

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แขนกล

ROBOT ARM



รฟ.
พ635ช
9549

เลขหมู่.....

เลขทะเบียน.....72710.....

วัน,เดือน,ปี.....2.2.ค.ย. 2550.....

b. 1172120
i.

ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2549

สาขาวิชาวิศวกรรมศาสตราระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แขนกล

ROBOT ARM

ผู้จัดทำ

1. นางสาว พิจิตตรา ศรีสุนทรไท 46010515
2. นางสาว รัตนา จิรจิตติกาล 46010636
3. นาย ราชศักดิ์ วิเชตธรรมศักดิ์ 46010637

..... อาจารย์ที่ปรึกษา

(อาจารย์ สุमितร์ พนาอุดมทรัพย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แขนกล
ROBOT ARM

นางสาว พิจิตตรา ศรีสุนทรไท	46010515
นางสาว รัตนา จิริจิติกาล	46010636
นาย ราชศักดิ์ วิเชตธรรมศักดิ์	46010637

อาจารย์ที่ปรึกษา อาจารย์ สุมิตร พนาอุดมทรัพย์

บทคัดย่อ

โครงการนี้เป็นส่วนหนึ่งของการศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต (วิศวกรรมระบบควบคุม) ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง นำเสนอโครงการแขนกล สำหรับการวาดรูปโดยจะทำการวาดรูปจากคอมพิวเตอร์ แล้วให้ แขนกลทำการวาดรูปนั้นลงในกระดาษ โดยมีจุดมุ่งหมายเพื่อศึกษาและเรียนรู้ทฤษฎีหุ่นยนต์และนำมาประยุกต์ใช้งานจริง คณะผู้จัดทำหวังว่าโครงการแขนกลนี้จะถูกพัฒนา และนำไปใช้ในอุตสาหกรรมที่เป็นระบบอัตโนมัติอย่างแพร่หลายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แขนกล

ROBOT ARM

นางสาว พิจิตตรา ศรีสุนทรไท 46010515

นางสาว รัตนา จิรจิตติกาล 46010636

นาย ราชศักดิ์ วิเชตธรรมศักดิ์ 46010637

อาจารย์ที่ปรึกษา อาจารย์ สุมิตร พนาอุดมทรัพย์

ABSTRACT

This project is a part of Bachelor Degree of Engineering (Control Engineering) curriculum of King Mongkut's Institute of Technology Ladkrabang and presents the Robot Arm for drawing . By drawing a picture form Visual Basic 6 after that Robot Arm draw a picture follow that . For purpose that learning the Robotics Theory to render Arm and applying. We hope that, in the future, Robot Arm will be developed and widely used in automation industry.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์สุมิตร พนาอุดมทรัพย์ เป็นอย่างสูงที่ชี้แนะและเป็นที่ปรึกษาให้กับ
โครงการ รวมทั้งความเอื้อเฟื้อ อุปกรณ์การทำงานเป็นอย่างดี

ขอขอบคุณท่านอาจารย์ภาควิชาวิศวกรรมระบบควบคุมทุกๆ ท่าน ที่ประสิทธิประสาทวิชา
ความรู้ และคอยแนะนำแนวทางการแก้ปัญหาต่างๆ รวมทั้งความรักความเมตตาที่ท่านมอบให้

ขอขอบคุณคุณพ่อ คุณแม่ผู้ให้ความดูแลเลี้ยงดู อบรมสั่งสอนตลอดมา

ขอขอบคุณ พี่ๆ เพื่อนๆ และน้องๆ ทุกคนที่คอยเป็นกำลังใจ ให้ความช่วยเหลือ ให้ยืม
อุปกรณ์และให้คำแนะนำที่ดีๆ มีประโยชน์ยิ่ง

และสุดท้ายขอขอบคุณ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบังที่เป็นสถานศึกษา ที่เป็นแหล่งสร้างทักษะความรู้และประสบการณ์ต่างๆ ที่สำคัญ
ต่อผู้จัดทำ

คณะผู้จัดทำ

นางสาวพิจิตรา ศรีสุนทรไท

นางสาวรัตนา จิรจิตติกาล

นายราชศักดิ์ วิเชตธรรมศักดิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เนื้อหา	หน้า
บทคัดย่อ.....	I
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	V
สารบัญตาราง.....	VIII
บทที่ 1 บทนำ.....	1
บทที่ 2 ทฤษฎีและหลักการ.....	3
บทที่ 3 การออกแบบและการสร้าง.....	56
บทที่ 4 การทดลอง.....	66
บทที่ 5 บทวิจารณ์และสรุป.....	81
ภาคผนวก.....	85
เอกสารอ้างอิง.....	99



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
2.1 แสดงการเปรียบเทียบหุ่นยนต์แขนกลกับแขนมนุษย์.....	4
2.2-ก Gantry Robot.....	5
2.2-ข Work envelope of Gantry Robot.....	5
2.3-ก Cylindrical Robot.....	6
2.3-ข Work envelope of Cylindrical Robot.....	6
2.4-ก Spherical Robot.....	7
2.4-ข Work envelope of Spherical Robot.....	7
2.5-ก Scara Robot.....	8
2.5-ข Work envelope of Scara Robot.....	8
2.6-ก Articulated Robot.....	9
2.6-ข Work envelope of Articulated Robot.....	9
2.7 เซอร์โวมอเตอร์แบบต่างๆ.....	10
2.8 ส่วนประกอบของเซอร์โวมอเตอร์.....	11
2.9 สัญลักษณ์ที่สามารถป้อนให้กับ เซอร์โวมอเตอร์.....	12
2.10 แสดงความกว้างของพัลส์ที่สามารถป้อนแก่เซอร์โวมอเตอร์ได้.....	13
2.11 แสดงหน้าโปรแกรมการสร้างงานบน Visual Basic.....	15
2.12 ส่วนประกอบต่างๆของ Visual Basic 6.0.....	16
2.13 แสดงการจัดขาใช้งานของไมโครคอนโทรลเลอร์ dsPIC30F4011.....	18
2.14 แสดงไดอะแกรมการทำงานและส่วนประกอบทั้งหมด.....	18
2.15 แสดงการจัดสรรหน่วยความจำโปรแกรมใน dsPIC.....	22
2.16 แสดงโครงสร้างทางโปรแกรมของ dsPIC(1).....	23
2.17 แสดงโครงสร้างทางโปรแกรมของ dsPIC(2).....	25
2.18 แสดงตัวอย่างการตีความค่าตัวเลขข้อมูล ในกรณีที่บิต MSB เป็น “0” ทั้งในกรณีที่คิดเลขแบบจำนวนเต็มและเลขเศษส่วนทศนิยมในหน่วย DSP และแบบ ALU ของ dsPIC.....	28
2.19 แสดงตัวอย่างการตีความค่าตัวเลขข้อมูล ในกรณีที่บิต MSB เป็น “1” ทั้งในกรณีที่คิดเลขแบบจำนวนเต็มและเลขเศษส่วนทศนิยมในหน่วย DSP และแบบ ALU ของ dsPIC.....	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ(ต่อ)

รูปที่	หน้า
2.20	โครงสร้างการทำงานพื้นฐานของขาพอร์ตอินพุตเอาต์พุต ของไมโครคอนโทรลเลอร์ dsPIC.....31
2.21	โครงสร้างการทำงานของขาพอร์ตอินพุตเอาต์พุตดิจิตอลร่วมกับ โมดูลฟังก์ชันพิเศษ.....33
2.22	แสดงไคอะแกรมการทำงานของไทมเมอร์ 1 ฐานเวลาแบบ A.....37
2.23	แสดงไคอะแกรมการทำงานของไทมเมอร์ 2 ฐานเวลาแบบ B.....37
2.24	แสดงไคอะแกรมการทำงานของไทมเมอร์ 3 ฐานเวลาแบบ C.....38
2.25	แสดงไคอะแกรมการทำงานของไทมเมอร์ 4 ฐานเวลาแบบ B.....38
2.26	แสดงไคอะแกรมการทำงานของไทมเมอร์ 5 ฐานเวลาแบบ C.....39
2.27	แสดงการทำงาน ไทมเมอร์ใน dsPIC30F4011.....44
2.28	ไคอะแกรมส่วนประกอบหลักของโมดูล UART ในไมโครคอนโทรลเลอร์ dsPIC.....45
2.29	แสดงไคอะแกรมการทำงานของตัวส่งข้อมูลใน UART ของ dsPIC.....46
2.30	แสดงไคอะแกรมการทำงานของตัวรับข้อมูลใน UART ของ dsPIC.....47
2.31	แสดงวงจรของบอร์ด JX-dsPIC40.....49
3.1	รูปแสดงขนาดของแกนกล.....56
3.2	รูปแสดงความกว้างของแกนกล.....57
3.3	รูปแสดงความกว้างของฐาน.....57
3.4	รูปแสดงตำแหน่งที่ติดตั้งเซอร์โวมอเตอร์.....58
3.5	รูปแสดงการใช้สายพานช่วยในการเคลื่อนที่แกนท่อนบน.....59
3.6	รูปแสดงการติดตั้งเซอร์โวมอเตอร์ที่ใช้เคลื่อนที่ปากกา.....59
3.7	รูปแสดงการวางซัพพลาย(supply) และบอร์ด(board).....60
3.8	รูปแสดงตำแหน่งวางกระดาษ.....60
3.9	แกนกล.....61
3.10	รูปแสดงมุม θ_2, θ_3 ที่นำไปใช้กับแกนกล.....63
3.11	รูปแสดงมุม θ_1, θ_4 ที่นำไปใช้กับแกนกล.....63
3.12	รูปแสดงหน้าต่างวาดรูปในโปรแกรมวิซวลเบสิก 6 (Visual basic 6).....64
4.1	รูปแสดงปุ่ม pulse up , pulse down และ กรอบแสดงค่าพัลซ์ pulse.....67
4.2	รูปแสดงหน้าต่างวาดภาพ.....80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ(ต่อ)

รูปที่	หน้า
5.1	รูปแสดงการแก้ปัญหาโดยการเอียงระนาบที่วางกระดาษ.....83
5.2	รูปการเอียงระนาบที่วางกระดาษ.....83
รูปภาคผนวกที่ 1-2	รูปแสดงแขนกล.....85
รูปภาคผนวกที่ 3	รูปแสดงแขนกล.....86
รูปภาคผนวกที่ 4	หน้าต่างฟอร์ม(Form).....86



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงการทำงานของจุดต่อ (Joint) ของหุ่นยนต์อุตสาหกรรม.....	4
2.2 แสดงการแบ่งชนิดของหุ่นยนต์ตามลักษณะรูปทรงของพื้นที่ทำงาน.....	5
2.3 แสดงการเปรียบเทียบข้อดี-ข้อเสียของคาร์ทีเซียนหรือกานทรี โรบอท.....	6
2.4 แสดงการเปรียบเทียบข้อดี-ข้อเสียของ ไชลิน ไดรคอลล.....	7
2.5 แสดงการเปรียบเทียบข้อดี-ข้อเสียของสเฟอริคอลล โรบอท.....	8
2.6 แสดงการเปรียบเทียบข้อดี-ข้อเสียของ ไชลิน ไดรคอลล.....	9
2.7 แสดงการเปรียบเทียบข้อดี-ข้อเสียของ ไชลิน ไดรคอลลอาร์ติคิวเลทอาร์ม.....	10
2.8 แสดงขอบเขตของข้อมูลในรีจิสเตอร์ที่มีขนาดต่างกัน ในไมโครคอนโทรลเลอร์ dsPIC ทั้งในแบบเลขจำนวนเต็มและเศษส่วน/ทศนิยม.....	27
4.1-4.4 ตารางแสดงผลการทดลอง.....	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมา

ในยุคเริ่มต้นของการส่งเสริมอุตสาหกรรมในประเทศไทยจะเห็นได้ว่ามีโรงงานต่าง ๆ เข้ามาตั้งฐานผลิตในเมืองไทยจำนวนมากทำให้เกิดนิคมอุตสาหกรรมขึ้นหลายแห่ง ทั้งนี้เนื่องจากรัฐบาลมีนโยบายส่งเสริมอุตสาหกรรมที่ชัดเจน ค่าแรงงานถูก ลดรายจ่ายเนื่องจากภาษีการนำเข้าของสินค้าและวัตถุดิบบางตัว แต่ ณ ปัจจุบันนี้ค่าแรงบ้านเราสูงขึ้นและสูงกว่าประเทศเพื่อนบ้าน เช่น จีน เวียดนาม อินโดนีเซีย ฯลฯ ในขณะที่เดียวกันคุณภาพแรงงานไม่ได้มาตรฐาน ขาดความรู้และทักษะ จึงทำให้หลายบริษัทได้ย้ายฐานการผลิตไปยังประเทศที่มีค่าแรงงานถูกกว่า และอีกหลายบริษัทที่พยายามปรับตัวเอง โดยมีการนำเทคโนโลยีระบบอัตโนมัติ (Automation Technology) เข้ามาใช้งาน เพื่อให้สินค้าสามารถแข่งขันในตลาดโลกได้ทั้งในเรื่องราคา และคุณภาพ โดยเฉพาะในเรื่องคุณภาพ ปัจจุบันโรงงานที่ผลิตสินค้าส่งออกหรือส่งให้กับลูกค้าที่เป็นบริษัทของต่างประเทศมักจะประสบปัญหาในเรื่องคุณภาพ มีทั้ง ผลิตสินค้าไม่ได้มาตรฐานตามที่ลูกค้ากำหนด หรือ ผลิตสินค้าไม่ทันตามกำหนดเวลา อาจเนื่องจากการเปลี่ยนรุ่นผลิตภัณฑ์อยู่บ่อยๆ ต้องใช้เวลาในการ Set up ปัจจุบันจึงมีการนำเทคโนโลยีต่าง ๆ เข้ามาใช้ หนึ่งในเทคโนโลยีที่มีความยืดหยุ่นสูง ได้แก่ หุ่นยนต์อุตสาหกรรมหรือแขนกลอุตสาหกรรม เนื่องจากการเปลี่ยนการทำงานสามารถทำได้โดยการเปลี่ยนโปรแกรม นอกจากนี้คุณภาพของผลิตภัณฑ์ก็ได้มีความสม่ำเสมอเป็นมาตรฐานเดียวกัน

การตัดสินใจเลือกทำโครงการนี้ก็เพื่อที่จะศึกษาพื้นฐานหลักการทำงานของแขนกล โดยให้แขนกลวาดภาพตามที่เราได้วาดไว้ในโปรแกรมทางจอคอมพิวเตอร์ มีการทำงานแบบอัตโนมัติ และหวังเป็นอย่างยิ่งว่า โครงการแขนกลชิ้นนี้จะทำให้เกิดแนวคิดที่จะพัฒนาแขนกลให้สามารถทำงานอื่นๆ ตามที่ต้องการได้ รวมถึงการเพิ่มความสามารถในการทำงานที่ซับซ้อนขึ้นไปในอนาคต

1.2 วัตถุประสงค์ของงาน

- 1.2.1 เพื่อศึกษาโครงสร้างการทำงานของแขนกล และสามารถนำมาออกแบบแขนกลสำหรับวาดภาพได้
- 1.2.2 เพื่อศึกษาการทำงานของเซอร์โวมอเตอร์ที่ใช้สำหรับการเคลื่อนที่ของแขนกล
- 1.2.3 เพื่อศึกษาการใช้โปรแกรม วิชาลเบสิก 6 สำหรับการวาดภาพผ่านทางจอคอมพิวเตอร์
- 1.2.4 เพื่อศึกษาการเชื่อมต่อระหว่างคอมพิวเตอร์กับแขนกล

เอกสารนี้ 1.2.5 เพื่อศึกษาการทำงานของ dsPIC เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตการศึกษา

1.3.1 สำหรับการออกแบบแขนกลจะออกแบบให้แขนกลวาดภาพลงในกระดาษขนาด A4 ซึ่งวางอยู่ในแนวราบ

1.3.2 การเคลื่อนที่ของแขนกลจะใช้เซอร์โวมอเตอร์ดังนั้นจะทำการศึกษาความสัมพันธ์ระหว่างพัลส์ที่ป้อนให้กับเซอร์โวมอเตอร์กับการหมุนของเซอร์โวมอเตอร์

1.3.3 สำหรับการวาดภาพผ่านทางคอมพิวเตอร์จะใช้โปรแกรม วิวอลเบสิก 6 สร้างหน้าต่างสำหรับการวาดรูป นอกจากนี้ยังใช้ในการเก็บข้อมูลจุดจากหน้าต่างวาดรูปเพื่อนำมาคำนวณหามุมที่ใช้สำหรับการเคลื่อนที่ของแขนกล

1.3.4 การเชื่อมต่อระหว่างคอมพิวเตอร์กับแขนกล จะเชื่อมต่อผ่านทางซีเรียลพอร์ต โดยจะทำการส่งข้อมูลของมุมจากโปรแกรม วิวอลเบสิก 6 ผ่านทางซีเรียลพอร์ตส่งไปยัง dsPIC เพื่อให้ dsPIC ทำการประมวลผลและสร้างพัลส์ส่งให้เซอร์โวมอเตอร์



บทที่ 2

ทฤษฎีและหลักการ

2.1 บทนำ

โครงการนี้เป็นแขนงสำหรับวาดภาพ โดยจะวาดภาพผ่านทางหน้าจอคอมพิวเตอร์จากโปรแกรมวิซวลเบสิก 6 จะทำการสร้างหน้าจอสำหรับวาดภาพขึ้นและใช้โปรแกรม วิซวลเบสิก 6 นำค่าจุดจากภาพวาดในหน้าต่างวาดภาพมาคำนวณหาค่ามุมสำหรับการเคลื่อนที่ของแขนกล นอกจากนี้จะใช้โปรแกรม วิซวลเบสิก 6 ส่งข้อมูลผ่านทางซีเรียลพอร์ต(serial port)มายัง dsPIC เพื่อทำการประมวลผลและสร้างพัลส์(pulse) เพื่อส่งไปยังเซอร์โวมอเตอร์ (servo motor)

สำหรับการประมวลผลและสร้างพัลส์ของ dsPIC นั้นจะใช้การเขียนโปรแกรมภาษา C เก็บข้อมูลความสัมพันธ์ระหว่างพัลส์กับค่ามุมของเซอร์โวมอเตอร์ที่ได้จากการทดลองแล้วก็เขียนโปรแกรมสำหรับรับค่าอินพุต(input) ซึ่งเป็นค่ามุมที่ส่งมาจากโปรแกรม วิซวลเบสิก 6 ผ่านทางซีเรียลพอร์ตและ ส่งค่าเอาต์พุต(output) ออกเป็นค่าพัลส์ที่ต้องส่งให้กับเซอร์โวมอเตอร์

2.2 ทฤษฎีแขนกล

2.2.1 ความหมายของแขนกล

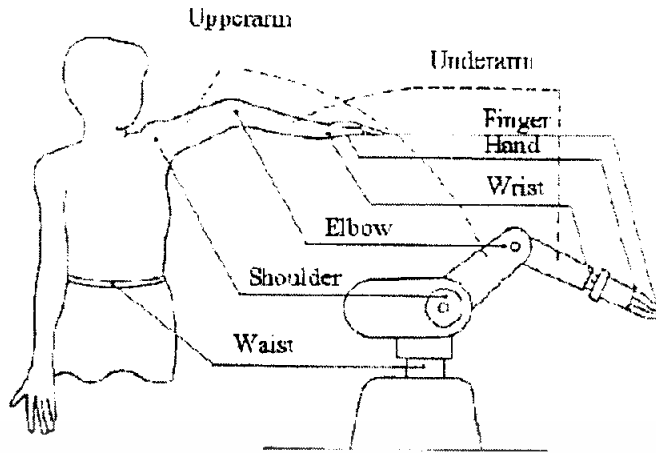
หุ่นยนต์ หรือ ภาษาอังกฤษเขียนว่า Robot มาจากบทละครของ นายคาเรล คาเปก (Karel Kapek) นักแต่งนิยายชาว เช็ก เรื่อง R.U.R (Rossum's Universal Robots) ซึ่งหมายถึงคนงานจำกััดความของ หุ่นยนต์ตามมาตรฐาน ISO 8373 An automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes which may be either fixed in place or mobile for use in industrial automation application หุ่นยนต์ คือ เครื่องจักรที่ถูกควบคุมอัตโนมัติ สามารถเขียนโปรแกรมใหม่ได้ ใช้งานเอนกประสงค์ โปรแกรมการเคลื่อนที่ที่จะต้องสามารถโปรแกรมให้เคลื่อนที่ได้อย่างน้อย 3 แกนหรือมากกว่า หุ่นยนต์อาจจะยึดอยู่กับที่หรือย้ายตำแหน่ง (Mobile) เพื่อใช้ในงานอุตสาหกรรม

2.2.2 แขนกลอุตสาหกรรม

การทำงานของหุ่นยนต์อุตสาหกรรมจะเลียนแบบร่างกายของมนุษย์โดยจะเลียนแบบเฉพาะส่วนของร่างกายที่จะนำไปใช้ประโยชน์ ในอุตสาหกรรมเท่านั้น นั่นคือช่วงแขนของมนุษย์ ดังนั้น บางคนอาจจะได้อินคำว่าแขนกล ซึ่งก็หมายถึงหุ่นยนต์อุตสาหกรรม

การทำงานของหุ่นยนต์อุตสาหกรรมเปรียบเทียบกับแขนมนุษย์ แสดงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงการเปรียบเทียบหุ่นยนต์แขนกลกับแขนมนุษย์

ปัจจุบันและในอนาคตหุ่นยนต์อุตสาหกรรมจะเข้ามามีบทบาทในอุตสาหกรรมมากขึ้น โดยจะทำงานแทนมนุษย์ในงานต่าง ๆ เหล่านี้งานที่อันตราย เช่น งานยกเหล็กเข้าเตาหลอม งานที่เกี่ยวข้องกับสารเคมีงานซ้ำซากน่าเบื่อ เช่น งานยกสินค้าจากสายงานการผลิต งานประกอบ งานบรรจุผลิตภัณฑ์งานที่ต้องการคุณภาพมาตรฐานเดียวกัน เช่น งานเชื่อม งานตัด งานที่ต้องใช้ทักษะความชำนาญสูง เช่น งานเชื่อมแนว เชื่อมเลเซอร์งานที่ต้องใช้ความละเอียดประณีต เช่น งานประกอบชิ้นส่วนอิเล็กทรอนิกส์ งานตรวจสอบ (Inspection) ฯลฯ

2.2.3 การแบ่งประเภทของแขนกล

โดยทั่วไปการแบ่งชนิดของหุ่นยนต์จะแบ่งตามลักษณะรูปทรงของพื้นที่ทำงาน (Envelope Geometric) แต่ก่อนจะอธิบายชนิดของหุ่นยนต์ขออธิบายการทำงานของจุดต่อ (Joint) ของหุ่นยนต์อุตสาหกรรมซึ่งในขั้นพื้นฐานมี 2 ชนิดด้วยกัน ดังนี้

ตารางที่ 2.1 แสดงการทำงานของจุดต่อ (Joint) ของหุ่นยนต์อุตสาหกรรม

ชนิด	สัญลักษณ์	หมายเหตุ
Revolute (R)		เป็นการหมุนรอบแกน (Rotary)
Prismatic (P)		การเคลื่อนที่เชิงเส้น (Linear motion)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดต่อ (Joint) ทั้งสองแบบเมื่อนำมาต่อเข้าด้วยกันอย่างน้อย 3 แกนหลักจะได้พื้นที่ทำงาน (Work envelope) ที่มีลักษณะแตกต่างกันไป ซึ่งสามารถนำมาแบ่งชนิดของหุ่นยนต์ได้ดังต่อไปนี้

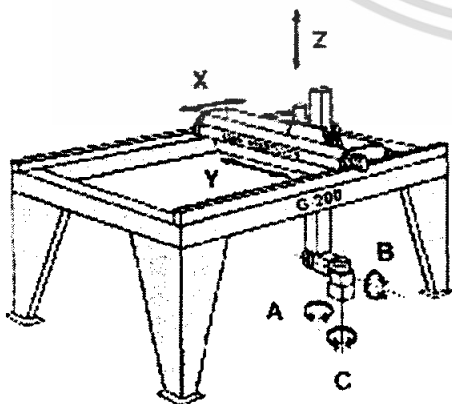
ตารางที่ 2.2 แสดงการแบ่งชนิดของหุ่นยนต์ตามลักษณะรูปทรงของพื้นที่ทำงาน

ชนิดของหุ่นยนต์	แกนที่ 1 (เอว)	แกนที่ 2 (เอว)	แกนที่ 3 (ข้อศอก)
Cartesian (gantry)	P	P	P
Cylindrical	R	P	P
Spherical (Polar)	R	R	P
SCARA	R	P	R
Articulated	R	R	R

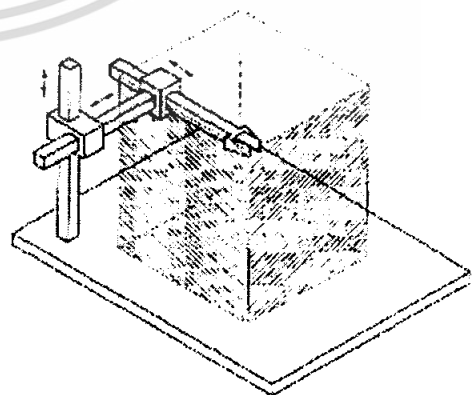
R = Revolute, P = Prismatic

2.2.3.1 คาร์ทีเซียนหรือกานทรีโรบอต (Cartesian or gantry Robot)

แกนทั้ง 3 ของหุ่นยนต์จะเคลื่อนที่เป็นแบบเชิงเส้น (Prismatic) ถ้าโครงสร้างมีลักษณะคล้าย Overhead Crane จะเรียกว่าเป็นหุ่นยนต์ชนิดกานทรี แต่ถ้าหุ่นยนต์ไม่มีขาตั้งหรือขาเป็นแบบอื่น เรียกว่า ชนิด Cartesian



รูปที่ 2.2-ก Gantry Robot



รูปที่ 2.2-ข Work envelope of Gantry Robot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงการเปรียบเทียบข้อดี-ข้อเสียของคาร์ทีเซียนหรือกานทรโรบอท

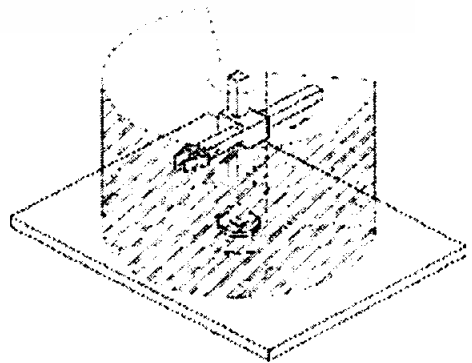
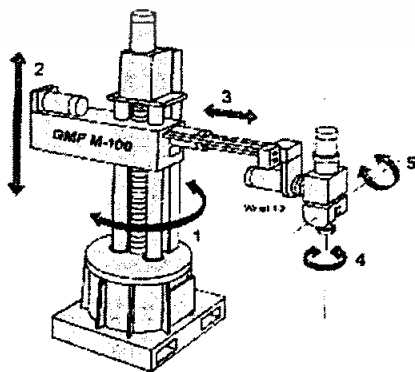
ข้อดี	ข้อเสีย
<ul style="list-style-type: none"> เคลื่อนที่เป็นแนวเส้นตรงทั้ง 3 มิติ การเคลื่อนที่สามารถทำความเข้าใจง่าย มีส่วนประกอบง่าย ๆ โครงสร้างแข็งแรงตลอดการเคลื่อนที่ 	<ul style="list-style-type: none"> ต้องการพื้นที่ติดตั้งมาก บริเวณที่หุ่นยนต์เข้าไปทำงานได้ จะเล็กกว่าขนาดของตัวหุ่นยนต์ ไม่สามารถเข้าถึงวัตถุจากทิศทางข้างใต้ได้ แกนแบบเชิงเส้นจะ Seal เพื่อป้องกันฝุ่นและของเหลวได้ยาก

การประยุกต์ใช้งาน

เนื่องจากโครงสร้างมีความแข็งแรงตลอดแนวการเคลื่อนที่ ดังนั้นจึงเหมาะกับงานเคลื่อนย้ายของหนัก ๆ หรือเรียกว่างาน Pick-and-Place เช่น ใช้โหลดชิ้นงานเข้าเครื่องจักร (Machine loading) , ใช้จัดเก็บชิ้นงาน (Stacking) นอกจากนี้ยังสามารถใช้ในงานประกอบ (Assembly) ที่ไม่ต้องการเข้าถึงในลักษณะที่มีมุมหมุน เช่น ประกอบอุปกรณ์อิเล็กทรอนิกส์ และงาน Test ต่าง ๆ

2.2.3.2 ไซลินโดรคอด (Cylindrical)

หุ่นยนต์ประเภทนี้จะมีแกนที่ 2 (ไหล) และแกนที่ 3 (ข้อศอก) เป็นแบบ prismatic ส่วนแกนที่ 1 (เอว) จะเป็นแบบหมุน (revolute) ทำให้การเคลื่อนที่ได้พื้นที่การทำงานเป็นรูปทรงกระบอก ดังรูป



เอกสารนี้ รูปที่ 2.3-ก Cylindrical Robot ใช้งานเพื่อการ รูปที่ 2.3-ข Work envelope of Cylindrical Robot

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงการเปรียบเทียบข้อดี-ข้อเสียของโซลินไดรคอลล

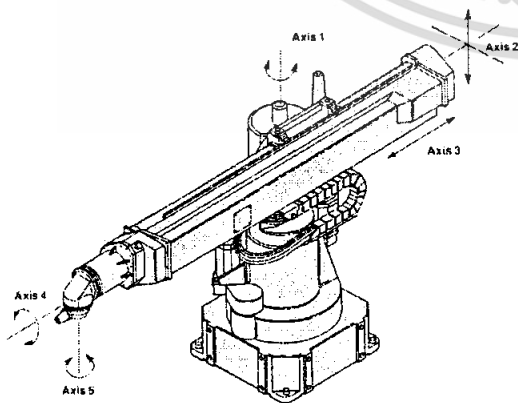
ข้อดี	ข้อเสีย
<ul style="list-style-type: none"> มีส่วนประกอบไม่ซับซ้อน การเคลื่อนที่สามารถเข้าใจได้ง่าย สามารถเข้าถึงเครื่องจักรที่มีการเปิด-ปิด หรือเข้าไปในบริเวณที่เป็นช่องหรือโพรงได้ง่าย (Loading) เช่น การโหลดชิ้นงานเข้าเครื่อง CNC 	<ul style="list-style-type: none"> มีพื้นที่ทำงานจำกัด แกนที่เป็นเชิงเส้นมีความยุ่งยากในการ seal เพื่อป้องกันฝุ่นและของเหลว

การประยุกต์ใช้งาน

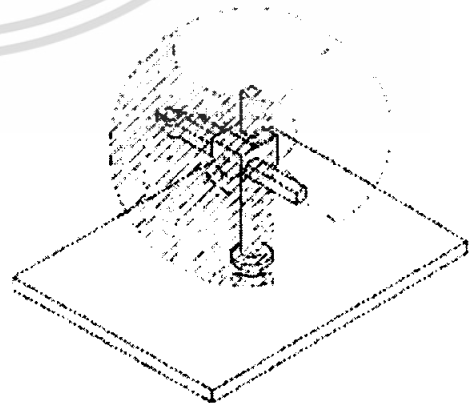
โดยทั่วไปจะใช้ในการหยิบยกชิ้นงาน (Pick-and-Place) หรือป้อนชิ้นงานเข้าเครื่องจักร เพราะสามารถเคลื่อนที่เข้าออกบริเวณที่เป็นช่อง โพรงเล็ก ๆ ได้สะดวก

2.2.3.3 สเตอริคคอลลโรบอท (Spherical Robot (Polar))

มีสองแกนที่เคลื่อนในลักษณะการหมุน (Revolute Joint) คือแกนที่ 1 (เอว) และแกนที่ 2 (ไหล่) ส่วนแกนที่ 3 (ข้อศอก) จะเป็นลักษณะของการเคลื่อนที่แนวเส้นตรง ดังรูปที่ 6-a ซึ่งทำให้ได้พื้นที่การทำงาน ดังรูปที่ 6-b



รูปที่ 2.4-ก Spherical Robot



รูปที่ 2.4-ข Work envelope of Spherical Robot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 แสดงการเปรียบเทียบข้อดี-ข้อเสียของสเฟอริคอลโรบอท

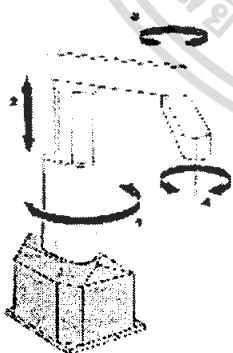
ข้อดี	ข้อเสีย
<ul style="list-style-type: none"> มีปริมาตรการทำงานมากขึ้นเนื่องจากการหมุนของแกนที่ 2 (ใหญ่) สามารถที่จะก้มลงมาจับชิ้นงานบนพื้นได้สะดวก 	<ul style="list-style-type: none"> มีระบบพิกัด (Coordinate) และส่วนประกอบที่ซับซ้อน การเคลื่อนที่และระบบควบคุมมีความซับซ้อนขึ้น

การประยุกต์ใช้งาน

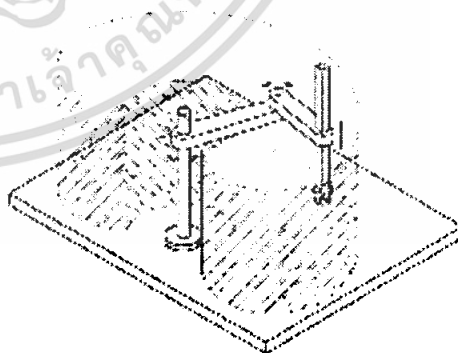
ใช้ในงานที่มีการเคลื่อนที่ในแนวตั้ง (Vertical) เพียงเล็กน้อย เช่น การโหลดชิ้นงานเข้าออกจากเครื่องปั๊ม (Press) หรืออาจจะใช้งานเชื่อมจุด (Spot Welding)

2.2.3.4 สกาลาร์โรบอท (SCARA Robot)

หุ่นยนต์ SCARA (Selective Compliance Assembly Robot Arm) จะมีลักษณะแกนที่ 1 (เอว) และแกนที่ 3 (ข้อศอก) หมุนรอบแกนแนวตั้ง และแกนที่ 2 จะเป็นลักษณะการเคลื่อนที่ขึ้นลง (Prismatic) ดังรูปที่ 7-a ทำให้ได้พื้นที่การทำงานดัง รูปที่ 7-b หุ่นยนต์ SCARA จะเคลื่อนที่ได้รวดเร็วในแนวระนาบ และมีความแม่นยำสูง



รูปที่ 2.5-ก Scara Robot



รูปที่ 2.5-ข Work envelope of Scara Robot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 แสดงการเปรียบเทียบข้อดี-ข้อเสียของโซลินไดรคอลล

