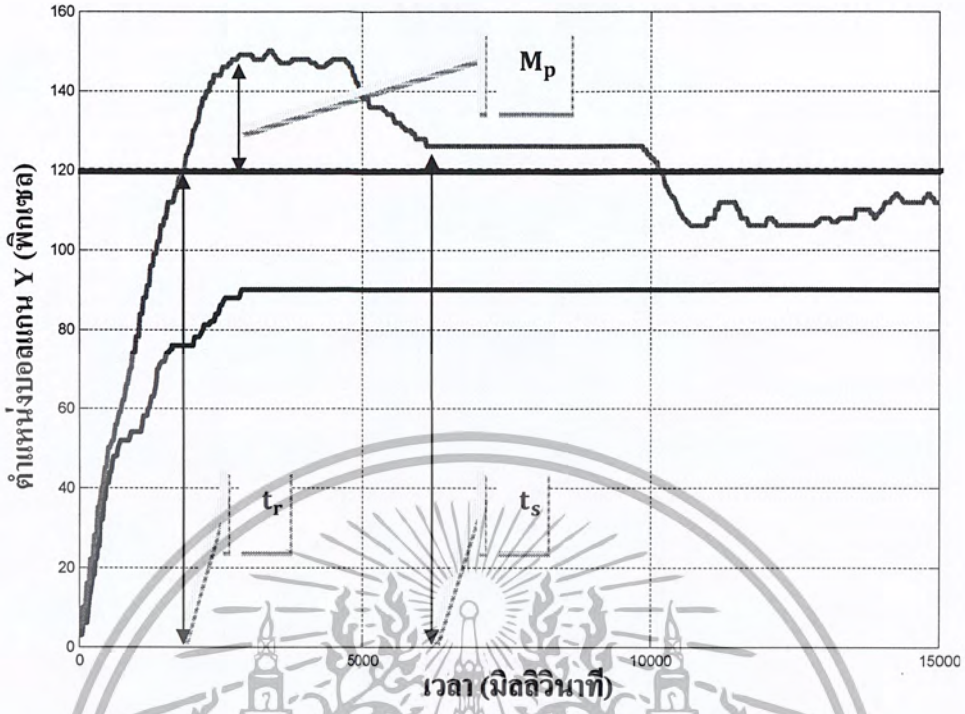
4.6 การทดลองควบคุมตำแหน่งลูกบอลบนแผ่นระนาบ

ทดลองปรับใช้ตัวควบคุมแบบพีดี และพีไอดี เพื่อเปรียบเทียบผล ให้ควบคุมตำแหน่งลูกจาก 0 ไปยัง 120 พิกเซล โดยแกน X ตัวควบคุมพีดีใช้เกน $K_p = 0.5, K_d = 0.5$ และตัวควบคุมพีไอดีใช้เกน $K_p = 0.5, K_i = 0.5, K_d = 0.5$ และแกน Y ตัวควบคุมพีดีใช้เกน $K_p = 0.5, K_d = 0.05$ และตัวควบคุมพีไอดีใช้เกน $K_p = 0.5, K_i = 0.5, K_d = 0.5$ เช่นเดียวกัน ได้ผลตอบสนองดังแสดงรูปที่ 4.7 และ 4.8



รูปที่ 4.7 ผลการทดลองควบคุมตำแหน่งลูกบอลแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ผลการทดลองควบคุมตำแหน่งลูกบอลแกน Y

จากการทดลองในรูปที่ 4.7 และ 4.8 พบว่าตัวควบคุมพีดีไม่สามารถควบคุมตำแหน่งลูกบอลไปยัง 120 พิกเซลได้ ซึ่งตัวควบคุมพีไอดีให้ตำแหน่งลูกบอลเข้าสู่ตำแหน่ง 120 พิกเซล อันเนื่องมาจากไอคอนโทรลทำหน้าที่เป็นตัวสะสมค่าความผิดพลาดทำให้ลูกไปยังตำแหน่งที่ต้องการได้ซึ่งผลตอบสนองในแกน X มีค่าสมรรถนะ $t_r = 1.79s,\ t_s = 6.79s,\ M_p = 25\%$ และในแกน Y มีค่า $t_r = 1.79s,\ t_s = 6.07s,\ M_p = 25\%$

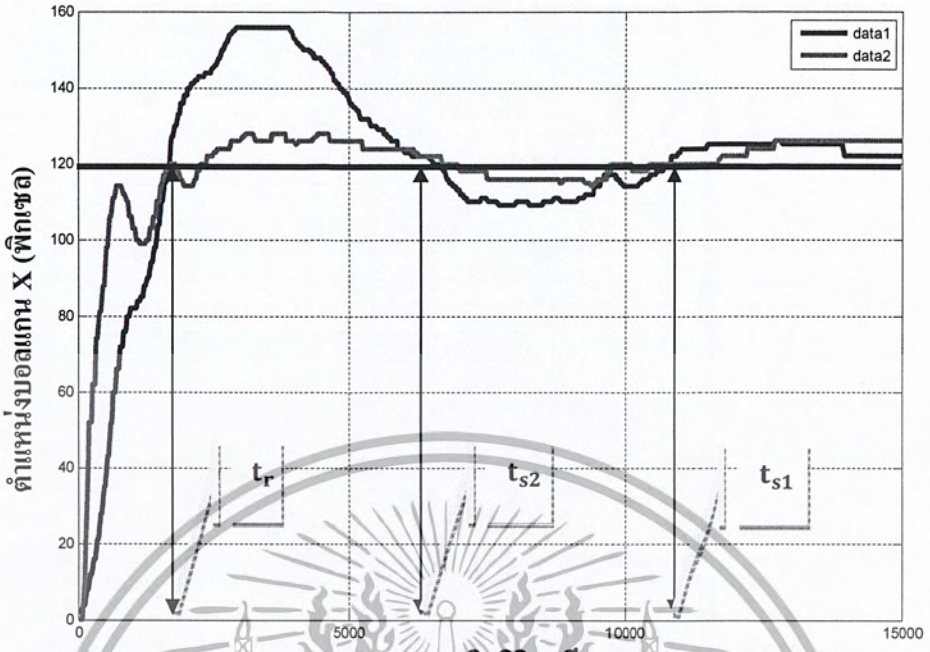
พบว่าตัวควบคุมพีไอดี สามารถควบคุมตำแหน่งลูกบอลได้ จึงนำค่าเกนที่ได้จากการทดลองปรับตัวควบคุมพีไอดีมาใช้ทดลองควบคุมตำแหน่งลูกบอล โดยปล่อยลูกบอลจากมุมต่าง ๆ และควบคุมลูกบอลให้หยุดที่ตำแหน่ง (120,120) พิกเซล

4.6.1 การทดลองปล่อยลูกบอลจริงครั้งที่ 1

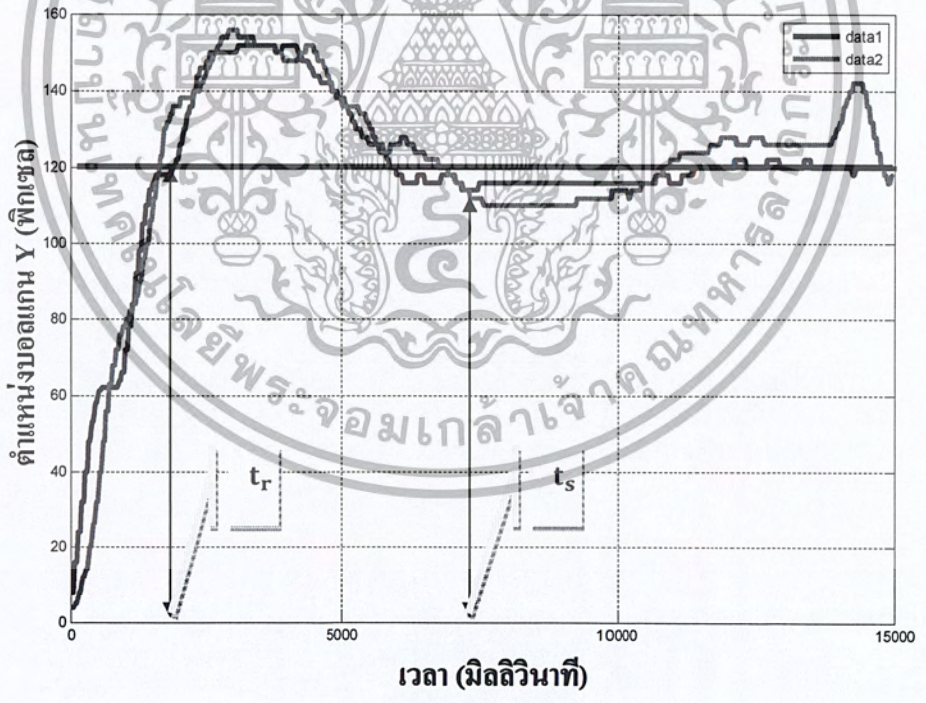
การทดลองนี้เป็นการปล่อยลูกบอลจากมุมซ้ายล่าง (0,0) และควบคุมลูกบอลไปยังตำแหน่ง (120,120) ของระนาบ ใช้เกน $K_p = 0.5, K_i = 0.5, K_d = 0.5$ โดยทำการทดลอง 2 ครั้ง ได้ผลตอบสนองดังรูปที่ 4.9

จากการทดลองระบบสามารถรักษาตำแหน่งลูกบอลที่ 120 พิกเซลได้ โดยทำการทดลอง 2 ครั้ง สามารถหาค่าเฉลี่ยเวลาลู่เข้าได้ที่เวลา 9 วินาที โดยระบบพยายามควบคุมลูกบอลให้ได้ตำแหน่งที่ 120 พิกเซล ตามที่กำหนด ทั้งในแนวแกน X และ Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) แกน X



(ข) แกน Y

รูปที่ 4.9 ผลการทดลองควบคุมตำแหน่งลูกบอลโดยปล่อยลูกบอลครั้งที่ 1

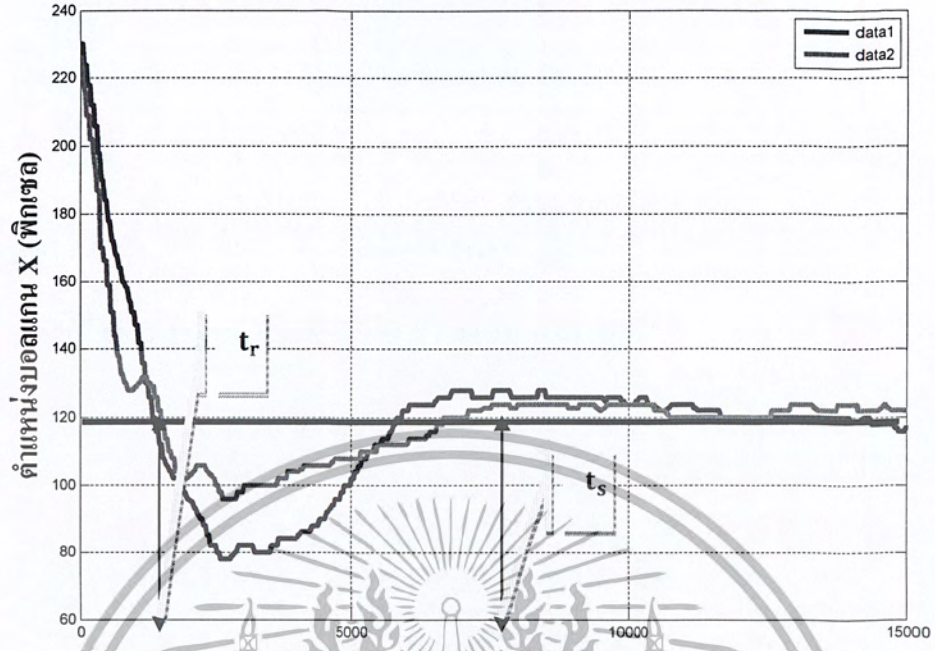
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2 การทดลองปล่อยลูกบอลจริงครั้งที่ 2

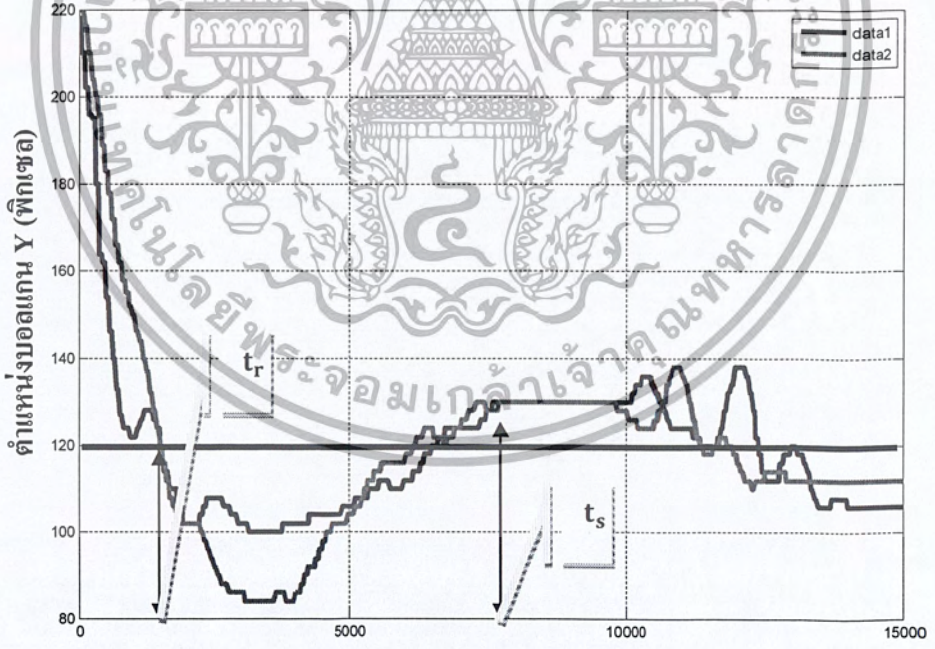
ทดลองปล่อยลูกบอลจากมุมขวาบน (240,240) และควบคุมลูกบอลไปยังตำแหน่ง (120,120) ของระนาบ ใช้เกน $K_p = 0.5, K_i = 0.5, K_d = 0.5$ โดยทำการทดลอง 2 ครั้งเช่นกัน ได้ผลตอบสนองดังรูปที่ 4.10

จากการทดลองสามารถหาค่าเฉลี่ยเวลาลู่เข้าได้ที่เวลาประมาณ 7 วินาที ซึ่งเร็วกว่าการปล่อยลูกบอลในตำแหน่ง (0,0) แม้ระบบจะมีค่าพุ่งเกินในช่วงแรกไปมาก แต่สามารถบังคับให้ลูกบอลกลับสู่ตำแหน่ง (120,120) ได้ จากการทดลองพบว่าลูกบอลเคลื่อนที่เลยตำแหน่งที่กำหนดไปสูงสุดไม่เกิน 40 พิกเซลหรือ 67 มิลลิเมตร คิดเป็นค่าพุ่งเกินสูงสุด $M_p = 33\%$ ซึ่งสมรรถนะที่ได้นี้ยังเพียงพอในการใช้งานสำหรับแผ่นระนาบซึ่งมีขนาด 400x400 มิลลิเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) แกน X



(ข) แกน Y

รูปที่ 4.10 ผลการทดลองควบคุมตำแหน่งลูกบอลโดยปล่อยลูกบอลครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุปผล

5.1 สรุปผลการทดลอง

จากการทดลองทำโครงงานเรื่องระบบควบคุมลูกบอลบนแผ่นระนาบนั้น สามารถแบ่งออกได้เป็น 2 ส่วนหลัก ๆ คือ ส่วนควบคุมบนแผ่นระนาบ และส่วนควบคุมตำแหน่งลูกบอล ทั้งสองส่วนทำการควบคุมด้วยตัวควบคุมพีไอดี