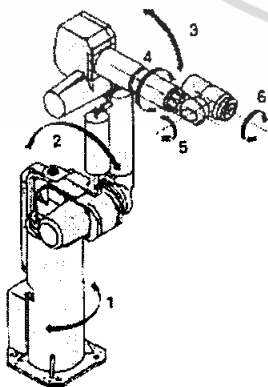
ข้อดี	ข้อเสีย
<ul style="list-style-type: none"> • สามารถเคลื่อนที่ในแนวระนาบ และ ขึ้นลงได้รวดเร็ว • มีความแม่นยำสูง 	<ul style="list-style-type: none"> • มีพื้นที่ทำงานจำกัด • ไม่สามารถหมุน (rotation) ในลักษณะ มุมต่างๆ ได้ • สามารถยกน้ำหนัก (Payload) ได้ไม่มากนัก

การประยุกต์ใช้งาน

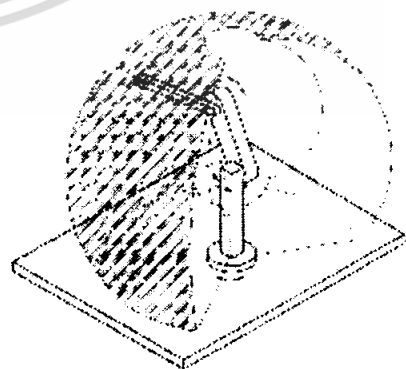
เนื่องจากการเคลื่อนที่ในแนวระนาบและขึ้นลงได้รวดเร็วจึงเหมาะกับ งานประกอบ ชิ้นส่วนทางอิเล็กทรอนิกส์ซึ่ง ต้องการความเร็วและการเคลื่อนที่ที่ไม่ต้องการ การหมุนมากนัก แต่จะไม่เหมาะกับงานประกอบชิ้นส่วนทางกล (Mechanical part) ซึ่งส่วนใหญ่การประกอบจะอาศัย การหมุน(rotation)ใน

2.2.3.5 อาร์ติคิวเลทอรัม (Articulated Arm (Revolute))

ทุกแกนการเคลื่อนที่จะเป็นแบบหมุน (Revolute) รูปแบบการเคลื่อนที่จะคล้ายกับแขน คน ซึ่งจะประกอบด้วยช่วงเอว ท่อนแขนบน ท่อนแขนล่าง ข้อมือ การเคลื่อนที่ทำให้ได้พื้นที่การทำงาน ดังรูปที่ 8-b



รูปที่ 2.6-ก Articulated Robot



รูปที่ 2.6-ข Work envelope of Articulated Robot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 แสดงการเปรียบเทียบข้อดี-ข้อเสียของโซลินอยด์คอลลอยด์อิควิลเทอร์ม

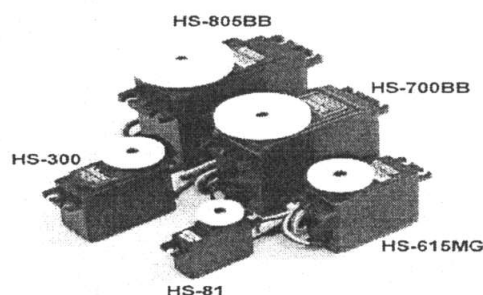
ข้อดี	ข้อเสีย
<ul style="list-style-type: none"> • เนื่องจากทุกแกนจะเคลื่อนที่ในลักษณะของการหมุนทำให้มีความยืดหยุ่นสูงในการเข้าไปยังจุดต่าง ๆ • บริเวณข้อต่อ (Joint) สามารถ Seal เพื่อป้องกันฝุ่น ความชื้น หรือน้ำ ได้ง่าย • มีพื้นที่การทำงานมาก • สามารถเข้าถึงชิ้นงานทั้งจากด้านบน ด้านล่าง • เหมาะกับการใช้มอเตอร์ไฟฟ้า 	<ul style="list-style-type: none"> • มีระบบพิกัด (Coordinate) ที่ซับซ้อน • การเคลื่อนที่และระบบควบคุมทำความเข้าใจได้ยากขึ้น • ควบคุมให้เคลื่อนที่ในแนวเส้นตรง (Linear) ได้ยาก • โครงสร้างไม่มั่นคงตลอดช่วงการเคลื่อนที่ เพราะบริเวณขอบ Work envelope ปลายแขนจะมีการสั่น ทำให้ความแม่นยำลดลง

การประยุกต์ใช้งาน

หุ่นยนต์ชนิดนี้สามารถใช้งานได้กว้างขวางเพราะสามารถเข้าถึงตำแหน่งต่าง ๆ ได้ดี เช่นงานเชื่อม Spot Welding, Path Welding, งานยกของ, งานตัด, งานทากาว, งานที่มีการเคลื่อนที่ ยาก ๆ เช่น งานพันสี งาน sealing ฯลฯ

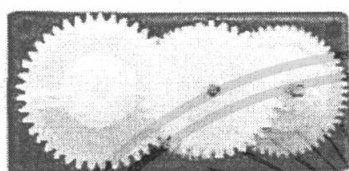
2.3 เซอร์โวมอเตอร์ (Servo Motor)

ในการทำโครงการโปรเจกต์เกี่ยวกับแขนกล ต้องอาศัยความรู้พื้นฐาน และ หลักการทำงาน ของ เซอร์โวมอเตอร์และบอร์ดที่ใช้ในการควบคุมเซอร์โวมอเตอร์นั้นคือ Mini SSC II เสียก่อน

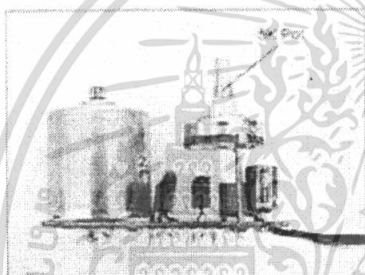


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **รูปที่ 2.7 เซอร์โวมอเตอร์แบบต่างๆ** ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซอร์โวมอเตอร์ คือ มอเตอร์ ไฟฟ้ากระแสตรง (DC MOTOR) ภายใน เซอร์โวมอเตอร์ ก็จะมี ส่วนของ เฟืองเกียร์และบอร์ดควบคุม มีสายสัญญาณในการควบคุมการหมุน ของเซอร์โวมอเตอร์ 1 เส้น และอีก 2 เส้นเป็น VCC กับ กราวนด์ โดยตัวของเซอร์โวมอเตอร์จะสามารถหมุนได้ 180 องศา แต่จะมีบางรุ่นที่สามารถหมุนได้ถึง 210 องศา โดยสามารถที่จะหมุนไปทางซ้าย 90 องศา ทางขวา อีก 90 องศา



รูปที่ 2.8 ส่วนประกอบหลักของ Servo motor มีน้ำหนักเบา และส่งแรงบิดสูง



รูปที่ 2.8 ทำหน้าที่คำนวณตำแหน่งการ หมุนของ Servo motor ตามสัญญาณ pulse ที่ เข้ามา โดยไม บอร์ดก็จะมี ตัวต้านทานที่ปรับ ค่าได้อยู่ด้วย

รูปที่ 2.8 ส่วนประกอบของเซอร์โวมอเตอร์

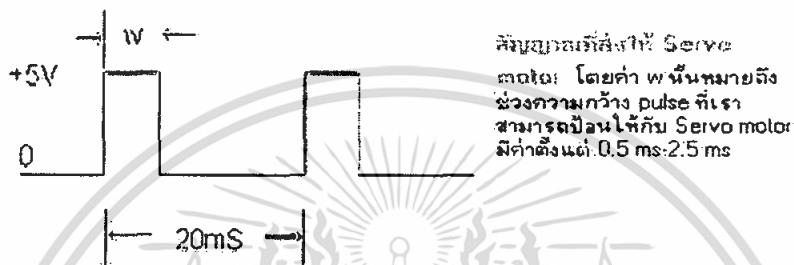
ตามท้องตลาดทั่วไปก็จะมีเซอร์โวมอเตอร์อยู่หลายยี่ห้อและราคาก็แตกต่างกันออก อย่างไรก็ตาม เซอร์โวมอเตอร์ก็จะมีคุณสมบัติคล้าย ๆ กันดังนี้

1. เซอร์โวมอเตอร์ จะหมุนเมื่อมีสัญญาณพัลส์เข้าไป โดยสัญญาณความกว้างพัลส์นี้จะทำหน้าที่ควบคุมการหมุนและทิศทางของเซอร์โวมอเตอร์
2. กินไฟประมาณ 4-6 volt กินกระแส 9.7 mA (Idle) และ 130 mA (Moving)
3. สามารถหมุนได้ 180 องศา (บางรุ่นหมุนได้ 210 องศา) ที่เป็นเช่นนี้ก็เพราะป้องกันความเสียหายที่อาจเกิดขึ้นกับวงจรภายใน ดังจะได้ศึกษาต่อไปในหลักการทำงาน
4. ขนาดเล็ก น้ำหนักเบา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 หลักการทำงานของเซอร์โวมอเตอร์

เราสามารถควบคุมการทำงานของเซอร์โวมอเตอร์ได้โดยป้อนสัญญาณความกว้างพัลส์ให้กับ เซอร์โวมอเตอร์ซึ่งความกว้างของ สัญญาณพัลส์นี้จะไปกำหนดตำแหน่งของเซอร์โวมอเตอร์ว่าจะหมุนไปอยู่ตำแหน่งใด ซึ่งเราสามารถป้อนความกว้างพัลส์ได้ตั้งแต่ 1.0 ms -2.0 ms โดยที่คาบเวลา (Period) ในการส่งสัญญาณความกว้างพัลส์เท่ากับ 20 ms ดังรูป

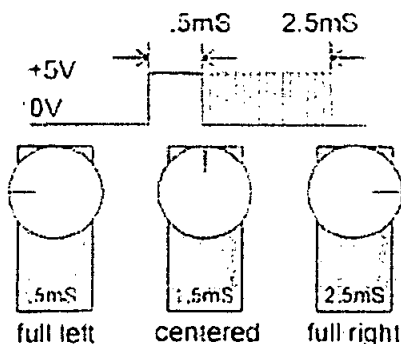


รูปที่ 2.9 สัญญาณที่สามารถป้อนให้กับ เซอร์โวมอเตอร์

โดยจะมีจุดที่อ้างอิง 3 จุดด้วยกันคือ

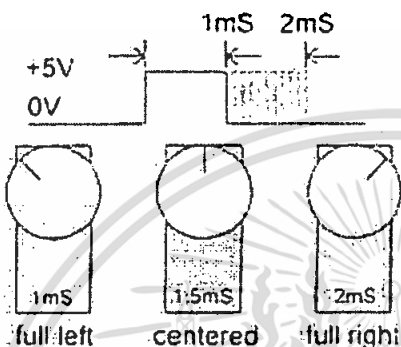
1. ตำแหน่งซ้ายสุด เราสามารถกำหนดให้เซอร์โวมอเตอร์เคลื่อนที่ไปยังตำแหน่งซ้ายสุด(-90 องศา)ได้ โดยการป้อน ความกว้างพัลส์เท่ากับ 0.5 ms
2. ตำแหน่งตรงกลาง เราสามารถกำหนดให้เซอร์โวมอเตอร์เคลื่อนที่ไปยังตำแหน่งตรงกลาง(0 องศา)ได้ โดยการป้อนความกว้างพัลส์เท่ากับ 1.5 ms
3. ตำแหน่งขวาสุด เราสามารถกำหนดให้เซอร์โวมอเตอร์เคลื่อนที่ไปยังตำแหน่งขวาสุด (90 องศา)ได้ โดยการป้อนความกว้างพัลส์เท่ากับ 2.5 ms

จากจุดอ้างอิงทั้ง 3 เราสามารถที่จะคำนวณตำแหน่งที่ต้องการให้เซอร์โวมอเตอร์หมุนไปได้ เช่นถ้าเราต้องการให้เซอร์โวมอเตอร์หมุนไปยังตำแหน่ง -45 องศา เราก็ทำการป้อน $(0.5+1.5)/2=1.0$ ms ให้กับเซอร์โวมอเตอร์ดังรูป



ความกว้าง pulse ที่สามารถป้อนให้กับ Servo motor ได้

1. ป้อน 0.5 ms Servo จะหมุนไปด้านหนึ่ง ข้ายสุด
2. ป้อน 1.5 ms Servo จะหมุนไปด้านตรงกลาง
3. ป้อน 2.5 ms Servo จะหมุนไปยังด้านหนึ่ง ขวาวสุด



สัญญาณ อนุภาค ก็สามารถป้อนใช้ เราสามารถป้อนสัญญาณความกว้าง pulse ให้กับ Servo motor ได้ จากรูป

1. ป้อน 1.0 ms Servo หมุนไปยังด้านหนึ่ง -45 องศา
2. ป้อน 2.0 ms Servo หมุนไปยังด้านหนึ่ง +45 องศา

รูปที่ 2.10 แสดงความกว้างของพัลส์ที่สามารถป้อนแก่เซอร์โวมอเตอร์ได้

2.3.2 ข้อควรระวัง

ความกว้างของสัญญาณพัลส์ที่กล่าวไว้นั้นเป็นเพียงค่าประมาณเท่านั้น ซึ่งเซอร์โวมอเตอร์ที่มีอยู่ตามท้องตลาดอาจจะใช้สัญญาณพัลส์ที่แตกต่างจากนี้ก็ได้ และแต่ละยี่ห้ออาจจะใช้ความกว้างของสัญญาณแตกต่างกันก็ได้ ดังนั้นเพื่อให้แน่ใจควรรศึกษาคู่มือ ของเซอร์โวมอเตอร์นั้นๆ ให้เข้าใจก่อน

อีกประเด็นหนึ่งที่ต้องกล่าวถึงก็คือ ทิศทางการหมุนของเซอร์โวมอเตอร์กล่าวคือ เซอร์โวมอเตอร์แต่ละยี่ห้ออาจจะไม่หมุนไป ในทิศทางเดียวกันถ้าหากเราป้อนสัญญาณพัลส์ให้เซอร์โวมอเตอร์ขนาดเท่ากัน เช่น สมมุติว่าถ้าเราป้อนสัญญาณพัลส์ 2.0 ms บางยี่ห้ออาจจะหมุนไปตามเข็มนาฬิกาไปที่ +45 องศา เช่นยี่ห้อ Hitec และมีบางยี่ห้ออาจจะหมุนทวนเข็มนาฬิกาไปที่ -45 องศา เช่น ยี่ห้อ GWS เป็นต้น

ส่วนการกำหนดความเร็วของเซอร์โวมอเตอร์เราก็สามารถทำได้โดย การป้อนค่าความกว้างของสัญญาณพัลส์ไปเป็นสเต็ป(step) ทุก 20 ms กล่าวคือ ถ้าเราอยากจะให้เซอร์โวมอเตอร์

เคลื่อนอย่างรวดเร็วก็สามารถทำได้โดย ป้อน สัญญาณพัลส์โดยใช้สเต็ปน้อย ๆ ยกตัวอย่างเช่น ต้องการให้ เซอร์โวมอเตอร์เคลื่อนที่ จากตำแหน่งตรงกลางไปซ้ายสุดอย่างรวดเร็ว

ป้อน 1.5 -----> 0.5 ms

ถ้าต้องการให้ช้าลง ก็สามารทำได้ 1.5 ms----->1.0 ms ----->0.5 ms

อีกนัยหนึ่งคือ ถ้าต้องการให้ช้ามากเท่าใดก็ต้องเพิ่ม Step ให้มากขึ้นตามไปด้วย โดยเรา ต้อง ป้อนให้กับ เซอร์โวมอเตอร์ ทุก ๆ 20 ms นั้นเอง

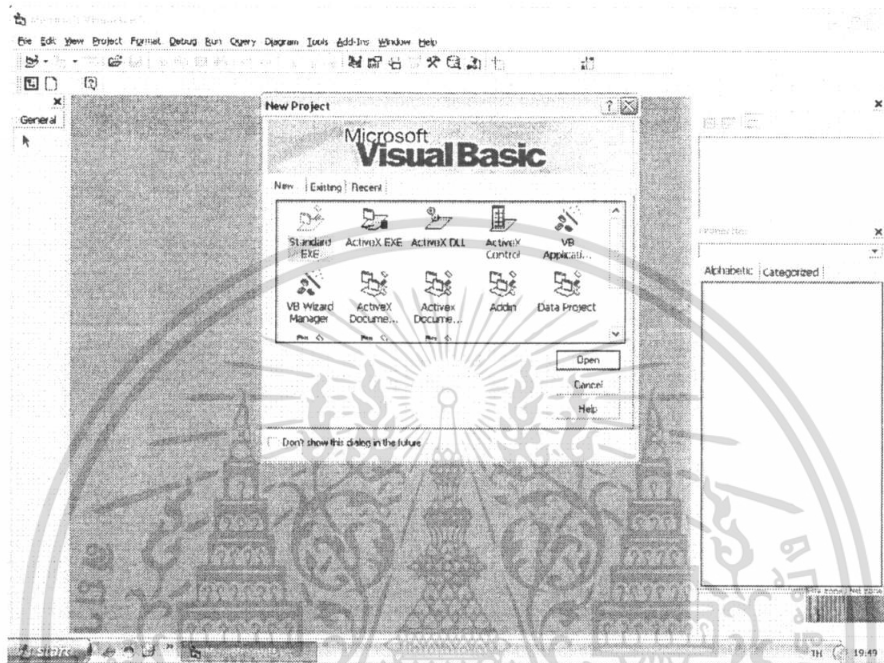
2.4 โปรแกรมวิซวลเบสิก 6 (Visual Basic 6)

โปรแกรมวิซวลเบสิก (VB) เป็นภาษาคอมพิวเตอร์ (Programming Language) ที่พัฒนาโดย บริษัท ไมโครซอฟท์ปัจจุบัน โดยตัวภาษาเองมีรากฐานมาจากภาษาเบสิก (Basic ย่อมาจาก Beginner's All-Purpose Symbolic Instruction) ถ้าแปลให้ได้ความหมายคือ “ชุดคำสั่งหรือภาษาคอมพิวเตอร์สำหรับผู้เริ่มต้น” วิซวลเบสิก 6.0 เป็นเครื่องมือในการสร้างโปรแกรมบนระบบปฏิบัติการวินโดวส์ที่ใช้งานง่าย โปรแกรม วิซวลเบสิก 6 มีจุดเด่นคือ ผู้ที่ไม่มีพื้นฐานด้านการเขียนโปรแกรมเลย ก็สามารถเรียนรู้และนำไปใช้งานได้อย่างง่ายดายและรวดเร็วเมื่อเทียบกับการเรียนภาษาคอมพิวเตอร์อื่นๆ

โปรแกรมวิซวลเบสิก เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียน โปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล(Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น

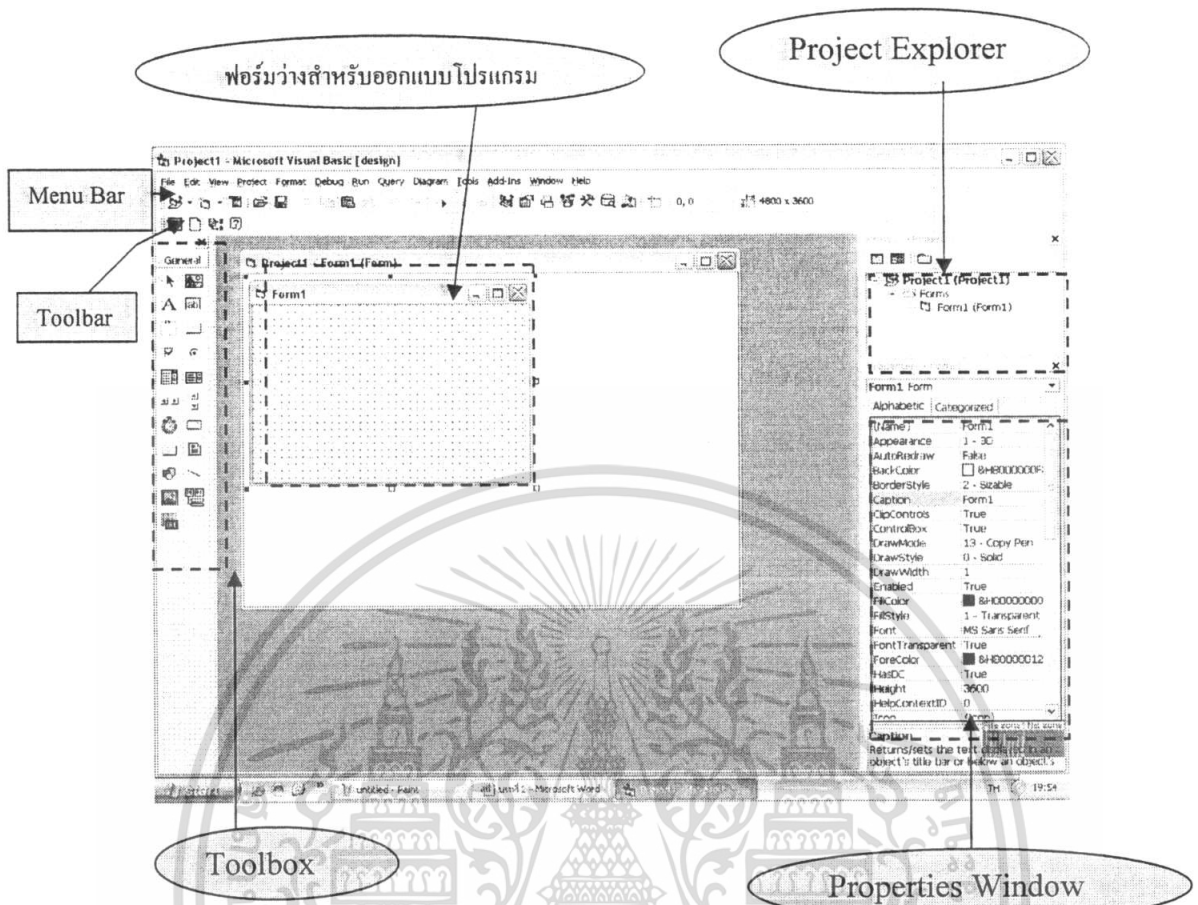
เครื่องมือ หรือ คอนโทรล ต่าง ๆ ที่วิซวลเบสิก ได้เตรียมไว้ให้ ไม่ว่าจะเป็น Form TextBox Label ฯลฯ ถือว่าเป็นวัตถุ (Object ในที่นี้ขอใช้คำว่า ออบเจกต์) นั้นหมายความว่า ไม่ว่าจะเป็นเครื่องมือใด ๆ ใน วิซวลเบสิก 6 จะเป็นออบเจกต์ทั้งสิ้น สามารถที่จะควบคุมการทำงาน แก่ไขคุณสมบัติของออบเจกต์นั้นได้โดยตรง ในทุกๆ ออบเจกต์จะมีคุณสมบัติ (properties) และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจกต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือต่างกันก็ได้ ขึ้นอยู่กับชนิดของออบเจกต์

ในการพัฒนาโปรแกรมประยุกต์ด้วยวิซวลเบสิกการเขียนโค้ดจะถูกแบ่งออกเป็นส่วนๆ เรียกว่า โพรซีเจอร์ (procedure) แต่ละโพรซีเจอร์จะประกอบไปด้วย ชุดคำสั่งที่พิมพ์เข้าไปแล้ว ทำให้คอนโทรลหรือออบเจกต์นั้น ๆ ตอบสนองการกระทำของผู้ใช้



รูปที่ 2.11 แสดงหน้าโปรแกรมการสร้างงานบน วิซวลเบสิก 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 ส่วนประกอบต่างๆของ วิววลเบสิก 6.0

โดยรายละเอียดต่างๆของหน้าจอข้างต้นมีดังนี้

- **Menu Bar** เก็บคำสั่งที่เราสามารถใช้งานได้ทั้งหมดใน vb6 ประกอบไปด้วยเมนูทำงานกับ File, View, Window ฯลฯ
- **Toolbar** ประกอบด้วยปุ่มคำสั่งต่างๆที่ช่วยให้เราใช้งานคำสั่งของ VB6 ได้เร็วยิ่งขึ้น
- **Toolbox** เป็นเครื่องแสดงเครื่องมือต่างๆที่เรียกว่า คอนโทรล ซึ่งเป็นเครื่องมือที่เราสามารถเลือกไปวางลงบนฟอร์มได้เพื่อออกแบบหน้าจอของโปรแกรม(เรียกว่าส่วนติดต่อกับผู้ใช้งาน User Interface)
- **Project Explorer** เป็นหน้าต่างที่แสดง modules ต่างๆที่มีอยู่ในโปรเจกของเราทั้งหมด
- **Properties Window** เป็นหน้าต่างที่แสดงคุณสมบัติของคอนโทรลที่เลือกอยู่ในขณะนั้น
- **Form** ฟอร์มที่ใช้ในการออกแบบ เป็นหน้าต่างที่ใช้ในการออกแบบหน้าจอโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้การเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักของการพัฒนาโปรแกรมประยุกต์ด้วย วิชาลเมสิก ก็คือ การสร้างองค์ประกอบต่าง ๆ ของโปรแกรมประยุกต์ด้วยคอนโทรล โดยมีรูปแบบที่สื่อด้วยภาพ หรือ ที่เรียกกันติดปากว่า การออกแบบอินเตอร์เฟซ ต่อมาก็คือการเขียนชุดคำสั่งเพื่อรองรับเหตุการณ์ต่างๆ ที่อาจเกิดขึ้นได้ ซึ่งจะแบ่งออกเป็น 5 ขั้นตอนใหญ่ ๆ คือ

1. เลือกชนิดของโปรแกรมประยุกต์
2. สร้างยูสเซอร์อินเตอร์เฟซ (หรือส่วนติดต่อกับผู้ใช้ อาจเรียกสั้นๆว่า อินเตอร์เฟซ)
3. เขียนชุดคำสั่งเพื่อรองรับเหตุการณ์ต่างๆ ที่อาจเกิดขึ้นกับแต่ละคอนโทรลหรืออ็อบเจกต์
4. การทดสอบ ตรวจสอบ และดักจับข้อผิดพลาด
5. คอมไพล์โปรแกรมให้เป็น โปรแกรมประยุกต์ที่สมบูรณ์ (เช่น *.exe หรือ *.dll เป็นต้น)

การใช้งานคอนโทรลในการสร้างอินเตอร์เฟซ

จุดเริ่มต้นของการพัฒนาโปรแกรมประยุกต์ด้วย วิชาลเมสิก 6 ก็คือการนำคอนโทรลชนิดต่างๆ ที่ วิชาลเมสิก 6 จัดเตรียมไว้นามาสร้างอินเตอร์เฟซ การพัฒนาโปรแกรมประยุกต์ที่ดี ทำได้โดยการออกแบบอินเตอร์เฟซที่ใช้งานง่าย เป็นมิตรกับผู้ใช้ จะส่งผลให้ระยะเวลาในการพัฒนาโปรแกรมประยุกต์ลดลงไปได้มากทีเดียว เพราะสิ่งที่เหลืออยู่คือการเขียนโค้ดเพื่อให้โปรเจกต์ทำงานให้สมบูรณ์มากที่สุด

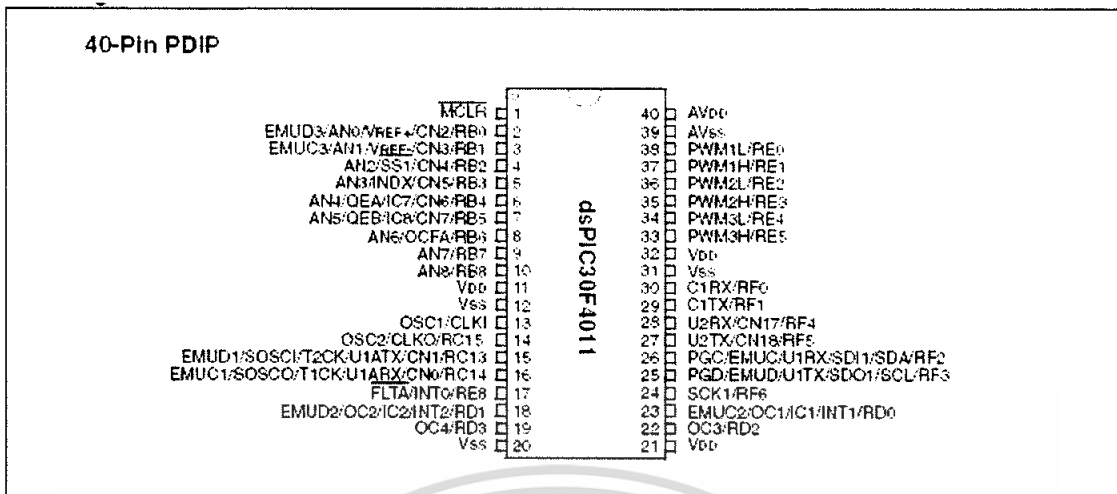
2.5 dsPIC

2.5.1 ข้อมูลเบื้องต้นของ dsPIC

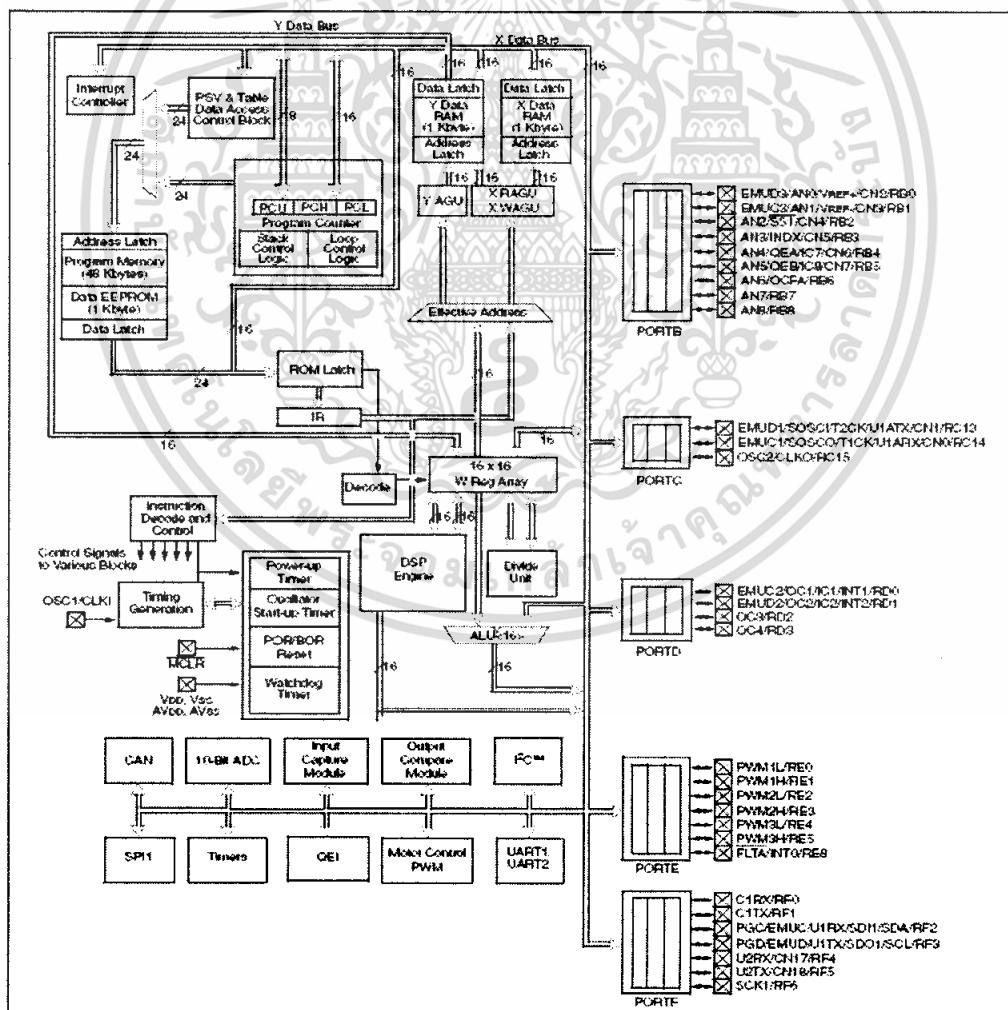
dsPIC คือชื่อของไมโครคอนโทรลเลอร์ 16 บิตจาก Microchip Technology Inc. ผู้ผลิตไมโครคอนโทรลเลอร์ PIC ซึ่งรู้จักกันเป็นอย่างดีในแวดวงนักพัฒนาระบบไมโครคอนโทรลเลอร์ โดย Microchip Technology ได้กำหนดชื่ออย่างเป็นทางการสำหรับไมโครคอนโทรลเลอร์อนุกรมใหม่นี้ว่า Digital Signal Controller หรือ DSC นั้นหมายความว่า dsPIC เป็นไมโครคอนโทรลเลอร์ที่ได้รับการออกแบบเป็นพิเศษเพื่องานประมวลผลสัญญาณดิจิทัลสำหรับสร้างระบบควบคุมอัตโนมัติที่มีความสามารถสูง

คุณสมบัติเด่นโดยรวมของ dsPIC

72710



รูปที่ 2.13 แสดงการจัดขาใช้งานของไมโครคอนโทรลเลอร์ dsPIC30F4011



รูปที่ 2.14 แสดงไดอะแกรมการทำงานและส่วนประกอบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี โดยผู้จัดทำเห็นชอบที่จะเผยแพร่เอกสารนี้เพื่อการศึกษาค้นคว้าวิจัยโดยไม่หวังผลกำไร หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.1 คุณสมบัติของชิพพียู

- มี 84 คำสั่งมาตรฐาน สามารถรองรับรูปแบบการอ้างแอดเดรสได้อย่างอิสระ
- ชุดคำสั่งมีขนาด 24 บิต สามารถประมวลผลข้อมูลได้ 16 บิต
- มีหน่วยความจำโปรแกรมแบบแฟลช ความจุ 48 กิโลไบต์ ลบและเขียนใหม่ได้ไม่น้อยกว่า 10,000 ครั้ง สามารถป้องกันการอ่านได้
- สามารถโปรแกรมหน่วยความจำโปรแกรมได้ด้วยตัวเอง โดยใช้กระบวนการทางซอฟต์แวร์
- มีหน่วยความจำข้อมูลอีพีรอม 1 กิโลไบต์ ลบและเขียนใหม่ได้ไม่น้อยกว่า 100,000 ครั้ง
- มีหน่วยความจำข้อมูลแรม 2 กิโลไบต์
- รีจิสเตอร์ W จัดในรูปของอะเรย์ มีขนาด 16 บิต จำนวน 16 ตัว
- ความเร็วในการทำงานสูงถึง 30 ล้านคำสั่งต่อวินาที
- ความถี่สัญญาณพิกษาภายนอก ตั้งแต่ย่าน ไฟตรงจนถึง 40 MHz
- ความถี่สัญญาณพิกษาในกรณีใช้งานร่วมกับวงจรเฟลลือกู๊ปภายใน ตั้งแต่ 4MHz ถึง 10 MHz เลือกได้ 3 ระดับคือ 4 , 8 หรือ 16 เท่า
- รองรับแหล่งกำเนิดสัญญาณอินเตอร์รัปต์ได้สูงสุด 30 แหล่ง รวมทั้งการอินเตอร์รัปต์จากภายนอก 3 แหล่ง
- สามารถกำหนดระดับความสำคัญในการตอบสนองอินเตอร์รัปต์ได้ 8 ระดับ
- มีอินเตอร์รัปต์เวกเตอร์ 30 ตำแหน่ง
- มีวงจรตรวจจับแรงดันไฟเลี้ยงต่ำกว่ากำหนดแบบโปรแกรมได้
- มีเพาเวอร์-อนรีเซต , เพาเวอร์-อัปไทเมอร์ และออสซิลเลเตอร์สตาร์ท-อัปไทเมอร์
- มีวอตช์ด็อกไทเมอร์แบบโปรแกรมได้
- มีวงจรตรวจสอบการทำงานของวงจรมินิเดสัญญาณพิกษา หากผิดพลาดจะเข้าสู่โหมดสัญญาณพิกษา RC พลังงานต่ำทันที
- รองรับการโปรแกรมในวงจรแบบอนุกรม (ICRP : In-Circuit Serial Programming)
- สามารถเลือกโหมดการใช้พลังงานได้
- ย่านไฟเลี้ยง 2.5 ถึง 5.5V กระแสไฟฟ้า 2.6 ถึง 44mA ที่ไฟเลี้ยง +5V ขึ้นอยู่กับการกำหนดความเร็วในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.2 คุณสมบัติด้านการประมวลผลสัญญาณดิจิทัล

- มีแอมพลิจูดสัญญาณขนาด 40 บิต 2 ตัว รองรับการประมวลผลทางคณิตศาสตร์ได้เป็นอย่างดี
- มีหน่วยประมวลผลด้านการคูณและหารเลข 17 บิตในรูปของฮาร์ดแวร์ จึงทำให้สามารถคูณและหารเลขได้อย่างรวดเร็ว
- ทำการคูณเลข 16 บิตได้ภายในสัญญาณนาฬิกาเพียง 1 ไชเกิล
- มีตัวเลื่อนข้อมูลบาร์เรล 40 สเตจ ช่วยให้การประมวลผลข้อมูลที่มีจำนวนบิตมากๆ สามารถทำได้อย่างรวดเร็ว
- มีวงจรมัลติเพล็กซ์ข้อมูล จึงทำให้สามารถประมวลผลข้อมูลได้อย่างรวดเร็ว

2.5.1.3 คุณสมบัติของโมดูลฟังก์ชันพิเศษ

- สามารถจ่ายกระแสออกทางขาพอร์ตได้ 25 mA ทั้งแบบกระแสซิงก์และซอร์ส
- ไทมเมอร์/เคาน์เตอร์มีขนาด 16 บิต 5 ตัว ต่อใช้งานร่วมกันเป็นไทมเมอร์ 32 บิตได้
- มีโมดูลตรวจจับและเปรียบเทียบสัญญาณดิจิทัลขนาด 16 บิต 4 ชุด
- มีโมดูลเปรียบเทียบข้อมูลและกำเนิดสัญญาณ PWM ความละเอียด 16 บิต 4 ชุด
 - ในการเปรียบเทียบข้อมูลสามารถเลือกการทำงานได้ 2 โหมด
- มีส่วนเชื่อมต่ออุปกรณ์อนุกรมแบบ SPI
- มีส่วนเชื่อมต่ออุปกรณ์ผ่านระบบบัส I²C ทั้งแบบ 7 และ 10 บิต กำหนดเป็นมาสเตอร์หรือสเลฟได้
- มีโมดูลสื่อสารข้อมูลอนุกรม UART พร้อมบัฟเฟอร์แบบ FIFO
- มีโมดูลสร้างสัญญาณ PWM สำหรับควบคุมมอเตอร์ 6 ช่อง
 - เลือกรูปแบบเอาต์พุตได้ทั้งแบบคอมพลิเมนต์และแบบอิสระ
 - มีโหมดปรับตำแหน่งการหมุนทั้งแบบปรับขอมสัญญาณและแบบกึ่งกลาง
 - มีส่วนกำเนิดควิต์ไชเกิล 4 ชุด
 - กำหนดฐานเวลาได้ 4 โหมด
 - สามารถเลือกขั้วของสัญญาณทางเอาต์พุตได้
 - มีสัญญาณกระตุ้นเพื่อให้ทำงานสัมพันธ์กับวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัลภายในไมโครคอนโทรลเลอร์
 - สามารถควบคุมสัญญาณเอาต์พุตได้
- มีโมดูลเชื่อมต่อตัวเข้ารหัสแบบควอดราเจอร์
 - มีอินพุต Phase A , Phase B และรับสัญญาณพัลส์เพื่อกำหนดตำแหน่ง
 - มีตัวนับตำแหน่งขนาด 16 บิต นับได้ทั้งขึ้นและลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แสดงสถานะของทิศทางการนับได้
- กำหนดโหมดของการวัดตำแหน่งได้ 2 โหมดคือ x2 และ x4
- มีวงจรกรองสัญญาณรบกวนแบบดิจิตอลจากอินพุตแบบโปรแกรมได้
- สำหรับกำหนดให้ทำงานเป็นไทเมอร์/เคาเตอร์ขนาด 16 บิตได้
- กำหนดสัญญาณอินเตอร์รัปต์จากตำแหน่งที่นับเกิน (rollover) หรือนับขาด (underflow)
- มีวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล ความละเอียด 10 บิต 9 ช่อง
 - อัตราการสุ่มและแปลงสัญญาณ 500 กิโลแซมเปิลต่อวินาที
 - สามารถแปลงสัญญาณเมื่อไมโครคอนโทรลเลอร์ทำงานในโหมดสลีปและไอเดิลได้

2.5.2 สถาปัตยกรรมโดยสรุปของ dsPIC30F4011

2.5.2.1 หน่วยประมวลผลกลาง

หน่วยประมวลผลของ dsPIC30F4011 ใช้คำสั่งที่มีความยาว 1 เวิร์ด ขนาด 24 บิต โดยมีโปรแกรมเคาเตอร์ขนาด 23 บิต (จริงๆ แล้วโดยโครงสร้างมี 24 บิต แต่ไม่สนใจบิต MSB ซึ่งก็คือบิต 23 และบิต LSB หรือบิต 0 กำหนดให้เป็น "0" จึงทำให้สามารถติดต่อกับหน่วยความจำโปรแกรมได้สูงสุด 4 เมกะเวิร์ด) เพื่อแจ้งแอดเดรสของหน่วยความจำโปรแกรมที่เข้าไปประมวลผล dsPIC30F2010 มีความจุของหน่วยความจำโปรแกรม 48 กิโลไบต์ เมื่อคำสั่งมีความยาว 24 บิต จึงบรรจุคำสั่งได้จริง 16 กิโลเวิร์ด

รีจิสเตอร์หลักที่ใช้ในการทำงานคือ รีจิสเตอร์ W (Working register) สำหรับใน dsPIC จะแตกต่างจากไมโครคอนโทรลเลอร์ PIC อย่างมาก โดยรีจิสเตอร์ W ได้รับการจัดโครงสร้างเป็นอะเรย์ขนาด 16 บิต จึงทำให้สามารถรองรับทั้งข้อมูล, ค่าแอดเดรส หรือค่าของรีจิสเตอร์ใดๆ ที่ต้องนำมาประมวลผลโดยใน dsPIC มีรีจิสเตอร์ W ให้ใช้งานถึง 16 ตัว ส่วนใหญ่ใช้ในการประมวลผลหลัก ส่วนอีกตัวหนึ่งคือรีจิสเตอร์ W15 จะใช้ทำงานร่วมกับตัวชี้สแต็กในการทำงานของโปรแกรมย่อยและบริการอินเตอร์รัปต์

ด้านการตอบสนองอินเตอร์รัปต์นั้น dsPIC30F4011 มีการจัดสรรพื้นที่เก็บค่าอินเตอร์รัปต์เวกเตอร์ไว้มากถึง 30 ตำแหน่ง และยังสามารถกำหนดระดับความสำคัญได้อีก 8 ระดับด้วย

2.5.2.2 หน่วยความจำ

dsPIC30F2010 มีหน่วยความจำโปรแกรม 16 กิโลเวิร์ด แอดเดรสอยู่ในช่วง 0x000100 ถึง 0x007FFE สามารถโปรแกรมหรือเขียนข้อมูลลงไปได้ 2 วิธีคือ

1. โดยใช้การโปรแกรมในวงจรแบบอนุกรมหรือ ICSP ผ่านทางขา PGD และ PGC (ขาที่ 17 และ 18) แล้วป้อนสัญญาณพัลส์แรงดันสูงสำหรับโปรแกรมผ่านเข้ามาทางขา MCLR

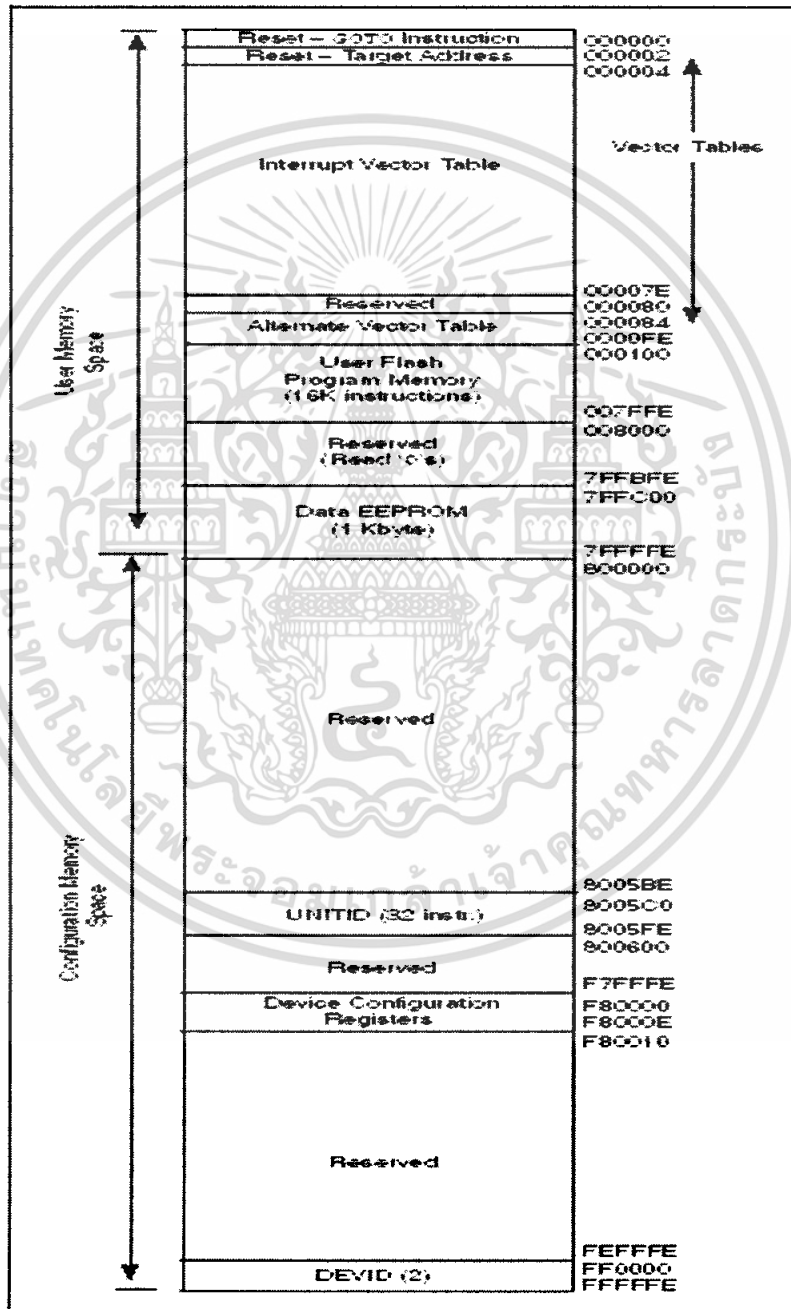
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ บริษัท ไมโครเทคโนโลยี จำกัด การนำเอกสารนี้ไปใช้โดยไม่ผ่านการอนุญาตจากบริษัทฯ ถือว่าผิดกฎหมาย

2. โปรแกรมตัวเองในขณะที่ทำงานหรือ RTSP (Run Time Self-Programming)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนหน่วยความจำข้อมูลแรมนั้น dsPIC30F4011 ได้จัดสรรเป็น 2 ส่วนคือ หน่วยความจำข้อมูลแรม X และ Y แต่ละส่วนมีขนาด 16 บิต ความจุ 256 ไบต์ รวมเป็น 512 ไบต์ โดยในแต่ละส่วนจะมีตัวกำหนดแอดเดรสแยกออกจากกัน เรียกว่า AGU (Address Generation Unit)

ในขณะที่หน่วยความจำข้อมูลอีอีพรอม dsPIC30F4011 จัดสรรไว้ที่แอดเดรส 0x7FFC00 ถึง 0x7FFFFE มีความจุ 1 กิโลไบต์

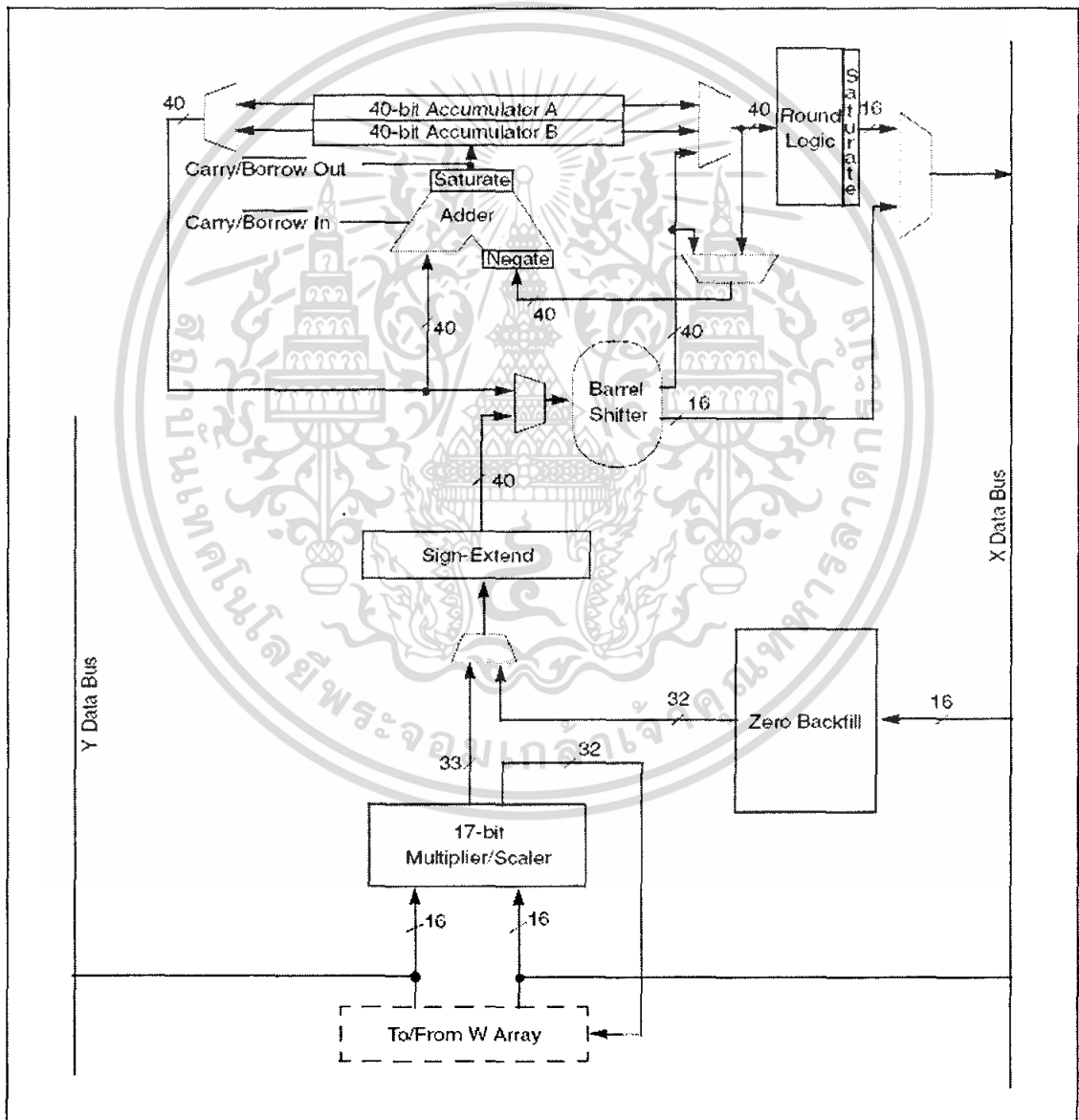


รูปที่ 2.15 แสดงการจัดสรรหน่วยความจำโปรแกรมใน dsPIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2.3 ส่วนประมวลผลสัญญาณดิจิทัล (DSP Engine)

นับเป็นส่วนประกอบที่สำคัญอย่างยิ่งของ dsPIC เนื่องจาก dsPIC ได้รับการออกแบบมาให้ทำงานในด้านการประมวลผลสัญญาณดิจิทัลเป็นหลัก ดังนั้นจึงต้องมีการเพิ่มความสามารถในหน่วยคำนวณทางคณิตศาสตร์และลอจิกอย่างมาก โดยในส่วนประมวลผลสัญญาณดิจิทัลมีหน่วยจัดการคูณเลขขนาด 17 x 17 บิตความเร็วสูง, หน่วยประมวลผลทางคณิตศาสตร์และลอจิก หรือ ALU ขนาด 40 บิต, แอ็กคิวเมเตอร์ขนาด 40 บิต อีก 2 ตัว และตัวเลื่อนข้อมูล 2 ทิศทางแบบบาร์เรล (barrel shifter) ขนาด 40 บิต จึงทำให้สามารถจัดการข้อมูลขนาด 16 บิต ได้เสร็จสิ้นภายในสัญญาณนาฬิกาเพียงไซเคิลเดียว



รูปที่ 2.16 แสดงโครงสร้างทางโปรแกรมของ dsPIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2.4 โมดูลฟังก์ชันพิเศษ

dsPIC30F4011 ได้รวมเอาโมดูลสำหรับทำงานเฉพาะทางไว้อย่างมากมาย ไม่ว่าจะเป็นโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัล ความละเอียด 10 บิต, โมดูลเชื่อมต่ออุปกรณ์อนุกรมหรือ SPI, โมดูลสื่อสารข้อมูลบนระบบบัส I²C, โมดูลสื่อสารข้อมูลผ่านพอร์ตอนุกรมหรือ UART, ไทมเมอร์ขนาด 16 บิตถึง 3 ตัว และที่เป็นพิเศษอีก 2 โมดูลคือ โมดูลสร้างสัญญาณ PWM เพื่อการควบคุมมอเตอร์และโมดูลเข้ารหัสแบบควอดราเจอร์ โดยสามารถใช้งานร่วมกันเพื่อสร้างระบบควบคุมมอเตอร์แบบปิดประสิทธิภาพสูง

2.5.2.5 พอร์ตอินพุตเอาต์พุต

dsPIC30F4011 มีพอร์ตให้ใช้งานมากถึง 5 พอร์ต รวม 29 ขาดังนี้

พอร์ต B มี 9 ขาคือ RB0-RB8 โดยทุกขาสามารถกำหนดให้เป็นพอร์ตอินพุตหรือเอาต์พุตได้ และยังสามารถขับกระแสทั้งแบบซิงก์และซอร์สได้สูงถึง 25 mA

พอร์ต C มี 2 ขาคือ RC13-RC14

พอร์ต D มี 4 ขาคือ RD0 - RD3

พอร์ต E มี 6 ขาคือ RE0-RE5

พอร์ต F มี 8 ขาคือ RF0- RF7

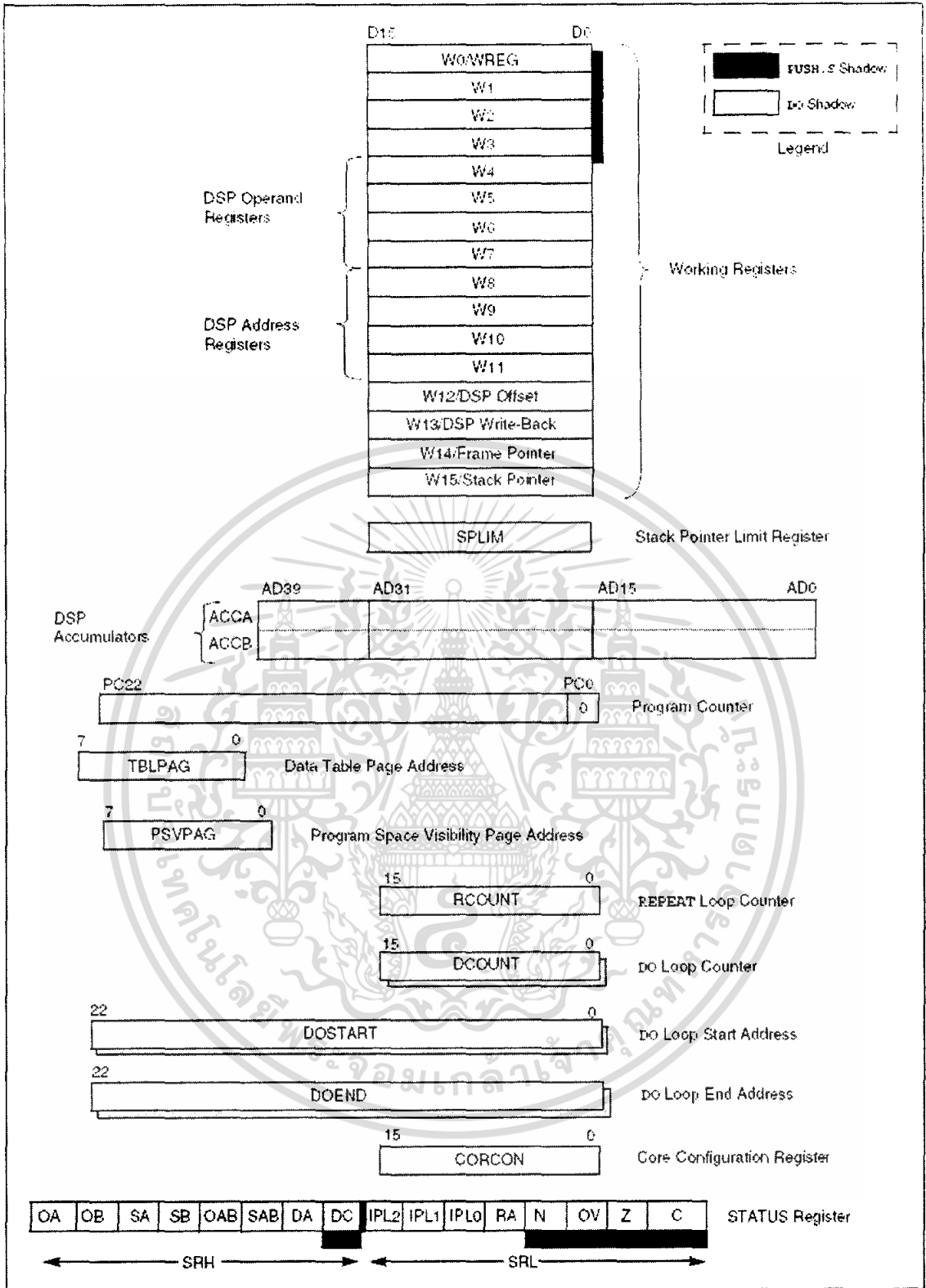
2.5.3 โครงสร้างทางโปรแกรมที่ควรทราบ

2.5.3.1 สแต็กพอยน์เตอร์และเฟรมพอยน์เตอร์ทางซอฟต์แวร์

ใน dsPIC ได้บรรจุสแต็กทางซอฟต์แวร์ขึ้นมา โดยใช้รีจิสเตอร์ W15 เป็นตัวชี้ตำแหน่งของสแต็กหรือสแต็กพอยเตอร์ (software attack pointer) ซึ่งค่าของ W15 จะเปลี่ยนแปลงอย่างอัตโนมัติตามการกระโดดไปทำงานหรือออกจากโปรแกรมแกรมย่อย อย่างไรก็ตาม ก็ยังสามารถใช้งาน W15 เหมือนกับรีจิสเตอร์ W ตัวหนึ่งได้ด้วย แต่ถ้าไม่จำเป็นจริงๆ แนะนำให้สงวนรีจิสเตอร์ W15 นี้ไว้เพื่อทำหน้าที่เป็นตัวชี้สแต็กหรือสแต็กพอยเตอร์จะเหมาะสมกว่า

W15 จะมีค่าเท่ากับ 0x0800 หลังจากการรีเซต ผู้พัฒนาสามารถกำหนดค่าใหม่ได้ตามต้องการภายในขอบเขตของหน่วยความจำข้อมูล

นอกจากนั้นใน dsPIC ยังมีตัวชี้กรอบหรือขอบเขตของสแต็กหรือเรียกว่า เฟรมพอยน์เตอร์ (software frame pointer) ด้วย โดยใช้รีจิสเตอร์ W14 ทำหน้าที่นี้ โดยการกำหนดค่าของ W14 สามารถกระทำผ่านคำสั่ง LNK และ ULNK (คำสั่งภาษาแอสเซมบลี) เช่นเดียวกันก็ยังสามารถใช้งาน W14 เหมือนกับรีจิสเตอร์ W ตัวหนึ่งได้ด้วย แต่ควรสงวนรีจิสเตอร์ W14 นี้ไว้เพื่อทำหน้าที่เป็นเฟรมพอยน์เตอร์จะเหมาะสมกว่า



รูปที่ 2.17 แสดงโครงสร้างทางโปรแกรมของ dsPIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.2 รีจิสเตอร์ STATUS

ใน dsPIC รีจิสเตอร์ตัวนี้มีความสำคัญมาก เพราะใช้ในการแสดงสถานะการทำงานของซอฟต์แวร์โดยรีจิสเตอร์ STATUS (SR) มีขนาด 16 บิต แบ่งออกเป็น 2 ส่วนคือ SRH (บิต 8 ถึง 15) และ SRL (บิต 0 ถึง 7)

โดยใน SRH บรรจุนสถานะการบวกและลบค่าของข้อมูลในส่วนประมวลผลสัญญาณดิจิทัลเป็นหลักที่ 6 บิตบน ส่วนบิต DA ใช้แสดงสถานะการทำงานของกระบวนการวนทำงานหรือทำลูป (DO loop active) สุดท้ายเป็นบิตแฟลคแสดงการทศข้ามหลักหรือ Digit Carry (DC)

ส่วน SRL บรรจุนสถานะการทำงานของหน่วยประมวลผลทางคณิตศาสตร์และลอจิกหรือ ALU โดยแสดงในรูปของแฟลค (flag) ที่บิต C (เกิดการทศ), Z (ค่าเป็นศูนย์), OV (ค่าเกิน), N (เกิดค่าลบ) นอกจากนี้ยังเก็บค่าของการจัดลำดับความสำคัญของการตอบสนองอินเทอร์รัปต์ไว้ที่บิต IPL0-IPL2 สุดท้ายมีบิตแสดงสถานะการทำงานของกระบวนการ REPEAT ซึ่งแสดงที่บิต RA

2.5.3.3 รีจิสเตอร์ควบคุมเพิ่มเติมที่เกี่ยวข้องกับการทำงานของหน่วยประมวลผลกลาง ประกอบด้วย

(1) MODCON (Modulo Control Register) : เป็นรีจิสเตอร์ที่ใช้เอ็นเอเบิลและกำหนดการเข้าถึงหน่วยความจำข้อมูลแบบโมดูลอ (modulo)

(2.) XMODSRT และ XMODEND (X Modulo Start Address Register และ X Modulo End Address Register) : เป็นรีจิสเตอร์ที่ใช้เก็บค่าแอดเดรสตั้งต้น (XMODSRT) และปลายทาง (XMODEND) ของหน่วยความจำข้อมูล X ที่ใช้การเข้าถึงแบบโมดูลอ

(3) YMODSRT และ YMODEND (Y Modulo Start Address Register และ Y Modulo End Address Register) : เป็นรีจิสเตอร์ที่ใช้เก็บค่าแอดเดรสตั้งต้น (YMODSRT) และปลายทาง (YMODEND) ของหน่วยความจำข้อมูล Y ที่ใช้การเข้าถึงแบบโมดูลอ

(4) XBREV (X Modulo Bit-Reverse Register) : เป็นรีจิสเตอร์ที่ใช้กำหนดขนาดของบัพเฟอร์หรือหน่วยความจำข้อมูลชั่วคราวเพื่อใช้ในการเข้าถึงหน่วยความจำข้อมูลแบบกลับบิต

(5) DISICNT (Disable Interrupts Count Register) : เป็นรีจิสเตอร์ที่ใช้ในการทำงานของคำสั่ง DISI เพื่อดิสเอเบิลการอินเทอร์รัปต์ที่มีระดับความสำคัญในระดับ 1 ถึง 6

2.5.4 หน่วยประมวลผลสัญญาณดิจิทัล (DSP Engine)

ตารางที่ 2.8 แสดงขอบเขตของข้อมูลในรีจิสเตอร์ที่มีขนาดต่างกันในไมโครคอนโทรลเลอร์ dsPIC ทั้งในแบบเลขจำนวนเต็มและเศษส่วน/ทศนิยม

ขนาดของรีจิสเตอร์	ย่านของตัวเลข กรณีเป็นเลขจำนวนเต็ม	ย่านของตัวเลข กรณีเป็นเลขเศษส่วน หรือทศนิยม	ความละเอียดของ เลขเศษส่วนหรือ ทศนิยม
16 บิต	-32,768 ถึง 32,767	-1.0 ถึง $(1.0 - 2^{-15})$	3.052×10^{-5}
32 บิต	-2,147,483,648 ถึง 2,147,483,647	-1.0 ถึง $(1.0 - 2^{-31})$	4.657×10^{-10}
40 บิต	-549,755,813,888 ถึง 549,755,813,887	-256.0 ถึง $(256.0 - 2^{-31})$	4.657×10^{-10}

ข้อมูลที่น่ามาประมวลผลในหน่วยประมวลผลสัญญาณดิจิทัล (DSP) นี้มาได้จาก 2 แหล่ง
คือ

- (1) เข้ามาโดยตรงจากรีจิสเตอร์ W4 ถึง W7
 - (2) จากบัสข้อมูลของหน่วยความจำข้อมูล X
- ส่วนข้อมูลเอาต์พุตที่ได้จากหน่วย DSP นี้จะถูกส่งไปยัง 2 แหล่ง
- (1) ส่งไปยังแอกคิวมูลเตอร์เป้าหมาย
 - (2) หน่วยความจำข้อมูล X

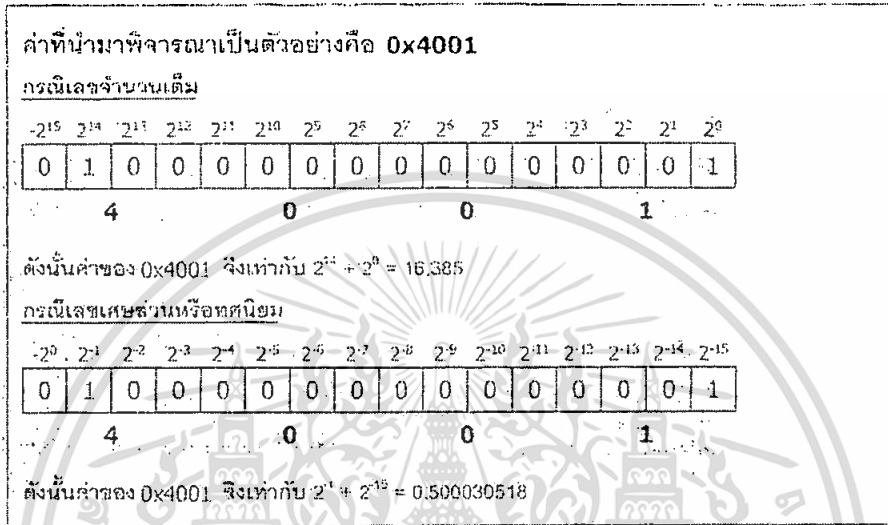
ในตารางที่ 1-2 แสดงถึงขอบเขตของข้อมูลในรีจิสเตอร์ที่มีขนาดต่างกัน ทั้งในแบบเลขจำนวนเต็มและเลขเศษส่วน

2.5.4.1 รูปแบบของตัวเลขที่ใช้ประมวลผลในหน่วย DSP และ ALU

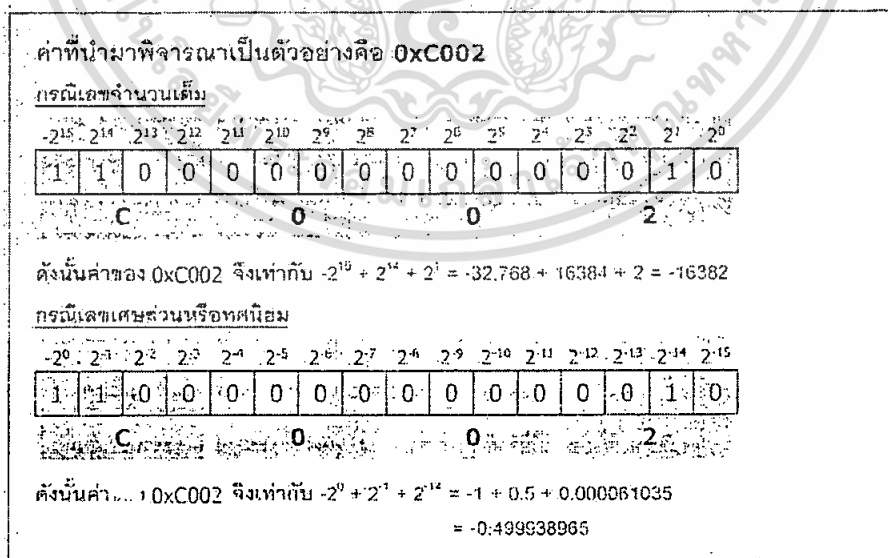
ในรูปที่ 1-4 และ 1-5 แสดงให้เห็นถึงการกำหนดรูปแบบของข้อมูลเมื่อนำประมวลผลในหน่วย DSP และ ALU ทั้งในแบบเลขจำนวนเต็มและเลขเศษส่วนหรือทศนิยม

ในกรณีเลขจำนวนเต็ม จะคำนวณตามปกติ โดยบิต LSB จะมีค่านำหนักประจำหลักเป็น 2^0 และที่บิต MSB หากเป็นเลขคิดเครื่องหมาย บิตนี้จะเป็นบิตแสดงเครื่องหมายว่าเป็นค่าบวกหรือลบ (ถ้าเป็น “0” หมายถึงเป็นค่าบวก ถ้าเป็น “1” หมายถึงเป็นค่าลบ) ส่วนน้ำหนักประจำหลักจะเท่ากับ 2^{15} ในกรณีเป็นเลข 16 บิต, 2^{31} กรณีเป็นเลข 32 บิต และ 2^{39} กรณีเป็นเลข 40 บิต