จากการทดลองส่วนควบคุมแผ่นระนาบ ซึ่งเป็นการควบคุมมอเตอร์กระแสตรง ผ่านไมโครคอนโทรลเลอร์ และทำการตรวจจับค่ามุมที่เปลี่ยนไปโดยอาศัยโพเท็นทิโอมิเตอร์ขนาด 10 กิโลโอห์ม ในการอ่านค่ามุมระนาบ พบว่าเมื่อใช้ตัวควบคุมแบบพีดี สามารถควบคุมมุมเอียงของแผ่นระนาบได้ตามที่กำหนด นั่นคือระบบเข้าสู่สถานะคงตัวอย่างรวดเร็วโดยใช้เวลาประมาณ 400 มิลลิวินาที มีค่าพุ่งเกินสูงสุดและสัญญาณความผิดพลาดในปริมาณที่น้อย ทั้งการเคลื่อนที่ในแกน X และแกน Y

ในส่วนการทดลองควบคุมลูกบอลบนระนาบ อาศัยตัวควบคุมพีไอดี เพื่อควบคุมลูกบอลเพื่อไปยังตำแหน่งที่ต้องการ จากผลการทดลองพบว่า ลูกบอลสามารถถูกบังคับไปยังตำแหน่งที่ต้องการได้โดยมีค่าความผิดพลาดเพียงเล็กน้อยและมีค่าพุ่งเกินสูงสุดไม่เกิน 33% ซึ่งไม่เกินขนาดของแผ่นระนาบ และเข้าสู่สถานะคงตัว ภายในเวลาประมาณ 3 วินาที

5.2 ปัญหาที่พบและแนวทางการแก้ไข

เนื่องจากปัญหาทางกล ทำให้ไม่สามารถหาแบบจำลองโมเดลทางคณิตศาสตร์ที่แน่นอนของระบบโดยรวม จึงไม่สามารถทำการออกแบบระบบควบคุมโดยอาศัยทฤษฎีโดยตรง ทางผู้จัดทำจึงเปลี่ยนไปใช้วิธีลองผิดลองถูก

5.3 ข้อเสนอแนะ

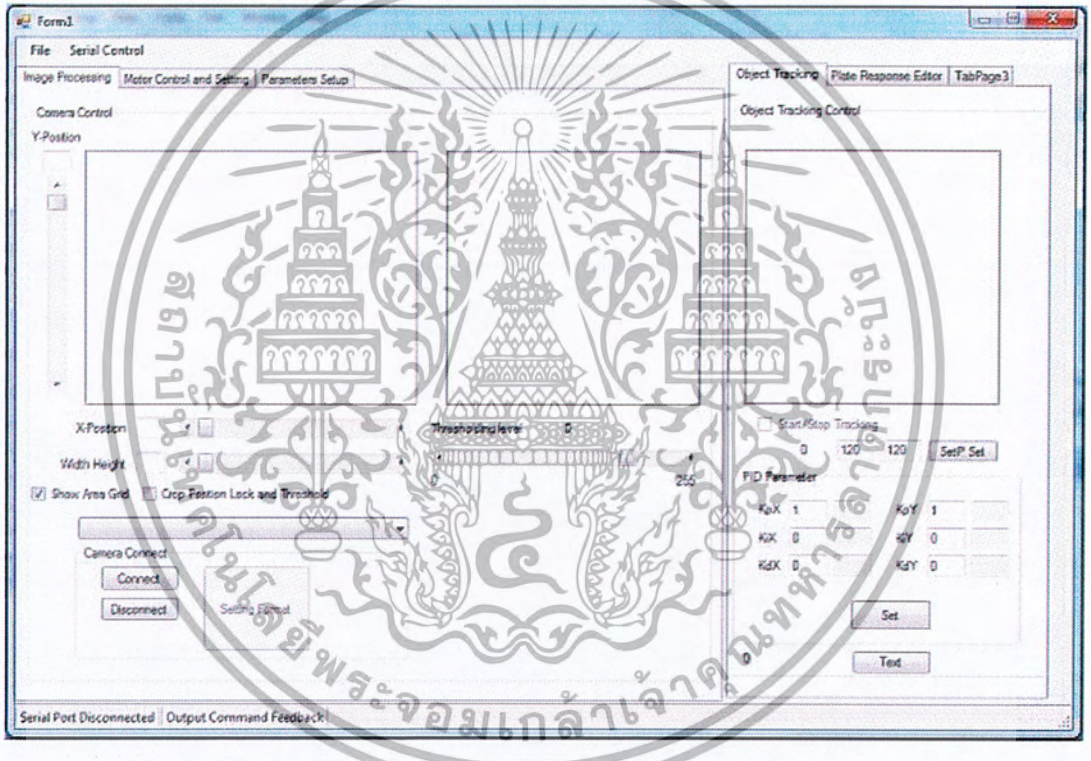
จากปัญหาที่พบดังกล่าวนั้น จึงควรจะศึกษาโครงสร้างทางกลให้มากขึ้น และเลือกวิธีที่เหมาะสมในการสร้างโครงสร้างทางกลให้มีความผิดพลาดน้อยที่สุด เพื่อให้คำนวณหาฟังก์ชั่นถ่ายโอนได้ดียิ่งขึ้น อีกทั้งยังควรทดลองใช้ระบบควบคุมแบบอื่นนอกจากตัวควบคุมพีไอดี ที่สามารถชดเชยความไม่แน่นอนของโครงสร้างทางกลได้ จึงควรศึกษาเพิ่มเติมเพื่อให้ระบบมีทั้งเสถียรภาพและประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

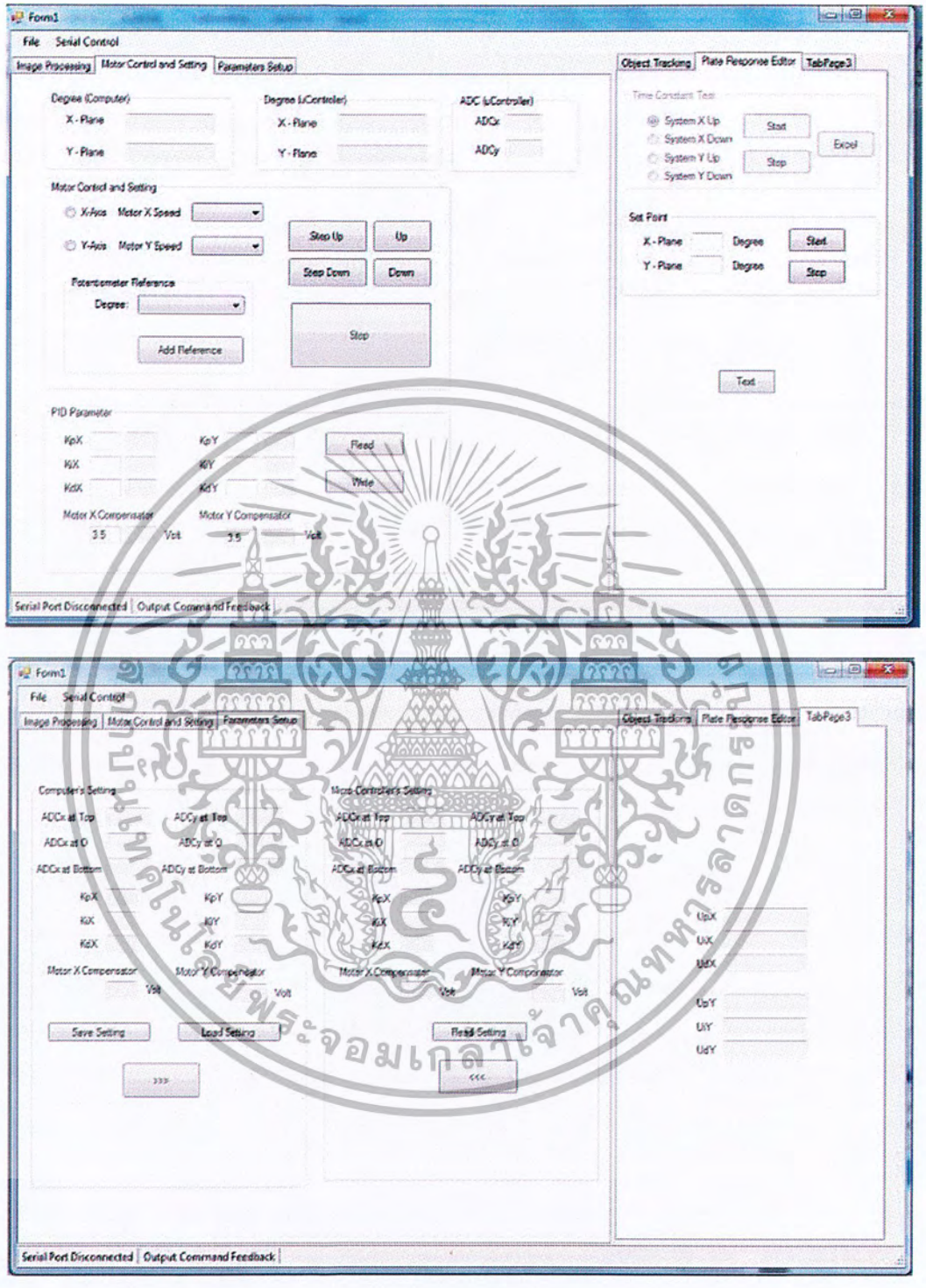
โปรแกรมการประมวลผลภาพและโปรแกรมควบคุม

โปรแกรมวิชชวลเบสิก วัตถุประสงค์ในการเขียนโปรแกรมนี้ เพื่อตรวจสอบตำแหน่งลูกบอลและเป็นหน้าต่าง ๆ เชื่อมต่อระหว่างผู้ใช้และอุปกรณ์ (Graphical User Interface) ดังรูปที่ ก.1 และโค้ดโปรแกรมดังต่อไปนี้



รูปที่ ก.1 โปรแกรมอินเตอร์เฟสของวิชชวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 หน้าต่างการปรับพารามิเตอร์พีไอดีของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.1 โปรแกรมในส่วนคอมพิวเตอร์