ในกรณีเป็นเลขเศษส่วนหรือทศนิยม บิตซ้ายสุดซึ่งปกติเป็นบิต MSB แต่ในกรณีเลขเศษส่วนหรือทศนิยมแล้ว บิตซ้ายสุดจะเป็นเลขจำนวนเต็มหลักหน่วยและถ้าเป็นเลขแบบคิดเครื่องหมายบิตนี้ก็จะใช้ในการแสดงเครื่องหมายด้วย นั่นคือ บิตซ้ายสุดจะมีค่าน้ำหนักประจำหลักเป็น 2^0 ในกรณีเป็นค่าบวก และ -2^0 ในกรณีเป็นเลขคิดเครื่องหมาย ส่วนบิตถัดมาทางขวาจะมีค่าน้ำหนักเป็น 2^{-1} ไต่เรียงไปจนถึง 2^{-15} ในกรณีเป็นเลข 16 บิต เป็นต้น



รูปที่ 2.18 แสดงตัวอย่างการตีความค่าตัวเลขข้อมูล ในกรณีที่บิต MSB เป็น “0” ทั้งในกรณีที่คิดเลขแบบจำนวนเต็มและเลขเศษส่วนทศนิยมในหน่วย DSP และแบบ ALU ของ dsPIC



รูปที่ 2.19 แสดงตัวอย่างการตีความค่าตัวเลขข้อมูล ในกรณีที่บิต MSB เป็น “1” ทั้งในกรณีที่คิดเลขแบบจำนวนเต็มและเลขเศษส่วนทศนิยมในหน่วย DSP และแบบ ALU ของ dsPIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์แล้ว
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4.2 แยกคิวมูลเตอร์ 40 บิต

ในหน่วย DSP มีแยกคิวมูลเตอร์ข้อมูลขนาด 40 บิต เพื่อใช้ในการคำนวณ 2 ตัวคือ AccA และ AccB โดยจัดแบ่งออกเป็น 3 ส่วนคือ

AccxL มีขนาด 16 บิต คือ แยกคิวมูลเตอร์บิต 0 ถึง 15 (Accx0-Accx15 โดยที่ x หมายถึง A หรือ B)

AccxH มีขนาด 16 บิต คือ แยกคิวมูลเตอร์บิต 16 ถึง 31 (Accx16-Accx31)

AccxU มีขนาด 8 บิต คือ แยกคิวมูลเตอร์บิต 32 ถึง 39 (Accx32-Accx39)

เมื่อประมวลผลเชิงเศษส่วน แยกคิวมูลเตอร์จะสามารถเก็บข้อมูลได้ตั้งแต่ -256.0 ถึง $256 \cdot 2^{-31}$

เมื่อประมวลผลเลขจำนวนเต็ม แยกคิวมูลเตอร์สามารถเก็บข้อมูลได้ตั้งแต่ $-549,755,813,888$ ถึง $549,755,813,887$

2.5.4.3 ตัวคูณหรือมัลติพลายเออร์ 17 x 17 บิต ความเร็วสูง

เป็นส่วนประกอบที่มีบทบาทสำคัญอย่างยิ่งในหน่วย DSP และ ALU มัลติพลายเออร์นี้สามารถรองรับการคำนวณค่าตัวเลขทั้งแบบคิดและไม่คิดเครื่องหมาย รวมไปถึงข้อมูลที่เป็นเลขเศษส่วน ด้านการให้ผลลัพธ์สามารถให้ผลได้ทั้งเลขจำนวนเต็มขนาด 31 บิต และเลขเศษส่วน

มัลติพลายเออร์รับข้อมูลขนาด 16 บิตเข้ามาประมวลผล จากนั้นแปลงเป็นข้อมูล 17 บิต โดยการเพิ่มบิตเครื่องหมายเข้าไปเพื่อช่วยให้การคำนวณสามารถทำได้ง่ายและรวดเร็วขึ้น ด้วยการกำหนดสถานะที่บิต IF (บิต 0 ของรีจิสเตอร์ (CORCON) จะเป็นตัวกำหนดการทำงานว่า ต้องการให้ประมวลผลในรูปของเลขจำนวนเต็มหรือเศษส่วน นอกจากนั้นบิต IF ยังไม่ส่งผลกระทบต่อคำสั่งการคูณ 5 คำสั่งของ dsPIC คือ

MUL กับ MUL.UU เป็นคำสั่งคูณเลขจำนวนเต็ม 2 จำนวนแบบไม่คิดเครื่องหมาย

MUL.SS เป็นคำสั่งคูณเลขจำนวนเต็ม 2 จำนวนแบบคิดเครื่องหมาย

MUL.SU กับ MUL.US เป็นคำสั่งคูณเลขจำนวนเต็มคิดเครื่องหมายกับเลขจำนวนเต็มที่ไม่คิดเครื่องหมาย

อย่างไรก็ตาม มัลติพลายเออร์นี้จะได้รับการกำหนดให้ทำงานกับเลขเศษส่วนเป็นหลัก หลังจากเกิดการรีเซตไมโครคอนโทรลเลอร์

การให้ผลลัพธ์ของมัลติพลายเออร์เป็นดังนี้

(1) เมื่อทำงานในโหมดเลขจำนวนเต็ม จะให้ผลลัพธ์เป็นค่าทศนิยมพืเมนต์แบบคิดเครื่องหมาย (signed two's complement value) โดยบิตนัยสำคัญสูงสุดหรือ MSB ใช้แสดงเครื่องหมาย ดังนั้นค่าที่ได้จึงอยู่ระหว่าง -2^{N-1} ถึง $2^{N-1} - 1$

(2.) เมื่อทำงานในโหมดเลขเศษส่วน จะให้ผลลัพธ์เป็นค่าทศนิยมเต็มของเลขเศษส่วนแบบคิดเครื่องหมายเช่นกัน โดยบิตนัยสำคัญสูงสุดหรือ MSB ใช้แสดงเครื่องหมาย ค่าที่ได้จะอยู่ระหว่าง -1.0 ถึง

$(1-2^{-N})$

2.5.4.4 ตัวเลื่อนข้อมูลบาร์เรล (Barrel Shifter)

เป็นตัวเลื่อนข้อมูลขนาด 40 บิตจึงทำให้สามารถจัดการเลื่อนข้อมูลขนาด 16 บิตไปทางซ้ายหรือขวาได้ภายในสัญญาณนาฬิกาเพียงไซเคิลเดียว นับเป็นคุณสมบัติที่โดดเด่นอีกประการหนึ่งของหน่วยคำนวณใน dsPIC อย่างไรก็ตามตัวเลื่อนข้อมูลนี้ต้องการข้อมูลที่มีการกำหนดเครื่องหมายเพื่อช่วยให้ทราบจำนวนบิตและทิศทางการเลื่อนข้อมูล ดังมีรายละเอียดต่อไปนี้

- | | |
|--------------------|-------------------------|
| (1) ถ้าเป็นค่าบวก | จะเลื่อนข้อมูลไปทางขวา |
| (2) ถ้าเป็นค่าลบ | จะเลื่อนข้อมูลไปทางซ้าย |
| (3) ถ้าค่าเป็น "0" | จะไม่มีการทำงานใดๆ |

ตัวเลื่อนข้อมูลจะให้ผลลัพธ์ออกมา 40 บิตเมื่อทำงานในหน่วย DSp และให้ผลลัพธ์ 16 บิตเมื่อทำการคำนวณในหน่วย ALU ของไมโครคอนโทรลเลอร์ การรองรับการหารเลขใน dsPIC

คุณสมบัติเด่นอีกประการหนึ่งของไมโครคอนโทรลเลอร์ dsPIC คือ การหารเลข (divide) ในหน่วยคำนวณของไมโครคอนโทรลเลอร์ทั่วไปมักจะไม่รองรับการหารเลข ทำให้ต้องสร้างโปรแกรมย่อยขึ้นมาจัดการคำนวณในตัวเอง หรือถ้าจะรองรับก็เป็นแบบพื้นฐาน ทั้งยังต้องใช้เวลาในการคำนวณนานพอควร แต่สำหรับ dsPIC แล้วสามารถรองรับการหารเลขได้อย่างเต็มรูปแบบ โดยแบ่งเป็น 5 รูปแบบคือ

- (1) การหารเลขเศษส่วนหรือทศนิยม 16 บิตแบบคิดเครื่องหมาย โดยใช้คำสั่ง DIVF
- (2) การหารเลขตัวตั้ง 32 บิตด้วยตัวหาร 16 บิต แบบคิดเครื่องหมาย โดยใช้คำสั่ง DIV. SD
- (3) การหารเลขตัวตั้ง 32 บิตด้วยตัวหาร 16 บิต แบบไม่คิดเครื่องหมาย โดยใช้คำสั่ง DIV. UD
- (4) การหารเลขจำนวนเต็ม 16 บิต แบบคิดเครื่องหมาย โดยใช้คำสั่ง DIV. SW
- (5) การหารเลขจำนวนเต็ม 16 บิต แบบไม่คิดเครื่องหมาย โดยใช้คำสั่ง DIV. UW

ผลหารของทุกคำสั่งหารเลขจะเก็บไว้ในรีจิสเตอร์ *W0* ส่วนเศษของการหารเก็บไว้ในรีจิสเตอร์ *W1* ในขณะที่ค่าของตัวหาร 16 บิตสามารถเก็บไว้ในรีจิสเตอร์ *W* ตัวใดก็ได้ เช่นเดียวกับ

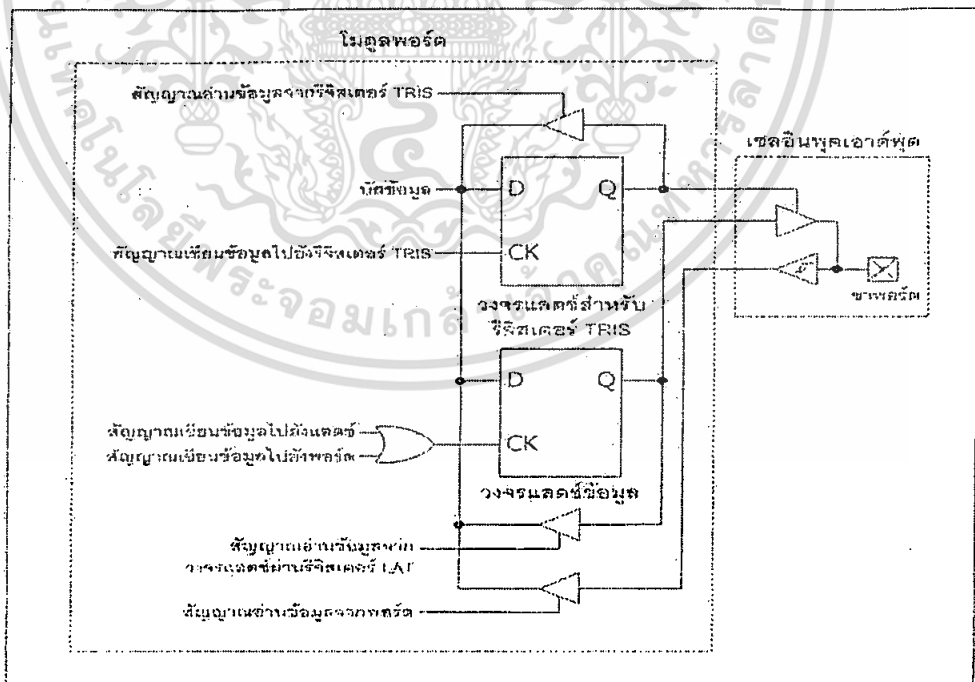
ตัวตั้ง แต่ถ้านหากเป็นตัวตั้ง 32 บิตจะต้องนำค่าไปเก็บไว้ในรีจิสเตอร์ W จำนวน 2 ตัวที่อยู่ติดกัน เช่น W2 กับ W3 โดย W3 เก็บค่าตัวตั้ง 16 บิตบน และ W2 เก็บค่าตัวตั้ง 16 บิตล่าง

เวลาที่ใช้ในการคำนวณเพื่อหาค่าการนั้นจะต้องใช้มากถึง 18 เท่าของเวลาที่ใช้ในการกระทำคำสั่งปกติ สามารถใช้กระบวนการทำโปรแกรมลูบซ้ำด้วยคำสั่ง REPEAT 18 รอบ ด้วยการ ใช้กระบวนการ REPEAT ทำให้การเขียนโปรแกรมด้วยภาษาแอสเซมบลีทำได้ง่ายขึ้นอย่างมาก ดังนั้นรวมเวลาทั้งหมดที่ต้องใช้ในการหารเลขคือ 19 ไชเกิล

2.5.5 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ dsPIC

ด้วยคุณสมบัติของไมโครคอนโทรลเลอร์สมัยใหม่มาต่อใช้งานหรือพอร์ตอินพุตเอาต์พุตนั้นมักมีความสามารถพิเศษร่วมอยู่ด้วย โดยไม่ได้เป็นเพียงพอร์ตอินพุตเอาต์พุตอย่างใดอย่างหนึ่งเพียงอย่างเดียวอีกต่อไป dsPIC ก็เช่นกัน ในแต่ละขาพอร์ตของ dsPIC ก็สามารถกำหนดให้ทำงานได้ทั้งเป็นพอร์ตอินพุต เอาต์พุต และขาสำหรับเชื่อมต่อใช้งานโมดูลพิเศษ

ในรูปที่ 2.20 แสดงโคแแกรมการทำงานพื้นฐานของขาพอร์ตอินพุตเอาต์พุตใน dsPIC แบ่งการทำงานเป็น 2 ส่วนคือ โมดูลพอร์ต (Port module) และเซลล์อินพุตเอาต์พุต (I/O Cell) โดยโมดูลพอร์ตจะใช้ในการกำหนดทิศทางของสัญญาณและเป็นที่พักข้อมูลชั่วคราวก่อนรับเข้าหรือส่งออก ส่วนเซลล์อินพุตเอาต์พุตเป็นส่วนที่เชื่อมต่อกับขาพอร์ตจริง บรรจวงจรบัฟเฟอร์เพื่อจัดการสัญญาณที่เข้าออกขาพอร์ตให้มีเสถียรภาพ



รูปที่ 2.20 โครงสร้างการทำงานพื้นฐานของขาพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์

dsPIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.5.1 รีจิสเตอร์ควบคุมขาพอร์ตอินพุตเอาต์พุต

ทุกขาพอร์ตอินพุตเอาต์พุตของ dsPIC มีรีจิสเตอร์ที่ใช้ในการควบคุมการทำงาน โดยตรง 3 ตัว คือ

TRISx รีจิสเตอร์กำหนดทิศทางข้อมูลของขาพอร์ต

PORTx รีจิสเตอร์ข้อมูลของขาพอร์ต

LATx รีจิสเตอร์แลตช์ข้อมูลของขาพอร์ต

(x เป็นค่าหรือตัวอักษรที่ใช้แทนตำแหน่งของขาพอร์ตใดๆ เนื่องจาก dsPIC มีกลุ่มของขาพอร์ตอินพุตเอาต์พุตจำนวนมาก)

(1) รีจิสเตอร์ TRIS

ใช้กำหนดว่าหน้าที่ของขาพอร์ตให้เป็นอินพุตหรือเอาต์พุตดิจิทัล

“0” = กำหนดเป็นเอาต์พุต

“1” = กำหนดเป็นอินพุต

หลังจากรีเซต ขาพอร์ตดิจิทัลทั้งหมดจะถูกกำหนดเป็นอินพุตก่อน หลังจากนั้นจึงเปลี่ยนแปลงตามการกำหนดที่รีจิสเตอร์ TRIS

(2) รีจิสเตอร์ PORT

เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่ปรากฏขึ้นที่ขาพอร์ตดิจิทัล โดยหากเป็นขาพอร์ตอินพุตข้อมูลที่ขาพอร์ตจะปรากฏที่รีจิสเตอร์ PORT สามารถอ่านได้โดยตรง ถ้ากำหนดเป็นขาพอร์ตเอาต์พุตข้อมูลที่รีจิสเตอร์ PORT จะเป็นข้อมูลที่ส่งออกไปยังอุปกรณ์ภายนอกที่ต่อกับขาพอร์ตจริง โดยสามารถเลือกให้ผ่านรีจิสเตอร์ LAT ก่อนก็ได้ เพื่อช่วยลดข้อผิดพลาดในกรณีที่ใช้ขาพอร์ตเดียวกันเป็นทั้งอินพุตและเอาต์พุต อย่างไรก็ตาม เพื่อความปลอดภัยควรเลือกใช้รีจิสเตอร์ PORT ในการอ่านค่าอินพุตเป็นหลัก ส่วนการส่งข้อมูลออกควรกระทำผ่านรีจิสเตอร์ LAT จะดีกว่า

(3) รีจิสเตอร์ LAT

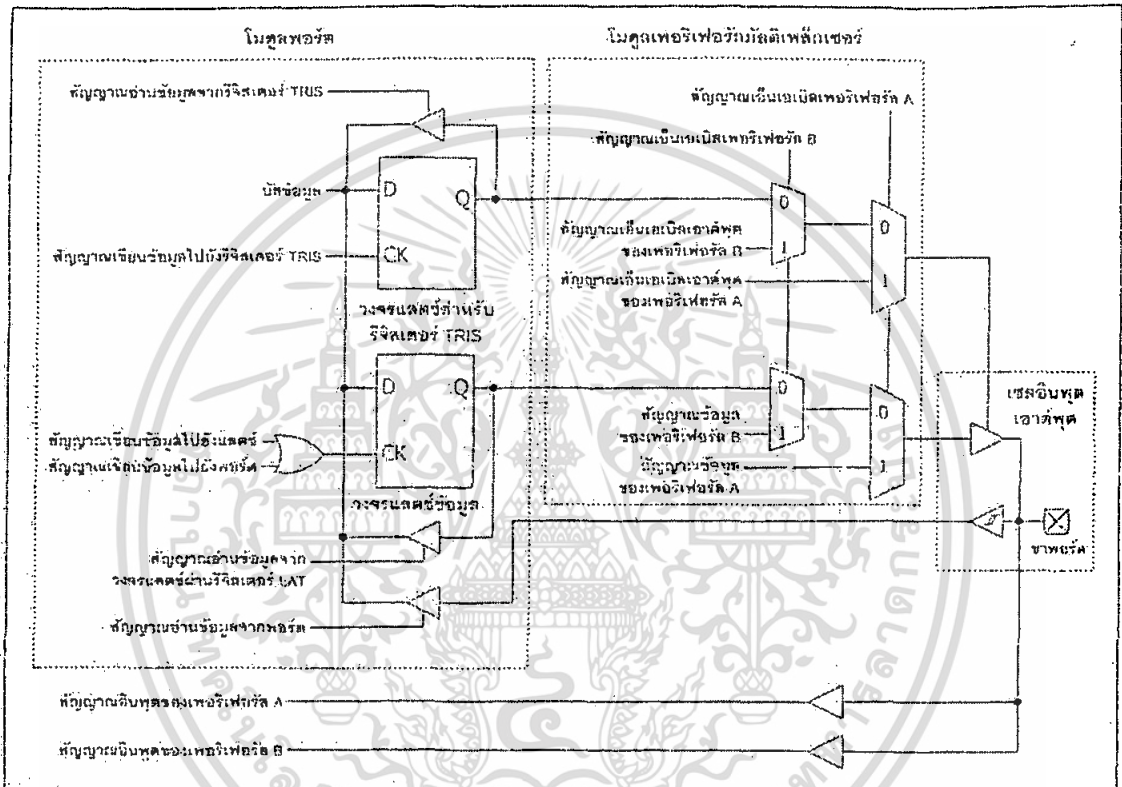
เป็นรีจิสเตอร์ที่ถูกใช้เก็บข้อมูลที่ต้องการส่งออกไปยังขาพอร์ตเมื่อทำงานเป็นเอาต์พุต ทั้งนี้เพื่อช่วยในกรณีที่ใช้ขาพอร์ตเป็นทั้งอินพุตและเอาต์พุตในเวลาใกล้เคียงกัน โดยมีความต้องการรักษาข้อมูลของเอาต์พุตเดิมไว้ แม้ว่าจะมีการเปลี่ยนแปลงที่ขาพอร์ตเกิดขึ้นเมื่อเป็นอินพุต ถ้าหากต้องการกลับสถานะลอจิกที่เอาต์พุตหรือ toggle จะสามารถทำได้โดยนำข้อมูลในรีจิสเตอร์ LAT นี้กลับลอจิกได้ทันที ซึ่งจะไม่สามารถทำเช่นนี้ได้หากใช้รีจิสเตอร์ PORT ดังนั้นผู้พัฒนาโปรแกรมสามารถเปลี่ยนแปลงค่าที่ต้องการส่งออกขาพอร์ตเอาต์พุตได้ตลอดเวลา พร้อมกันนั้นยังสามารถตรวจสอบข้อมูลเอาต์พุตได้ด้วย

การเปลี่ยนแปลงข้อมูลที่ขาพอร์ตหรือที่รีจิสเตอร์ PORT จะไม่ส่งผลกระทบใดๆ แก่รีจิสเตอร์ LAT เว้นแต่จะเกิดการรีเซตหรือเขียนข้อมูลใหม่มาทับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถสรุปความแตกต่างของการอ่านเขียนข้อมูลระหว่างรีจิสเตอร์ PORT และ LAT ได้ดังนี้

- เมื่อเขียนค่าไปยังรีจิสเตอร์ PORT ข้อมูลจะถูกส่งไปยังวงจรแลตช์ของพอร์ต
- เมื่อเขียนค่าไปยังรีจิสเตอร์ LAT ข้อมูลจะถูกส่งไปยังวงจรแลตช์ของพอร์ต
- ในการอ่านค่าของรีจิสเตอร์ PORT เป็นการอ่านค่าจากขาพอร์ตโดยตรง
- ในการอ่านค่าของรีจิสเตอร์ LAT ค่าที่ได้คือ ข้อมูลล่าสุดที่อยู่ในแลตช์ของพอร์ต



รูปที่ 2.21 โครงสร้างการทำงานของขาพอร์ตอินพุตเอาต์พุตดิจิตอลร่วมกับโมดูลฟังก์ชันพิเศษ

2.5.5.1 การสนับสนุนโมดูลพิเศษของขาพอร์ตอินพุตเอาต์พุต

เมื่อโมดูลฟังก์ชันพิเศษต่างๆ ที่ต้องใช้ขาพอร์ตอินพุตเอาต์พุตในการติดต่อได้รับการเอ็นเอเบิลให้ทำงาน ขาพอร์ตจะถูกจัดสรรให้เข้าไปทำงานร่วมกับโมดูลพิเศษนั้นๆ ทั้งนี้ โดยถ้าหากโมดูลนั้นต้องการใช้งานขาพอร์ตนั้นในลักษณะเป็นอินพุต ข้อมูลจะถูกอ่านผ่านขาพอร์ตเข้าไปยังบัสข้อมูลได้ ในขณะที่วงจรขับเอาต์พุตของขาพอร์ตนั้น จะถูกดิสเอเบิล

ในรูปที่ 1-13 แสดงให้เห็นถึงการใช้งานขาพอร์ตอินพุตเอาต์พุตดิจิตอลปกติร่วมกับโมดูลฟังก์ชันพิเศษซึ่งจะเรียกรวมกันว่า อุปกรณ์เพอร์ipheral (peripheral) โดยมีส่วนเพอร์ipheral

มัลติเพล็กซ์เซอร์ (Peripheral Multiplexer) เข้ามาจัดการให้ขาพอร์ตเชื่อมต่อระหว่างโมดูลพอร์ตกับโมดูลเพริเฟอรัล

อย่างไรก็ตาม ขาพอร์ต B ของไมโครคอนโทรลเลอร์ dsPIC ซึ่งเป็นขาพอร์ตอินพุตอนาลอกสำหรับวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล เมื่อต้องการกำหนดให้ทำงานกับโมดูลเพริเฟอรัลใดๆ ที่เป็นสัญญาณดิจิทัล จะต้องเซตบิต PCFGx ในรีจิสเตอร์ ADPCFG เป็น “1” ที่ตรงกับตำแหน่งของขาพอร์ตที่ต้องการใช้งานเพื่อเลือกให้ขาพอร์ตอนาลอกนั้นทำงานเป็นขาพอร์ตดิจิทัล เสมือนหนึ่งเป็นการปิดการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลไปด้วย

2.5.5.3 ตรวจสอบการเปลี่ยนแปลงลอจิกของขาพอร์ต

อีกหนึ่งคุณสมบัติที่น่าสนใจของขาพอร์ตอินพุตเอาต์พุตในไมโครคอนโทรลเลอร์ dsPIC คือ ความสามารถในการตรวจสอบการเปลี่ยนแปลงลอจิกหรือ Change Notification (CN) เมื่อเอ็นเอเบิลความสามารถนี้จะทำให้เสมือนกับว่า dsPIC มีอินพุตอินเตอร์รัปต์เพิ่มขึ้นทันที ทำให้การเขียนโปรแกรมเพื่อตรวจสอบการเปลี่ยนแปลงที่ขาพอร์ตทำได้ง่ายขึ้นอย่างมาก โดยไม่ต้องมีการวนตรวจสอบตลอดเวลา เพราะหน้าที่นี้ส่วนจัดการ CN จะจัดการเองทางฮาร์ดแวร์ เมื่อตรวจสอบพบความเปลี่ยนแปลงก็จะทำให้อินเตอร์รัปต์ จากนั้นซีพียูจึงไปตอบสนองการอินเตอร์รัปต์ต่อไป

2.5.6 การทดลองใช้งานไทมเมอร์ใน dsPIC

ไทมเมอร์/เคาน์เตอร์หลักใน dsPIC มี 5 ตัวคือ ไทมเมอร์ 1 ถึง ไทมเมอร์ 5 โดยแต่ละตัวมีขนาด 16 บิต สำหรับไทมเมอร์ 2 และ 3 กับไทมเมอร์ 4 และ 5 สามารถทำงานร่วมกันเป็นไทมเมอร์ขนาด 32 บิต เมื่อไทมเมอร์แต่ละตัวทำงานแยกกันสามารถกำหนดการทำงานได้อีก 3 แบบตามลักษณะของฐานเวลา คือ ฐานเวลาแบบ A B และ C ซึ่งจะได้อีกกล่าวถึงในลำดับถัดไป

2.5.6.1 คุณสมบัติของไทมเมอร์

ใน dsPIC30F4011 มีไทมเมอร์/เคาน์เตอร์ขนาด 16 บิต ให้ใช้งานรวม 3 ตัว คือ ไทมเมอร์ 1 (T1), ไทมเมอร์ 2 (T2) และ ไทมเมอร์ 3 (T3) , ไทมเมอร์ 4 (T4) และ ไทมเมอร์ 5 (T5)

(1) คุณสมบัติของไทมเมอร์ 1

- รีจิสเตอร์ตัวนับความละเอียด 16 บิต
- ทำงานได้ทั้งแบบซิงโครนัสและอะซิงโครนัสเคาน์เตอร์
- ทำงานร่วมกับขาอินพุตประจำตัวของไทมเมอร์ได้
- มีปริสเกลเลอร์สำหรับหารความถี่การนับ
- สามารถกำหนดอินเตอร์รัปจากการนับหรือจากการตรวจพบสัญญาณขอบขาลงที่ขาอินพุตของไทมเมอร์

(2) คุณสมบัติของไทมเมอร์ 2,3,4 และ 5

- ไทมเมอร์ 2,3,4 และ 5เมื่อทำงานแยกอิสระต่อกัน มีคุณสมบัติคล้ายกับไทมเมอร์ 1
- เมื่อนำไทมเมอร์ 2 และ 3 มาทำงานร่วมกัน รีจิสเตอร์ตัวนับมีความละเอียดเพิ่มขึ้น 32 บิต
- เมื่อนำไทมเมอร์ 4 และ 5 มาทำงานร่วมกัน รีจิสเตอร์ตัวนับมีความละเอียดเพิ่มขึ้น 32 บิต
- ทำงานร่วมกับขาอินพุตประจำตัวไทมเมอร์ได้ (ขา TxCKI)
- มีปริสเกลเลอร์สำหรับหารความถี่การนับ
- สามารถกำหนดอินเตอร์รัปจากการนับหรือจากการตรวจพบสัญญาณขอบขาลงที่ขาอินพุตของไทมเมอร์
- สามารถกำเนิดสัญญาณกระตุ้นการทำงานไปยังโมดูล ADC ได้

2.5.6.2 รีจิสเตอร์ในโมดูลไทมเมอร์ของ dsPIC30F

ในโมดูลไทมเมอร์ของ dsPIC30F4011 มีรีจิสเตอร์ขนาด 16 บิตที่สำคัญอยู่ 3 กลุ่ม คือ กลุ่มที่ 1 TMRx (16-bit timer counter register) เป็นกลุ่มของรีจิสเตอร์เก็บค่าไทมเมอร์กลุ่มนี้มี 5 ตัว คือ TMR1, TMR2, TMR3, TMR4 และ TMR5

กลุ่มที่ 2 PRx (16-bit period register associated with the timer) เป็นกลุ่มของรีจิสเตอร์คาบเวลาที่สัมพันธ์กับไทมเมอร์ มี 5 ตัวเช่นกัน PR1 ถึง PR5 แยกกันตามไทมเมอร์หลักทั้ง 5 ตัว (ไทมเมอร์ 1 ถึง ไทมเมอร์ 5)

กลุ่มที่ 3 TxCON (16-bit control register associated with the timer) เป็นกลุ่มของรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไทมเมอร์ มี 5 ตัวเช่นกัน T1CON ถึง T5CON แยกกันตามไทมเมอร์หลักทั้ง 5 ตัว สำหรับกลุ่มนี้จะมีความพิเศษตรงที่ยังสามารถแยกย่อยออกไปได้ 3 แบบภายใต้ชื่อรีจิสเตอร์เดียวกัน เนื่องจากไทมเมอร์สามารถทำงานได้อีก 3 แบบตามลักษณะของฐานเวลา จึงทำให้รีจิสเตอร์ TxCON สามารถระบุได้เป็น TxCON ในแบบ A, B และ C เวลาเขียนโปรแกรมเพื่อกำหนดค่าให้รีจิสเตอร์ TxCON จะต้องระมัดระวังในจุดนี้ด้วย

2.5.6.3 การทำงานของไทมเมอร์ตามลักษณะของฐานเวลา

มีทั้งสิ้น 3 แบบคือ ไทมเมอร์ที่ทำงานด้วยฐานเวลาแบบ A, B และ C มีข้อมูลโดยสรุปดังนี้

(1) ไทมเมอร์ที่ทำงานด้วยฐานเวลาแบบ A

เป็นการทำงานในแบบมาตรฐานของ dsPIC โดยไทมเมอร์ที่ทำงานในแบบนี้คือ ไทมเมอร์ 1 ซึ่งมีคุณสมบัติพิเศษเพิ่มเติมดังนี้

- กำหนดให้ทำงานกับแหล่งจ่ายสัญญาณนาฬิกาจำกัดค่า 32 kHz จากภายนอกได้
- กำหนดทำงานในแบบอะซิงโครนัส โดยรับสัญญาณนาฬิกาจากภายนอกได้

ดังนั้นไทมเมอร์ 1 จึงเหมาะอย่างยิ่งในการนำมาใช้สร้างฐานเวลานาฬิกาจริง (RTC)

(2) ไทมเมอร์ที่ทำงานด้วยฐานเวลาแบบ B

ไทมเมอร์ที่สามารถทำงานในลักษณะนี้ได้มี 2 ตัวคือ ไทมเมอร์ 2 และ ไทมเมอร์ 4 คุณสมบัติที่เพิ่มเติมเข้ามาสำหรับฐานเวลาแบบ B คือ

- สามารถต่อกับไทมเมอร์ฐานเวลาแบบ C เพื่อใช้งานเป็นไทมเมอร์ขนาด 32 บิตได้ โดยการกำหนดสถานะที่บิต T32 ในรีจิสเตอร์ TxCON เมื่อทำงานด้วยฐานเวลาแบบ B ให้เป็นลอจิก “1”
- มีวงจรกำหนดจังหวะของสัญญาณนาฬิกาให้สัมพันธ์กัน (ซิงโครไนซ์) หลังจากผ่านการปรับอัตราลดทอนความถี่ด้วยพรีสเกลเลอร์แล้ว

(3) ไทมเมอร์ที่ทำงานด้วยฐานเวลาแบบ C

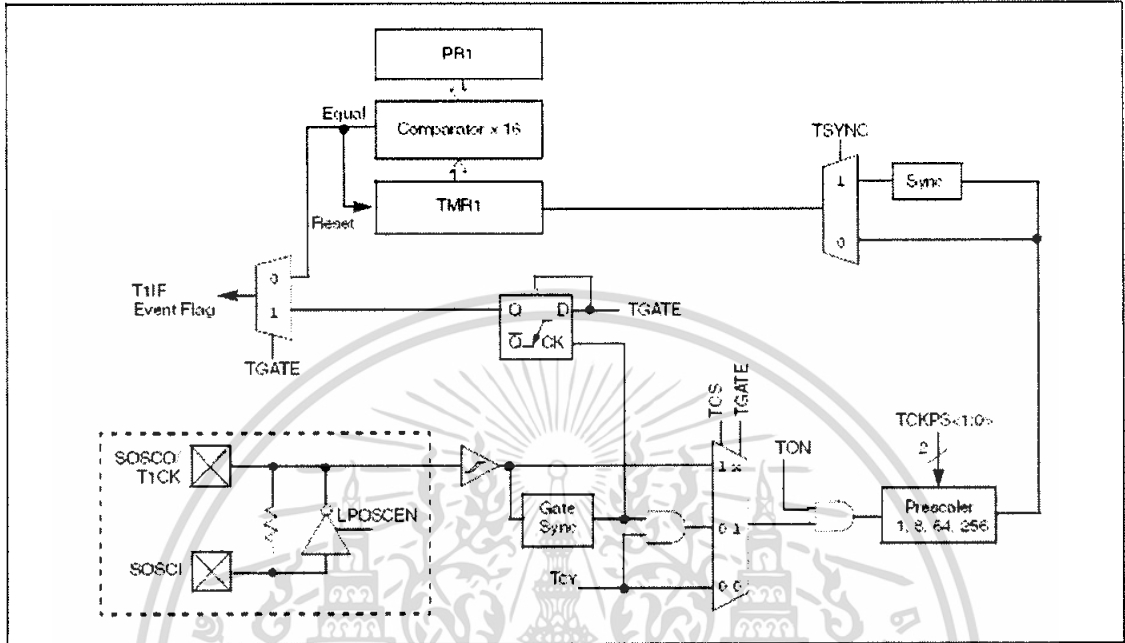
ไทมเมอร์ที่สามารถทำงานในลักษณะนี้ได้มี 2 ตัวคือ ไทมเมอร์ 3 และไทมเมอร์ 5 คุณสมบัติที่เพิ่มเติมเข้ามาสำหรับฐานเวลาแบบ C คือ

- สามารถต่อกับไทมเมอร์ฐานเวลาแบบ B เพื่อใช้งานเป็นไทมเมอร์ 32 บิตได้ โดยการกำหนดสถานะที่บิต T32 ในรีจิสเตอร์ TxCON เมื่อทำงานด้วยฐานเวลาแบบ B

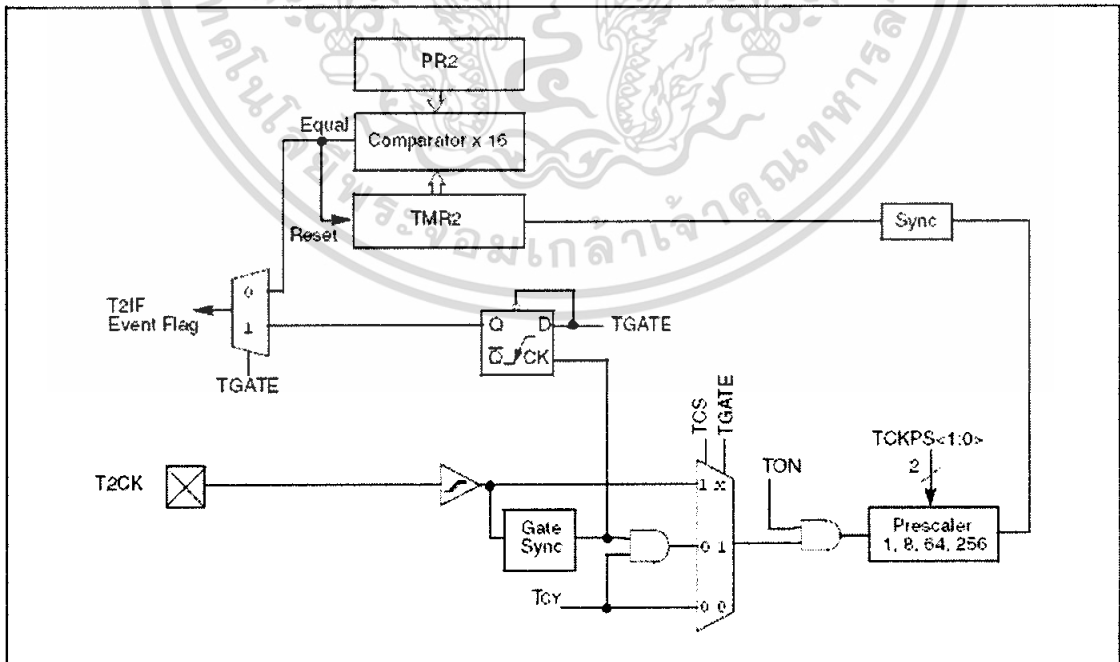
ให้เป็นลอจิก “1” ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถสร้างสัญญาณกระตุ้นการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลได้ ทำให้ไทเมอร์และโมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลทำงานสัมพันธ์กัน

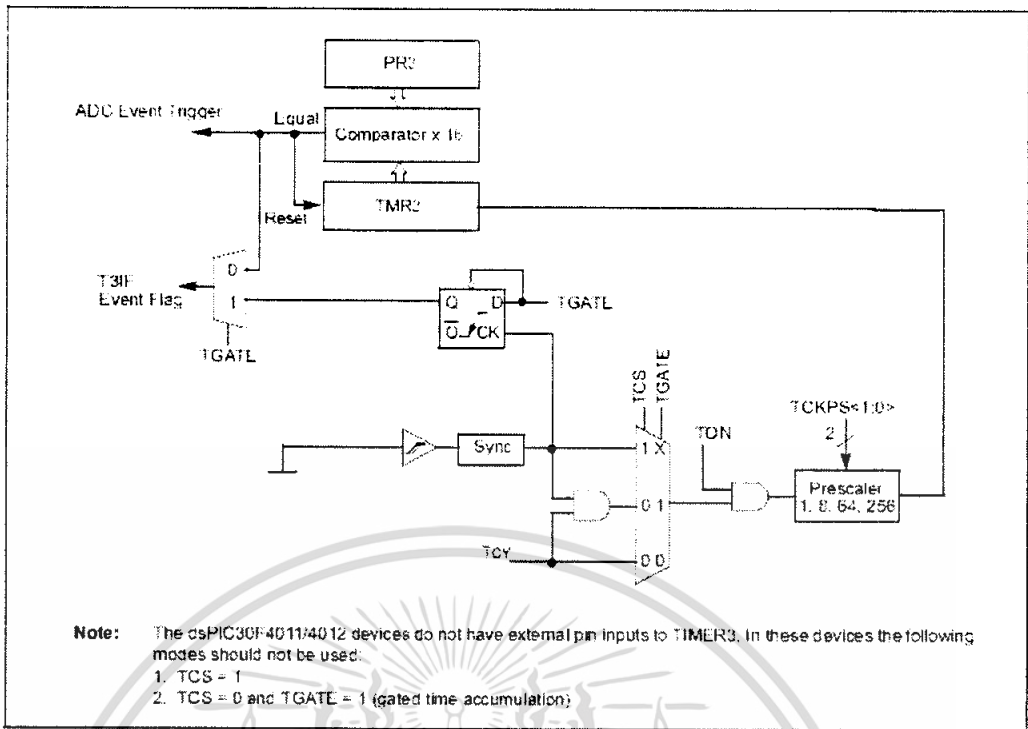


รูปที่ 2.22 แสดงไต่แกรมการทำงานของไทเมอร์ 1 ฐานเวลาแบบ A

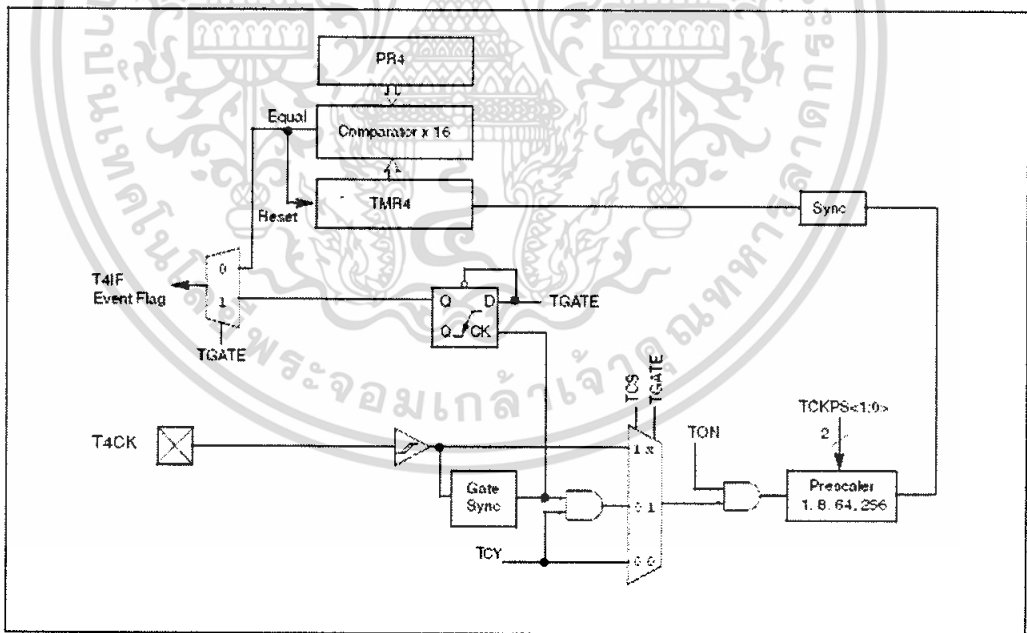


รูปที่ 2.23 แสดงไต่แกรมการทำงานของไทเมอร์ 2 ฐานเวลาแบบ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

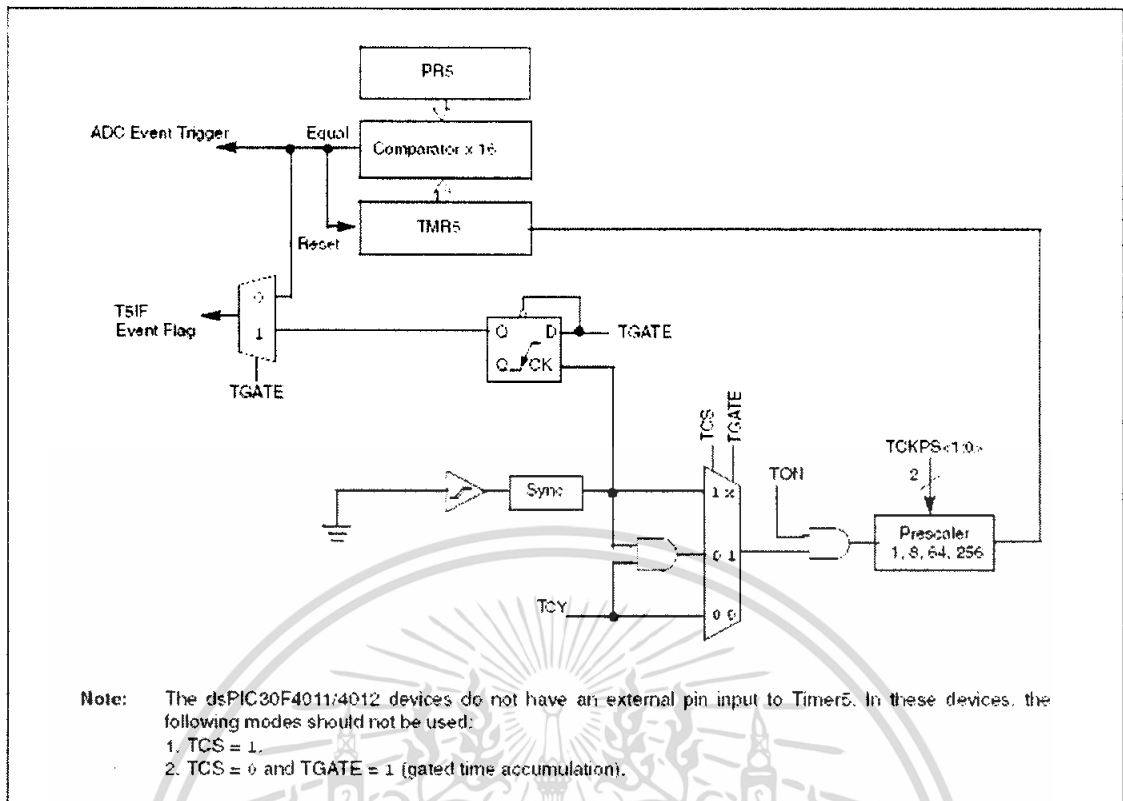


รูปที่ 2.24 แสดงไดอะแกรมการทำงานของไทมเมอร์ 3 ฐานเวลาแบบ C



รูปที่ 2.25 แสดงไดอะแกรมการทำงานของไทมเมอร์ 4 ฐานเวลาแบบ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 แสดงไดอะแกรมการทำงานของไทมเมอร์ 5 ฐานเวลาแบบ C

2.5.6.4 โหมดการทำงานของไทมเมอร์/เคาน์เตอร์

ไทมเมอร์/เคาน์เตอร์ทุกตัวใน dsPIC สามารถกำหนดโหมดการทำงานได้ 4 แบบ ดังนี้

- 1) ไทมเมอร์แบบซิงโครนัส (synchronous timer)
- 2) เคาน์เตอร์แบบซิงโครนัส (synchronous counter)
- 3) ไทมเมอร์แบบควบคุมการทำงานจากสัญญาณจากภายนอกหรือเกตไทมเมอร์ (Gated timer)
- 4) เคาน์เตอร์แบบอะซิงโครนัส (asynchronous counter)

การกำหนดโหมดการทำงานของเคาน์เตอร์จะขึ้นอยู่กับสถานะของบิตควบคุม 3 บิต

ดังต่อไปนี้

- 1) TCS (บิต 1 ของรีจิสเตอร์ TxCON) ใช้เลือกแหล่งกำเนิดสัญญาณนาฬิกาของไทมเมอร์ ว่าต้องการใช้จากภายในหรือภายนอก
- 2) TSYNC (บิต 2 ของรีจิสเตอร์ TxCON) ใช้ควบคุมการซิงโครไนซ์ของสัญญาณ เมื่อทำงานในฐานเวลาแบบ A เท่านั้น

3) TGATE (บิต 6 ของรีจิสเตอร์ TxCON) ใช้ควบคุมการเปิดปิดเกตเมื่อทำงานกับ สัญญาณนาฬิกาจากภายนอก

แต่บิตควบคุมหลักที่สำคัญที่สุดของไทเมอร์คือ บิต TON ซึ่งเป็นบิต 15 ของรีจิสเตอร์ TxCON เนื่องจากเป็นบิตสำหรับเปิดปิดการทำงานของไทเมอร์ ดังนั้นต้องทำการเซตบิตนี้ก่อนเสมอ มิฉะนั้นไทเมอร์จะไม่สามารถทำงานได้

(1) ไทเมอร์แบบซิงโครนัส

ในโหมดนี้สามารถใช้กับฐานเวลาได้ทุกแบบ โดยสัญญาณนาฬิกาอินพุตได้มาจาก สัญญาณนาฬิกาภายในของระบบ มีความถี่เท่ากับ ความถี่สัญญาณนาฬิกาหลักหาร 4 ($F_{osc}/4$) เมื่อ เอนเนเบิลให้ทำงาน ค่าของรีจิสเตอร์ TMRx จะเพิ่มค่าขึ้น 1 ค่าในทุกๆ ไซเคิลการทำงานในกรณีที่ เลือกอัตราปรีสเกลเลอร์เป็น 1:1 การเลือกโหมดนี้ทำได้โดยการเคลียร์บิต TCS (บิต 1 ของรีจิสเตอร์ TxCON ตัวอักษร x เป็นเลขของรีจิสเตอร์มีค่า 1 ถึง 5)

(2) เคน์เตอร์แบบซิงโครนัส ใช้สัญญาณนาฬิกาจากภายนอก

ไทเมอร์จะเข้าสู่โหมดนี้เมื่อบิต TCS ถูกเซตจึงสามารถใช้สัญญาณนาฬิกาจากภายนอก ผ่านทางขาอินพุต TxCK โดยค่าของเคน์เตอร์จะเพิ่มขึ้นที่ทุกขอบขาขึ้นของสัญญาณนาฬิกา

หากเลือกใช้ฐานเวลาแบบ A จะต้องมีการกำหนดให้สัมพันธ์กับจังหวะการทำงาน ภายในหรือการซิงโครไนซ์ด้วยการเซตบิต TSYNC (บิต 2 ของรีจิสเตอร์ TxCON) ในขณะที่ฐาน เวลาแบบ B และ C จะได้รับการซิงโครไนซ์กับสัญญาณนาฬิกาการทำงาน (TCY) โดยอัตโนมัติ

เมื่อไทเมอร์ถูกกำหนดให้ทำงานเป็นเคน์เตอร์แบบซิงโครนัส ค่าเวลาลอจิกสูง (high time) และลอจิกต่ำ (low time) ของสัญญาณนาฬิกาจากภายนอกจะต้องมีค่าตามที่กำหนดในค่า ซีตของ dsPIC เนื่องจากจะต้องกำหนดให้สัมพันธ์กับสัญญาณนาฬิกาของการทำงานภายใน dsPIC นั้นเอง

หากไมโครคอนโทรลเลอร์ dsPIC เข้าสู่โหมดสลีป การทำงานของไทเมอร์ในโหมดนี้ จะหยุดลงทันที

(3) การทำงานของไทเมอร์กับแหล่งกำเนิดสัญญาณนาฬิกาความถี่สูงจากภายนอก

ในการทำงานบางครั้งมีความต้องการให้ไทเมอร์นับค่าสัมพันธ์กับสัญญาณนาฬิกาจาก แหล่งกำเนิดสัญญาณนาฬิกาภายนอกความถี่สูง จึงทำให้ต้องใช้ไทเมอร์ 2 ตัวทำงานร่วมกัน นั่นคือ ใช้ไทเมอร์ตัวหนึ่งทำงานด้วยฐานเวลาแบบ A และอีกตัวหนึ่งทำงานด้วยฐานเวลาแบบ B โดยต้อง ใช้ปรีสเกลเลอร์ช่วยในการลดทอนสัญญาณนาฬิกาจากแหล่งกำเนิดที่สูงกว่า เพื่อให้สามารถทำงาน ได้ทันและสัมพันธ์กัน

(4) ไทเมอร์แบบควบคุมการทำงานจากสัญญาณจากภายนอกหรือเกตไทเมอร์

ในโหมดนี้ค่าของรีจิสเตอร์ไทเมอร์ TMRx จะเพิ่มค่าขึ้นเมื่อเกิดสัญญาณลอจิกสูงที่ขาพอร์ตอินพุต TxCK นั่นหมายความว่า ไทเมอร์จะสามารถทำการนับค่าขึ้นได้เมื่อขา TxCK เป็นลอจิก “1” และไทเมอร์ยังคงนับค่าขึ้นอย่างต่อเนื่องตราบเท่าที่ขา TxCK ยังเป็นลอจิก “1” เมื่อค่าของรีจิสเตอร์ไทเมอร์ยังเท่ากับค่าของรีจิสเตอร์คาบเวลาหรือ PRx ไทเมอร์ก็จะรีเซ็ต แล้วเริ่มต้นการนับค่าใหม่ โดยไม่มีการอินเตอร์รัปต์เกิดขึ้น แต่ถ้าหากขา TxCK เปลี่ยนสถานะจากลอจิกสูงเป็นลอจิกต่ำ ไทเมอร์ก็จะหยุดการทำงาน บิตแฟล็ก TxIF ก็จะเซต และเกิดการอินเตอร์รัปต์ขึ้นหากได้รับการเอ็นเอเบิลไว้

การกำหนดโหมดการทำงานนี้ทำได้โดยการ

- 1) เซตบิต TGATE (บิต 6 ของรีจิสเตอร์ TxCON) เพื่อเอ็นเอเบิลการทำงานในโหมดนี้
- 2) เซตบิต TON (บิต 15 ของรีจิสเตอร์ TxCON) เพื่อเปิดการทำงานของไทเมอร์
- 3) เซตบิต TCS (บิต 1 ของรีจิสเตอร์ TxCON) เพื่อเลือกแหล่งกำเนิดสัญญาณนาฬิกาจากสัญญาณนาฬิกาในการทำงานหรือ T_{DY}

การทำงานโหมดนี้จะเริ่มต้นขึ้นเมื่อเกิดขอบขาขึ้นของสัญญาณเกต (ลอจิกสูง) ที่ขาพอร์ต TxCK ไทเมอร์จะเริ่มทำการนับค่า เมื่อที่ขาพอร์ต TxCK เกิดขอบขาลงของสัญญาณเกต (ลอจิกต่ำ) การนับค่าของไทเมอร์จะหยุดลงทันที แต่ยังไม่มีการรีเซ็ตไทเมอร์ ผู้พัฒนาโปรแกรมจะต้องรีเซ็ตค่าของไทเมอร์ เนื่องจากไทเมอร์ต้องเริ่มต้นด้วยค่าศูนย์เมื่อเริ่มทำงานทุกครั้ง แต่ถ้าหากการนับค่าของไทเมอร์เพิ่มไปจนถึงค่าที่กำหนดไว้ในรีจิสเตอร์คาบเวลา PRx ไทเมอร์ก็จะสามารถรีเซ็ตค่าได้เอง แล้วเริ่มต้นการนับใหม่

ความละเอียดของการนับค่าของไทเมอร์จะขึ้นอยู่กับคาบเวลาของสัญญาณนาฬิกาของไทเมอร์ ซึ่งสามารถกำหนดได้จากอัตราปรีสเกลเลอร์ หากเลือกอัตราปรีสเกลเลอร์เป็น 1:1 คาบเวลาสัญญาณนาฬิกาของไทเมอร์จะเท่ากับ 1 ไซเคิลการทำงาน ถ้าหากเลือกอัตราปรีสเกลเลอร์เป็น 1:256 คาบเวลาสัญญาณนาฬิกาของไทเมอร์จะเท่ากับ 256 เท่าของไซเคิลการทำงาน ความละเอียดของสัญญาณนาฬิกาของไทเมอร์สามารถนำมากำหนดค่าความกว้างของสัญญาณเกตได้

5. เคาน์เตอร์แบบอะซิงโครนัสฐานเวลาแบบ A โดยใช้สัญญาณนาฬิกาจากภายนอก

ในโหมดนี้ใช้ไทเมอร์ฐานเวลาแบบ A นั่นคือใช้ไทเมอร์ 1 ทำงานเป็นเคาน์เตอร์แบบอะซิงโครนัส โดยสัญญาณนาฬิกาที่มาจากภายนอกส่งเข้ามาทางขา TxCK การเลือกโหมดนี้ทำได้โดยการเคลียร์บิต TSYNC (บิต 2 ของรีจิสเตอร์ TxCON) ในโหมดนี้สัญญาณนาฬิกาจากภายนอกจะไม่สัมพันธ์หรือซิงโครไนซ์กับสัญญาณนาฬิกาหลักของระบบ

ฐานเวลาของไทเมอร์ในโหมดนี้สามารถทำงานได้อย่างต่อเนื่อง แม้ว่าไมโครคอนโทรลเลอร์จะเข้าสู่โหมดสลีปและสามารถกำหนดสัญญาณอินเตอร์รัปต์เมื่อค่าของรีจิสเตอร์ TMRx เท่ากับ PRx ซึ่งจะเป็นการกระตุ้นให้ไมโครคอนโทรลเลอร์เกิดการเวก-อัปออกจากโหมดสลีป

นอกจากนั้นยังสามารถเลือกใช้สัญญาณความถี่ 32kHz จากวงจรออสซิลเลเตอร์กำลังงานต่ำ เพื่อสามารถนำมาใช้สร้างฐานเวลาพิกางจรหรือรีลไทม์คล็อกได้

2.5.6.5 การอินเตอร์รัปต์ในไทเมอร์

การอินเตอร์รัปต์เนื่องจากการทำงานของไทเมอร์มีบิตที่เกี่ยวข้องโดยตรง 5 บิตคือ

- 1) TxIE บิตเอ็นเอเบิลการอินเตอร์รัปต์เนื่องจากการทำงานของไทเมอร์
- 2) TxIP2 ถึง TxIP0 บิตเลือกระดับความสำคัญของการอินเตอร์รัปต์
- 3) TxIF บิตเฟลทกแสดงสถานการณ์เกิดอินเตอร์รัปต์

ในไทเมอร์ 16 บิตสามารถกำเนิดสัญญาณอินเตอร์รัปต์เมื่อ

- 1) ค่าของรีจิสเตอร์ TMRx เท่ากับรีจิสเตอร์คาบเวลา PRx
- 2) มีการเปลี่ยนจากลอจิกสูงไปต่ำหรือเกิดขอบขาลงของสัญญาณเกิดในโหมดเกตไทเมอร์บิต TxIF จะถูกเซตเมื่อเงื่อนไขดังต่อไปนี้เป็นจริง
 - 1) ค่าของรีจิสเตอร์ TMRx เท่ากับรีจิสเตอร์คาบเวลา PRx และไทเมอร์ต้องไม่ทำงานในโหมดเกตไทเมอร์
 - 2) มีการเปลี่ยนลอจิกจากสูงไปต่ำหรือเกิดขอบขาลงของสัญญาณเกิดในโหมดเกตไทเมอร์

การเคลียร์บิต TxIF ต้องกระทำด้วยกระบวนการทางซอฟต์แวร์เท่านั้น

2.5.6.6 การอ่านและเขียนข้อมูลกับรีจิสเตอร์ของไทเมอร์

การเขียนข้อมูลลงในรีจิสเตอร์ที่ใช้งานในไทเมอร์สามารถเขียนได้ทั้งในแบบ 8 และ 16 บิต ในขณะที่การอ่านค่าจะอ่านได้ในแบบ 16 บิตเท่านั้น

ผู้พัฒนาสามารถเขียนข้อมูลไปยังรีจิสเตอร์เก็บค่าของไทเมอร์ TMRx และรีจิสเตอร์คาบเวลา PRx ในขณะที่ไทเมอร์ได้รับการเอ็นเอเบิลให้ทำงาน แต่มีข้อควรระวังที่ควรทราบในกรณีเขียนข้อมูลแบบไบนารีไปยังรีจิสเตอร์ดังนี้

- 1) ถ้าไทเมอร์กำลังเพิ่มค่าขึ้น แล้วมีการเขียนข้อมูล ไบนารีไปตรงกลาง ข้อมูลในไบนารีของจิสเตอร์จะไม่ได้รับผลกระทบ เช่น ถ้าหากมีการเขียนข้อมูล 0xFF ไปยังไบนารีตรงกลางของรีจิสเตอร์เก็บค่าไทเมอร์ ในการนับครั้งต่อไป ค่าของไทเมอร์ที่ไบนารีตรงกลางจะเปลี่ยนเป็น 0x00 และเกิดการทศค่าไปยังไบนารีสูงของรีจิสเตอร์ไทเมอร์ต่อไป
- 2) ถ้าไทเมอร์กำลังเพิ่มค่าขึ้น แล้วมีการเขียนข้อมูล ไบนารีไปตรงบน ข้อมูลในไบนารีตรงกลางของ

รีจิสเตอร์จะไม่ได้รับผลกระทบ เช่น ถ้าข้อมูลไบต์ล่างเป็น 0xFF ในขณะที่เกิดการเขียนข้อมูลไปยังไบต์บน ในการนับค่าครั้งต่อไป ค่าของไทเมอร์ที่ไบต์ล่างจะเปลี่ยนเป็น 0x00 และเกิดการทดค่าไปยังไบต์สูงของรีจิสเตอร์ไทเมอร์ ทำให้เกิดการเพิ่มค่าข้อมูลไบต์บนจากค่าที่เขียนไปก่อนหน้านี้

2.5.6.7 อินพุตสัญญาณนาฬิกาความถี่ 32kHz จากคริสตอลออสซิลเลเตอร์กำลังงานต่ำ

ไทเมอร์จะสามารถทำงานร่วมกับอินพุตนี้เมื่อกำหนดให้ทำงานด้วยฐานเวลาแบบ A และเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาของไทเมอร์จากแหล่งภายนอก ส่วนโหมดของสัญญาณนาฬิกานั้นต้องเลือกเป็นโหมด LP หรือคริสตอลกำลังงานต่ำด้วยการเซตบิต LPOSCEN ในรีจิสเตอร์ OSCCON และต้องต่อคริสตอลความถี่ 32kHz เข้าที่ขา SOSC1 และ SOSC2

2.5.6.8 การกำหนดให้ทำงานเป็นไทเมอร์ขนาด 32 บิต

ไทเมอร์ขนาด 32 บิตเกิดจากการนำไทเมอร์ขนาด 16 บิตที่ทำงานด้วยฐานเวลาแบบ B และ C มาทำงานร่วมกัน โดยไทเมอร์ฐานเวลาแบบ C (ไทเมอร์ 3 หรือ 5) ใช้เก็บข้อมูล 16 บิตบนหรือเวิร์ดสูง (MS word) ส่วนไทเมอร์ฐานเวลาแบบ B (ไทเมอร์ 2 หรือ 4) ใช้เก็บ 16 บิตล่างหรือเวิร์ดต่ำ (LS word)

เมื่อทำงานเป็นไทเมอร์ขนาด 32 บิต การควบคุมการทำงานของไทเมอร์จะกำหนดผ่านทางไทเมอร์ฐานเวลา B นั่นคือกำหนดผ่านค่ารีจิสเตอร์ T2CON หรือ T4CON แล้วแต่กรณี สำหรับ dsPIC30F2010 จะใช้ T2CON เว้นแต่กรณีอินเทอร์รัปต์ จะไปกำหนดค่าที่ไทเมอร์ฐานเวลา C ซึ่งก็คือรีจิสเตอร์ T3CON หรือ T5CON สำหรับ dsPIC30F2010 จะใช้ T3CON

ยกตัวอย่างการกำหนดให้ไทเมอร์ใน dsPIC30F2010 ทำงานเป็นแบบ 32 บิต นั่นคือใช้ไทเมอร์ 2 และไทเมอร์ 3 โดยการควบคุมปกติจะกระทำที่รีจิสเตอร์ T2CON ส่วนการจัดการอินเทอร์รัปต์กระทำผ่านรีจิสเตอร์ T3CON ขึ้นตอนโดยสรุปดังนี้

- 1) เซตบิต TON (บิต 15 ในรีจิสเตอร์ T2CON) เพื่อเปิดการทำงานของไทเมอร์
- 2) เซตบิต T32 (บิต 3 ในรีจิสเตอร์ T2CON) เพื่อเลือกการทำงานแบบ 32 บิต
- 3) กำหนดอัตราปรีสเกลเลอร์ (ถ้าต้องการ) ที่บิต TCKPS1 และ TCKPS0 (บิต 5 และ 4 ของรีจิสเตอร์ T2CON)
- 4) รีจิสเตอร์ PR3 และ PR2 ถูกใช้เก็บค่าคาบเวลาขนาด 32 บิต สำหรับเปรียบเทียบกับค่าของรีจิสเตอร์ไทเมอร์ TMR3 และ TMR2
- 5) บิต T3IE (บิต 7 ในรีจิสเตอร์ IEC0) ถูกใช้ในการเอ็นเอเบิลการอินเทอร์รัปต์ของไทเมอร์ 32 บิต
- 6) บิต T3IF (บิต 7 ในรีจิสเตอร์ IFS0) ถูกใช้แสดงสถานการณ์อินเทอร์รัปต์ของไทเมอร์ 32 บิต

- 7) กำหนดค่าลงในบิต T3IP2 ถึง T3IP0 (บิต 14 ถึง 12 ของรีจิสเตอร์ IPC1) เพื่อ

เลือกระดับความสำคัญของการตอบสนองอินเทอร์รัทซ์ของไทเมอร์ 32 บิต

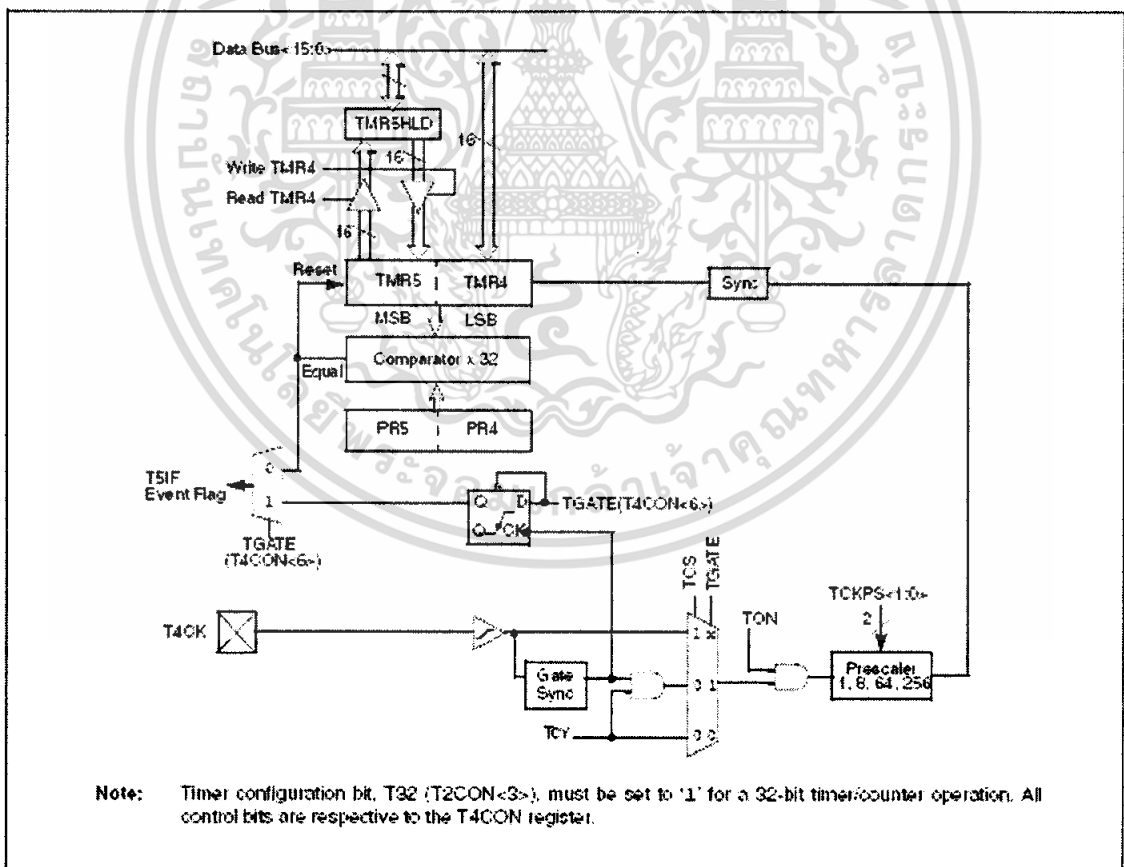
8) ไม่สนใจการทำงานของรีจิสเตอร์ T3CON

2.5.6.9 การอ่านและเขียนข้อมูลในไทเมอร์ 32 บิต

การอ่านและเขียนข้อมูลในไทเมอร์ 32 บิตจะต้องกำหนดให้สัมพันธ์กันระหว่างข้อมูลในเวิร์ดสูงและเวิร์ดต่ำร่วมกับรีจิสเตอร์ที่ใช้ในการพักค่าของรีจิสเตอร์ที่ทำงานด้วยฐานเวลาแบบ C ซึ่งใน dsPIC30F4011 คือ ไทเมอร์ 3 ซึ่งก็คือ TMR3HLD

ในการอ่านค่า จะต้องเริ่มอ่านจากเวิร์ดต่ำ ซึ่งก็คือค่าของรีจิสเตอร์ TMR2 ในจังหวะเดียวกันนั้นค่าของรีจิสเตอร์ TMR3 จะถูกโหลดไปยังรีจิสเตอร์ TMR3HLD โดยอัตโนมัติ ดังนั้นจึงสามารถอ่านค่าเวิร์ดสูงของไทเมอร์ 32 บิตได้จากรีจิสเตอร์ TMR3HLD ได้ทันที

ส่วนการเขียนข้อมูลนั้น จะต้องเริ่มต้นด้วยการเขียนข้อมูลเวิร์ดสูงไปยังรีจิสเตอร์ TMR3HLD ก่อน จากนั้นจึงเขียนข้อมูลเวิร์ดต่ำไปยังรีจิสเตอร์ TMR2 ข้อมูลจากรีจิสเตอร์ TMR3HLD จะถูกถ่ายทอดไปยังรีจิสเตอร์ TMR3 โดยอัตโนมัติ ทำให้การเขียนข้อมูลของไทเมอร์ 32 บิตเสร็จสิ้นลง