ก.1.1 โปรแกรมควบคุมตำแหน่งลูกบอล

```
Imports Microsoft.Office.Interop
'Imports Microsoft.Office.Interop.Excel
Imports System.IO
Public Class Form1
    Dim Str As Char = "#" '# Start Code
    Dim fin2 As Char = "S"   'Ending Code2
    Dim fin As Char = vbCr   'Ending Code
    Dim mode As Char
    Dim Send As String
    Dim Send2 As String
    Dim ReadBuffer As String
    Dim StartSystemTest As Boolean = False
    ''''''''''''''''''''''''''''''''''''''''ส่วนของ PID Ball
    Dim BallKpX As Double = 1
    Dim BallKiX As Double = 0
    Dim BallKdX As Double = 0
    Dim BallKpY As Double = 1
    Dim BallKiY As Double = 0
    Dim BallKdY As Double = 0
    Dim BallXSetPoint As Double
    Dim BallYSetPoint As Double
    Dim X_err_now As Double
    Dim Y_err_now As Double
    Dim X_err_Previous As Double
    Dim Y_err_Previous As Double
    Dim X_err_Accumulate As Double
    Dim Y_err_Accumulate As Double
    Dim BallXPIDGain As Double
    Dim BallYPIDGain As Double
    Dim Ts As Double = 0.05 's
    Dim UdX As Double
    Dim UdY As Double
    Dim UiX As Double
    Dim UiY As Double
    Dim UiX0 As Double
    Dim UiY0 As Double
    ''''''''''''''''''''''''''''''''''''''''''

    '////////////////////////////////// ส่วนของDegree Calculate//////////////////////////////
    Public Structure PlateData
        Dim ADC As Integer
        Dim VBDeg As Double
        Dim Deg As Double
        Dim ADCTop As Integer
        Dim ADCOrigin As Integer
        Dim ADCBottom As Integer
        Dim MPos As Double
        Dim MNeg As Double
        Dim CPos As Double
        Dim CNeg As Double
        Dim MMean As Double
        Dim CMean As Double
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Dim inverseADC_Deg As Boolean
End Structure
Dim X_Plate As PlateData
Dim Y_Plate As PlateData
" ย่อได้รีเปล่า??
Dim X_max As Integer = 7
Dim X_min As Integer = -7
Dim Y_max As Integer = 7
Dim Y_min As Integer = -7
Dim X_Limitmax As Integer = 5
Dim X_Limitmin As Integer = -5
Dim Y_Limitmax As Integer = 5
Dim Y_Limitmin As Integer = -5
Dim NewDegXCheck As Boolean
Dim NewDegYCheck As Boolean
Dim Temp_Str As String
Dim Timer2Count As Integer = 0
Dim TimerStart As Long
Dim KpX As Double = 1
Dim KiX As Double = 0
Dim KdX As Double = 0
Dim KpY As Double = 1
Dim KiY As Double = 0
Dim KdY As Double = 0
Dim MotorX_Compensator As Double = 0
Dim MotorY_Compensator As Double = 0
Dim KpXLimit As Double = 200
Dim KiXLimit As Double = 100
Dim KdXLimit As Double = 100
Dim KpYLimit As Double = 100
Dim KiYLimit As Double = 100
Dim KdYLimit As Double = 200
Dim MotorX_Limit As Double = 5
Dim MotorY_Limit As Double = 5
'////////////////////////////////////////////////////////////////////////
'////////////////////////////// ส่วนของ Image Processing//////////////////////////
Public Structure SqrGroup
    Dim Top As Integer
    Dim Bottom As Integer
    Dim Left As Integer
    Dim Right As Integer
    Dim X1Pre As Integer
    Dim Y1Pre As Integer
    Dim X2Pre As Integer
    Dim Y2Pre As Integer
    Dim Area As Integer
End Structure
Public Structure Position
    Dim X As Integer
    Dim Y As Integer
End Structure
Dim CropArea As New CROP_SIZE
Dim Position_String As String
Dim Position_A(300, 300) As Char
Dim ObjPos As Position
Public iDevice As Integer = 0 ' device ID ปัจจุบันในเครื่อง
'////////////////////////////////////////////////////////////////////////
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    SerialStat1.Text = "Serial Port Disconnected"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Timer1.Stop()
        LoadDeviceList() 'เรียก Subฟังก์ชั่น LoadDevicelust
    StrTrackingChkB.Enabled = False
    btnVidFormat.Enabled = False
End Sub
'////////////////////////////////////////หน้า Image Process/////////////////////////////////
Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCamConnect.Click
    OpenPreviewWindow()
    Timer3.Enabled = True  'Start Transfer
    btnSaveFromCam.Enabled = True
    btnCamStop.Enabled = True
    btnCamConnect.Enabled = False
    btnVidFormat.Enabled = True
End Sub
Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCamStop.Click
    Timer3.Enabled = False
    ClosePreviewWindow()
    btnCamConnect.Enabled = True
    btnCamStop.Enabled = False
    btnSaveFromCam.Enabled = False
End Sub
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSaveFromCam.Click
    CopyImgToClipboard()
End Sub
Private Sub btnVidFormat_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnVidFormat.Click
    VideoFormat()
End Sub
Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer3.Tick
    Dim data As IDataObject
    Dim bmap As Bitmap
    ' Copy ภาพ ไป clipboard
    SendMessage(hHwnd, WM_CAP_GET_FRAME, 0, 0) 'ดึงภาพจากกล้อง
    SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
    'นำ ภาพ จาก clipboard และ convert มัน ไปเป็น bitmap
    data = Clipboard.GetDataObject()
    If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
        bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)), Image)
        If ShwAreaCB.Checked Then
            Dim gr As Graphics = Graphics.FromImage(bmap)
            Dim p1 As New Pen(Color.Red)
            Dim rect As New Rectangle(CropArea.X - 1, CropArea.Y - 1, CropArea.Width + 1, CropArea.Height + 1)
            gr.DrawRectangle(p1, rect)
        End If
        PicCapture.Image = bmap
    End If
    If StrTrackingChkB.Checked Then
        Dim ImgClass As New ImgClass
        Dim temp_bmp As Bitmap
        Dim Threshold As Single
        Threshold = HScrollBar1.Value
        temp_bmp = ImgClass.CropBitmap(PicCapture.Image, CropArea.X, CropArea.Y, CropArea.Width, CropArea.Height)
        temp_bmp = ImgClass.ResizeBitmap(temp_bmp, 240, 240)
        temp_bmp = ImgClass.Thresholding(temp_bmp, Threshold)
        Position_Check(temp_bmp)
        'PicProcess.Image = temp_bmp
        BallPIDControl()
    Else
        X_err_Accumulate = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Y_err_Accumulate = 0
    X_err_Previous = 0
    Y_err_Previous = 0
    UiX0 = 0
    UiY0 = 0
    UdX = 0
    UdY = 0
    UiX = 0
    UiY = 0
  End If

End Sub

Private Sub HScrollBar1_Scroll(ByVal sender As System.Object, ByVal e As System.Windows.Forms.ScrollEventArgs)
  txtThr.Text = HScrollBar1.Value
  'ThrValue =

End Sub

Public Sub OpenPreviewWindow() ' เปิดแสดงผลที่ picturebox .
  Dim iHeight As Integer = PicCapture.Height 'PicCapture คือ picturebox
  Dim iWidth As Integer = PicCapture.Width
  hHwnd = capCreateCaptureWindowA(iDevice, WS_CHILD, 0, 0, 320, 240, PicCapture.Handle.ToInt32, 0)

  If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then ' คิดต่อกับ device
    SendMessage(hHwnd, WM_CAP_SET_SCALE, 1&, 0&)
    SetWindowPos(hHwnd, HWND_BOTTOM, 0, 0, PicCapture.Width, PicCapture.Height, SWP_NOMOVE Or SWP_NOZORDER) ' ปรับขนาด window ให้เท่ากับใน picturebox

  Else ' การติดต่อdevice Error ให้ปิด window
    DestroyWindow(hHwnd)
    'btnSaveFromCam.Enabled = False

  End If
End Sub

Public Sub LoadDeviceList()
  Dim strName As String = Space(100)
  Dim strVer As String = Space(100)
  Dim bReturn As Boolean
  Dim x As Integer = 0
  Do ' เรียกชื่อ Driver และ version
    bReturn = capGetDriverDescriptionA(x, strName, 100, strVer, 100)
    If bReturn Then lstDevices.Items.Add(strName.Trim) ' ถ้าใช่ device ให้เพิ่มชื่อเข้าไปใน list box
    x += 1
  Loop Until bReturn = False

End Sub

Public Sub ClosePreviewWindow() ' เลิกติดต่อกับ device
  SendMessage(hHwnd, WM_CAP_SET_PREVIEW, False, 0)
  SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0) ' ขั้นตอนสุดท้าย, การปิดแสดงภาพใน โปรแกรมของเรา
  ',เลิกทำการติดต่อ device และ destroy (ทำลาย Object) ที่ เราสร้างขึ้นมาตอนต้นการเขียนโปรแกรม ที่ให้แสดงภาพ และปิดโปรแกรม
  DestroyWindow(hHwnd) ' ปิด window
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Public Sub CopyImgToClipboard()
  SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
End Sub
Public Sub VideoFormat()
  SendMessage(hHwnd, WM_CAP_DLG_VIDEOFORMAT, 0, 0)
End Sub
Private Sub PAXPosScrll_ValueChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles PAXPosScrll.ValueChanged
  CropArea.X = PAXPosScrll.Value
  If CropArea.Width > PicCapture.Width - (CropArea.X - 1) Then
    PAXPosScrll.Value = PicCapture.Width - CropArea.Width
    CropArea.X = PAXPosScrll.Value
  End If
End Sub
Private Sub PAYPosScrll_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PAYPosScrll.ValueChanged
  CropArea.Y = PAYPosScrll.Value
  If CropArea.Height > PicCapture.Height - (CropArea.Y - 1) Then
    PAYPosScrll.Value = PicCapture.Height - CropArea.Height
    CropArea.Y = PAYPosScrll.Value
  End If
End Sub
Private Sub PAWdtScr_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PAWdtScr.ValueChanged
  CropArea.Width = PAWdtScr.Value
  CropArea.Height = PAWdtScr.Value
  If CropArea.Width > PicCapture.Width - (CropArea.X - 1) Then
    PAWdtScr.Value = PicCapture.Width - CropArea.X
    CropArea.Width = PAWdtScr.Value
  End If
  If CropArea.Height > PicCapture.Height - (CropArea.Y - 1) Then
    PAWdtScr.Value = PicCapture.Height - CropArea.Y
    CropArea.Height = PAWdtScr.Value
  End If
End Sub
Private Sub CropLock_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CropLock.CheckedChanged
  If CropLock.Checked Then
    StrTrackingChkB.Enabled = True
    PAXPosScrll.Enabled = False
    PAYPosScrll.Enabled = False
    PAWdtScr.Enabled = False
  Else
    StrTrackingChkB.Enabled = False
    StrTrackingChkB.Checked = False
    PAXPosScrll.Enabled = True
    PAYPosScrll.Enabled = True
    PAWdtScr.Enabled = True
  End If
End Sub
Public Sub Position_Check(ByVal Picture As Image)
  Dim bmp As New Bitmap(Picture)
  Dim x As Integer = 0
  Dim y As Integer = -1 ' เพราะจะไปโดน+=1 จน = 0ในลูปครั้งแรก
  Dim x1(240 * 240) As Integer
  Dim y1(240 * 240) As Integer
  Dim x2(240 * 240) As Integer
  Dim y2(240 * 240) As Integer
  Dim X1Check As Boolean = False 'True คือเจอสีขาว False คือเจอสีดำ
  Dim i As Long = 0
  ' Lock the bitmap's bits.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim rect As New Rectangle(0, 0, bmp.Width, bmp.Height)
Dim bmpData As System.Drawing.Imaging.BitmapData = bmp.LockBits(rect, _
   Drawing.Imaging.ImageLockMode.ReadWrite, bmp.PixelFormat)
' Get the address of the first line.
Dim ptr As IntPtr = bmpData.Scan0
' Declare an array to hold the bytes of the bitmap.
' This code is specific to a bitmap with 24 bits per pixels.
Dim bytes As Integer = bmpData.Stride * bmp.Height
Dim rgbValues(bytes - 1) As Byte
Dim Group(240 * 240) As SqrGroup
' Copy the RGB values into the array.
System.Runtime.InteropServices.Marshal.Copy(ptr, rgbValues, 0, bytes)
' Set every third value to 255. A 24bpp image will look red.
For counter As Integer = 0 To rgbValues.Length - 1 Step 4
  If (counter Mod (240 * 4) = 0) Then ' ถ้าขึ้นแถวใหม่
    If X1Check = True Then ' ถ้าวนครบรอบแล้ว เจอแต่สีขาว แต่ไม่เจอสีดำ ให้ทำดังนี้
      x2(i) = x
      y2(i) = y
      i += 1 'ครบคู่ แล้ว จึง บวก i
    End If
    x = 0
    y += 1
    'rgbValues(counter + 1) = 255 ' แถวแรกจะขึ้นสีเขียว
    X1Check = False
    'Picture_String = Picture_String & vbCrLf
  End If
  If rgbValues(counter + 1) > 200 Then 'ตรวจสีเขียว
    If X1Check = False Then ' ก่อนหน้าเป็นสีดำ
      x1(i) = x
      y1(i) = y
      X1Check = True
    End If
  Else
    If X1Check = True Then ' ก่อนหน้าเป็นสีขาว
      x2(i) = x
      y2(i) = y
      X1Check = False
      i += 1 'ครบคู่ แล้ว จึง บวก i
    End If
  End If
  x += 1
Next
'Position_A = Picture_Array
' Copy the RGB values back to the bitmap
System.Runtime.InteropServices.Marshal.Copy(rgbValues, 0, ptr, bytes)
' Unlock the bits.
bmp.UnlockBits(bmpData)
'เริ่มGroup
Dim a As Integer = 1
Dim b As Integer = 1
Group(0).Top = (240 / 2) - 1 ' Initial Group เป็นGroupสามัญ กำหนดให้อยู่ตรงกลางขนาด 2 pix
Group(0).Bottom = (240 / 2) + 1
Group(0).Left = (240 / 2) - 1
Group(0).Right = (240 / 2) + 1
Group(a).Top = y1(0) ' เซ็ตข้อมูลแรก
Group(a).Bottom = y1(0)
Group(a).Left = x1(0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Group(a).Right = x2(0)
Group(a).X1Pre = x1(0)
Group(a).X2Pre = x2(0)
Group(a).Y1Pre = y1(0)
Group(a).Y2Pre = y2(0)
a += 1
For i = 1 To 240 * 240 ' ไปเริ่มเช็คข้อมูลที่ 2 ได้เลย
  If x1(i) + x2(i) + y1(i) + y2(i) = 0 Then ' End of Data
    Exit For
  End If
  If Group(a).Y1Pre = y1(i) Then ' Case ที่ค่าที่บันทึกอยู่ในชั้น Y เดิม ให้แยกเป็นข้อมูลใหม่
    a += 1 ' Create New group
    Group(a).Top = y1(i)
    Group(a).Bottom = y1(i)
    Group(a).Left = x1(i)
    Group(a).Right = x2(i)
    Group(a).X1Pre = x1(i)
    Group(a).X2Pre = x2(i)
    Group(a).Y1Pre = y1(i)
    Group(a).Y2Pre = y2(i)
  ElseIf (y1(i) - Group(a).Y1Pre) > 10 Then ' Case ที่ขึ้นชั้น Y ใหม่ แต่ ค่า Y อยู่ห่างจากชั้นเดิมเกิน 20 pixel
    a += 1 ' Create New group
    Group(a).Top = y1(i)
    Group(a).Bottom = y1(i)
    Group(a).Left = x1(i)
    Group(a).Right = x2(i)
    Group(a).X1Pre = x1(i)
    Group(a).X2Pre = x2(i)
    Group(a).Y1Pre = y1(i)
    Group(a).Y2Pre = y2(i)

    'b = a ' เพื่อตัดการตรวจGroup ที่จบไปแล้ว
  Else ' ไม่ตรงตามเงื่อนไขของY
    For c = 1 To a ' เทียบข้อมูลกับทุกๆGroup(กรณี Groupที่จะตรวจมีมากกว่า1 ถ้า มีกรุปเดียว จะ ไม่เข้าลูป)
      If Math.Abs(x1(i) - Group(c).X1Pre) < 10 And Math.Abs(x2(i) - Group(c).X2Pre) < 10 _
      And Math.Abs(y1(i) - Group(c).Y1Pre) < 10 Then 'ตรวจระยะเยื้องมากกว่า 10 pixel ไหม
        If x1(i) < Group(c).Left Then
          Group(c).Left = x1(i)
        End If
        If x2(i) > Group(c).Right Then
          Group(c).Right = x2(i)
        End If
        Group(c).Bottom = y1(i)
        Group(c).X1Pre = x1(i)
        Group(c).X2Pre = x2(i)
        Group(c).Y1Pre = y1(i)
        Group(c).Y2Pre = y2(i)
        Exit For
      ElseIf c = a Then
        a += 1 ' Create New group
        Group(a).Top = y1(i)
        Group(a).Bottom = y1(i)
        Group(a).Left = x1(i)
        Group(a).Right = x2(i)
        Group(a).X1Pre = x1(i)
        Group(a).X2Pre = x2(i)
        Group(a).Y1Pre = y1(i)
        Group(a).Y2Pre = y2(i)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Exit For
            End If
        Next
    End If
Next 'ปิดลูปการGroup
Dim Group_select As Integer = 0
'หา Group ที่พื้นที่มากที่สุด
For count = 0 To a
    Group(count).Area = (Group(count).Right - Group(count).Left) * (Group(count).Bottom - Group(count).Top)
    If Group(count).Area > Group(Group_select).Area Then
        Group_select = count
    End If
    'Dim tem_rect As New Rectangle(Group(d).Left, Group(d).Top, Group(d).Right - Group(d).Left, Group(d).Bottom - Group(d).Top)
    'gr.DrawRectangle(p1, tem_rect)
Next
Dim gr As Graphics = Graphics.FromImage(bmp)
Dim p1 As New Pen(Color.Red)
Dim tem_rect As New Rectangle(Group(Group_select).Left, Group(Group_select).Top, Group(Group_select).Right - Group(Group_select).Left, Group(Group_select).Bottom -
Group(Group_select).Top)
gr.DrawRectangle(p1, tem_rect)
'ส่งค่าPosition ของ Groupนั้นออกมา
ObjPos.X = (Group(Group_select).Left + Group(Group_select).Right) / 2
ObjPos.Y = (Group(Group_select).Top + Group(Group_select).Bottom) / 2
ObjPosLabel.Text = "(" & ObjPos.X & "," & ObjPos.Y & ")"
PicProcess.Image = bmp
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.1.2 โปรแกรมการควบคุมมุมระนาบ

```
'//////////////////////////////////////////////////////////////////////////////////
'////////////////////////////////////หน้าCalibration/////////////////////////////////
Private Sub StartToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StartToolStripMenuItem.Click
  SerialPort1.ReadTimeout = 200
  SerialPort1.WriteTimeout = 100
  SerialPort1.Open()
  SerialStat1.Text = "Serial Port Connected"
  Timer1.Start()
  'Timer1.Interval = 50
End Sub
Private Sub StopToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StopToolStripMenuItem.Click
  Timer1.Stop()
  SerialStat1.Text = "Serial Port Disconnected"
  SerialPort1.Close()
End Sub
Private Sub Button50_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDegSave.Click
  If RBXcalib.Checked Then
    mode = "1"  '# mode Code
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    If DegreeComboBox1.SelectedIndex = 0 Then
      command2 = "2"
      X_Plate.ADCTop = X_Plate.ADC 'ADCx
    ElseIf DegreeComboBox1.SelectedIndex = 1 Then
      command2 = "1"
      X_Plate.ADCOrigin = X_Plate.ADC
    ElseIf DegreeComboBox1.SelectedIndex = 2 Then
      command2 = "3"
      X_Plate.ADCBottom = X_Plate.ADC
    End If
    Linear_Predict()
    command1 = "3"
    Send = String.Concat(Str, mode, command1, command2, fin2, fin)
    SerialPort1.Write(Send)
  ElseIf RBYcalib.Checked Then
    mode = "1"  '# mode Code
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    If DegreeComboBox1.SelectedIndex = 0 Then
      command2 = "2"
      Y_Plate.ADCTop = Y_Plate.ADC
    ElseIf DegreeComboBox1.SelectedIndex = 1 Then
      command2 = "1"
      Y_Plate.ADCOrigin = Y_Plate.ADC
    ElseIf DegreeComboBox1.SelectedIndex = 2 Then
      command2 = "3"
      Y_Plate.ADCBottom = Y_Plate.ADC
    End If
    Linear_Predict()
    command1 = "4"
    Send = String.Concat(Str, mode, command1, command2, fin2, fin)
    SerialPort1.Write(Send)
  End If
End Sub
Public Sub Linear_Predict()
  If X_Plate.ADCOrigin <> 0 And X_Plate.ADCTop <> 0 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If X_Plate.ADCTop > X_Plate.ADCOrigin Then
      X_Plate.MPos = X_max / (X_Plate.ADCTop - X_Plate.ADCOrigin)
      X_Plate.CPos = (-1) * X_Plate.ADCOrigin * X_Plate.MPos
      X_Plate.inverseADC_Deg = True
    Else
      X_Plate.MPos = (-1) * (X_max / (X_Plate.ADCOrigin - X_Plate.ADCTop))
      X_Plate.CPos = (-1) * X_Plate.ADCOrigin * X_Plate.MPos
      X_Plate.inverseADC_Deg = False
    End If
  End If
  If X_Plate.ADCOrigin <> 0 And X_Plate.ADCBottom <> 0 Then
    If X_Plate.ADCOrigin > X_Plate.ADCBottom Then
      X_Plate.MNeg = (0 - X_min) / (X_Plate.ADCOrigin - X_Plate.ADCBottom)
      X_Plate.CNeg = (-1) * X_Plate.ADCOrigin * X_Plate.MNeg
      X_Plate.inverseADC_Deg = True
    Else
      X_Plate.MNeg = X_min / (X_Plate.ADCBottom - X_Plate.ADCOrigin)
      X_Plate.CNeg = (-1) * X_Plate.ADCOrigin * X_Plate.MNeg
      X_Plate.inverseADC_Deg = False
    End If
  End If
  If X_Plate.MPos <> 0 And X_Plate.MNeg <> 0 Then
    X_Plate.MMean = (X_Plate.MPos + X_Plate.MNeg) / 2
    X_Plate.CMean = (X_Plate.CPos + X_Plate.CNeg) / 2
  End If
  If Y_Plate.ADCOrigin <> 0 And Y_Plate.ADCTop <> 0 Then
    If Y_Plate.ADCTop > Y_Plate.ADCOrigin Then
      Y_Plate.MPos = Y_max / (Y_Plate.ADCTop - Y_Plate.ADCOrigin)
      Y_Plate.CPos = (-1) * Y_Plate.ADCOrigin * Y_Plate.MPos
      Y_Plate.inverseADC_Deg = True
    Else
      Y_Plate.MPos = (-1) * (Y_max / (Y_Plate.ADCOrigin - Y_Plate.ADCTop))
      Y_Plate.CPos = (-1) * Y_Plate.ADCOrigin * Y_Plate.MPos
      Y_Plate.inverseADC_Deg = False
    End If
  End If
  If Y_Plate.ADCOrigin <> 0 And Y_Plate.ADCBottom <> 0 Then
    If Y_Plate.ADCOrigin > Y_Plate.ADCBottom Then
      Y_Plate.MNeg = (0 - Y_min) / (Y_Plate.ADCOrigin - Y_Plate.ADCBottom)
      Y_Plate.CNeg = (-1) * Y_Plate.ADCOrigin * Y_Plate.MNeg
      Y_Plate.inverseADC_Deg = True
    Else
      Y_Plate.MNeg = Y_min / (Y_Plate.ADCBottom - Y_Plate.ADCOrigin)
      Y_Plate.CNeg = (-1) * Y_Plate.ADCOrigin * Y_Plate.MNeg
      Y_Plate.inverseADC_Deg = False
    End If
  End If
  If Y_Plate.MPos <> 0 And Y_Plate.MNeg <> 0 Then
    Y_Plate.MMean = (Y_Plate.MPos + Y_Plate.MNeg) / 2
    Y_Plate.CMean = (Y_Plate.CPos + Y_Plate.CNeg) / 2
  End If
  TBcomADCxTop.Text = X_Plate.ADCTop
  TBcomADCxOri.Text = X_Plate.ADCOrigin
  TBcomADCxBottom.Text = X_Plate.ADCBottom