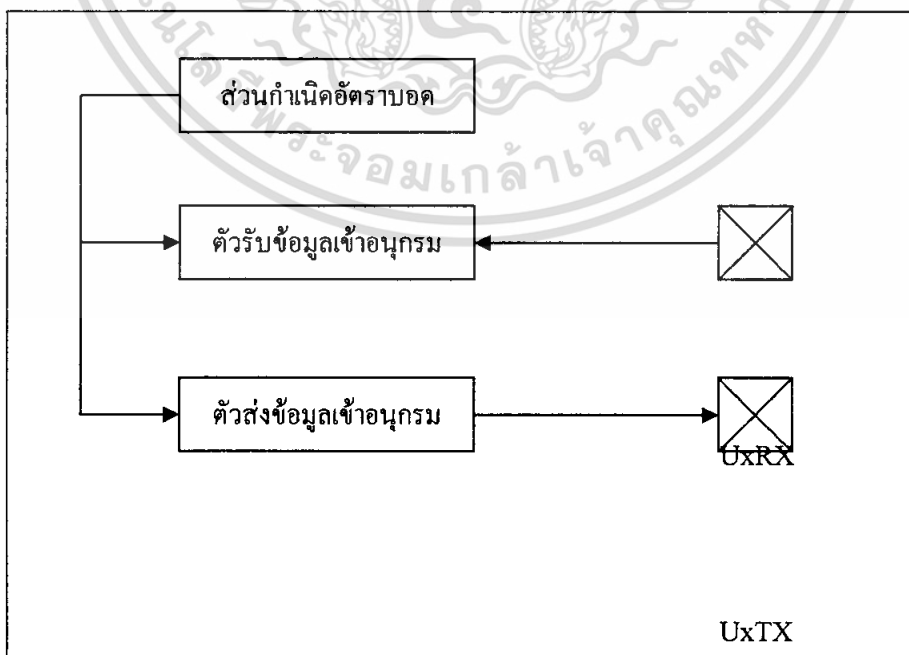
รูปที่ 2.27 แสดงการทำงานของไทเมอร์ใน dsPIC30F4011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7 การสื่อสารข้อมูลอนุกรมโดยใช้โมดูล UART ใน dsPIC30F4011

2.5.7.1 คุณสมบัติโดยสรุปของโมดูล UART ใน dsPIC30F4011

- สื่อสารข้อมูลแบบสองทิศทาง (ฟูลดูเพล็กซ์ : full-duplex) ในแบบ 8 และ 9 บิต
- เลือกการสื่อสารข้อมูลแบบตรวจสอบบิตพาริตีคู่ (Even) หรือคี่ (Odd) และไม่ตรวจสอบบิตพาริตี (None) สำหรับรูปแบบสื่อสารข้อมูลในแบบ 8 บิต มี บิตหยุด (Stop bit) 1 หรือ 2 บิต
- มีส่วนกำเนิดอัตราบอด (Baud Rate Generator) ขนาด 16 บิต สำหรับกำหนดจังหวะและอัตราเร็วในการสื่อสารข้อมูลอนุกรมแยกอิสระเพื่อลดภาระการทำงานของโมดูลไทมเมอร์
- กำหนดอัตราบอดได้ตั้งแต่ 38 บิตต่อวินาที (bps) ถึง 1.875 เมกะบิตต่อวินาที (Mbps)
- บัฟเฟอร์ข้อมูลขาส่ง (TX) และขารับ (RX) ขนาด 4 เวิร์ด แยกส่วนกัน
- มีบิตแฟล็กแจ้งข้อผิดพลาดในกรณีต่างๆ ของการสื่อสาร สามารถตรวจจับความผิดพลาดในการสื่อสารข้อมูลอนุกรม ได้แก่
 - ความผิดพลาดทางพาริตี (Parity Error : PE)
 - รับข้อมูลไม่ทัน (Buffer Overrun Error : OE)
 - เฟรมข้อมูลผิดพลาด (Framing Error : FE)
- สนับสนุนความสามารถในการอินเตอร์รัปต์แอสแตเรส (ข้อมูลบิต 9 เป็น "1")
- อินเตอร์รัปต์เวกเตอร์แยกตำแหน่งกันระหว่างการส่งและรับข้อมูล (RX)
- สามารถทำงานในโหมด Loopback



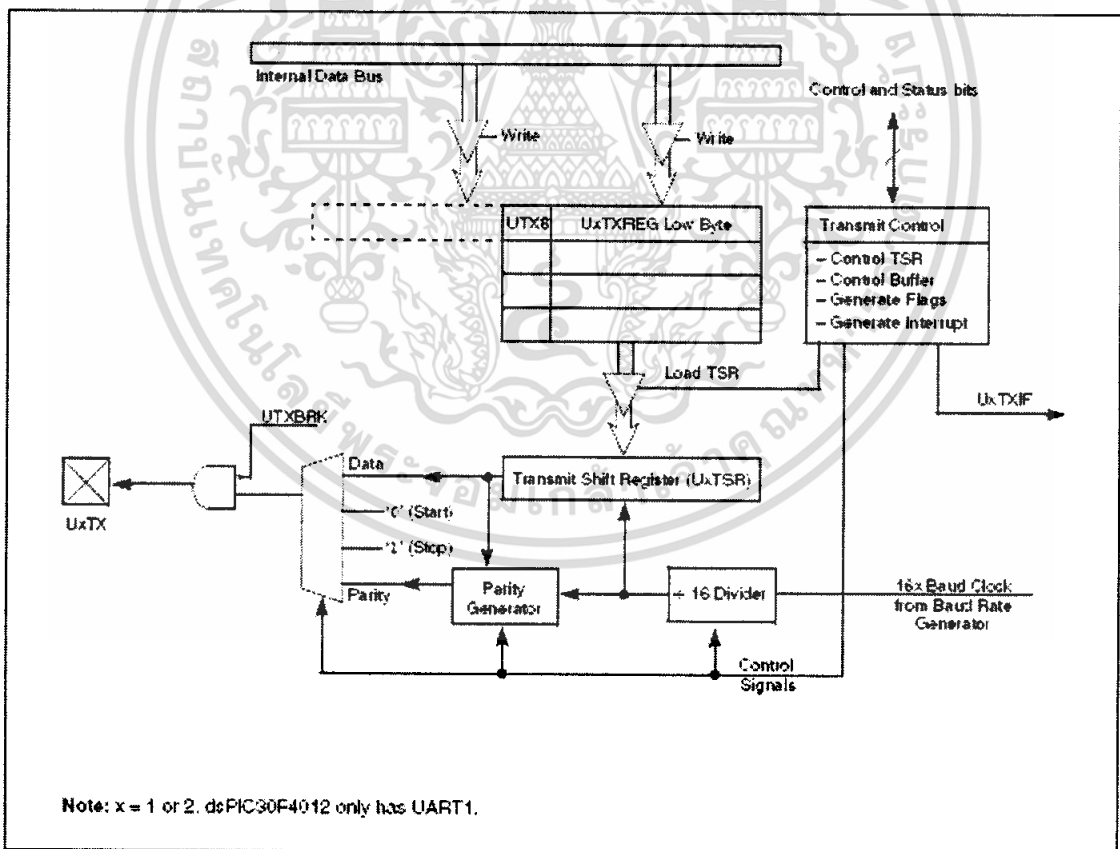
รูปที่ 2.28 ไลอะแกรมส่วนประกอบหลักของโมดูล UART ในไมโครคอนโทรลเลอร์ dsPIC

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาของบริษัทผู้ผลิตและจำหน่ายไมโครคอนโทรลเลอร์ dsPIC การค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7.2 รีจิสเตอร์ที่ใช้ในโมดูล UART

มีทั้งสิ้น 5 ตัวคือ

1. UxMODE ใช้กำหนดโหมดการทำงานของโมดูล UART โดยตัวอักษร x เป็นเลข 1 หรือ 2 เพราะใน dsPIC สามารถบรรจุโมดูล UART ได้ 2 ชุด
2. UxSTA ใช้แสดงสถานะการทำงานของโมดูล UART โดยตัวอักษร x เป็นเลข 1 หรือ 2 เพราะใน dsPIC สามารถบรรจุโมดูล UART ได้ 2 ชุด
3. UxRXREG ใช้เก็บข้อมูลที่รับเข้ามาทางขาเชื่อมต่อพอร์ตอนุกรม โดยตัวอักษร x เป็นเลข 1 หรือ 2 เพราะใน dsPIC สามารถบรรจุโมดูล UART ได้ 2 ชุด
4. UxTXREG ใช้เก็บข้อมูลสำหรับส่งออกทางขาเชื่อมต่อพอร์ตอนุกรม โดยตัวอักษร x เป็นเลข 1 หรือ 2 เพราะใน dsPIC สามารถบรรจุโมดูล UART ได้ 2 ชุด
5. UxBRG ใช้เก็บค่าสำหรับกำหนดอัตราบอด โดยตัวอักษร x เป็นเลข 1 หรือ 2 เพราะใน dsPIC สามารถบรรจุโมดูล UART ได้ 2 ชุด



รูปที่ 2.29 แสดงไดอะแกรมการทำงานของตัวส่งข้อมูลใน UART ของ dsPIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1) การเอนเอเบิลโมดูล UART ใน dsPIC30F4011 ให้ทำงาน

ทำได้โดยการเซตบิต UAERTEN ซึ่งเป็นบิต 15 ในรีจิสเตอร์ UxMODE และเซตบิต UTXEN ซึ่งเป็นบิต 10 ในรีจิสเตอร์ U1STA ทั้งนี้ที่เอนเอเบิล ขา UxTX และ UxRX จะถูกกำหนดให้ทำงานเป็นขาพอร์ตเอาต์พุตและอินพุตตามลำดับ โดยไม่สนใจการกำหนดทิศทางที่เกิดขึ้นก่อนหน้า

(2) การดิสเอเบิลโมดูล UART ใน dsPIC30F4011

ทำได้โดยการเคลียร์บิต UARTEN ซึ่งเป็นบิต 15 ในรีจิสเตอร์ UxMODE โดยปกติโมดูล UART จะถูกดิสเอเบิลหลังจากที่เกิดการรีเซต ซึ่งเป็นการกำหนดสถานะในเบื้องต้น จากนั้นหากต้องการให้ทำงานจะต้องมีการเอนเอเบิลภายหลัง เมื่อโมดูล UART ถูกดิสเอเบิล ขา UxTX และ UxRX จะถูกปลดออกจากโมดูล UART สามารถนำไปใช้งานเป็นพอร์ตอินพุตเอาต์พุตทั่วไปได้ทันที

นอกจากนั้นในกรณีที่โมดูล UART ถูกดิสเอเบิล ค่าในบัพเฟอร์ทั้งหมดจะถูกเคลียร์ เช่นเดียวกับบิตแฟล็กแจ้งสถานะความผิดพลาดทั้งหมดจะถูกเคลียร์ด้วย ส่วนตัวนับอัตราบอดจะอยู่ในภาวะรีเซต หลังจากทีดิสเอเบิลโมดูล UART แล้วกลับมาเอนเอเบิลใหม่ การทำงานทั้งหมดจะเริ่มต้นใหม่ทั้งหมด

2.5.7.4 ความสัมพันธ์ระหว่างสัญญาณนาฬิกาของระบบกับอัตราบอด

ความสัมพันธ์ระหว่างความถี่สัญญาณนาฬิกาของระบบกับอัตราบอดในการสื่อสารข้อมูลอนุกรมนั้นสามารถแสดงในรูปของสมการดังนี้

$$\text{อัตราบอด} = \frac{F_{CY}}{16 \times (BRG + 1)} \quad (***)$$

และ

$$F_{CY} = \frac{F_{osc}}{4} \quad (***)$$

โดยที่

BRG คือ ค่าข้อมูลของจิสเตอร์ UxBRG (x เท่ากับ 1 หรือ 2 ขึ้นอยู่กับใช้โมดูล UART ชุดใด)

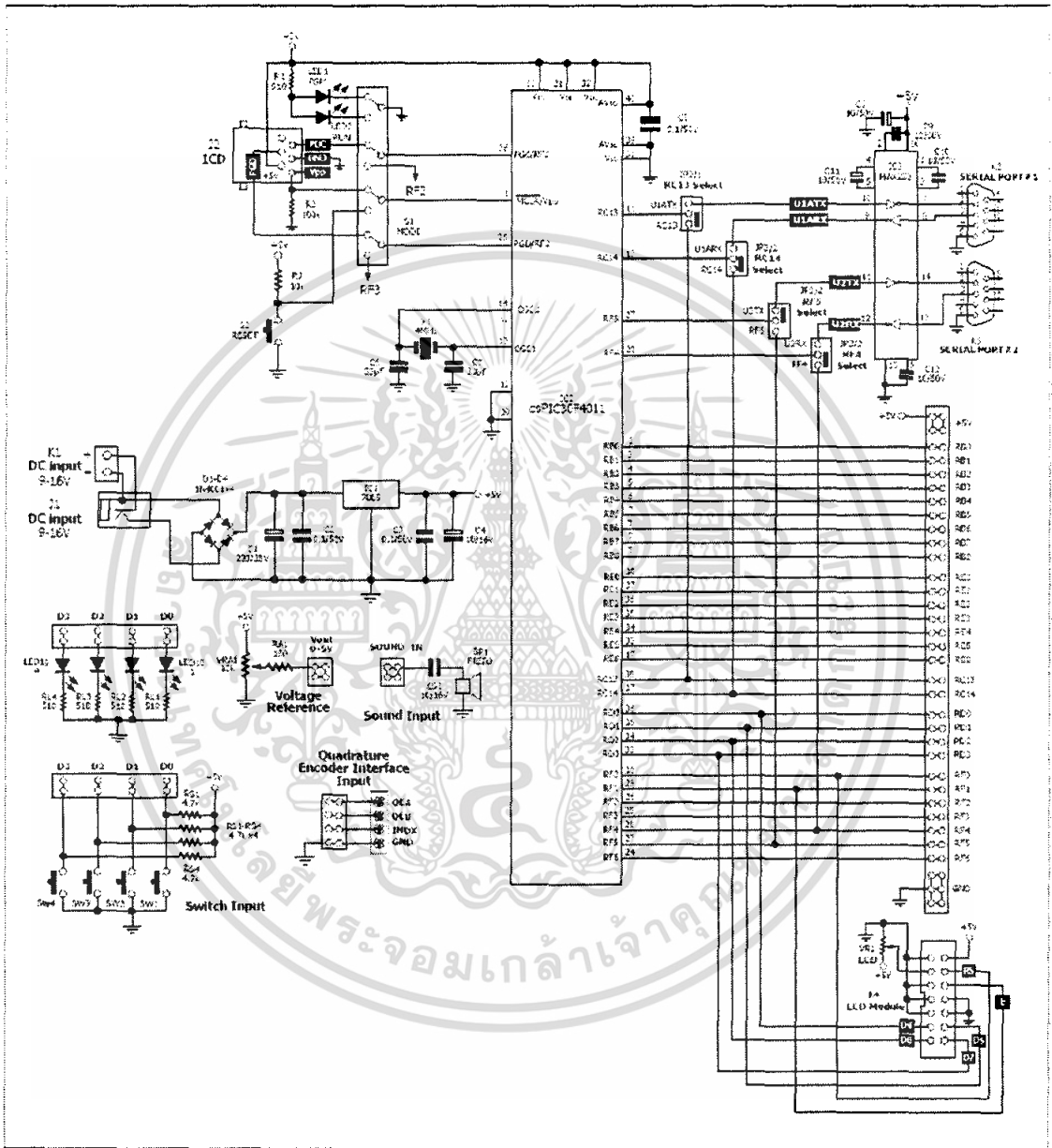
F_{osc} คือ ความถี่สัญญาณนาฬิกาหลัก

2.5.7.5 ขาพอร์ตเสริมของโมดูล UART (Alternate UART I/O Pins)

ในไมโครคอนโทรลเลอร์ dsPIC บางเบอร์ได้บรรจุขาพอร์ตสำหรับรับและส่งข้อมูลกับโมดูล UART เสริมเพิ่มเติมจากที่มีอยู่เดิมเป็นขา UxATX และ UxARX ทั้งนี้เพื่ออำนวยความสะดวกและเพิ่มช่องทางในการใช้งานขาพอร์ตของ dsPIC ที่อาจพบข้อจำกัดด้านการออกแบบแผ่นวงจรพิมพ์หรือด้านการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน dsPIC30F4011 ได้บรรลุความสามารถนี้ไว้ด้วย นั่นคือขาพอร์ต UxATX และ UxARX โดยการเลือกใช้งานต้องทำการเซตบิต ALTIO ซึ่งเป็นบิต 10 ในรีจิสเตอร์ UxMODE เมื่อเลือกใช้ขาพอร์ตเสริมนี้แล้ว ขาพอร์ตหลักคือ UxTX และ UxRX จะกลายเป็นขาพอร์ตอินพุตเอาต์พุตสำหรับใช้ในงานอื่นแทน



รูปที่ 2.31 แสดงวงจรของบอร์ด JX-dsPIC40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรุณาไปใช้

2.6 การเขียนโปรแกรมติดต่อและควบคุมซีเรียลพอร์ต (Serial Port) หรือพอร์ตอนุกรม

2.6.1 พื้นฐานการสื่อสารแบบอนุกรม

ถึงแม้ว่าการสื่อสารแบบอนุกรมในเครื่องคอมพิวเตอร์นั้นจะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน ทั้งนี้ก็เพราะว่าการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตขนานนั้นสามารถส่งข้อมูลได้ครั้งละหลายๆบิตพร้อมๆกัน ส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน

แต่ว่าการส่งข้อมูลแบบอนุกรมนั้นมีข้อที่เหนือกว่าการส่งข้อมูลแบบขนานคือ การสามารถส่งข้อมูลได้ในระยะที่ไกลกว่าแบบขนาน อีกทั้งสายสัญญาณที่ใช้ยังมีน้อยกว่าการส่งข้อมูลแบบขนานอีกด้วย การสื่อสารแบบอนุกรมสามารถแบ่งออกเป็น 3 รูปแบบดังนี้

1. **Simplex** สามารถส่งข้อมูลได้อย่างเดียวเป็นการสื่อสารแบบทางเดียว

2. **Half duplex** สามารถส่งข้อมูลไปยังปลายทางและสามารถรับข้อมูลจากปลายทางได้ แต่มาสามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกัน

3. **Full Duplex** สามารถรับส่งข้อมูลได้ในเวลาเดียวกัน

นอกจากนี้แล้วยังสามารถแบ่งประเภทของการสื่อสารแบบอนุกรมตามลักษณะสัญญาณในการส่งได้ 2 แบบคือ

- การสื่อสารแบบซิงโครนัส (Synchronous) สำหรับการสื่อสารแบบซิงโครนัสนี้จะใช้สัญญาณนาฬิกาควบคุมการรับส่งสัญญาณ เช่น สายเคเบิลคอมพิวเตอร์ โดยจะมีสายสัญญาณเส้นหนึ่งเป็นสายสัญญาณนาฬิกา ส่วนอีกเส้นหนึ่งเป็นสายข้อมูล(และมักมีสายกราวด์ด้วย)

สำหรับการสื่อสารแบบซิงโครนัสนี้เหมาะสำหรับการทำงานในระยะใกล้ ข้อมูลที่จะส่งมีไม่มากนัก เพราะถ้าระยะทางไกลขึ้นจะทำให้สัญญาณมีปัญหา อีกทั้งต้องมีสายหลายเส้นทำให้สิ้นเปลืองมาก

- การสื่อสารแบบอะซิงโครนัส(Asynchronous) สำหรับการสื่อสารแบบอะซิงโครนัสนั้นจะใช้ข้อมูลเพียงตัวเดียว แต่จะใช้รูปแบบการส่งข้อมูล หรือ bit pattern เป็นตัวกำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล และส่วนไหนเป็นส่วนปิดท้ายข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาคส่ง และภาครับ ซึ่งจะมีอุปกรณ์พิเศษที่ชื่อว่า URAT หรือ Universal Asynchronous Receiver/ Transmitter ควบคุมการรับ และส่งข้อมูล

มาตรฐานของการส่งข้อมูลแบบอนุกรมอีกแบบที่ได้รับความนิยมอย่างสูงตั้งแต่อดีตจนถึงปัจจุบัน โดยใช้งานกันอย่างแพร่หลายทั้งการสื่อสาร และการควบคุมอุตสาหกรรมนั้นก็คือมาตรฐาน RS-232C

2.6.2 มาตรฐาน RS-232C

มาตรฐาน RS-232C เป็นมาตรฐานที่ได้รับการออกแบบมาเพื่อที่จะทำให้อุปกรณ์ต่อพ่วงจากผู้ผลิตต่างกันสามารถใช้งานร่วมกันได้ มาตรฐานหลายชนิดได้รับการออกแบบขึ้นมา แต่มาตรฐานที่ได้รับความนิยมและใช้กันขวางมากที่สุด คือ มาตรฐาน RS-232c ซึ่งถูกประกาศใช้ในปี 1969 โดยสมาคมอุตสาหกรรมเทคนิค (Electronic Industries Association :EIA) ในยุคแรกๆการอินเตอร์เฟซแบบ RS-232C ถูกออกแบบสำหรับเชื่อมต่อเทอร์มินอล

(DTE :Data Terminal Equipment) ทั้งนี้ก็เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลบนสายเส้นเดียวกัน

มาตรฐาน RS-232C ได้แบ่งอุปกรณ์ออกเป็น 2 ประเภท ซึ่งอุปกรณ์ทั้งสองประเภทนี้คือ

2.6.2.1 อุปกรณ์ DTE (Data Terminal Equipment) เป็นอุปกรณ์สำหรับส่งข้อมูล(output)

2.6.2.2 อุปกรณ์ DCE (Data Communication Equipment) เป็นอุปกรณ์สำหรับรับข้อมูล

(input)

ตามมาตรฐาน RS-232c แล้วคอนเนคเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเนคเตอร์ของ DCEจะเป็นตัวเมีย ซึ่งคอนเนคเตอร์ที่นิยมใช้กันอยู่จะเป็นชนิด D-type แบบ 9 ขา และแบบ 25 ขา โดยจะติดตั้งอยู่หลังคอมพิวเตอร์ ระดับแรงดันจะมีค่าระหว่าง -3V ถึง -15V

สำหรับลอจิก high และลอจิก low จะมีแรงดันระหว่าง +3v ถึง 15v สามารถรับส่งข้อมูลได้ด้วยความยาวของสายสัญญาณสูงสุด 50 ฟุต หรือ 150 เมตร แต่ถ้าเราต้องการสื่อสารกับอุปกรณ์อื่นที่อยู่ห่างกันมากๆเราจำเป็นต้องใช้อุปกรณ์อื่นๆ เข้าช่วย เช่นการใช้ โมเด็ม เป็นต้น

2.6.3 รายละเอียดของสายสัญญาณ

สำหรับรายละเอียดของสายสัญญาณนั้นประกอบไปด้วย

- **Transmit Data : TD** ใช้สำหรับส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์
- **Receiver Data : RD** ใช้สำหรับรับข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์
- **Request Data : RTS** ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทาง เพื่อร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลมา
- **Clear To send : CTS** ใช้สำหรับตรวจสอบอุปกรณ์ที่เชื่อมต่อด้วยพร้อมที่จะรับข้อมูลหรือไม่ โดยจะคอยรับสัญญาณ RTS เมื่อทุกอย่างพร้อมจะทำการส่งข้อมูลออกทางขา TD
- **Data set Ready : DSR** ใช้สำหรับตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง
- **Signal Ground** : เป็นกราวด์ของระบบ
- **Carrier Detect : CD** ขานี้จะ Active เมื่อมีการส่งสัญญาณ Carrier จาก โมเด็ม

- **Data Terminal Ready : DTR** ใช้สำหรับบอกให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อโดยขา DRS ของอุปกรณ์ปลายทาง
- **Ring Indicator : RI** ขานี้จะ Active เมื่อโมเด็มได้รับสัญญาณเรียกเข้าจากสายโทรศัพท์

2.6.4 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม

การสื่อสารแบบอนุกรมที่นิยมใช้กับคอมพิวเตอร์นั้น เป็นการสื่อสารข้อมูลแบบอะซิงโครนัส นั่นคือ ต้องใช้สัญญาณเคียวทำหน้าที่ทั้งส่วนที่เป็นข้อมูล และส่วนที่ใช้ควบคุมข้อมูลการส่งข้อมูล ดังนั้นข้อมูลที่อ่านได้แต่ละบิตจากการส่งแบบอนุกรม จึงต้องถูกแยกแยะว่าสำหรับใช้วัตถุประสงค์ใด โดยเราสามารถแบ่งเป็น 4 ส่วนคือ

1. **start bit** ขนาด 1 บิต
2. **บิตข้อมูล** ขนาด 7 หรือ 8 บิต
3. **parity** ขนาด 1 บิต
4. **Stop bit** ขนาด 1 หรือ 2 บิต

แต่ละตัวอักษรที่ส่งออกไปเป็นกลุ่มจะประกอบด้วยบิตเริ่มต้น บิตข้อมูล บิตพาริตี (จะมีหรือไม่มีก็ได้) และบิตจบ โดยสรุปหน้าที่ของแต่ละส่วนได้ต่อไปนี้

- **Start Bit หรือบิตเริ่มต้น** จะใส่ที่จุดเริ่มต้นเสมอ เพื่อเตือนอุปกรณ์ฝ่ายรับข้อมูลกำลังจะมาถึง
- **Data Character หรือบิตข้อมูล** การส่งบิตข้อมูลจะส่งออกเป็นกลุ่มๆ โดยทั่วไปจะส่งเป็น 7 หรือ 8 บิต ซึ่งเพียงพอสำหรับการส่งแอสกีเวิร์ด (Ascii Word)
- **Parity Bit หรือ บิตพาริตี** ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ส่ง เราจะใส่บิตพาริตีเข้าไป แต่ทั้งตัวรับและตัวส่งจะต้องรู้กันว่าใช้พาริตีแบบไหนในการส่งข้อมูลซึ่งหลักการในการกำหนด bit parity มีดังนี้
 - **พาริตีคู่ (even parity)** ค่าของพาริตีนี้เมื่อรวมกับทุกๆบิตของข้อมูลแล้วจะต้องมีจำนวนบิตที่เป็น 1 เป็นเลขคู่ ตัวอย่างเช่น 1000101 มีเลข 1 ทั้งหมด 3 ตัว ดังนั้นบิตพาริตีจะเป็น 0
 - **พาริตีคี่ (odd parity)** ค่าของบิตพาริตีนี้เมื่อรวมกับทุกๆบิตของข้อมูลแล้ว จะต้องมิจำนวนบิตที่เป็นเลข 1 เป็นเลขคี่ เช่น ข้อมูล 1000101 มีเลข 1 ทั้งหมด 3 ตัว ดังนั้นพาริตีบิตจะเป็น 1
 - **ไม่มีพาริตี (none)** ถ้าตั้งพาริตีเป็น none ทั้งภาครับและภาคส่งจะไม่มีตรวจสอบบิตพาริตี
- **Stop Bit หรือ บิตจบ** เป็นบิตที่ส่งมาปิดท้ายข้อมูล

2.6.5 อัตราเร็วในการส่งข้อมูลแบบอนุกรม

การที่อุปกรณ์ 2 อย่างจะติดต่อสื่อสารกันได้นั้น จะต้องทำงานด้วยอัตราเร็วที่เท่ากัน ซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสคือ ค่าบอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที ซึ่งอัตราเร็วในการสื่อสารแบบอนุกรมสำหรับมาตรฐาน RS-232C นั้นมีใช้ดังนี้ 110, 150, 300, 600, 1200, 2400, 4800, 9600, และ 19200 บิตต่อวินาที

2.6.6 พรีอพอร์เตอร์์ที่สำคัญในการใช้งาน MSComm

- **CommPort** ใช้ในการกำหนดเลขของพอร์ตอนุกรมที่เราจะต้องการจะติดต่อ โดยมีรูปแบบของการใช้งานดังนี้

```
object.Commport[ = value ]
```

ยกตัวอย่างเช่น เรากำหนดให้เขียน โปรแกรมติดต่อกับพอร์ตคอม 1 จะเขียนเป็น

```
MSComm1.CommPort = 1
```

- **Settings** ใช้ในการกำหนดอัตราบอด (Baud Rate) หรือความเร็วในการส่งข้อมูล มีหน่วยเป็นบิตต่อวินาที, พาริตี, จำนวนบิตของข้อมูล, จำนวนบิตปิดท้าย โดยมีรูปแบบการใช้งานดังนี้

```
object.Settings [ = value ]
```

ยกตัวอย่างเช่น เรากำหนดให้มีการเขียน โปรแกรมใช้งานที่อัตราบอด = 9600 บิตต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูลเท่ากับ 8 บิต และมีบิตปิดท้าย 1 บิต

```
MSComm1.Settings = " 9600, N, 8, 1"
```

- **PortOpen** ใช้สำหรับเปิดและปิดการใช้พอร์ตอนุกรม โดยมีรูปแบบของการใช้งานดังนี้

```
object.PortOpen [ = value ]
```

ยกตัวอย่างเช่น เราจะใช้เปิดใช้งานพอร์ตอนุกรม ให้กำหนดค่า value เป็น True เขียนโค้ดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSComm1.PortOpen = True
```

แต่ถ้าต้องการปิดพอร์ตคอนูกรม ให้กำหนดค่า value เป็น False เช่น

```
MSComm1.PortOpen = False
```

- **InBufferSize** เป็นการกำหนดขนาดของ Buffer ในการรับข้อมูลเข้ามา โดยมีรูปแบบการทำงานดังนี้

```
object.InBufferSize [ = value ]
```

- **OutBufferSize** เป็นการกำหนดขนาดของ Buffer ในการส่งข้อมูลออกไป โดยมีรูปแบบการทำงานดังนี้

```
object.OutBuffer [ = value ]
```

- **InputLen** เป็นการกำหนดค่าของข้อมูลที่อ่านจาก Buffer ในการส่งข้อมูลออกไปโดยมีรูปแบบการกำหนดค่าดังนี้

```
object.InputLen [ = value ]
```

- **InputMode** เป็นการกำหนดค่าชนิดของข้อมูลที่ได้รับเข้ามา โดยมีรูปแบบการกำหนดค่าดังนี้