  TBcomADCyTop.Text = Y_Plate.ADCTop
  TBcomADCyOri.Text = Y_Plate.ADCOrigin
  TBcomADCyBottom.Text = Y_Plate.ADCBottom
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub StpUPbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StpUPbtn.Click
  If RBXcalib.Checked Then
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    mode = "1"  '# mode Code
    command1 = "1"
    command2 = "1"
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
  ElseIf RBYcalib.Checked Then
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    mode = "1"  '# mode Code
    command1 = "2"
    command2 = "1"
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
  End If
End Sub
Private Sub StpDOWNbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StpDOWNbtn.Click
  If RBXcalib.Checked Then
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    command1 = "1"
    command2 = "2"
    mode = "1"  '# mode Code
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
  ElseIf RBYcalib.Checked Then
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    mode = "1"  '# mode Code
    command1 = "2"
    command2 = "2"
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
  End If
End Sub
Private Sub UPbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UPbtn.Click
  If RBXcalib.Checked Then
    mode = "1"  '# mode Code
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    command1 = "1"
    command2 = "3"
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
  ElseIf RBYcalib.Checked Then
    mode = "1"  '# mode Code
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    command1 = "2"
    command2 = "3"
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
  End If
End Sub
Private Sub DOWNbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DOWNbtn.Click
  If RBXcalib.Checked Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mode = "1"  '# mode Code
        Dim command1 As Char   'Command1 Code
        Dim command2 As Char   'Command2 Code
        command1 = "1"
        command2 = "4"
        Send = (Str & mode & command1 & command2 & fin2 & fin)
        SerialPort1.Write(Send)
    ElseIf RBYcalib.Checked Then
        mode = "1"  '# mode Code
        Dim command1 As Char   'Command1 Code
        Dim command2 As Char   'Command2 Code
        command1 = "2"
        command2 = "4"
        Send = (Str & mode & command1 & command2 & fin2 & fin)
        SerialPort1.Write(Send)
    End If
End Sub
Private Sub E_Stop1btn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles E_Stop1btn.Click
    mode = "2"  '# mode Code
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    command1 = "3"
    command2 = "1"
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
End Sub
Private Sub btnToExcel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnToExcel.Click
    Dim oExcel As Object = CreateObject("Excel.Application")
    Dim wBook As Object
    Dim wSheet As Object
    Dim Temp_Array() As String
    Dim Temp_Sub_Array() As String
    Dim i As Integer
    Dim rowIndex As Integer = 1
    wBook = oExcel.Workbooks.Add()
    wSheet = wBook.ActiveSheet()
    wSheet.Columns.AutoFit()
    wSheet.Cells(rowIndex, 1) = "mSec."
    If RBTimTestXUp.Checked Then
        wSheet.Cells(rowIndex, 2) = "Test System X Up"
    ElseIf RBTimTestYUp.Checked Then
        wSheet.Cells(rowIndex, 2) = "Test System Y Up"
    ElseIf RBTimTestXDown.Checked Then
        wSheet.Cells(rowIndex, 2) = "Test System X Down"
    ElseIf RBTimTestYDown.Checked Then
        wSheet.Cells(rowIndex, 2) = "Test System Y Down"
    End If
    Temp_Array = Temp_Str.Split(vbCrLf)
    For i = 0 To Temp_Array.Length - 2 ' Cut the Last data (might error because data after vbcrlf)
        rowIndex = rowIndex + 1
        Temp_Sub_Array = Temp_Array(i).Split(",")
        wSheet.Cells(rowIndex, 1) = Temp_Sub_Array(0)
        wSheet.Cells(rowIndex, 2) = Temp_Sub_Array(1)
    Next i
    oExcel.Visible = True
End Sub
Private Sub E_Stop2btn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles E_Stop2btn.Click
    mode = "2"  '# mode Code
    Dim command1 As Char   'Command1 Code
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Dim command2 As Char   'Command2 Code
    command1 = "3"
    command2 = "1"
    StartSystemTest = False
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
End Sub
Private Sub TimConStrbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TimConStrbtn.Click
    Dim myDateTime As DateTime = DateTime.Now
    Temp_Str = vbNullString
    TimerStart = myDateTime.Ticks
    'Timer2.Start()
    StartSystemTest = True
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Send = (Str & "01" & fin2 & fin)
    SerialPort1.Write(Send)
    Dim Received As String
    Dim Received2() As String
    Dim Received3() As String
    Dim i As Integer
    ReadBuffer = SerialPort1.ReadExisting
    While Not ReadBuffer.EndsWith(vbCr)
      If SerialPort1.BytesToRead > 0 Then
        ReadBuffer += SerialPort1.ReadExisting
      End If
    End While
    'Received = SerialPort1.ReadExisting ' Input Data From Controller
    Received = ReadBuffer
    If Received <> vbNullString Then
      Received2 = Received.Split(vbCr)
      For i = 0 To Received2.Length - 1
        If Received2(i).StartsWith("#") Then 'Header right
          If InStr(1, Received2(i), "D") = 2 Then ' Print_All
            If InStr(1, Received2(i), "x") = 3 Then ' X,Print_All
              Received3 = Received2(i).Split(",")
              X_Plate.ADC = Convert.ToInt16(Received3(0).Substring(3, 4))
              X_Plate.Deg = Convert.ToDouble(Received3(1).Substring(0, 5))
              NewDegXCheck = True
              ADCxTBu.Text = X_Plate.ADC
              XTBu.Text = X_Plate.Deg 'Convert.ToDouble(Received2(i).Substring(3, 5))
            ElseIf InStr(1, Received2(i), "y") = 3 Then ' Y,Print_All
              Received3 = Received2(i).Split(",")
              Y_Plate.ADC = Convert.ToInt16(Received3(0).Substring(3, 4))
              Y_Plate.Deg = Convert.ToDouble(Received3(1).Substring(0, 5))
              ADCyTBu.Text = Y_Plate.ADC
              YTBu.Text = Y_Plate.Deg 'Convert.ToDouble(Received2(i).Substring(3, 5))
              NewDegYCheck = True
            End If
          ElseIf InStr(1, Received2(i), "R") = 2 Then ' Setting Read
            TextBox31.Text = "a"
            If InStr(1, Received2(i), "x") = 3 Then
              If InStr(1, Received2(i), "T") = 4 Then
                TBuconADCxTop.Text = Convert.ToInt16(Received2(i).Substring(4, 4))
                TextBox31.Text = "a"
              ElseIf InStr(1, Received2(i), "O") = 4 Then
                TBuconADCxOri.Text = Convert.ToInt16(Received2(i).Substring(4, 4))
              ElseIf InStr(1, Received2(i), "B") = 4 Then
                TBuconADCxBottom.Text = Convert.ToInt16(Received2(i).Substring(4, 4))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
          ElseIf InStr(1, Received2(i), "y") = 3 Then
            If InStr(1, Received2(i), "T") = 4 Then
              TBuconADCyTop.Text = Convert.ToInt16(Received2(i).Substring(4, 4))
            ElseIf InStr(1, Received2(i), "O") = 4 Then
              TBuconADCyOri.Text = Convert.ToInt16(Received2(i).Substring(4, 4))
            ElseIf InStr(1, Received2(i), "B") = 4 Then
              TBuconADCyBottom.Text = Convert.ToInt16(Received2(i).Substring(4, 4))
            End If
          End If
        ElseIf InStr(1, Received2(i), "K") = 2 Then
          If InStr(1, Received2(i), "X") = 4 Then
            If InStr(1, Received2(i), "p") = 3 Then
              RBKpX.Text = Convert.ToDouble(Received2(i).Substring(4, 7))
            ElseIf InStr(1, Received2(i), "i") = 3 Then
              RBKiX.Text = Convert.ToDouble(Received2(i).Substring(4, 7))
            ElseIf InStr(1, Received2(i), "d") = 3 Then
              RBKdX.Text = Convert.ToDouble(Received2(i).Substring(4, 7))
            ElseIf InStr(1, Received2(i), "m") = 3 Then
              RBMotXCom.Text = Convert.ToDouble(Received2(i).Substring(4, 7))
            End If
          ElseIf InStr(1, Received2(i), "Y") = 4 Then
            If InStr(1, Received2(i), "p") = 3 Then
              RBKpY.Text = Convert.ToDouble(Received2(i).Substring(4, 7))
            ElseIf InStr(1, Received2(i), "i") = 3 Then
              RBKiY.Text = Convert.ToDouble(Received2(i).Substring(4, 7))
            ElseIf InStr(1, Received2(i), "d") = 3 Then
              RBKdY.Text = Convert.ToDouble(Received2(i).Substring(4, 7))
            ElseIf InStr(1, Received2(i), "m") = 3 Then
              RBMotYCom.Text = Convert.ToDouble(Received2(i).Substring(4, 7))
            End If
          End If
        ElseIf InStr(1, Received2(i), "S") = 5 Then
          LabelFeedbackCmd.Text = Received2(i)
        End If
        'End If
        If ReadBuffer.EndsWith(vbCr) Then
          ReadBuffer = ""
        End If ' เคลียค่าได้ไม่มีปัญหา? เพราะ เอาข้อมูลไปใส่ใน Received แล้ว
      End If
    Next i
    'Label9.Text = Received2(3)
  End If
  If StartSystemTest Then
    System_Test()
  End If
  If X_Plate.inverseADC_Deg Then
    If X_Plate.MMean <> 0 Then
      X_Plate.VBDeg = (X_Plate.MMean * X_Plate.ADC) + X_Plate.CMean
      XTBCom.Text = X_Plate.VBDeg
    ElseIf X_Plate.ADC > X_Plate.ADCOrigin Then
      X_Plate.VBDeg = (X_Plate.MPos * X_Plate.ADC) + X_Plate.CPos
      XTBCom.Text = X_Plate.VBDeg
    Else
      X_Plate.VBDeg = (X_Plate.MNeg * X_Plate.ADC) + X_Plate.CNeg
      XTBCom.Text = X_Plate.VBDeg
    End If
  Else
    If X_Plate.MMean <> 0 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            X_Plate.VBDeg = (X_Plate.MMean * X_Plate.ADC) + X_Plate.CMean
            XTBCom.Text = X_Plate.VBDeg
        ElseIf X_Plate.ADC < X_Plate.ADCOrigin Then
            X_Plate.VBDeg = (X_Plate.MPos * X_Plate.ADC) + X_Plate.CPos
            XTBCom.Text = X_Plate.VBDeg
        Else
            X_Plate.VBDeg = (X_Plate.MNeg * X_Plate.ADC) + X_Plate.CNeg
            XTBCom.Text = X_Plate.VBDeg
        End If
    End If
    If Y_Plate.inverseADC_Deg Then
        If Y_Plate.MMean <> 0 Then
            Y_Plate.VBDeg = (Y_Plate.MMean * Y_Plate.ADC) + Y_Plate.CMean
            YTBCom.Text = Y_Plate.VBDeg
        ElseIf Y_Plate.ADC > Y_Plate.ADCOrigin Then
            Y_Plate.VBDeg = (Y_Plate.MPos * Y_Plate.ADC) + Y_Plate.CPos
            YTBCom.Text = Y_Plate.VBDeg
        Else
            Y_Plate.VBDeg = (Y_Plate.MNeg * Y_Plate.ADC) + Y_Plate.CNeg
            YTBCom.Text = Y_Plate.VBDeg
        End If
    Else
        If Y_Plate.MMean <> 0 Then
            Y_Plate.VBDeg = (Y_Plate.MMean * Y_Plate.ADC) + Y_Plate.CMean
            YTBCom.Text = Y_Plate.VBDeg
        ElseIf Y_Plate.ADC < Y_Plate.ADCOrigin Then
            Y_Plate.VBDeg = (Y_Plate.MPos * Y_Plate.ADC) + Y_Plate.CPos
            YTBCom.Text = Y_Plate.VBDeg
        Else
            Y_Plate.VBDeg = (Y_Plate.MNeg * Y_Plate.ADC) + Y_Plate.CNeg
            YTBCom.Text = Y_Plate.VBDeg
        End If
    End If
End Sub
Public Sub System_Test()
    Dim myDateTime As DateTime = DateTime.Now
    Dim TimerCount As Long
    TimerCount = myDateTime.Ticks
    If RBTimTestXUp.Checked Or RBTimTestXDown.Checked Then
        If NewDegXCheck Then 'ถ้าเป็นDegreeที่รับเข้ามาใหม่ ไม่ใช่ค่าซ้ำ
            Temp_Str = Temp_Str & (TimerCount - TimerStart) / 10000 & "," & X_Plate.Deg & vbCrLf'แสดงผลของ ADCX ADCY
            NewDegXCheck = False
        End If
    ElseIf RBTimTestYUp.Checked Or RBTimTestYDown.Checked Then
        If NewDegYCheck Then 'ถ้าเป็นDegreeที่รับเข้ามาใหม่ ไม่ใช่ค่าซ้ำ
            Temp_Str = Temp_Str & (TimerCount - TimerStart) / 10000 & "," & Y_Plate.Deg & vbCrLf'แสดงผลของ ADCX ADCY
            NewDegYCheck = False
        End If
    End If
End Sub
'Dim ffname As String = "C:\Users\Neo\Desktop\"
Private Sub btnsav_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsav.Click
    Dim saveFileDialog1 As New SaveFileDialog()
    saveFileDialog1.Filter = "Text File|*.txt|Save File|*.SAV"
    saveFileDialog1.Title = "Save Setting's Parametes "
    saveFileDialog1.ShowDialog()
    ' If the file name is not an empty string open it for saving.
    If saveFileDialog1.FileName <> "" Then
        ' Saves the Text via a StreamWriter created by the OpenFile method.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Dim fs As New System.IO.StreamWriter(saveFileDialog1.OpenFile())
        ' Saves the Image in the appropriate Format based upon the
        ' file type selected in the dialog box.
        ' NOTE that the FilterIndex property is one-based.
        Select Case saveFileDialog1.FilterIndex
            Case 1
                fs.WriteLine(X_Plate.ADCTop)
                fs.WriteLine(X_Plate.ADCOrigin)
                fs.WriteLine(X_Plate.ADCBottom)
                fs.WriteLine(Y_Plate.ADCTop)
                fs.WriteLine(Y_Plate.ADCOrigin)
                fs.WriteLine(Y_Plate.ADCBottom)
            Case 2
                fs.WriteLine(X_Plate.ADCTop)
                fs.WriteLine(X_Plate.ADCOrigin)
                fs.WriteLine(X_Plate.ADCBottom)
                fs.WriteLine(Y_Plate.ADCTop)
                fs.WriteLine(Y_Plate.ADCOrigin)
                fs.WriteLine(Y_Plate.ADCBottom)
        End Select
        fs.Close()
    End If
End Sub
Private Sub btnLod_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLod.Click
    Dim myStream As Stream = Nothing
    Dim openFileDialog1 As New OpenFileDialog()
    openFileDialog1.InitialDirectory = "c:\"
    openFileDialog1.Filter = "txt files (*.txt)|*.txt|Save files (*.SAV)|*.SAV|All files (*.*)|*.*"
    openFileDialog1.FilterIndex = 3
    openFileDialog1.RestoreDirectory = True
    If openFileDialog1.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
        Try
            myStream = openFileDialog1.OpenFile()
            If (myStream IsNot Nothing) Then
                ' Insert code to read the stream here.
                Dim fo As New System.IO.StreamReader(myStream)
                X_Plate.ADCTop = CInt(fo.ReadLine())
                X_Plate.ADCOrigin = CInt(fo.ReadLine())
                X_Plate.ADCBottom = CInt(fo.ReadLine())
                Y_Plate.ADCTop = CInt(fo.ReadLine())
                Y_Plate.ADCOrigin = CInt(fo.ReadLine())
                Y_Plate.ADCBottom = CInt(fo.ReadLine())
                TBcomADCxTop.Text = X_Plate.ADCTop
                TBcomADCxOri.Text = X_Plate.ADCOrigin
                TBcomADCxBottom.Text = X_Plate.ADCBottom
                TBcomADCyTop.Text = Y_Plate.ADCTop
                TBcomADCyOri.Text = Y_Plate.ADCOrigin
                TBcomADCyBottom.Text = Y_Plate.ADCBottom
                Linear_Predict()
            End If
        Catch Ex As Exception
            MessageBox.Show("Cannot read file from disk. Original error: " & Ex.Message)
        Finally
            ' Check this again, since we need to make sure we didn't throw an exception on open.
            If (myStream IsNot Nothing) Then
                myStream.Close()
            End If
        End Try
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub btnTest2Text_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnTest2Text.Click
    Dim Temp_Array() As String
    Dim Temp_Sub_Array() As String
    Dim i As Integer
    Dim rowIndex As Integer = 1
    Dim time As String = "time = ["
    Dim degree As String = "degree = ["
    Dim TempText_string As String
    Temp_Array = Temp_Str.Split(vbCrLf)
    For i = 0 To Temp_Array.Length - 2 ' Cut the Last data (might error because data after vbcrlf)
        rowIndex = rowIndex + 1
        Temp_Sub_Array = Temp_Array(i).Split(",")
        time += Temp_Sub_Array(0) + vbCrLf
        degree += Temp_Sub_Array(1) + vbCrLf
    Next i
    time += "];" + vbCrLf
    degree += "];" + vbCrLf
    TempText_string = time + degree + "plot(time,degree)" + vbCrLf + "grid" + vbCrLf
    If RBTimTestXUp.Checked Then
        TempText_string += "xlabel('time(ms)')" + vbCrLf _
        + "ylabel('degree')" + vbCrLf _
        + "title('System X Up')" + vbCrLf
    ElseIf RBTimTestYUp.Checked Then
        TempText_string += "xlabel('time(ms)')" + vbCrLf _
        + "ylabel('degree')" + vbCrLf _
        + "title('System Y Up')" + vbCrLf
    ElseIf RBTimTestXDown.Checked Then
        TempText_string += "xlabel('time(ms)')" + vbCrLf _
        + "ylabel('degree')" + vbCrLf _
        + "title('System X Down')" + vbCrLf
    ElseIf RBTimTestYDown.Checked Then
        TempText_string += "xlabel('time(ms)')" + vbCrLf _
        + "ylabel('degree')" + vbCrLf _
        + "title('System Y Down')" + vbCrLf
    End If
    Dim saveFileDialog1 As New SaveFileDialog()
    saveFileDialog1.Filter = "Text File|*.txt"
    saveFileDialog1.Title = "Export to MATLAB"
    saveFileDialog1.ShowDialog()
    ' If the file name is not an empty string open it for saving.
    If saveFileDialog1.FileName <> "" Then
        ' Saves the Text via a StreamWriter created by the OpenFile method.
        Dim fs As New System.IO.StreamWriter(saveFileDialog1.OpenFile())
        ' Saves the Image in the appropriate Format based upon the
        ' file type selected in the dialog box.
        ' NOTE that the FilterIndex property is one-based.
        Select Case saveFileDialog1.FilterIndex
            Case 1
                fs.Write(TempText_string)
        End Select
        fs.Close()
    End If
End Sub
Private Sub ListMotorXSp_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListMotorXSp.SelectedIndexChanged
    mode = "2"  '# mode Code
    Dim command1 As Char    'Command1 Code
    Dim command2 As Char    'Command2 Code
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    command1 = "4"
    'command2 = "1"
    If ListMotorXSp.SelectedIndex = 0 Then
      command2 = "1"
    ElseIf ListMotorXSp.SelectedIndex = 1 Then
      command2 = "2"
    ElseIf ListMotorXSp.SelectedIndex = 2 Then
      command2 = "3"
    ElseIf ListMotorXSp.SelectedIndex = 3 Then
      command2 = "4"
    End If
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
  End Sub
  Private Sub ListMotorYSp_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListMotorYSp.SelectedIndexChanged
    mode = "2"  '# mode Code
    Dim command1 As Char   'Command1 Code
    Dim command2 As Char   'Command2 Code
    command1 = "4"
    'command2 = "1"
    If ListMotorYSp.SelectedIndex = 0 Then
      command2 = "5"
    ElseIf ListMotorYSp.SelectedIndex = 1 Then
      command2 = "6"
    ElseIf ListMotorYSp.SelectedIndex = 2 Then
      command2 = "7"
    ElseIf ListMotorYSp.SelectedIndex = 3 Then
      command2 = "8"
    End If
    Send = (Str & mode & command1 & command2 & fin2 & fin)
    SerialPort1.Write(Send)
  End Sub
  Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
    Send = (Str & "02" & fin2 & fin)
    SerialPort1.Write(Send)
  End Sub
  Private Sub btnPc2Con_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPc2Con.Click
    mode = "1"  '# mode Code
    Dim command1 As Char   'Command1 Code
    'Dim command2 As Char   'Command2 Code
    command1 = "5"
    'command2 = "1"
    Send = (Str & mode & command1 & "1" & String.Format("{0:D4}", X_Plate.ADCTop) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "2" & String.Format("{0:D4}", X_Plate.ADCOrigin) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "3" & String.Format("{0:D4}", X_Plate.ADCBottom) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "4" & String.Format("{0:D4}", Y_Plate.ADCTop) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "5" & String.Format("{0:D4}", Y_Plate.ADCOrigin) & fin2 & fin)
    SerialPort1.Write(Send)

    Send = (Str & mode & command1 & "6" & String.Format("{0:D4}", Y_Plate.ADCBottom) & fin2 & fin)
    SerialPort1.Write(Send)

  End Sub

  Private Sub btnCon2PC_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCon2PC.Click
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    X_Plate.ADCTop = CInt(TBuconADCxTop.Text)
    X_Plate.ADCOrigin = CInt(TBuconADCxOri.Text)
    X_Plate.ADCBottom = CInt(TBuconADCxBottom.Text)
    Y_Plate.ADCTop = CInt(TBuconADCyTop.Text)
    Y_Plate.ADCOrigin = CInt(TBuconADCyOri.Text)
    Y_Plate.ADCBottom = CInt(TBuconADCyBottom.Text)
    Linear_Predict()
End Sub
Private Sub btnSetPStr_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSetPStr.Click
    mode = "3"  '# mode Code
    'Dim command1 As Char = "3"
    Dim XSetP As String
    Dim YSetP As String
    If TBSetPX.Text = vbNullString Then
        TBSetPX.Text = 0
    End If
    If TBSetPY.Text = vbNullString Then
        TBSetPY.Text = 0
    End If
    If TBSetPX.Text >= 0 Then
        If TBSetPX.Text > X_max Then
            XSetP = "+" & String.Format("{0:D2}", X_max * 10)
            TBSetPX.Text = X_max
        Else
            XSetP = "+" & String.Format("{0:D2}", CInt(CDbl(TBSetPX.Text) * 10))
        End If
        'DesigX = "+"
    Else
        If TBSetPX.Text < X_min Then
            XSetP = String.Format("{0:D2}", X_min * 10)
            TBSetPX.Text = X_min
        Else
            XSetP = String.Format("{0:D2}", CInt(CDbl(TBSetPX.Text) * 10))
            'XSetP = TBSetPX.Text
        End If
    End If
    If TBSetPY.Text >= 0 Then
        If TBSetPY.Text > Y_max Then
            YSetP = "+" & String.Format("{0:D2}", Y_max * 10)
            TBSetPY.Text = Y_max
        Else
            YSetP = "+" & String.Format("{0:D2}", CInt(CDbl(TBSetPY.Text) * 10))
        End If
        'DesigX = "+"
    Else
        If TBSetPY.Text < Y_min Then
            YSetP = String.Format("{0:D2}", Y_min * 10)
            TBSetPX.Text = Y_min
        Else
            YSetP = String.Format("{0:D2}", CInt(CDbl(TBSetPY.Text) * 10))
            'XSetP = TBSetPX.Text
        End If
    End If
    'Send = (Str & mode & command1 & "X" & String.Format("{0:D2}", XSetP * 10) _
    '  & "Y" & String.Format("{0:D2}", YSetP * 10) & fin2 & fin)
    Send = (Str & mode & "X" & XSetP _
     & "Y" & YSetP & fin2 & fin)
    SerialPort1.Write(Send)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Send = (Str & "41" & fin2 & fin)
    SerialPort1.Write(Send)
    Dim myDateTime As DateTime = DateTime.Now
    Temp_Str = vbNullString
    TimerStart = myDateTime.Ticks
    'Timer2.Start()
    StartSystemTest = True
End Sub
Public Sub SetPoint_Send(ByVal SetpointX As Double, ByVal SetpointY As Double)
    mode = "3"  '# mode Code
    'Dim command1 As Char = "3"
    Dim SetPointXString As String
    Dim SetPointYString As String
    If SetpointX >= 0 Then
      If SetpointX > X_max Then
        SetPointXString = "+" & String.Format("{0:D2}", X_max * 10)
        'TBSetPX.Text = X_max
      Else
        SetPointXString = "+" & String.Format("{0:D2}", CInt(SetpointX * 10))
      End If
      'DesigX = "+"
    Else
      If SetpointX < X_min Then
        SetPointXString = String.Format("{0:D2}", X_min * 10)
        'TBSetPX.Text = X_min
      Else
        SetPointXString = String.Format("{0:D2}", CInt(SetpointX * 10))
        'XSetP = TBSetPX.Text
      End If
    End If
    If SetpointY >= 0 Then
      If SetpointX > Y_max Then
        SetPointYString = "+" & String.Format("{0:D2}", Y_max * 10)
        'TBSetPY.Text = Y_max
      Else
        SetPointYString = "+" & String.Format("{0:D2}", CInt(SetpointY * 10))
      End If
      'DesigX = "+"
    Else
      If SetpointY < Y_min Then
        SetPointYString = String.Format("{0:D2}", Y_min * 10)
        TBSetPX.Text = Y_min
      Else
        SetPointYString = String.Format("{0:D2}", CInt(SetpointY * 10))
        'XSetP = TBSetPX.Text
      End If
    End If
    'Send = (Str & mode & command1 & "X" & String.Format("{0:D2}", XSetP * 10) _
    '  & "Y" & String.Format("{0:D2}", YSetP * 10) & fin2 & fin)
    Send = (Str & mode & "X" & SetPointXString _
     & "Y" & SetPointYString & fin2 & fin)
    SerialPort1.Write(Send)
    Label66.Text = Send