```
object.InputMode [ = value ]
```

โดยที่เราสามารถเลือกชนิดของข้อมูลได้ 2 ประเภทคือ

- **comInputModeText** ข้อมูลที่รับเข้ามาเป็นข้อความปกติ เราสามารถตั้งค่าให้อยู่ในโหมดนี้ได้โดยการกำหนด Value ให้เป็น "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **comInputModeBinary** ข้อมูลที่รับเข้ามาเป็นข้อมูลไบนารี เราสามารถตั้งค่าให้อยู่ในโหมดนี้ได้โดยการกำหนด Value ให้เป็น “1”
- **Input** ใช้ในการอ่านค่าข้อมูลจากพอร์ตอนุกรม โดยมีรูปแบบการอ่านดังต่อไปนี้

object.Input

ตัวอย่างเช่น เราอ่านค่าจากบัฟเฟอร์ของพอร์ตอนุกรม แล้วนำมาเก็บไว้ในตัวแปรที่ชื่อว่า Data เราจะเขียนโค้ดดังนี้

Data = Mscomm1.Input

- **Output** ใช้ในการส่งข้อมูลออกไปจากพอร์ตอนุกรม โดยมีรูปแบบของการเขียนดังนี้

object.Output [= value]

- **EOFEnable** เป็นการบอกสิ้นสุดของไฟล์ End of file [EOF] โดยมีรูปแบบการใช้งานดังนี้

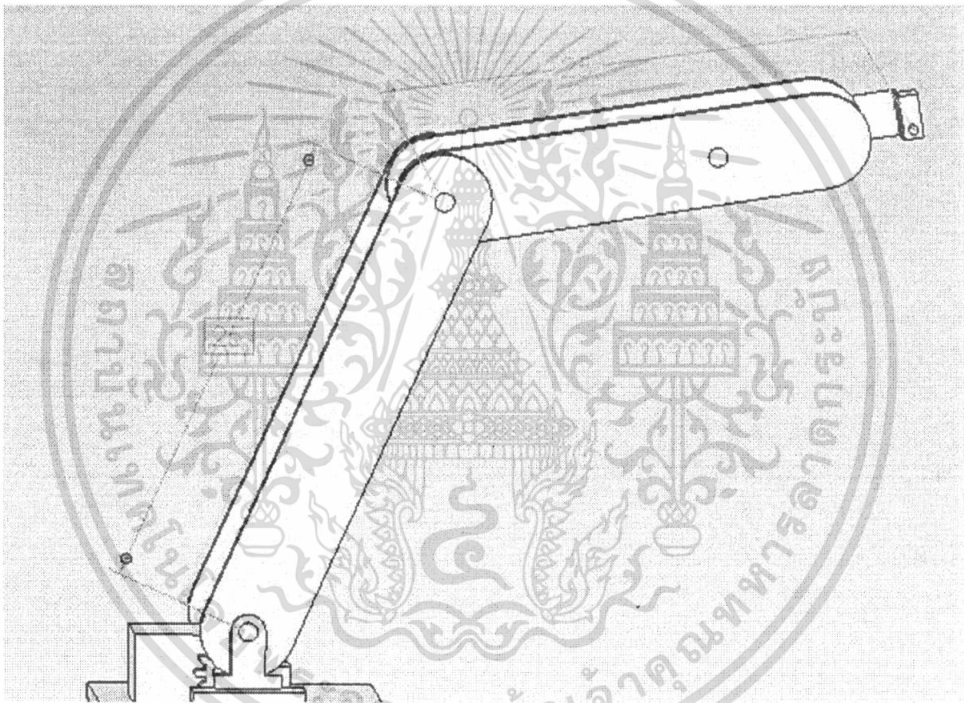
Object.EOFEnable [= valued]

บทที่ 3

การออกแบบและการสร้าง

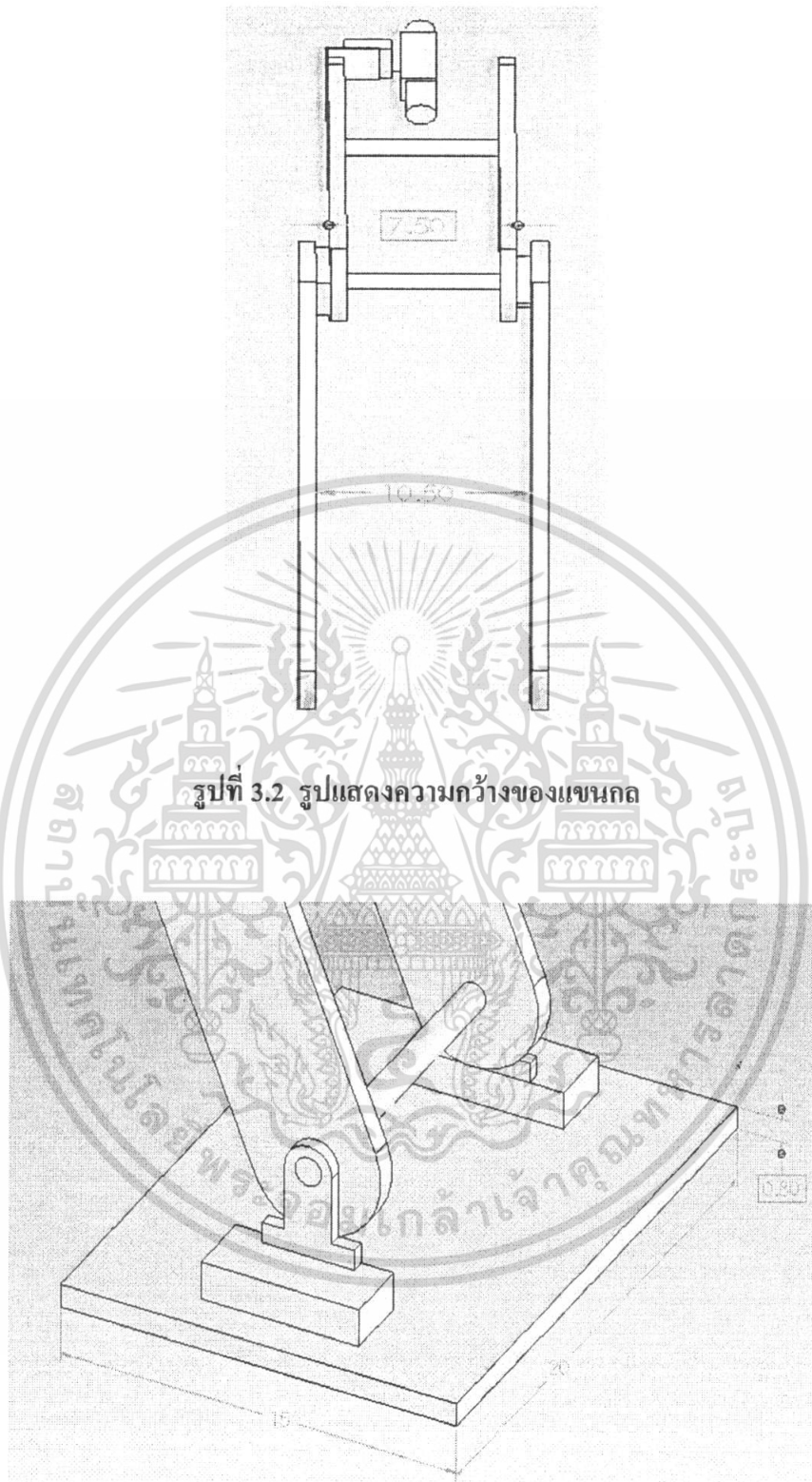
3.1 การออกแบบแขนกล

3.1.1 การกำหนดขนาดของแขนกล จะต้องสามารถให้แขนกลวาดภาพครอบคลุมพื้นที่กระดาษ A4 ได้ทั้งหมด สามารถตรวจสอบได้ว่าแขนที่กำหนดนั้นสามารถวาดภาพได้ทั้งกระดาษ A4 หรือไม่ โดยการทดสอบวาดภาพในหน้าต่างวาดภาพในจุดที่มีระยะไกลที่สุดโดยใช้ค่าของแขนที่กำหนดขึ้น แล้วดูค่ามุมต่างที่แขนใช้เคลื่อนที่ว่ามีค่าเกินมุมที่ เซอร์โวมอเตอร์ทำงานหรือไม่



รูปที่ 3.1 รูปแสดงขนาดของแขนกล

จะเห็นว่าได้ออกแบบให้แขนท่อนมีขนาดสั้นกว่าแขนท่อนล่าง เนื่องจากต้องเหลือความยาวไว้สำหรับปากกาที่ใช้สำหรับวาดภาพเพราะต้องการให้ระยะห่างของจุดหมุนของแขนทั้ง 2 ท่อน มีขนาดเท่ากัน



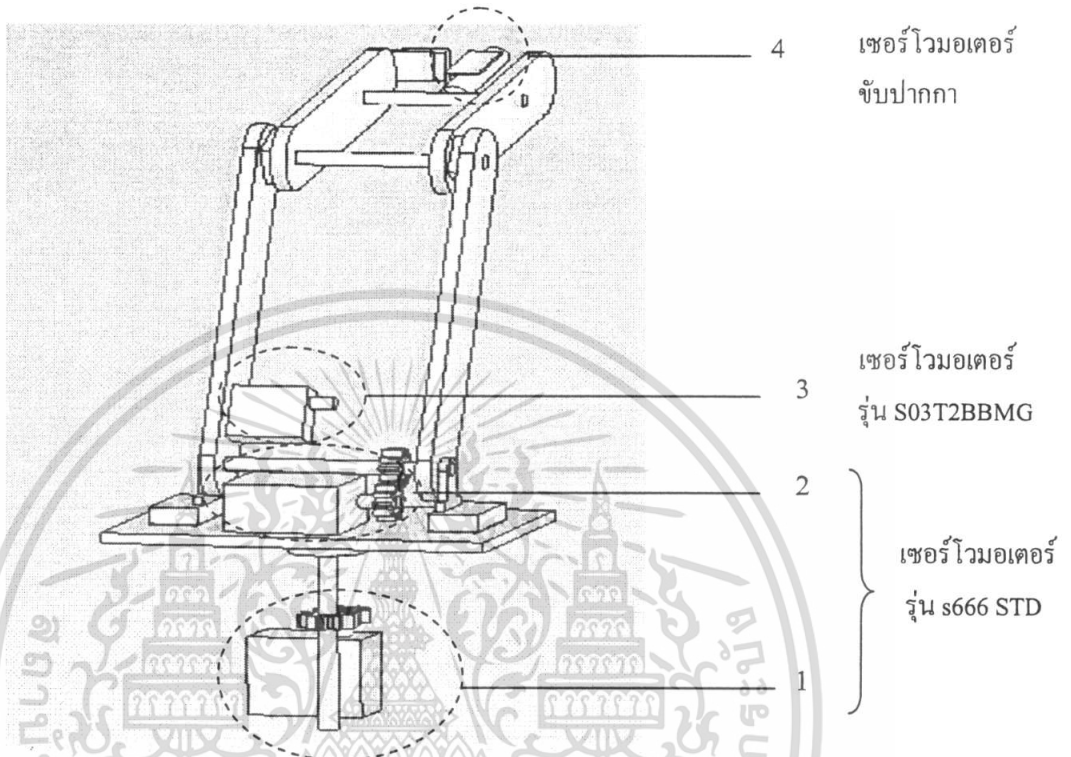
รูปที่ 3.2 รูปแสดงความกว้างของแขนกล

รูปที่ 3.3 รูปแสดงความกว้างของฐาน

สำหรับการกำหนดขนาดความกว้างของแขนและขนาดของฐานจะต้องคำนึงถึงการติดตั้งเซอร์โวมอเตอร์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การติดตั้งเซอร์โวมอเตอร์

สำหรับแขนกลวาดภาพนี้จะใช้เซอร์โวมอเตอร์ในการเคลื่อนแขนกล ทั้งหมด 4 ตัว ซึ่งมีขนาดแตกต่างกันไป ขึ้นกับการรับน้ำหนักของเซอร์โวมอเตอร์ในแต่ละจุด ดังนี้



รูปที่ 3.4 รูปแสดงตำแหน่งที่ติดตั้งเซอร์โวมอเตอร์

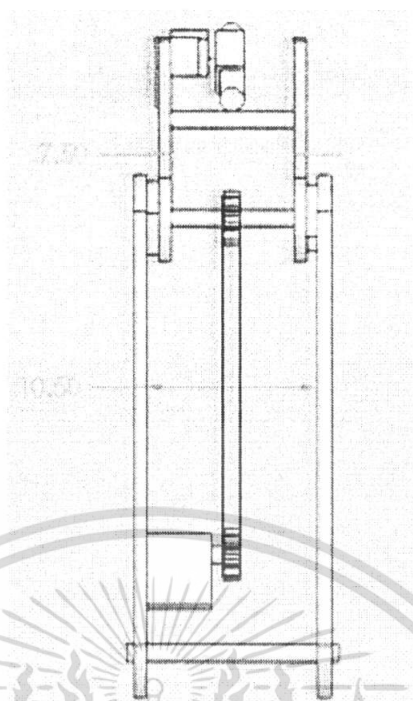
- เซอร์โวมอเตอร์ รุ่น S666 STD torque 15 kg/cm จะเป็นเซอร์โวมอเตอร์ที่ให้ทอร์ก(torque)มากที่สุด เนื่องจากทั้ง 2 จุดที่ติดตั้งเซอร์โวมอเตอร์นี้จะเป็นจุดที่จะต้องรับน้ำหนักมาก โดย

ตัวที่ 1 จะต้องใช้หมุนแกนเหล็ก ซึ่งแกนเหล็กนี้จะรับน้ำหนักแขนกลทั้ง 2 ท่อน เซอร์โวมอเตอร์อีก 3 ตัว รวมถึงฐานที่ไว้ตั้งแขนด้วย ถึงแม้ว่าเซอร์โวมอเตอร์จะไม่ได้รับน้ำหนักโดยตรง แต่การหมุนแกนเหล็กที่รับน้ำหนักมาก ก็จำเป็นต้องใช้เซอร์โวมอเตอร์ที่มีทอร์กมาก

ตัวที่ 2 จะเป็นตัวที่ใช้เคลื่อนที่แขนท่อนล่าง ซึ่งก็ต้องรับน้ำหนักของแขนทั้ง 2 ท่อน และเซอร์โวมอเตอร์อีก 2 ตัว

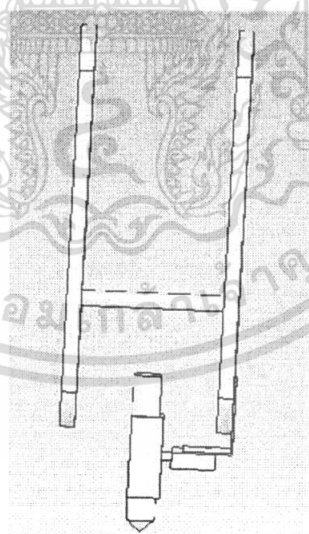
- เซอร์โวมอเตอร์ รุ่น S03T2BBMG torque 8 kg/cm จะใช้สำหรับเคลื่อนที่แขนท่อนบน ซึ่งจะมีขนาดของทอร์กน้อยลงเนื่องจากจะรับน้ำหนักเฉพาะแขนท่อนบนในการเคลื่อนที่แขนท่อนบนจะมีการใช้สายพานช่วย เนื่องจากการติดเซอร์โวมอเตอร์ที่จุดหมุนของแขนท่อนบนนั้นจะทำให้ค่าทอร์กของแขนกลนั้นมีค่ามากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 รูปแสดงการใช้สายพานช่วยในการเคลื่อนที่แกนท่อนบน

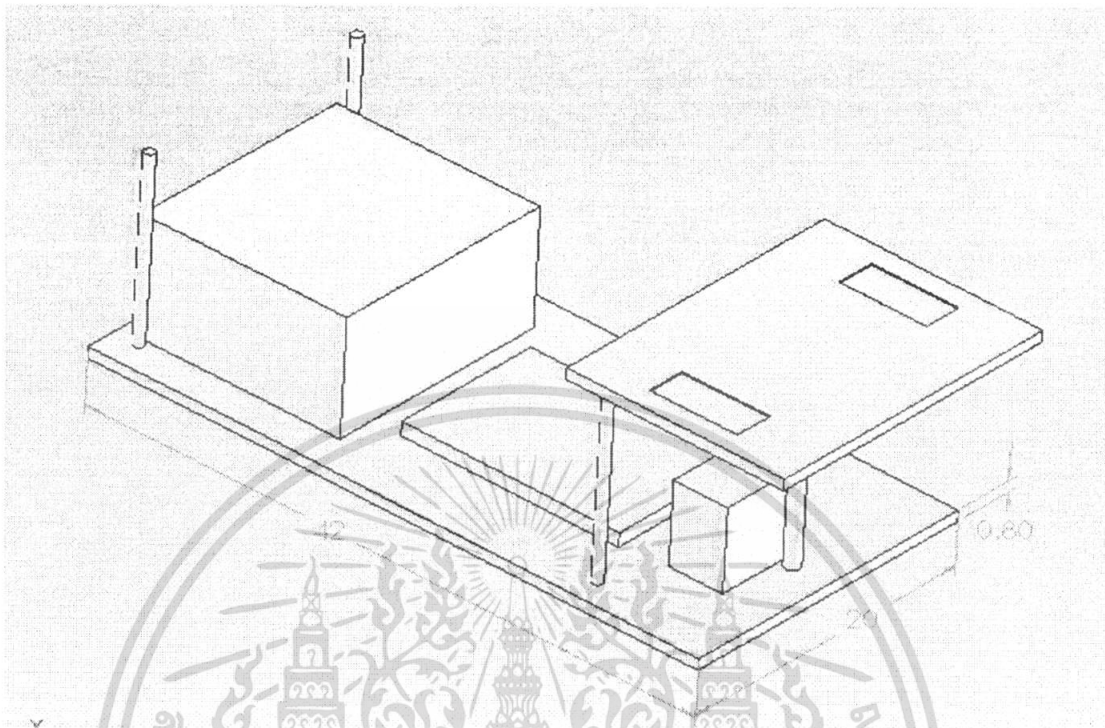
- เซอร์โวมอเตอร์ตัวเล็ก



รูปที่ 3.6 รูปแสดงการติดตั้งเซอร์โวมอเตอร์ที่ใช้เคลื่อนที่ปากกา

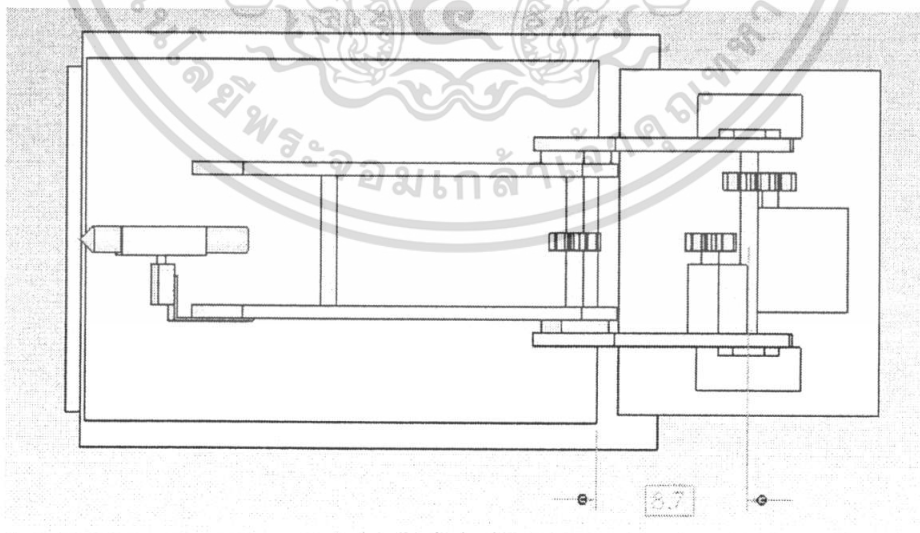
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ออกแบบที่วาง supply และ Board



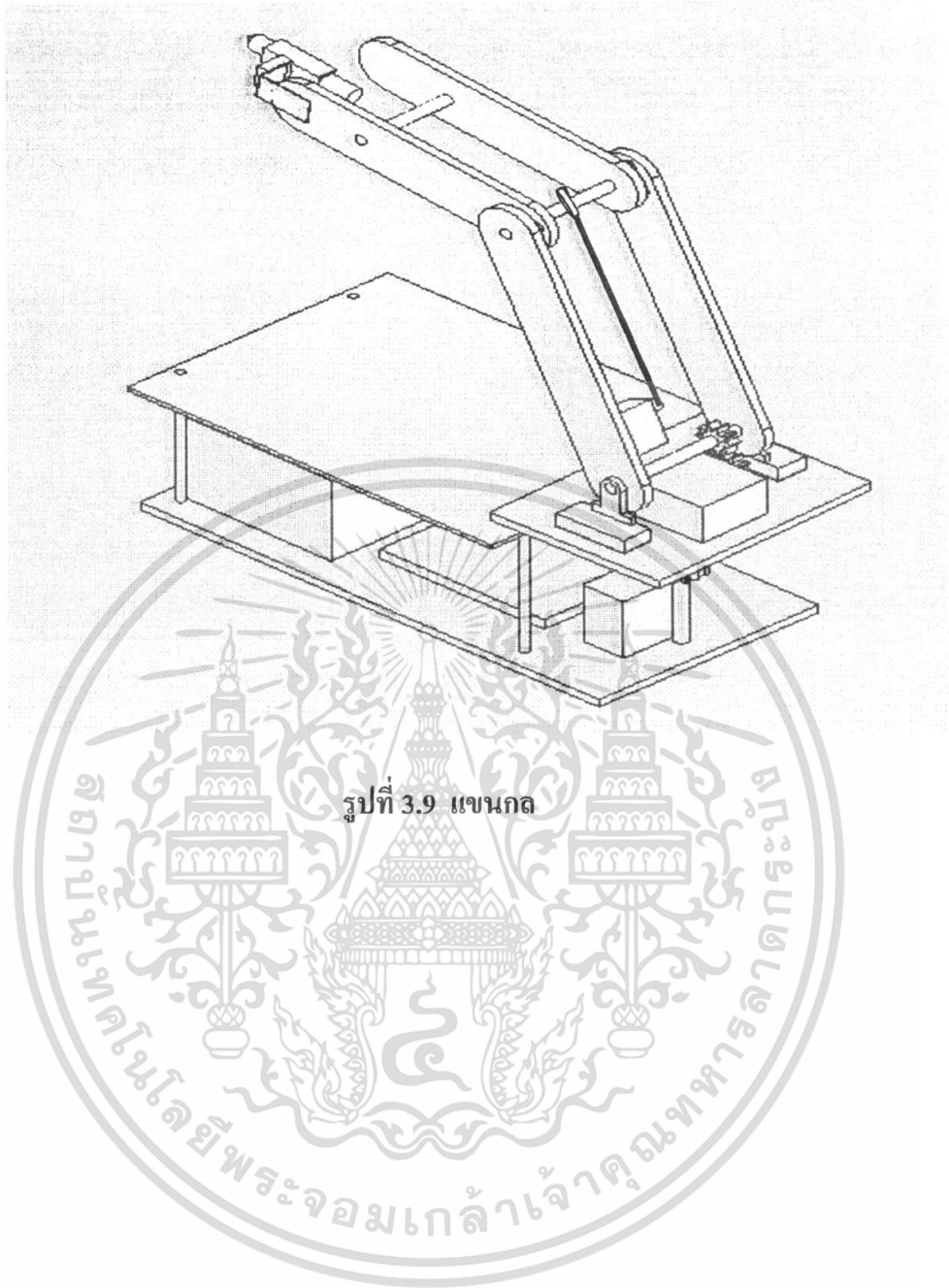
รูปที่ 3.7 รูปแสดงการวางซัพพลาย(supply) และบอร์ด(board)

3.1.4 ออกแบบที่วางกระดาษ



รูปที่ 3.8 รูปแสดงตำแหน่งวางกระดาษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

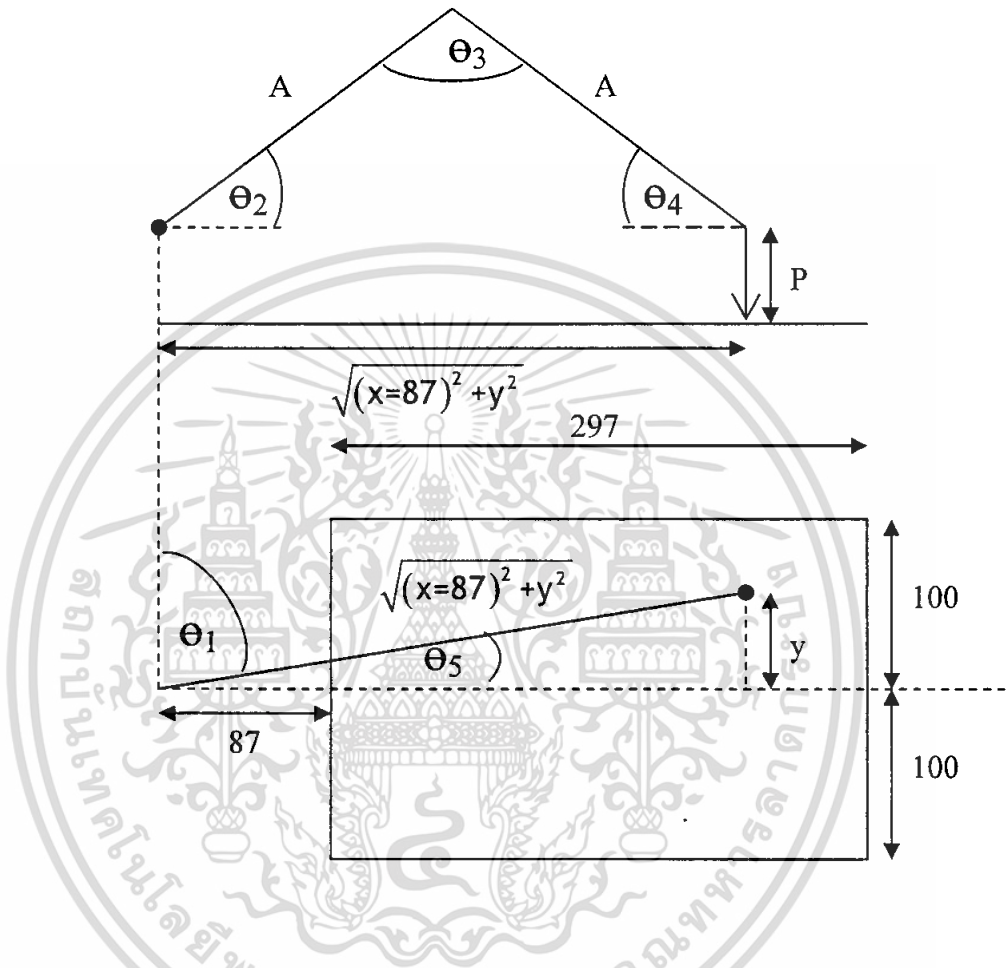


รูปที่ 3.9 แขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบวิชวลเบสิก (Visual Basic 6)

3.2.1 การคำนวณ



$$A^2 = A^2 + (x + 87)^2 + y^2 - 2A\sqrt{(x + 87)^2 + y^2} \cos \theta_2$$

$$\theta_2 = \cos^{-1} \frac{(x + 87)^2 + y^2}{2A\sqrt{(x + 87)^2 + y^2}}$$

$$\theta_2 = \theta_4$$

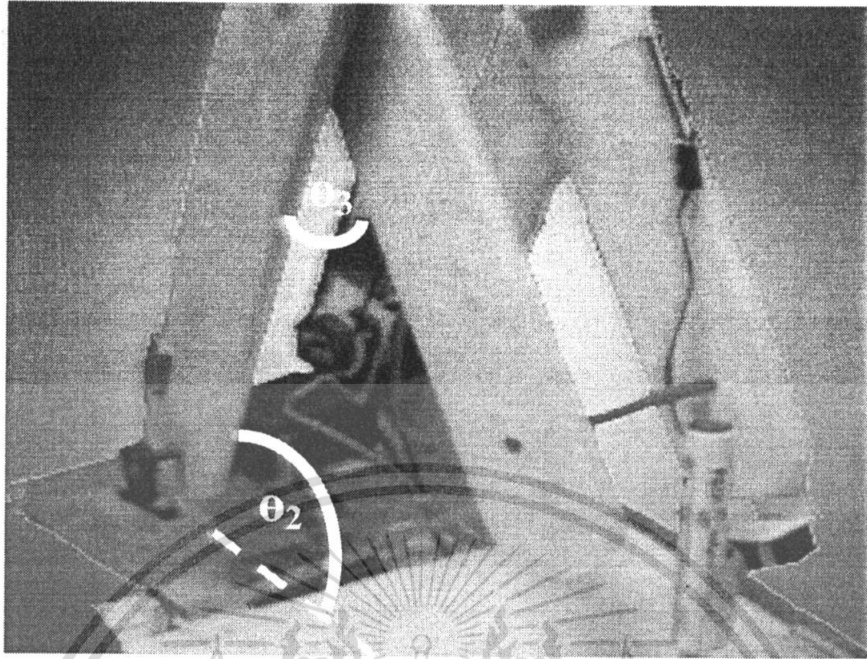
$$(x + 87)^2 + y^2 = A^2 + A^2 - 2A^2 \cos \theta_3$$

$$\theta_3 = \cos^{-1} \frac{[2A^2 - (x + 87)^2 - y^2]}{2A^2}$$

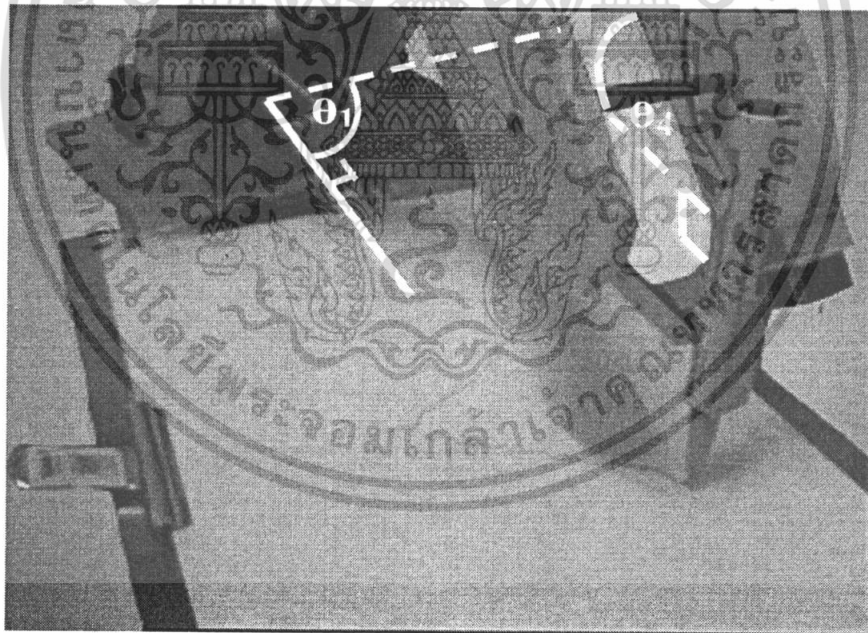
$$\theta_1 = \tan^{-1} \frac{y}{x}$$

$$\theta_5 = 90 - \theta_1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 รูปแสดงมุม θ_2 , θ_3 ที่นำไปใช้กับแขนกล

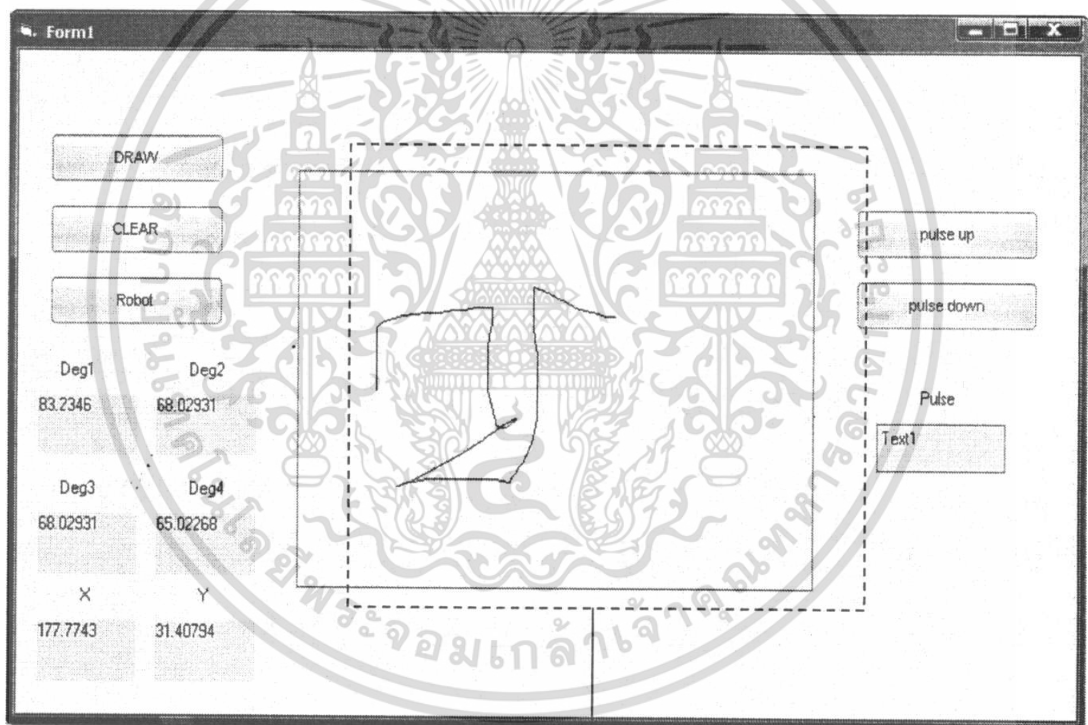


รูปที่ 3.11 รูปแสดงมุม θ_1 , θ_4 ที่นำไปใช้กับแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- A คือ ความยาวของแขน ซึ่งแขนทั้ง 2 ท่อนมีขนาดเท่ากัน
 X,Y คือ จุดพิกัดในหน้าต่างวาดรูป
 P คือ ความยาวของปากกา
 θ_1 คือ มุม $90 - \theta_5$
 θ_2 คือ มุมระหว่างแขนท่อนล่างกับพื้นระนาบ
 θ_3 คือ มุมระหว่างแขนท่อนบนกับแขนท่อนล่าง
 θ_4 คือ มุมระหว่างแขนท่อนบนกับปากกา - 90
 θ_5 คือ มุมระหว่างแนวแขนกับเส้นกึ่งกลางกระดาษ A4

3.2.2 ส่วนสำหรับวาดภาพผ่านทางหน้าจอคอมพิวเตอร์



หน้าต่างสำหรับวาดภาพ

รูปที่ 3.12 รูปแสดงหน้าต่างวาดรูปในโปรแกรมวิชวลเบสิก 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายส่วนประกอบต่างๆ

- ปุ่ม **Draw** ใช้เมื่อต้องการวาดรูปใหม่หลังจากทำการลบรูปเก่าแล้ว
- ปุ่ม **Clear** ใช้เมื่อต้องการลบรูปภาพทั้งหมด
- กรอบแสดงค่ามุมที่คำนวณได้จากพิกัดของภาพ เพื่อส่งออกผ่านทาง ซีเรียลพอร์ทไปให้ dsPIC ทำการประมวลผลและสร้างพัลส์ให้เซอร์โวมอเตอร์สำหรับเคลื่อนที่ โดยมีทั้งหมด 4 มุม ได้แก่

$$\text{Deg1} = \theta_1$$

$$\text{Deg2} = \theta_2$$

$$\text{Deg3} = \theta_3$$

$$\text{Deg4} = \theta_4$$

- กรอบแสดงค่าพิกัด X , Y ของภาพเพิ่มเติม
 - ปุ่ม **Pulse up** , **Pulse Down** และกรอบแสดงค่า **Pulse** จะใช้สำหรับการทดลองหาความสัมพันธ์ของพัลส์กับมุมของเซอร์โวมอเตอร์
- 3.2.3 ใช้วีลเวลบสิก 6 สำหรับส่งข้อมูลที่คำนวณได้ผ่านซีเรียลพอร์ท

บทที่ 4

การทดลอง

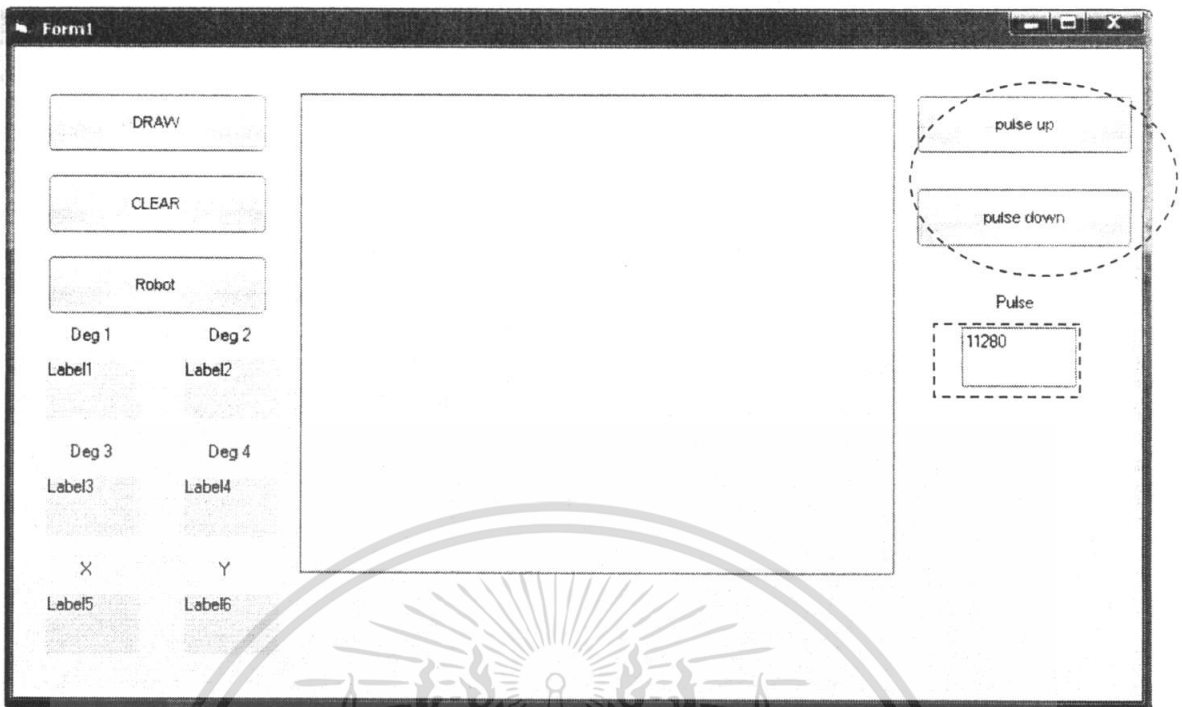
4.1 ทดลองหาความสัมพันธ์ระหว่างพัลส์กับมุมที่เซอร์โวมอเตอร์เคลื่อนที่

สำหรับในการทดลองนี้จะทดลองขณะที่ประกอบเซอร์โวมอเตอร์เป็นแกนกลเรียบร้อย เนื่องจากการทดลองเซอร์โวมอเตอร์ขณะมีโหลด(load)และไม่มีโหลดจะให้ค่าพัลส์ที่แตกต่างกัน

ดังนั้นการทดลองเซอร์โวมอเตอร์ที่ประกอบเป็นแกนกลเรียบร้อยจะสามารถนำค่าความสัมพันธ์ระหว่างพัลส์กับมุมที่เซอร์โวมอเตอร์เคลื่อนที่ไปใช้ขณะที่แกนกลเคลื่อนที่ได้ดีกว่า เสมือนเป็นการชดเชยค่ามุมที่เกิดจากค่าความผิดพลาดของเซอร์โวมอเตอร์ขณะมีโหลด

ขั้นตอนการทดลอง

1. ประกอบแกนกล
2. ทำเครื่องกลมโดยมีเส้นแสดงค่าทุกๆ องศา ตั้งแต่ 0 – 180 องศา
3. เขียนโปรแกรมสร้างพัลส์ในภาษา C
4. เขียนโปรแกรมวิซวลเบสิก 6 สำหรับทำการเปลี่ยนค่าดีวตี้ไซเคิล(duty cycle) ของ พัลส์
5. เซ็ท (set) เครื่องกลมสำหรับวัดค่ามุมของเซอร์โวมอเตอร์แต่ละจุด โดยจะติดไม้ปลายแหลมสำหรับชี้ค่าที่ 0 องศา ของเซอร์โวมอเตอร์จากนั้นทำการรีเซ็ท (reset) dspic แล้วเซอร์โวมอเตอร์จะหมุนให้ไม้ปลายแหลมชี้อยู่ที่ 90 องศา ซึ่งจะติดเครื่องกลม โดยให้ 90 องศา ของเครื่องกลมอยู่ตำแหน่งที่ไม้ปลายแหลมชี้อยู่
6. จากนั้นส่งพัลส์จาก โปรแกรมวิซวลเบสิก 6 จากปุ่ม pulse up และ pulse down และอ่านค่าพัลส์ที่นำมาให้มุมเปลี่ยนไปทุกๆองศาจากกรอบแสดงค่าพัลส์



รูปที่ 4.1 รูปแสดงปุ่ม pulse up , pulse down และ กรอบแสดงค่าพัลส์

จากนั้นอ่านค่าพัลส์ที่ทำให้มุมเปลี่ยนไปทุกๆ องศา เก็บค่าพัลส์กับค่ามุมที่อ่านได้

ตารางที่ 4.1 ผลการทดลองความสัมพันธ์ระหว่างพัลส์(pulse)กับมุมของเซอร์โวมอเตอร์
(servo motor) รุ่น S666 STD ตัวที่2 (มอเตอร์ตัวฐาน)

มุม ครั้งที่	ทวนเข็ม				ตามเข็ม			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
40	16850	16860	16830	16847	16850	16860	16830	16847
41	16440	16450	16420	16437	16580	16570	16590	16580
42	16340	16350	16330	16340	15940	15930	15950	15940
43	16270	16250	16250	16257	15820	15830	15840	15830
44	16170	16160	16150	16160	15720	15730	15740	15730
45	16110	16100	16100	16103	15680	15670	15690	15680
46	16070	16060	16060	16063	15600	15610	15610	15607
47	16000	16010	16000	16003	15510	15500	15520	15510
48	15900	15910	15920	15910	15460	15470	15480	15470
49	15780	15770	15790	15780	15400	15420	15410	15410
50	15700	15720	15710	15710	15180	15170	15190	15180
51	15470	15480	15480	15477	15110	15100	15120	15110
52	15440	15430	15450	15440	15060	15070	15080	15070
53	15370	15380	15360	15370	14980	14960	14990	14977
54	15310	15300	15300	15303	14930	14950	14940	14940
55	15220	15210	15220	15217	14860	14870	14850	14860
56	15130	15140	15120	15130	14760	14740	14770	14757
57	15050	15060	15040	15050	14680	14670	14690	14680
58	14960	14970	14950	14960	14590	14580	14590	14587
59	14860	14870	14850	14860	14500	14520	14510	14510
60	14660	14650	14650	14653	14450	14430	14460	14447
61	14590	14580	14590	14587	14390	14370	14400	14387
62	14500	14520	14510	14510	14310	14300	14320	14310
63	14430	14400	14420	14417	14250	14230	14260	14247
64	14350	14340	14360	14350	14160	14150	14180	14163
65	14290	14280	14290	14287	14100	14120	14100	14107
66	14230	14200	14220	14217	14040	14020	13950	14003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1(ต่อ)

क्रम क्रम	ทวนเข็ม				ตามเข็ม			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
67	14130	14120	14110	14120	13960	13950	13960	13957
68	13920	13900	13910	13910	13890	13880	13880	13883
69	13810	13820	13800	13810	13810	13800	13800	13803
70	13740	13730	13740	13737	13710	13700	13720	13710
71	13660	13640	13670	13657	13650	13640	13670	13653
72	13600	13610	13620	13610	13570	13590	13580	13580
73	13520	13510	13530	13520	13490	13480	13490	13487
74	13440	13430	13450	13440	13420	13400	13410	13410
75	13350	13330	13570	13417	13340	13330	13340	13337
76	13270	13250	13260	13260	13270	13250	13280	13267
77	13190	13180	13190	13187	13180	13190	13190	13187
78	13080	13060	13090	13077	13130	13120	13150	13133
79	13000	13020	13010	13010	13030	13020	13040	13030
80	12920	12900	12930	12917	12940	12920	13960	13273
81	12820	12800	12830	12817	12840	12830	12850	12840
82	12750	12740	12760	12750	12750	12760	12770	12760
83	12670	12650	12680	12667	12670	12660	12670	12667
84	12590	12570	12590	12583	12600	12610	12610	12607
85	12520	12500	12510	12510	12450	12440	12470	12453
86	12420	12400	12420	12413	12400	12410	12410	12407
87	12330	12310	12320	12320	12350	12360	12360	12357
88	12240	12230	12240	12237	12250	12270	12260	12260
89	12180	12170	12190	12180	12180	12160	12190	12177
90	12000	12020	12020	12013	12120	12110	12110	12113
91	11970	11970	11980	11973	12050	12040	12060	12050
92	11910	11900	11910	11907	11940	11940	11950	11943
93	11880	11870	11870	11873	11870	11850	11850	11857
94	11850	11850	11850	11850	11770	11770	11760	11767

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1(ต่อ)

ครั้งที่ มุม	ทวนเข็ม				ตามเข็ม			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
95	11810	11830	11830	11823	11690	11680	11690	11687
96	11760	11790	11780	11777	11600	11580	11580	11587
97	11710	11730	11710	11717	11460	11440	11440	11447
98	11660	11670	11670	11667	11350	11370	11360	11360
99	11570	11560	11590	11573	11270	11250	11270	11263
100	11510	11500	11520	11510	11180	11160	11170	11170
101	11410	11400	11430	11413	11110	11100	11120	11110
102	11310	11300	11330	11313	11060	11050	11070	11060
103	11260	11250	11270	11260	11010	11000	11010	11007
104	11210	11220	11230	11220	10870	10860	11870	11200
105	11130	11100	11140	11123	10770	10780	10780	10777
106	11000	11010	11010	11007	10730	10720	10740	10730
107	10910	10920	10920	10917	10650	10640	10670	10653
108	10850	10880	10870	10867	10420	10400	10430	10417
109	10780	10760	10790	10777	10330	10310	10340	10327
110	10680	10670	10680	10677	10240	10250	10260	10250
111	10610	10600	10610	10607	10180	10160	10190	10177
112	10530	10520	10540	10530	11120	11140	11130	11130
113	10430	10440	10440	10437	11060	11040	11060	11053
114	10340	10330	10350	10340	11010	11000	11020	11010
115	10280	10270	10290	10280	9940	9960	9950	9950
116	10210	10200	10230	10213	9870	9880	9980	9910
117	10120	10110	10120	10117	9830	9840	9840	9837
118	10040	10030	10030	10033	9730	9720	9740	9730
119	9990	9980	9990	9987	9640	9650	9660	9650
120	9910	9900	9900	9903	9530	9540	9540	9537
121	9820	9800	9810	9810	9430	9450	9450	9443
122	9730	9740	9730	9733	9320	9340	9330	9330

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1(ต่อ)

มุ่ม ครั้งที่	ทวนเข้ม				ตามเข้ม			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
123	9610	9600	9610	9607	9220	9210	9230	9220
124	9520	9530	9530	9527	9150	9140	9170	9153
125	9440	9430	9450	9440	9110	9100	9130	9113
126	9350	9360	9360	9357	9000	9020	9010	9010
127	9260	9240	9270	9257	8930	8950	8940	8940
128	9200	9210	9220	9210	8850	8860	8850	8853
129	9100	9110	9110	9107	8630	8650	8640	8640
130	9040	9030	9060	9043	8520	8530	8530	8527
131	8950	8960	8960	8957	8440	8450	8440	8443
132	8880	8890	8890	8887	8350	8320	8340	8337
133	8810	8800	8120	8577	8270	8260	8280	8270
134	8740	8730	8760	8743	8200	8210	8210	8207
135	8660	8670	8670	8667	8130	8140	8130	8133
136	8580	8560	8590	8577	8070	8060	8060	8063
137	8500	8510	8510	8507	8010	8000	8010	8007
138	8410	8400	8430	8413	7950	7970	7960	7960
139	8360	8370	8380	8370	7890	7870	7880	7880
140	8240	8250	8270	8253	7840	7850	7840	7843
141	7800	7810	7830	7813	7800	7820	7830	7817

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการทดลองความสัมพันธ์ระหว่างพัลส์(pulse)กับมุมของเซอร์โวมอเตอร์
(servo motor) รุ่น S666 STD ตัวที่2(แขนฐาน)

มุม	ครั้งที่	ทวนเข็ม				ตามเข็ม			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
45		7570	7590	7600	7588	7990	7970	7970	7977
46		7730	7720	7730	7727	8140	8160	8130	8143
47		7820	7810	7810	7813	8210	8200	8190	8200
48		7910	7900	7930	7913	8300	8300	8290	8297
49		8000	7980	7990	7990	8380	8390	8390	8387
50		8130	8130	8140	8133	8550	8450	8480	8493
51		8190	8220	8240	8217	8610	8570	8600	8593
52		8310	8320	8320	8317	8710	8710	8710	8710
53		8380	8420	8410	8403	8810	8810	8820	8813
54		8470	8510	8510	8497	8910	8920	8920	8917
55		8570	8600	8590	8587	8960	8960	8970	8963
56		8620	8670	8680	8657	9080	9070	9070	9073
57		8760	8770	8780	8770	9200	9210	9200	9203
58		8880	8880	8890	8883	9330	9340	9350	9340
59		8980	8980	9000	8987	9400	9410	9430	9413
60		9090	9090	9110	9097	9480	9490	9480	9483
61		9150	9180	9180	9170	9610	9580	9600	9597
62		9230	9260	9250	9247	9710	9720	9690	9707
63		9350	9350	9370	9357	9760	9760	9740	9753
64		9410	9420	9440	9423	9880	9890	9880	9883
65		9540	9530	9560	9543	10010	10020	10020	10017
66		9630	9630	9650	9637	10060	10070	10090	10073
67		9720	9740	9770	9743	10180	10180	10170	10177
68		9810	9850	9850	9837	10260	10270	10270	10267