    Send = (Str & "41" & fin2 & fin)
    SerialPort1.Write(Send)
End Sub

Private Sub E_Stop3btn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles E_Stop3btn.Click
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Send = (Str & "42" & fin2 & fin)
    SerialPort1.Write(Send)
    StartSystemTest = False
End Sub
Private Sub btnPIDRd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPIDRd.Click
    Send = (Str & "03" & fin2 & fin)
    SerialPort1.Write(Send)
End Sub
Private Sub btnPIDWr_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPIDWr.Click
    mode = "1"  '# mode Code
    Dim command1 As Char   'Command1 Code
    'Dim command2 As Char   'Command2 Code
    command1 = "6"
    'command2 = "1"
    If TBKpX.Text = vbNullString Then
        TBKpX.Text = KpX
    End If
    If TBKiX.Text = vbNullString Then
        TBKiX.Text = KiX
    End If
    If TBKdX.Text = vbNullString Then
        TBKdX.Text = KdX
    End If
    If TBKpY.Text = vbNullString Then
        TBKpY.Text = KpY
    End If
    If TBKiY.Text = vbNullString Then
        TBKiY.Text = KiY
    End If
    If TBKdY.Text = vbNullString Then
        TBKdY.Text = KdY
    End If
    If TBMotXCom.Text = vbNullString Then
        TBMotXCom.Text = MotorX_Compensator
    End If
    If TBMotYCom.Text = vbNullString Then
        TBMotYCom.Text = MotorY_Compensator
    End If
    If TBKpX.Text >= KpXLimit Then
        TBKpX.Text = 1
    End If
    If TBKiX.Text >= KiXLimit Then
        TBKiX.Text = 0
    End If
    If TBKdX.Text >= KdXLimit Then
        TBKdX.Text = 0
    End If
    If TBKpY.Text >= KpYLimit Then
        TBKpY.Text = 1
    End If
    If TBKiY.Text >= KiYLimit Then
        TBKiY.Text = 0
    End If
    If TBKdY.Text >= KdYLimit Then
        TBKdY.Text = 0
    End If
    If TBMotXCom.Text >= MotorX_Limit Then
        TBMotXCom.Text = 0
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If TBMotYCom.Text >= MotorY_Limit Then
      TBMotYCom.Text = 0
    End If
    KpX = TBKpX.Text
    KiX = TBKiX.Text
    KdX = TBKdX.Text
    KpY = TBKpY.Text
    KiY = TBKiY.Text
    KdY = TBKdY.Text
    MotorX_Compensator = TBMotXCom.Text
    MotorY_Compensator = TBMotYCom.Text
    Send = (Str & mode & command1 & "1" & String.Format("{0:00.00000}", KpX) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "2" & String.Format("{0:00.00000}", KiX) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "3" & String.Format("{0:00.00000}", KdX) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "4" & String.Format("{0:00.00000}", KpY) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "5" & String.Format("{0:00.00000}", KiY) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "6" & String.Format("{0:00.00000}", KdY) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "7" & String.Format("{0:00.00000}", MotorX_Compensator) & fin2 & fin)
    SerialPort1.Write(Send)
    Send = (Str & mode & command1 & "8" & String.Format("{0:00.00000}", MotorY_Compensator) & fin2 & fin)
    SerialPort1.Write(Send)
  End Sub
  Public Sub BallPIDGain()
    BallKpX = CDbl(TBBallKpX.Text)
    BallKiX = CDbl(TBBallKiX.Text)
    BallKdX = CDbl(TBBallKdX.Text)
    BallKpY = CDbl(TBBallKpY.Text)
    BallKiY = CDbl(TBBallKiY.Text)
    BallKdY = CDbl(TBBallKdY.Text)
    BallXSetPoint = CDbl(TBBallXSetpoint.Text)
    BallYSetPoint = CDbl(TBBallYSetpoint.Text)
    X_err_now = BallXSetPoint - ObjPos.X
    Y_err_now = BallYSetPoint - ObjPos.Y
    X_err_Accumulate = (X_err_Previous + X_err_now) * (Ts / 2)
    Y_err_Accumulate = (Y_err_Previous + Y_err_now) * (Ts / 2)
    UiX = UiX0 + (BallKiX * X_err_Accumulate)
    UiY = UiY0 + (BallKiY * Y_err_Accumulate)
    TBUiX.Text = UiX
    TBUiY.Text = UiY
    UdX = BallKdX * (X_err_now - X_err_Previous) / Ts
    UdY = BallKdY * (Y_err_now - Y_err_Previous) / Ts
    If UdX <> 0 Then
      TBUdX.Text = X_err_Previous
    End If
    If UdY <> 0 Then
      TBUdY.Text = X_err_Previous
    End If
    TBUdX.Text = (X_err_now - X_err_Previous)
    TBUdY.Text = (X_err_now - X_err_Previous)
    TBUpX.Text = (BallKpX * X_err_now)
    TBUpY.Text = (BallKpY * Y_err_now)
    BallXPIDGain = (-1) * ((BallKpX * X_err_now) + UiX + UdX) '(ballKiX*X_err_Accumulate) ;
    BallYPIDGain = (-1) * ((BallKpY * Y_err_now) + UiY + UdY) '(KiY*Y_err_Accumulate) + );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        X_err_Previous = X_err_now
        Y_err_Previous = Y_err_now
        UiX0 = UiX
        UiY0 = UiY
    End Sub
    Public Sub BallPIDControl()
        BallPIDGain()
        If BallXPIDGain >= X_Limitmax Then
            BallXPIDGain = X_Limitmax
        ElseIf BallXPIDGain <= X_Limitmin Then
            BallXPIDGain = X_Limitmin
        End If
        If BallYPIDGain >= Y_Limitmax Then
            BallYPIDGain = Y_Limitmax
        ElseIf BallYPIDGain <= Y_Limitmin Then
            BallYPIDGain = Y_Limitmin
        End If
        SetPoint_Send(BallXPIDGain, BallYPIDGain)
        'Label66.Text = BallXPIDGain.ToString + BallYPIDGain.ToString
    End Sub
End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 โปรแกรมในส่วนไมโครคอนโทรลเลอร์

ก.2.1 โปรแกรม ADC chennel

```
volatile unsigned int temp_ADC[8];
volatile unsigned char adc_choose = 0;
/*
void init_ADC(void)
{
        //ADCSRA = (1 << 7)|(0 << 5);                                   //ADEN,ADATE
        //ADCSRA |= (0 << ADPS2)|(1 << ADPS1)|(1 << ADPS0);
        //ADCSRA = ADCSRA|0x40; //start the next conversion
        ADCSRA = 0xCE; //ADC on, /64, interrupt unmasked,and started
//      ADCSRA = 0xCB; //ADC on, /8, interrupt unmasked,and started
        ADMUX = (0 << REFS1)|(1 << REFS0)|(adc_choose);
}*/
void init_ADC(void)
{
        ADCSRA = (1 << 7)|(0 << 5);                                   //ADEN,ADATE
        ADCSRA |= (1<< ADPS2)|(1 << ADPS1)|(0 << ADPS0);
}
unsigned int read_ADC(unsigned char adc_Ch)
{
        ADMUX = (0 << REFS1)|(1 << REFS0)|(adc_Ch);
        //delay_ms(10);
        ADCSRA |= 0x40;                                                                 // Start ADC
        while ((ADCSRA & 0x10)==0);                                     // Wait for the AD conversion to complete
        ADCSRA |=0x10;
        return ADCW;
}

void GetAllADCData(void)                                                //If you want ADCData ,get it from uiADCDAta variable ,don't directly call read_ADC(x).
{
        temp_ADC[0] = read_ADC(0);
        temp_ADC[1] = read_ADC(1);
//      uiADCData[2] = read_ADC(2);
//      uiADCData[3] = read_ADC(3);
//      uiADCData[4] = read_ADC(4);
//      uiADCData[5] = read_ADC(5);
//      uiADCData[6] = read_ADC(6);
//      uiADCData[7] = read_ADC(7);
}
```

ก-2.2 โปรแกรมการควบคุมลูกบอลบนระนาบ

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define DIPPIN          PINC
#define DIPDDR                  DDRC
#define DIPPORT         PORTC
#define PulseDDR        DDRD
#define         PulsePIN        PIND
#define PulsePORT       PORTD
#define MtrDDR                  DDRC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MtrPIN PINC
#define MtrPORT PORTC
#define LEDPIN PINB
#define LEDDDR DDRB
#define LEDPORT PORTB
#define ADCPIN PINA
#define ADCDDR DDRA
#define ADCPORT PORTA
#define DIPPIN PINC
#define DIPDDR DDRC
#define DIPPORT PORTC
#define PulseDDR DDRD
#define PulsePIN PIND
#define PulsePORT PORTD
#define MtrDDR DDRC
#define MtrPIN PINC
#define MtrPORT PORTC
#define LEDPIN PINB
#define LEDDDR DDRB
#define LEDPORT PORTB
#define ADCPIN PINA
#define ADCDDR DDRA
#define ADCPORT PORTA
#define MtrX1 4
#define MtrX2 5
#define MtrY1 6
#define MtrY2 7
#define Top 8000 // Pulse freq = 1 kHz
#include "main.h"
#include "delay.h"
#include "usart.h"
#include "ADC.h"
#include "Motor.h"
#define Top_Limit 900
#define Bottom_Limit 150
volatile float DegX_Max = 7.0,DegX_Min = -7.0,DegY_Max = 7.0,DegY_Min= -7.0;
volatile float KpX = 1 ,KpY = 1,KiX = 0 ,KiY = 0,KdX = 0 ,KdY = 0;
volatile float UpX = 0 ,UpY = 0,UiX = 0 ,UiY = 0,UiX0 = 0 ,UiY0 = 0,UdX = 0 ,UdY = 0;
volatile float X_err_Now,Y_err_Now;
volatile float X_err_Previous,Y_err_Previous;
volatile float X_err_Accumulate,Y_err_Accumulate;
volatile float X_PID_Gain = 0,Y_PID_Gain = 0;
volatile float X_Set_Point = 0,Y_Set_Point = 0;
volatile int Timer_Counter = 0;
volatile char PID_Timer_Flag = 0;
volatile float Ts = 0.0011; //Sampling Time 1.1 ms
float MotorX_Volt_Compensator = 2; // Volt
float MotorY_Volt_Compensator = 2; // Volt
volatile char code[100],X_Norm = 0,Y_Norm = 0,ledc;
//char KxOper=1,KyOper=1,CxOper = 1,CyOper = 1,XOper = 1,YOper=1;
volatile int SaveX[3],SaveY[3];
volatile float KxPos=0,KyPos=0,CxPos=0,CyPos=0,KxNev=0,KyNev=0,CxNev=0,CyNev=0,KxMean=0,KyMean=0,CxMean=0,CyMean=0;
volatile float XDeg=0,YDeg=0;
char System_Test_Select = 'N';
volatile char PID_Enable = 0;
volatile char Emergency_Flag = 0;

void Plate_X_Control(void);
void Plate_Y_Control(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Save_X(void);
void Save_Y(void);
void Degree_Cal(void);
void Degree_Print(void);
void Toggle_LED(void);
void Linear_Function_Predict(void);
void SetPoint(void);
void PID_Gain_Print(void);
void Write_PID(void);
void Emergency_Stop(void);

void Plate_Test(void);
oid Go_to_Home(void);
//**************************************Main Program*************************************************//

nt main(void){
        volatile unsigned char i;
        init_ADC();
        usart_init(USART_BAUD_SELECT(USART_BAUD_RATE,F_CPU));
        TCCR0 = (0<<CS02)(1<<CS01)(0<<CS00);//prescaler = 8
        TIMSK = (1<<TOIE0);
        sei();
        ADCDDR = 0x00;
        ADCPORT = 0x00;
        DIPDDR = 0x00;                  //For Dip Switch and Ext Interrupt
        DIPPORT |= 0x00;
        LEDDDR = 0x0F;
        LEDPORT |= 0x00;
        MtrDDR |= 0xF0;                 // 0-3 input , 4-7 output
        PulseDDR = 0xF0;                // 0-3 ,4-7 output
        TCCR1A = 0xA0;
        TCCR1B = 0x11;                  //00010001 Prescaler = 1
        ICR1 = Top;                     //Top
        OCR1A = Top/2;                  // Pulse X 50%
        OCR1B = Top/2;                  // Pulse Y 50%
        Motor_Stop();
        //printf("Hello World!! \r");
while(1){
//      Print_All();
        Degree_Cal();
        cbi(LEDPORT,3);
        //Toggle_LED();
        if(!(Emergency_Flag))
        {
                if (PID_Enable)
                {
                        if(PID_Timer_Flag)
                        {
                        PID_Control();
//              PID_Gain();
                //printf("#x%04d,+%04f\S\r",temp_ADC[0],XDeg);
                        PID_Timer_Flag = 0;
                        }
                }
                else
                {
                //MotorX_L0;
                //MotorY_L0;
                X_err_Accumulate = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Y_err_Accumulate = 0;
                X_err_Previous = 0;
                Y_err_Previous = 0;
                UiX0=0;
                UiY0=0;
                }
}
else
{
                Emergency_Stop();
}

while(usart_ready()){

                code[i++] = usart_getc();

                if(code[0] == '#'){

                        if(code[i - 1] == '*'){
                                usart_puts(code);

                                switch (code[1])
                                {
                                case '0':// Monitor mode

                                switch (code[2]) // Command1
                                {
                                case '1':

                                Degree_ADC_Print();
                                break ;

                                case '2':// Start

                                Setting_Print();

                                break ;

                                case '3':// Start

                                PID_Gain_Print();

//Toggle_LED0;
                                break ;


default:; //LEDPORT = 0x00;
                                                }


                                break ;

                                case '1':// Callibrate mode
                                switch (code[2]) // Command1
                                {
                                case '1':

                                Plate_X_Control();
                                break ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case '2': // Start

Plate_Y_Control();
break ;

case '3': // Command1 Save X Position

Save_X();
break ;

case '4': // Command1 Save Y Position

Save_Y();
break ;

case '5': // Command1 Save Y Position

Load_Setting();
break ;

case '6': // Command1 Save Y Position

Write_PID();
break ;

default:;//LEDPORT = 0x00;
}
break ;

case '2':// Time Constant Test Mode
switch (code[2])// Command1
{
case '1':

Go_to_Home();
break ;

case '2': // Direction Control

Plate_Test();
break ;

case '3': // Emergency Stop

Motor_Lock();
break ;

case '4': // Choose Motor Speed for test

Choose_Motor_Speed();

break;


default:; //LEDPORT = 0x00;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        break ;

                                        case '3'://Set point

                                        SetPoint();//PID_Control();

                                        break ;

                                        case '4':// PID Enable
                                        switch (code[2]) // Command1
                                        {
                                        case '1': // PID Enable

                                        PID_Enable = 1;
                                        break ;
                                        case '2': // PID Disable

                                        PID_Enable = 0;

                                        Motor_Lock();
                                        break ;


                                        default:; //LEDPORT = 0x00;
                                        }
                                        break ;

                                        default:; //LEDPORT= 0x00;
                                        }

                                memset(code,0,sizeof(code));                    //Clear ucBuffer
                                i = 0;
                        //              uiTimeout = 0;
                                        }
                                        }

                                else{
                                                memset(code,0,sizeof(code));                    // Clear ucBuffer
                                                i = 0;
                                                }
                        } // Close of loop while(usart ready)

}               // Close of loop while(1)

                return 0;

}
ISR (TIMER0_OVF_vect)
{

                                        if(Timer_Counter >=7 ) // 0-7 = 8
                                                {
                                                //Toggle_LED0;
                                                Timer_Counter = 0;
                                                PID_Timer_Flag = 1;
                                                }
                                                else {Timer_Counter++;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        if(temp_ADC[0] > Top_Limit || temp_ADC[0] < Bottom_Limit)
                                        {
                                        Emergency_Flag= 1;
                                        }
                                        else if (temp_ADC[1] > Top_Limit || temp_ADC[1] < Bottom_Limit)
                                        {
                                        Emergency_Flag= 1;
                                        }
                                        else
                                        {
                                        Emergency_Flag = 0;
                                        }
}

void Emergency_Stop()
{
//              Toggle_LED();
                PID_Enable = 0;

                X_err_Accumulate = 0;
                Y_err_Accumulate = 0;
                X_err_Previous = 0;
                Y_err_Previous = 0;
                UiX0 = 0;
                UiY0 = 0;
                MotorX_L();
                MotorY_L();
}
void Toggle_LED(void)
{
                if(ledc)
                {
                sbi(LEDPORT,0);
                ledc =0;
                }
                else
                {
                cbi(LEDPORT,0);
                ledc=1;
                }
}
////////////////////////////////////Data Send//////////////////////////////////////////////
void PID_Gain_Print()
{
                                        printf("#KpX%.5fS\r",KpX);
                                        printf("#KiX%.5fS\r",KiX);
                                        printf("#KdX%.5fS\r",KdX);
                                        printf("#KpY%.5fS\r",KpY);
                                        printf("#KiY%.5fS\r",KiY);
                                        printf("#KdY%.5fS\r",KdY);
                                        printf("#KmX%.5fS\r",MotorX_Volt_Compensator);
                                        printf("#KmY%.5fS\r",MotorY_Volt_Compensator);
}
void Degree_ADC_Print()
{
                if(XDeg>=0)
                                {
                                //printf("#x+%04f,%04dS\r",XDeg,temp_ADC[0]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                printf("#Dx%04d,+%04f\S\r",temp_ADC[0],XDeg);
            }
        else
            {
            //printf("#x-%04f,%04dS\r",(-1)*XDeg,temp_ADC[0]);
                printf("#Dx%04d,-%04f\S\r",temp_ADC[0],(-1)*XDeg);
            }
        if(YDeg>=0)
            {
            //printf("#y-%04f,%04dS\r",(-1)*YDeg,temp_ADC[1]);
                printf("#Dy%04d,+%04f\S\r",temp_ADC[1],YDeg);
            }
        else
            {
            //printf("#y-%04f,%04dS\r",(-1)*YDeg,temp_ADC[1]);
                printf("#Dy%04d,-%04f\S\r",temp_ADC[1],(-1)*YDeg);
            }

}
void Setting_Print()
{
                printf("#RxO%04dS\r",SaveX[0]);
                printf("#RxT%04dS\r",SaveX[1]);
                printf("#RxB%04dS\r",SaveX[2]);
                printf("#RyO%04dS\r",SaveY[0]);
                printf("#RyT%04dS\r",SaveY[1]);
                printf("#RyB%04dS\r",SaveY[2]);
}
////////////////////////////////////////////////////////////////////////////////

////////////////////////////////Plate Control////////////////////////////////////
void Plate_X_Control()
{
switch (code[3]) // Command1 X Control
        {
        case '1': // Plate X Step Up
        MotorX_Up();
        delay_ms(25);
        MotorX_L();
        break ;

        case '2': //Plate X Step Down
        MotorX_Down();
        delay_ms(25);
        MotorX_L();
        break ;

        case '3': // Plate X Up
        MotorX_Up();
        delay_ms(20);
//      MotorX_L();
        break ;

        case '4': //Plate X Down
        MotorX_Down();
        delay_ms(20);
//      MotorX_L();
        break ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    case '5': //Plate X Stop
                    MotorY_L();
                    MotorX_L();
                    delay_ms(20);
                    break ;

                    default: LEDPORT = 0x00;
                    }
}

void Plate_Y_Control()
{
switch (code[3]) // Command1 Y Control
                    {
                    case '1': //Plate Y Step Up
                    MotorY_Up();
                    delay_ms(25);
                    MotorY_L();
                    break ;
                    case '2': //Plate Y Step Down
                    MotorY_Down();
                    delay_ms(25);
                    MotorY_L();
                    break ;
                    case '3': // Plate Y Up
                    MotorY_Up();
                    delay_ms(20);
          //        MotorX_L();
                    break ;
                    case '4': //Plate Y Down
                    MotorY_Down();
                    delay_ms(20);
          //        MotorX_L();
                    break ;
                    case '5': //Plate Y Stop
                    MotorY_L();
                    MotorX_L();
                    delay_ms(20);
                    break ;
                    default:; //LEDPORT = 0x00;
                    }
}
////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////Calculation////////////////////////////////////
void Degree_Cal()
{
                    GetAllADCData();
          //        temp_ADC[0]=uiADCData[0];
          //        temp_ADC[1]=uiADCData[1];
          if (X_Norm) // X is Normal gain
          {
                    if (KxMean != 0)
                              {
                                        XDeg = (KxMean*(float)temp_ADC[0]) + CxMean;
                              }
                    else if (temp_ADC[0]>SaveX[0]) // X is Positive
                              {
                                        XDeg = (KxPos*(float)temp_ADC[0]) + CxPos;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
        else
                {
                        XDeg = (KxNev*(float)temp_ADC[0]) + CxNev;
                }
}
else // X is Reverse Gain
{
        if (KxMean != 0)
                {
                        XDeg = (KxMean*(float)temp_ADC[0]) + CxMean;
                }
        else if (temp_ADC[0]<SaveX[0]) // X is Positive
                {
                        XDeg = (KxPos*(float)temp_ADC[0]) + CxPos;
                }
        else
                {
                        XDeg = (KxNev*(float)temp_ADC[0]) + CxNev;
                }
}

if (Y_Norm) // Y is Normal Gain
{
        if (KyMean != 0)
                {
                        YDeg = (KyMean*(float)temp_ADC[1]) + CyMean;
                }
        else if (temp_ADC[1]>SaveY[0]) // Y is Positive
                {
                        YDeg = (KyPos*(float)temp_ADC[1]) + CyPos;
                }
        else
                {
                        YDeg = (KyNev*(float)temp_ADC[1]) + CyNev;
                }
}
else // Y is Reverse Gain
{
        if (KyMean != 0)
                {
                        YDeg = (KyMean*(float)temp_ADC[1]) + CyMean;
                }
        else if (temp_ADC[1]<SaveY[0]) // Y is Positive
                {
                        YDeg = (KyPos*(float)temp_ADC[1]) + CyPos;
                //      printf("#2y+%04dS\r",YDeg);
                }
        else
                {
                        YDeg = (KyNev*(float)temp_ADC[1]) + CyNev;
                //      printf("#2y-%04dS\r",(-1)*YDeg);
                }
}

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Save_X()
{
        switch (code[3]) // Command2
                {
                case '1': // 0 degree
                SaveX[0]= temp_ADC[0];
                break ;

                case '2': // Max X degree
                SaveX[1]= temp_ADC[0];
                break ;
                case '3': // Min X degree
                SaveX[2]= temp_ADC[0];
                break ;
                default: ;
                }
                Linear_Function_Predict();
}
void Save_Y()
{
        switch (code[3]) // Command2
                {
                case '1': // 0 degree
                SaveY[0]= temp_ADC[1];
                break ;
                case '2': // 45 degree
                SaveY[1]= temp_ADC[1];
                break ;
                case '3': // -45 degree
                SaveY[2]= temp_ADC[1];
                break ;
                default: ;
                }
                Linear_Function_Predict();
}
void Load_Setting()
{
        switch (code[3]) // Command2
                {
                case '1': //
                SaveX[1]= ((code[4]-0x30)*1000)+((code[5]-0x30)*100)+((code[6]-0x30)*10)+(code[7]-0x30);
                break ;
                case '2': //
                SaveX[0]= ((code[4]-0x30)*1000)+((code[5]-0x30)*100)+((code[6]-0x30)*10)+(code[7]-0x30);
                break ;
                case '3': //
                SaveX[2]= ((code[4]-0x30)*1000)+((code[5]-0x30)*100)+((code[6]-0x30)*10)+(code[7]-0x30);
                break ;
                case '4': //
                SaveY[1]= ((code[4]-0x30)*1000)+((code[5]-0x30)*100)+((code[6]-0x30)*10)+(code[7]-0x30);
                break ;
                case '5': //
                SaveY[0]= ((code[4]-0x30)*1000)+((code[5]-0x30)*100)+((code[6]-0x30)*10)+(code[7]-0x30);
                break ;
                case '6': //
                SaveY[2]= ((code[4]-0x30)*1000)+((code[5]-0x30)*100)+((code[6]-0x30)*10)+(code[7]-0x30);
                break ;

                default: ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
        Linear_Function_Predict();
}
void Linear_Function_Predict()
{
        if((SaveX[0] != 0 )&&(SaveX[1] != 0))
        {
                if(SaveX[1] > SaveX[0]) // Normal case
                        {
                                KxPos = DegX_Max/(float)(SaveX[1]-SaveX[0]);
                        //      printf("//////////////////////%f//////////////\r",KxPos);
                                CxPos = (-1)*((float)SaveX[0] * KxPos);
                                //printf("//////////////////////%f//////////////\r",CxPos);
                        //      KxOper = 1;
                        //      CxOper = -1;
                                X_Norm = 1;
                        }
                else // Abnormal case
                        {
                                KxPos = (-1)*(DegX_Max/(float)(SaveX[0]-SaveX[1]));
                        //      printf("//////////////////////%d//////////////\r",SaveX[0]);
                        //      printf("//////////////////////%d//////////////\r",SaveX[1]);
                        //      printf("//////////////////////%d//////////////\r",KxPos);
                                CxPos = (-1)*((float)SaveX[0] * KxPos);
                        //      printf("//////////////////////%d//////////////\r",CxPos);
                        //      KxOper = -1;
                        //      CxOper = 1;
                                X_Norm = 0;
                        }
        }
        if((SaveX[0] != 0 )&&(SaveX[2] != 0))
        {
                if(SaveX[0] > SaveX[2]) // Normal case
                        {
                                KxNev = (0 - DegX_Min)/(float)(SaveX[0]-SaveX[2]);
                                CxNev = (-1)*((float)SaveX[0] * KxNev);
                        //      KxOper = 1;
                        //      CxOper = -1;
                                X_Norm = 1;
                        }
                else // Abnormal case
                        {
                                KxNev = DegX_Min/(float)(SaveX[2]-SaveX[0]);
                                CxNev = (-1)*((float)SaveX[0] * KxNev);
                        //      KxOper = -1;
                        //      CxOper = 1;
                                X_Norm = 0;
                        }
        }

        if (KxPos != 0 && KxNev != 0)
        {
                KxMean = (KxPos+ KxNev)/2;
                CxMean = (CxPos+ CxNev)/2;
        }

        if((SaveY[0] != 0 )&&(SaveY[1] != 0))
        {
                if(SaveY[1] > SaveY[0]) // Normal case
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        KyPos = DegY_Max/(float)(SaveY[1]-SaveY[0]);
                        CyPos = (-1)*((float)SaveY[0] * KyPos);
                //      KxOper = 1;
                //      CxOper = -1;
                        Y_Norm = 1;
                }
        else // Abnormal case
                {
                        KyPos = (-1)*(DegY_Max/(float)(SaveY[0]-SaveY[1]));
                        CyPos = (-1)*(float)(SaveY[0] * KyPos);
                        Y_Norm = 0;
                }
    }
    if((SaveY[0] != 0 )&&(SaveY[2] != 0))
    {
        if(SaveY[0] > SaveY[2]) // Normal case
                {
                        KyNev = (0 - DegY_Min)/(float)(SaveY[0]-SaveY[2]);
                        CyNev = (-1)*((float)SaveY[0] * KyNev);
                        Y_Norm = 1;
                }
        else // Abnormal case
                {
                        KyNev = DegY_Min/(float)(SaveY[2]-SaveY[0]);
                        CyNev = (-1)*((float)SaveY[0] * KyNev);
                        Y_Norm = 0;
                }
    }
    if (KyPos != 0 && KyNev != 0)
    {
        KyMean = (KyPos+ KyNev)/2;
        CyMean = (CyPos+ CyNev)/2;
    }
}
////////////////////////////////////////////////////////////////////
/*****************************Manuaver*****************************/
void SetPoint()
{
    if (code[2] == 'X')
    {
    if (code[3] == '+')
                {
                X_Set_Point = (code[4]-0x30)+((float)(code[5]-0x30)*0.1);
                }
    else
                {
                X_Set_Point = (-1)*((code[4]-0x30)+((float)(code[5]-0x30)*0.1));
                }
    }

    if (code[6] == 'Y')
    {
    if (code[7] == '+')
                {
                Y_Set_Point = (code[8]-0x30)+((float)(code[9]-0x30)*0.1);
                }
    else
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        Y_Set_Point = (-1)*((code[8]-0x30)+((float)(code[9]-0x30)*0.1));
                        }
                }
}
void Write_PID()
{
        switch (code[3]) // Command2
                {
                case '1': //
                KpX = ((code[4]-0x30)*10)+(code[5]-0x30)+((code[7]-0x30)*0.1)+((code[8]-0x30)*0.01)+((code[9]-0x30)*0.001)+((code[10]-0x30)*0.0001)+((code[11]-
0x30)*0.00001);
                break ;
                case '2': //
                KiX = ((code[4]-0x30)*10)+(code[5]-0x30)+((code[7]-0x30)*0.1)+((code[8]-0x30)*0.01)+((code[9]-0x30)*0.001)+((code[10]-0x30)*0.0001)+((code[11]-
0x30)*0.00001);
                break ;
                case '3': //
                KdX = ((code[4]-0x30)*10)+(code[5]-0x30)+((code[7]-0x30)*0.1)+((code[8]-0x30)*0.01)+((code[9]-0x30)*0.001)+((code[10]-0x30)*0.0001)+((code[11]-
0x30)*0.00001);
                break ;
                case '4': //
                KpY = ((code[4]-0x30)*10)+(code[5]-0x30)+((code[7]-0x30)*0.1)+((code[8]-0x30)*0.01)+((code[9]-0x30)*0.001)+((code[10]-0x30)*0.0001)+((code[11]-
0x30)*0.00001);
                break ;
                case '5': //
                KiY = ((code[4]-0x30)*10)+(code[5]-0x30)+((code[7]-0x30)*0.1)+((code[8]-0x30)*0.01)+((code[9]-0x30)*0.001)+((code[10]-0x30)*0.0001)+((code[11]-
0x30)*0.00001);
                break ;
                case '6': //
                KdY = ((code[4]-0x30)*10)+(code[5]-0x30)+((code[7]-0x30)*0.1)+((code[8]-0x30)*0.01)+((code[9]-0x30)*0.001)+((code[10]-0x30)*0.0001)+((code[11]-
0x30)*0.00001);
                break ;
                case '7': //
                MotorX_Volt_Compensator = ((code[4]-0x30)*10)+(code[5]-0x30)+((code[7]-0x30)*0.1)+((code[8]-0x30)*0.01)+((code[9]-0x30)*0.001)+((code[10]-
0x30)*0.0001)+((code[11]-0x30)*0.00001);
                break ;
                case '8': //
                MotorY_Volt_Compensator = ((code[4]-0x30)*10)+(code[5]-0x30)+((code[7]-0x30)*0.1)+((code[8]-0x30)*0.01)+((code[9]-0x30)*0.001)+((code[10]-
0x30)*0.0001)+((code[11]-0x30)*0.00001);
                break ;
                default: ;
                }
}
void PID_Gain()
{
        X_err_Now = X_Set_Point - XDeg;
        Y_err_Now = Y_Set_Point - YDeg;
        X_err_Accumulate = (X_err_Previous + X_err_Now)*(Ts/2);
        Y_err_Accumulate = (Y_err_Previous + Y_err_Now)*(Ts/2);
        UiX = UiX0 + KiX*X_err_Accumulate;
        UiY = UiY0 + KiY*Y_err_Accumulate;
        UdX = KdX*(X_err_Now - X_err_Previous)/Ts;
        UdY = KdY*(Y_err_Now - Y_err_Previous)/Ts;
        X_PID_Gain = (KpX*X_err_Now) + UiX + UdX;
        Y_PID_Gain = (KpY*Y_err_Now) + UiY + UdY;
        UiX0 = UiX;
        UiY0 = UiY;
        Y_err_Previous = Y_err_Now;
        X_err_Previous = X_err_Now;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void PID_Control()
{
                PID_Gain();
                //MotorX_Pulse((int)((X_PID_Gain) * 8.33));
        //      MotorY_Pulse((int)((Y_PID_Gain) * 8.33));
                if (X_PID_Gain >= 0 )
                {
                MotorX_Pulse((float)((X_PID_Gain + MotorX_Volt_Compensator) * (Top/12)));
                MotorX_Up();}
                else
                {
                MotorX_Pulse((float)(((-1)*(X_PID_Gain) + MotorX_Volt_Compensator) * (Top/12)));
                MotorX_Down();}
                if (Y_PID_Gain >= 0 )
                {
                MotorY_Pulse((float)((Y_PID_Gain + MotorY_Volt_Compensator) * (Top/12)));
                MotorY_Up();}
                else
                {
                MotorY_Pulse((float)(((-1)*(Y_PID_Gain) + MotorY_Volt_Compensator)* (Top/12)));
                MotorY_Down();}
}
void Go_to_Home()
        {
                while((XDeg >= 1)||(XDeg <= (-1)))
                {
                        Degree_Cal();
                        if(XDeg > 0)
                        {MotorX_Down();}
                        else if(XDeg<0)
                        {MotorX_Up();}
                        else MotorX_L();
                        delay_ms(1);
                }
                MotorX_L();
                while((YDeg >= 1)||(YDeg <=(-1)))
                {
                        Degree_Cal();
                        if(YDeg > 0)
                        {MotorY_Down();}
                        else if(YDeg < 0)
                        {MotorY_Up();}
                        else MotorY_L();
                        delay_ms(1);
                }
                MotorY_L();
        }
void Plate_Test()
{
                switch (code[3]) // Command2
                {
                case '1': //  X Up start
                                MotorX_Up();
                break ;
                case '2': // X Up stop
                                MotorX_L();
                break ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case '3': //  Y Up start
                        MotorY_Up();

                break ;

                case '4': // Y Up stop
                        MotorY_L();
                break ;

                case '5': //  X Down start
                        MotorX_Down();

                break ;

                case '6': // X Down stop
                        MotorX_L();
                break ;
                case '7': //  Y Down start
                        MotorY_Down();
                break ;
                case '8': // Y Down stop
                        MotorY_L();
                break ;
                default:; //System_Test_Select = 'N' ;
                }
}
void Choose_Motor_Speed()
{
        //Toggle_LED();
                switch (code[3]) // Command2
                {
                case '1': //  X 30
                        MotorX_Pulse(30*800);
                break ;
                case '2': // X 50
                        MotorX_Pulse(50*80);
                break ;
                case '3': //  X 70
                        MotorX_Pulse(70*80);
                break ;
                case '4': // X U90
                        MotorX_Pulse(90*80);
                break ;
                case '5': //  Y 30
                        MotorY_Pulse(30*80);
                break ;
                case '6': // Y 50
                        MotorY_Pulse(50*80);
                break ;
                case '7': //  Y 70
                        MotorY_Pulse(70*80);
                break ;
                case '8': // Y 90
                        MotorY_Pulse(90*80);
                break ;
                default:
                        MotorX_Pulse(50*80);
                        MotorY_Pulse(50*80);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

ก.2.2 โปรแกรมดีเลย์

```
/*    Precise Delay Functions
  V 0.5, Martin Thomas, 9/2004

  In the original Code from Peter Dannegger a timer-interrupt
  driven "timebase" has been used for precise One-Wire-Delays.
  My loop-approach is less elegant but may be more usable
  as library-function. Since it's not "timer-dependent"
  See also delay.h.
  Inspired by the avr-libc's loop-code
*/
#include <avr/io.h>
#include <avr/io.h>
#include <inttypes.h>
void delayloop32(uint32_t loops)
{
 __asm__ volatile ( "cp  %A0,__zero_reg__\n\t" \
            "cpc %B0,__zero_reg__\n\t" \
            "cpc %C0,__zero_reg__\n\t" \
            "cpc %D0,__zero_reg__\n\t" \
            "breq L_Exit_%=     \n\t" \
            "L_LOOP_%=:         \n\t" \
            "subi %A0,1         \n\t" \
            "sbci %B0,0         \n\t" \
            "sbci %C0,0         \n\t" \
            "sbci %D0,0         \n\t" \
            "brne L_LOOP_%=     \n\t" \
            "L_Exit_%=:         \n\t" \
            : "=w" (loops)              \
                                        : "0" (loops)       \
           );
               return;
}
```

ก.2.3 โปรแกรม USART

```
include <avr/interrupt.h>
#include <avr/io.h>
#include <stdio.h>
#include "usart.h"
#include "delay.h"
static int uart_putchar(char c, FILE *stream);
static int uart_getchar(FILE *stream) ;
static FILE uart_str = FDEV_SETUP_STREAM(uart_putchar,uart_getchar,_FDEV_SETUP_RW);
extern void usart_init ( unsigned int baudrate )
{
                /* Set the baud rate */
                UBRRH = (unsigned char) ( baudrate >> 8 );
                UBRRL = (unsigned char) baudrate;
                /* Enable UART receiver and transmitter */
                UCSRB = ((1<<RXCIE) | (1<<RXEN) | (1<<TXEN));
                /*For devices without Extended IO*/
                UCSRC = (1<<URSEL) | (1<<USBS) | (1<<UCSZ1) | (1<<UCSZ0);
                USART_RxTail = 0;
                USART_RxHead = 0;
                USART_TxTail = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        USART_TxHead = 0;
  stderr = stdout = &uart_str;
  stdin =  &uart_str;
}
SIGNAL( USART_RECEIVE_INTERRUPT )
{
        unsigned char tmphead;
  tmphead = ( USART_RxHead + 1 ) & USART_RX_BUFFER_MASK;
  if(tmphead != USART_RxTail )
  {
     if( (UCSRA & (1<<RXC)) )
     {
          USART_RxHead = tmphead;
          USART_RxBuf[tmphead] = UDR;
     }
  }else{
          UCSRB &= ~(1<<RXCIE);
  }
}
SIGNAL( USART_TRANSMIT_INTERRUPT )
{
        unsigned char tmptail;
        if ( USART_TxHead != USART_TxTail )
        {
          tmptail = ( USART_TxTail + 1 ) & USART_TX_BUFFER_MASK;
          USART_TxTail = tmptail;
          UDR = USART_TxBuf[tmptail];
        }
        else
        {
          UCSRB &= ~(1<<UDRIE);
        }
}
extern char usart_getc(void)
{
  unsigned char tmptail;
  unsigned char data;

  if ( USART_RxHead == USART_RxTail )
  {
    return 0x00;
  }else{
    tmptail = (USART_RxTail + 1) & USART_RX_BUFFER_MASK;
    USART_RxTail = tmptail;
    data = USART_RxBuf[tmptail];
          UCSRB |= (1<<RXCIE);
    return data;
  }
}
extern void usart_puts(const char *s )
{
  while (*s)
    usart_putc(*s++);
}
extern void usart_putc(char data)
{
  unsigned char tmphead;
  tmphead  = (USART_TxHead + 1) & USART_TX_BUFFER_MASK;
  while ( tmphead == USART_TxTail );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  USART_TxBuf[tmphead] = data;
  USART_TxHead = tmphead;
  UCSRB |= (1<<UDRIE);
}
extern void usart_puts_p(const char *progmem_s )
{
  register char c;
  while ( (c = pgm_read_byte(progmem_s++)) )
   usart_putc(c);
}
extern unsigned char DataInReceiveBuffer( void )
{
        return ( USART_RxHead != USART_RxTail );
}
extern unsigned char usart_ready(void)
{
if ( USART_RxHead == USART_RxTail )
  return(FALSE);
else
  return(TRUE);
}
static int uart_putchar(char c, FILE *stream)
{
   //lcd_putc(c);
   usart_putc(c);
   return 0;
}
static int uart_getchar(FILE *stream)
{
  static uint16_t nTimeout;
  static uint8_t tempChar=0;
  static uint8_t c;

        nTimeout = 1024; // Timeout is 16 cycles per 10-bit char
  while (!usart_ready() && nTimeout )
  {
                --nTimeout;
     if(nTimeout==0)
        return _FDEV_ERR;
  }
  if(usart_ready())
  {
    tempChar = usart_getc();
  }
        switch (tempChar)
  {
    //case '\n': usart_putc(tempChar); break;
    //case '\r': usart_putc(tempChar); break;
    default  :  c = tempChar;  return (int)c;
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข.1 เอกสารคู่มือการใช้งาน ATMEL32

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 32 Kbytes of In-System Self-programmable Flash program memory
  - 1024 Bytes EEPROM
  - 2 Kbyte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7V - 5.5V for ATmega32L
  - 4.5V - 5.5V for ATmega32
- Speed Grades
  - 0 - 8 MHz for ATmega32L
  - 0 - 16 MHz for ATmega32
- Power Consumption at 1 MHz, 3V, 25°C for ATmega32L
  - **Active: 1.1 mA**
  - **Idle Mode: 0.35 mA**
  - **Power-down Mode: < 1 µA**

ATMEL®

8-bit AVR® Microcontroller with 32K Bytes In-System Programmable Flash

ATmega32
ATmega32L

Summary

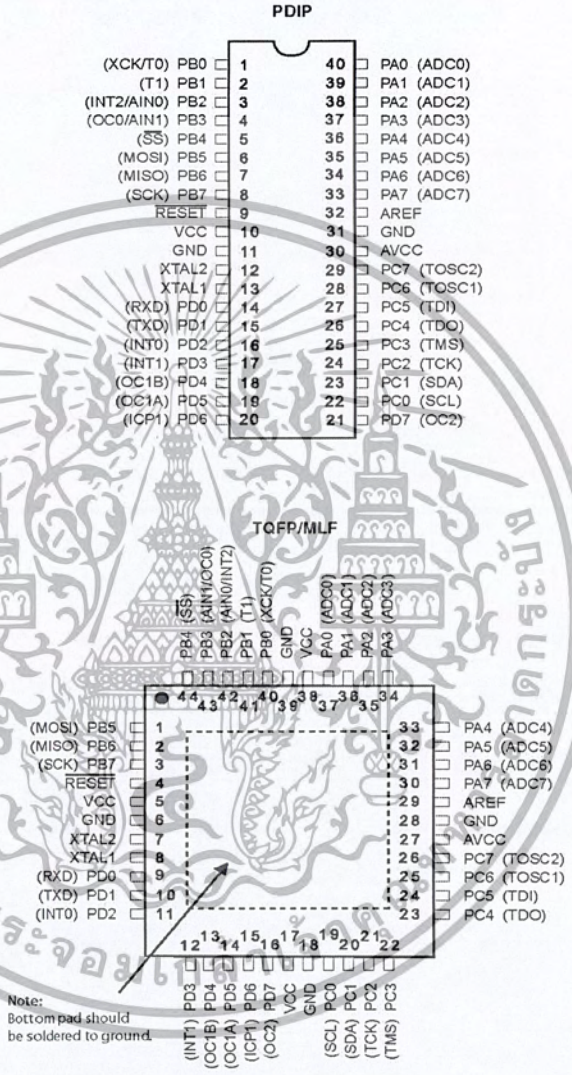
2503PS–AVR–07/10

ATMEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

Figure 1. Pinout ATmega32



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

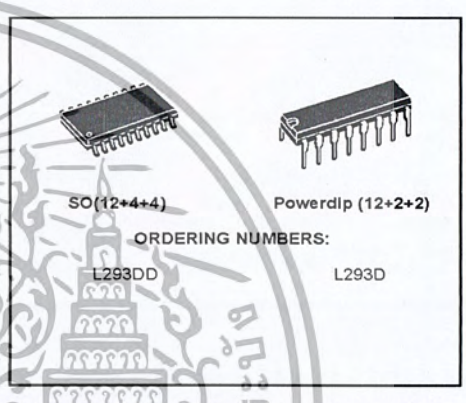
ข.2 เอกสารคู่มือการใช้งาน L293D

SGS-THOMSON MICROELECTRONICS

L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES



SO(12+4+4) Powerdip (12+2+2)

ORDERING NUMBERS:

L293DD L293D

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

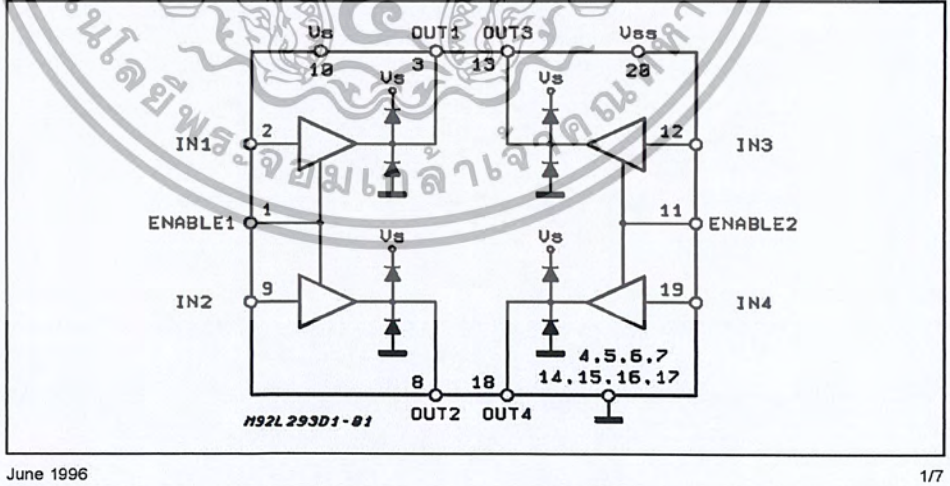
To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz.

The L293D is assembled in a 16 lead plastic packaage which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

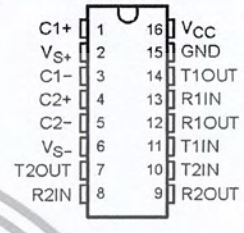
ข.3 เอกสารคู่มือการใช้งาน MAX232

MAX232, MAX232I
DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0-μF Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22
  – 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1-μF Charge-Pump Capacitors is Available With the MAX202
- Applications
  – TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
	SOP (NS)	Reel of 2000	MAX232NSR	MAX232
−40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2004, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.4 เอกสารคู่มือการใช้งาน 7805

FAIRCHILD
SEMICONDUCTOR®

www.fairchildsemi.com

KA78XX/KA78XXA

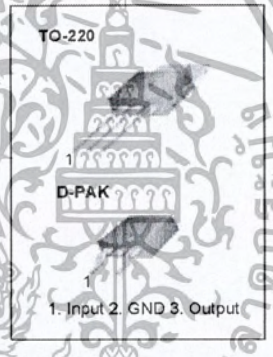
3-Terminal 1A Positive Voltage Regulator

Features

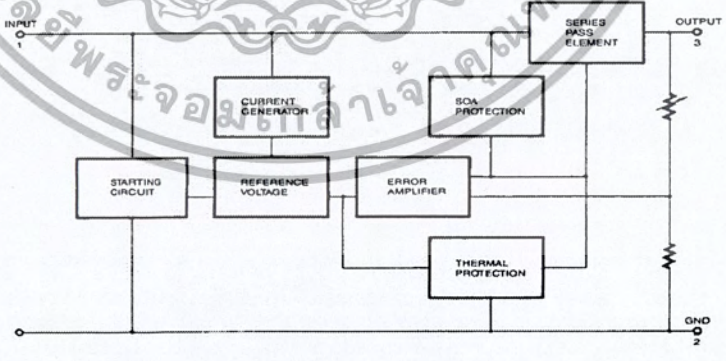
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The KA78XX/KA78XXA series of three-terminal positive regulator are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Digram



Rev. 1.0.0

©2001 Fairchild Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for V_O = 5V to 18V) (for V_O = 24V)	V_I V_I	35 40	V V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	°C/W
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	°C/W
Operating Temperature Range (KA78XX/A/R)	T_{OPR}	0 ~ +125	°C
Storage Temperature Range	T_{STG}	-65 ~ +150	°C

Electrical Characteristics (KA7805/KA7805R)

(Refer to test circuit ,0°C < T_J < 125°C, I_O = 500mA, V_I =10V, C_I= 0.33μF, C_O=0.1μF, unless otherwise specified)

Parameter	Symbol	Conditions		KA7805 Min.	KA7805 Typ.	KA7805 Max.	Unit
Output Voltage	V_O	T_J=+25 °C		4.8	5.0	5.2	V
		5.0mA ≤ I_O ≤ 1.0A, P_O ≤ 15W V_I = 7V to 20V		4.75	5.0	5.25	
Line Regulation (Note1)	Regline	T_J=+25 °C	V_O = 7V to 25V	-	4.0	100	mV
			V_I = 8V to 12V	-	1.6	50	
Load Regulation (Note1)	Regload	T_J=+25 °C	I_O = 5.0mA to1.5A	-	9	100	mV
			I_O =250mA to 750mA	-	4	50	
Quiescent Current	I_Q	T_J =+25 °C		-	5.0	8.0	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A		-	0.03	0.5	mA
		V_I= 7V to 25V		-	0.3	1.3	
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O= 5mA		-	-0.8	-	mV/ °C
Output Noise Voltage	V_N	f = 10Hz to 100KHz, T_A=+25 °C		-	42	-	μV/V_O
Ripple Rejection	RR	f = 120Hz V_O = 8V to 18V		62	73	-	dB
Dropout Voltage	V_{Drop}	I_O = 1A, T_J =+25 °C		-	2	-	V
Output Resistance	r_O	f = 1KHz		-	15	-	mΩ
Short Circuit Current	I_{SC}	V_I = 35V, T_A =+25 °C		-	230	-	mA
Peak Current	I_{PK}	T_J =+25 °C		-	2.2	-	A

Note:

1. Load and line regulation are specified at constant junction temperature. Changes In V_o due to heating effects must be taken into account separately. Pulse testing with low duty is used.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. เป็นเอก เพ็งวัน, สุริยันต์ โชตะศรี. **"การควบคุมตำแหน่งลูกบอลบนคาน."** ปริญญานิพนธ์ วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2548

2. กนก ทาสี, สฤษธิภัสร์ บู่ทอง. วศกร นาวีกิจ. **"ระบบควบคุมตำแหน่งลูกบอลบนระนาบ."** ปริญญานิพนธ์ วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2550

3. รศ. ธีรวัฒน์ ประกอบผล. **"คู่มือการพัฒนาแอพพลิเคชันด้วยVisual Basic 2008."** กรุงเทพฯ. ซิมพลิฟาย. 2553

4. Katsuhiko Ogata. "**Modern Control Engineering. Second Edition.** United States of America : Prentice-Hall, Inc 1990

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้