69		9900	9920	9940	9920	10390	10340	10370	10367
70		10020	10050	10080	10050	10430	10430	10450	10437
71		10120	10150	10160	10143	10550	10520	10530	10533

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ)

มูม ครั้งที่	ทวนเข็ม				ตามเข็ม			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
72	10210	10250	10250	10237	10610	10650	10640	10633
73	10310	10340	10370	10340	10740	10730	10720	10730
74	10420	10450	10470	10447	10840	10850	10850	10847
75	10520	10530	10560	10537	10920	10940	10940	10933
76	10620	10620	10660	10633	11030	11040	11030	11033
77	10720	10720	10770	10737	11130	11130	11120	11127
78	10810	10830	10870	10837	11220	11240	11210	11223
79	10910	10920	10960	10930	11310	11300	11320	11310
80	11010	11020	11040	11023	11390	11390	11400	11393
81	11100	11100	11160	11120	11470	11480	11470	11473
82	11190	11210	11250	11217	11610	11610	11600	11607
83	11320	11340	11390	11350	11690	11700	11700	11697
84	11410	11400	11460	11423	11790	11790	11790	11790
85	11510	11530	11570	11537	11880	11880	11900	11887
86	11590	11610	11660	11620	11950	12030	12020	12000
87	11710	11730	11790	11743	11220	12050	12050	11773
88	11780	11840	11870	11830	12070	12080	12100	12083
89	11860	11970	11980	11937	12150	12170	12160	12160
90	12000	12000	12000	12000	12290	12300	12270	12287

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลการทดลองความสัมพันธ์ระหว่างพัลส์(pulse)กับมุมของเซอร์โวมอเตอร์
(servo motor) รุ่น S666 STD ตัวที่ 2 (มอเตอร์แขนท่อนบน)

มุม / ครั้งที่	ทวนเข็ม				ตามเข็ม			
	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย
20	12310	12300	12320	12310	12000	12000	12000	12000
21	12370	12390	12380	12310	12030	12050	12060	12055
22	12450	12460	12460	12456	12080	12090	12110	12093
23	12540	12560	12550	12550	12160	12170	12190	12173
24	12620	12650	12630	12633	12240	12240	12260	12247
25	12700	12710	12700	12703	12300	12310	12320	12310
26	12800	12790	12810	12800	12400	12410	12420	12410
27	12860	12870	12870	12866	12460	12460	12470	12463
28	12950	12970	12940	12953	12540	12550	12560	12550
29	13010	13020	13000	13010	12610	12610	12630	12617
30	13110	13120	13110	13113	12680	12690	12690	12687
31	13200	13220	13210	13210	12750	12760	12760	12757
32	13260	13290	13270	13273	12830	12840	12830	12833
33	13330	13340	13350	13340	12920	12940	12930	12930
34	13380	13390	13390	13386	12990	13000	13010	13000
35	13470	13460	13460	13463	13050	13150	13060	13087
36	13530	13520	13540	13530	13140	13150	13160	13150
37	13560	13570	13560	13563	13200	13210	13210	13207
38	13630	13650	13640	13640	13280	13280	13290	13283
39	13750	13770	13760	13760	13340	13330	13350	13340
40	13800	13810	13810	13806	13420	13420	13450	13430
41	13860	13850	13860	13856	13500	13510	13520	13510
42	13940	13930	13940	13936	13560	13570	13580	13570
43	14000	14010	14020	14010	13630	13620	13640	13630
44	14070	14090	14090	14083	13690	13680	13700	13690
45	14150	14170	14160	14160	13770	13760	13790	13773
46	14240	14260	14250	14250	13830	13840	13840	13837

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3(ต่อ)

ครั้งที่ มุม	ทวนเข็ม				ตามเข็ม			
	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย
47	14280	14290	14290	14286	13900	13910	13900	13903
48	14350	14360	14340	14350	13960	13400	13980	13780
49	14420	14410	14410	14413	14020	14010	14030	14020
50	14470	14470	14460	14466	14090	14100	14090	14093
51	14550	14560	14540	14550	14190	14190	14200	14193
52	14620	14630	14620	14623	14270	14260	14280	14270
53	14670	14680	14680	14676	14320	14310	14320	14317
54	14750	14770	14760	14760	14380	14390	14400	14390
55	14830	14850	14820	14833	14430	14420	14440	14430
56	14890	14900	14910	14900	14500	14520	14530	14517
57	14960	14970	14970	14966	14600	14610	14600	14603
58	15020	15030	15030	15026	14720	14710	14700	14710
59	15100	15100	15110	15103	14790	14780	14790	14787
60	15170	15170	15180	15173	14870	14870	14890	14877
61	15260	15250	15250	15253	14940	14930	14960	14943
62	15310	15300	15310	15303	15070	15080	15090	15080
63	15390	15380	15380	15383	15160	15150	15170	15160
64	15440	15440	15450	15443	15240	15220	15250	15237
65	15520	15530	15540	15530	15300	15280	15320	15300
66	15580	15580	15590	15583	15380	15370	15390	15380
67	15680	15680	15690	15683	15470	15460	15490	15473
68	15750	15740	15770	15753	15540	15550	15550	15547
69	15790	15800	15800	15796	15620	15620	15640	15627
70	15850	15860	15870	15860	15680	15670	15690	15680
71	15920	15930	15930	15926	15750	15740	15760	15750
72	16000	16010	16020	16010	15840	15850	15860	15850
73	16060	16070	16080	16070	15890	15890	15900	15893
74	16150	16130	16160	16146	15910	15920	15930	15920

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3(ต่อ)

ครั้งที่ มุม	ทวนเข็ม				ตามเข็ม			
	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย
75	16200	16190	16210	16200	15990	15600	15610	15733
76	16270	16260	16270	16266	16120	16130	16130	16127
77	16340	16340	16340	16340	16170	16180	16190	16180
78	16410	16400	16420	16410	16250	16240	16270	16253
79	16480	16470	16480	16476	16270	16280	16290	16285
80	16560	16550	16570	16560	16310	16320	16330	16320
81	16650	16660	16660	16656	16460	16480	16470	16470
82	16700	16710	16710	16706	16530	16560	16550	16547
83	16760	16770	16770	16766	16610	16630	16630	16623
84	16850	16850	16860	16853	16680	16690	16700	16690
85	16920	16920	16930	16923	16710	16730	16740	16727
86	17000	17010	17010	17006	16770	16780	16790	16780
87	17120	17130	17140	17130	16820	16840	16850	16837
88	17140	17150	17160	17150	16990	16990	16700	16893
89	17190	17190	17200	17193	17080	17090	17100	17090
90	17270	17260	17270	17266	17160	17190	17190	17180
91	17350	17340	17350	17346	17230	17250	17260	17247
92	17420	17410	17420	17416	17310	17320	17340	17323
93	17480	17490	17500	17490	17380	17390	17390	17387
94	17550	17560	17570	17560	17450	17440	17450	17447
95	17610	17630	17620	17620	17510	17500	17520	17510
96	17690	17700	17710	17700	17620	17610	17620	17617
97	17760	17760	17770	17763	17690	17690	17700	17693
98	17830	17820	17850	17833	17750	17760	17760	17757
99	17900	17890	17930	17906	17820	17820	17840	17827
100	17960	17950	17990	17966	17920	17910	17930	17920
101	18040	18030	18060	18043	17980	17970	17990	17980
102	18110	18100	18120	18110	18040	18030	18050	18040

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3(ต่อ)

มูม	ทวนเข็ม				ตามเข็ม			
	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย
103	18170	18170	18180	18176	18130	18110	18130	18123
104	18260	18250	18280	18263	18210	18220	18220	18217
105	18340	18350	18350	18346	18300	18310	18320	18310
106	18420	18420	18440	18426	18380	18390	18400	18390
107	18490	18490	18500	18493	18490	18480	18500	18490
108	18550	18560	18570	18560	18590	18580	18610	18593
109	18610	18620	18620	18617	18680	18690	18700	18690
110	18750	18790	18770	18770	18750	18790	18770	18770



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ผลการทดลองความสัมพันธ์ระหว่างพัลส์(pulse)กับมุมของเซอร์โวมอเตอร์
(servo motor) เล็ก(มอเตอร์ตัวจับปากกา)

มุม ครั้งที่	ทวนเข็ม				ตามเข็ม			
	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย
45	8910	8910	8900	8907	8910	8910	8880	8900
46	9000	8990	8970	8987	9050	9070	9040	9053
47	9060	9040	9030	9043	9110	9130	9140	9127
48	9110	9100	9120	9110	9170	9200	9210	9193
49	9170	9180	9200	9183	9250	9280	9260	9263
50	9280	9250	9290	9273	9310	9340	9320	9323
51	9360	9320	9370	9350	9370	9390	9370	9377
52	9460	9390	9460	9437	9450	9470	9440	9453
53	9540	9440	9540	9507	9560	9570	9540	9557
54	9600	9530	9590	9573	9630	9690	9620	9647
55	9680	9630	9640	9650	9720	9750	9730	9733
56	9740	9710	9740	9730	9780	9790	9800	9790
57	9820	9800	9830	9817	9850	9850	9880	9860
58	9890	9840	9880	9870	9920	9940	9940	9933
59	9980	9920	9950	9950	9990	10000	9990	9993
60	10050	10020	11010	10360	10070	10040	10050	10053
61	10100	10100	11080	10427	10130	10150	10130	10137
62	10150	10170	11150	10490	10180	10210	10200	10197
63	10190	10230	10200	10207	10240	10280	10270	10263
64	10260	10300	10260	10273	10300	10330	10350	10327
65	10320	10340	10340	10333	10370	10390	10410	10390
66	10380	10380	11400	10720	10440	10440	10470	10450
67	10420	10440	11470	10777	10490	10500	10520	10503
68	10470	10530	11510	10837	10560	10590	10600	10583
69	10510	10600	11550	10887	10620	10670	10650	10647
70	10550	10630	11590	10923	10700	10790	10680	10723
71	10600	10680	11630	10970	10740	10850	10730	10773

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

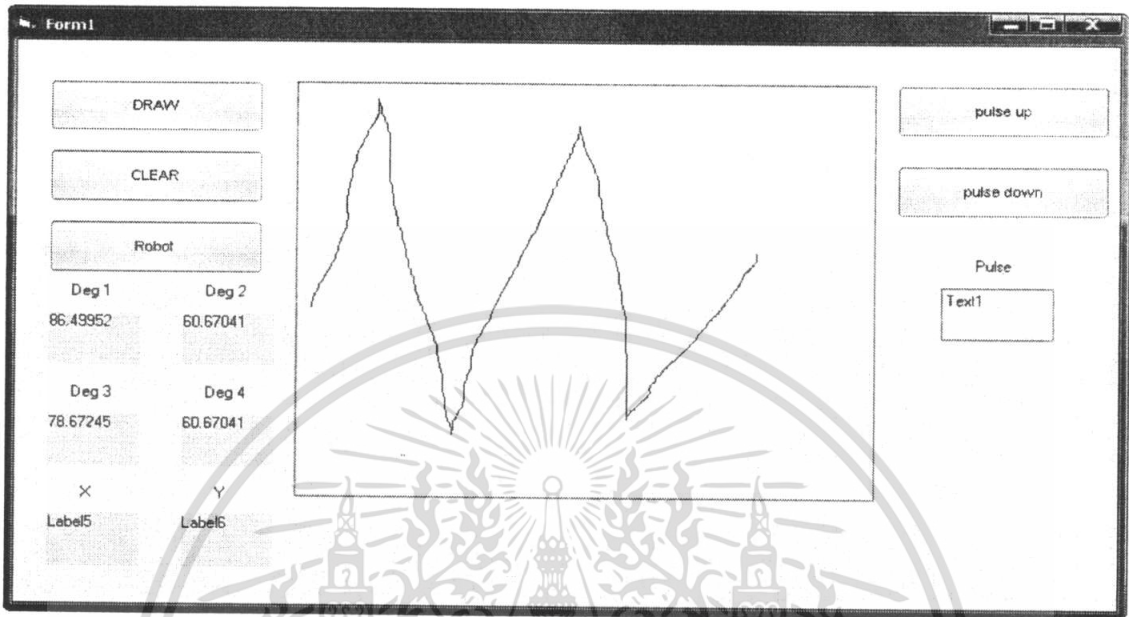
ตารางที่ 4.4(ต่อ)

ครั้งที่ มุม	ทวนเข็ม				ตามเข็ม			
	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย
72	10690	10730	11700	11040	10790	10900	10790	10827
73	10760	10780	11740	11093	10860	10920	10860	10880
74	10840	10850	11790	11160	10930	10970	10920	10940
75	10900	10930	11880	11237	10970	11020	10960	10983
76	10960	10980	11930	11290	11030	11070	11000	11033
77	11000	11020	11000	11007	11050	11130	11070	11083
78	11090	11090	11080	11087	11100	11170	11150	11140
79	11170	11150	11110	11143	11190	11220	11200	11203
80	11210	11220	11210	11213	11240	11260	11240	11247
81	11290	11310	11270	11290	11300	11350	11330	11327
82	11350	11350	11320	11340	11350	11390	11390	11377
83	11400	11430	11410	11413	11420	11450	11430	11433
84	11470	11490	11510	11490	11480	11500	11520	11500
85	11580	11570	11600	11583	11560	11550	11580	11563
86	11640	11670	11690	11667	11630	11610	11660	11633
87	11720	11740	11770	11743	11730	11720	11770	11740
88	11850	11830	11850	11843	11840	11820	11890	11850
89	11930	11930	11910	11923	11940	11930	11970	11947
90	12000	12000	12000	12000	12020	12050	12020	12030

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทำงานของแขนกล

ทำการวาดภาพผ่านโปรแกรมวิชวลเบสิก(visual basic 6)



รูปที่ 4.2 รูปแสดงหน้าต่างวาดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 การทดลองหาความสัมพันธ์ระหว่างพัลส์กับค่าอินพุตที่เซอร์โวมอเตอร์เคลื่อนที่ สรุป

1. จากการทดลองทั้ง 3 ครั้ง ของทั้งทิศทวนเข็มนาฬิกาและตามเข็มนาฬิกาของเซอร์โวมอเตอร์ แต่ละตัว จะเห็นว่าสำหรับค่ามุมใดๆ จะให้ค่าพัลส์ของการทดลองทั้ง 3 ครั้ง มีค่าใกล้เคียงกัน ดังนั้นจะสามารถนำค่าเฉลี่ยของพัลส์มาใช้ได้
2. จะเห็นว่าในแต่ละช่วงที่เซอร์โวมอเตอร์เปลี่ยนไป 1 องศา ค่าพัลส์ที่เปลี่ยนจะมีค่าไม่คงที่ จะมีค่าอยู่ในช่วง 30-100
3. จะเห็นว่าในช่วงค่ามุมเดียวกันของเซอร์โวมอเตอร์จะมีค่าพัลส์ในการหมุนทวนเข็มนาฬิกาและตามเข็มนาฬิกาไม่เท่ากัน มีค่าต่างกันค่อนข้างมาก

ปัญหา

1. ไม่สามารถหาความสัมพันธ์เป็นสมการเส้นตรงทางคณิตศาสตร์ระหว่างพัลส์เป็นค่ามุมที่ เซอร์โวมอเตอร์เคลื่อนที่ได้ เนื่องจากแต่ละช่วงที่เซอร์โวมอเตอร์เปลี่ยนไป 1 องศา ค่าพัลส์ที่เปลี่ยนจะมีค่าไม่คงที่ จะมีค่าอยู่ในช่วง 30-100
2. ในช่วงค่ามุมเดียวกันของเซอร์โวมอเตอร์จะมีค่าพัลส์ ในการหมุนทวนเข็มนาฬิกาและตามเข็มนาฬิกาไม่เท่ากัน มีค่าต่างกันค่อนข้างมาก

การแก้ปัญหา

1. ความสัมพันธ์ระหว่างพัลส์กับค่ามุมของการเคลื่อนที่ ในช่วงค่ามุมเดียวกันของเซอร์โวมอเตอร์จะนำไปใช้ในการเขียนโปรแกรม C ใน dsPIC เพื่อให้ dsPIC ประมวลผลจากค่ามุมที่ได้รับจากวิซวลเบสิกผ่านทางซีเรียลพอร์ตและสร้างพัลส์ให้เซอร์โวมอเตอร์

ถ้าไม่สามารถหาความสัมพันธ์เป็นสมการเป็นสมการเส้นตรงทางคณิตศาสตร์ได้ ดังนั้นจึงนำค่าเฉลี่ยที่ได้จากการทดลองทั้งชุดนำมาเขียนโปรแกรม C โดยเก็บเป็นอาร์เรย์ ระหว่างค่ามุมกับพัลส์

2. ในช่วงค่ามุมเดียวกันของ ในช่วงค่ามุมเดียวกันของเซอร์โวมอเตอร์จะมีค่าพัลส์ ในการหมุนทวนเข็มนาฬิกาและตามเข็มนาฬิกาไม่เท่ากัน มีค่าต่างกันค่อนข้างมาก

ดังนั้นในการเขียนโปรแกรม C ใน dsPIC ก็จะเก็บข้อมูลทั้งตามเข็มนาฬิกาและทวนเข็มนาฬิกา

5.2 การทดลองการทำงานของแขนกล

สรุป

จะเห็นว่าจากการวาดภาพผ่านทางหน้าจอคอมพิวเตอร์จากหน้าต่างวาดภาพในโปรแกรมวิซวลเบสิก 6 แล้วให้แขนกลวาดภาพตามจะเห็นว่าแขนกลมีแนวโน้มวาดภาพตามภาพที่ได้วาดไว้จากคอมพิวเตอร์แต่ภาพที่ได้ออกมานั้นยังคงมีค่าความผิดพลาด(error) เกิดขึ้นอยู่

ปัญหา

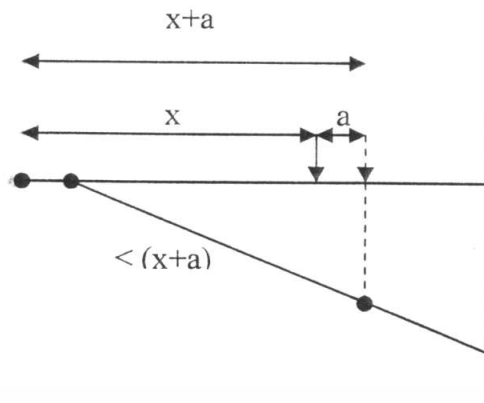
1. ภาพที่ได้มีค่าความผิดพลาดอยู่ จากภาพที่วาดผ่านคอมพิวเตอร์
2. จากการคำนวณ ในขณะที่แขนกลวาดภาพอยู่นั้น ส่วนของปากกาจะต้องตั้งฉากกับกระดาษเป็นค่าคงที่ค่าหนึ่ง

แต่ขณะทำการทดลองให้เคลื่อนที่โดยยังไม่ติดหัวปากกาจะเห็นว่าจุดปลายของแขนที่ไว้ติดปากกา มีค่าห่างจากกระดาษไม่คงที่ตามที่คำนวณไว้

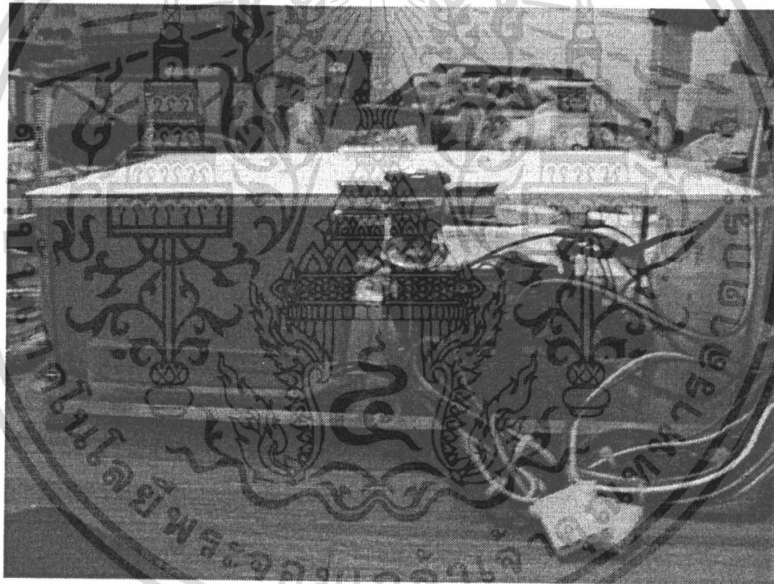
3. สำหรับบางภาพที่วางขีดขอบของหน้าต่างวาดภาพมากจนนั้น เมื่อแขนกลทำการวาดภาพที่ได้จะมีบางส่วนเลยออกมาจากกระดาษ A4

การแก้ปัญหา

1. ทำการแก้พื้นระนาบที่ใช้วางกระดาษให้ด้านปลายเอียงลงเล็กน้อย เนื่องจากสังเกตจากปัญหาที่จุดปลายของแขนที่ไว้ติดปากกา มีค่าห่างจากกระดาษไม่คงที่ลดลงจากที่คำนวณไว้ น่าจะเป็นผลมาจาก ในขณะที่แขนกลเคลื่อนที่วาดภาพอย่างต่อเนื่องก็จะเกิดความเร็วและแรงเฉื่อยเกิดขึ้น ส่งผลให้แขนกลเคลื่อนที่ลงจากมุมที่คำนวณไว้ ซึ่งทำให้พิกัดที่วาดภาพจะเคลื่อนห่างออกจากพิกัดของภาพที่วาดจากคอมพิวเตอร์ด้วย ถึงแม้จะทำการทดลองชดเชยค่ามุม โดยการทดลองหาความสัมพันธ์ระหว่างพัลส์กับค่ามุมของเซอร์โวมอเตอร์ด้วยการประกอบแขนกลก่อน แต่ก็ยังทำให้เกิดค่าความผิดพลาดอยู่ ดังนั้นการเอียงพื้นระนาบด้านปลายลงก็จะช่วยชดเชยค่าความผิดพลาดที่เกิดขึ้นได้ดังรูป



รูปที่ 5.1 รูปแสดงการแก้ปัญหาโดยการเอียงระนาบที่วางกระดาษ



รูปที่ 5.2 รูปการเอียงระนาบที่วางกระดาษ

2. จากปัญหาที่จุดปลายของแกนที่ไว้ติดปากกาซึ่งในขณะวาดภาพมีระยะห่างจากกระดาษไม่คงที่จากที่ได้คำนวณไว้ จะแก้ปัญหาโดยในส่วนของปากกาจะทำให้เคลื่อนที่อิสระไม่ยึดติดกับเซอร์โวมอเตอร์
3. สำหรับภาพวาดบางภาพเมื่อทำการวาดภาพจากหน้าจอคอมพิวเตอร์โดยภาพที่วาดอยู่ชิดขอบมากๆ อาจทำให้ภาพที่วาดโดยแกนกลลื่นออกจากกระดาษ A4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะแก้ไขโดยกำหนดหน้าต่างวาดรูปในวิหวลเบสิก 6 ให้มีขนาดเล็กลงจากขนาดกระดาษ A4 ขอบละ 1 เซนติเมตร ดังนั้นถ้าเกิดวาดภาพชิดขอบหน้าต่างวาดภาพมากๆ ก็ยังมีพื้นที่เหลือไว้อีก 1 เซนติเมตร

โดยกำหนดหน้าต่างวาดรูปให้มีพิกัดดังนี้

- ขอบล่างถึงขอบบน เท่ากับ -100 point ถึง 100 point
- ขอบซ้ายถึงขอบขวา มีค่าเท่ากับ 0 point ถึง 287 point

ข้อเสนอแนะ

1. จากการทำงานของเซอร์โวมอเตอร์มีค่าความผิดพลาดเกิดขึ้น เนื่องจากในเซอร์โวมอเตอร์มีการทำงานเป็นแบบ feedback control แบบ P ซึ่งเป็นระบบ control ที่ไม่สามารถจำกัดค่าความผิดพลาดได้

ซึ่งสามารถทำการแก้ไขเบื้องต้นจากการแก้ปัญหาที่ได้กล่าวมาแล้วคือ

- การทดลองหาความสัมพันธ์ระหว่างพัลส์ กับค่ามุมของเซอร์โวมอเตอร์

ด้วยการประกอบแขนกลก่อน

- ทำการเรียงพื่นระนาบที่ใช้วางกระดาษ

แต่ยังไม่สามารถจำกัดค่าความผิดพลาดได้หมด ซึ่งยังมีวิธีที่สามารถจำกัดได้อีกดังนี้

- ใช้เซอร์โวมอเตอร์ที่การทำงานเป็น feedback control แบบ PI ซึ่งจะสามารถ

จำกัดค่าความผิดพลาดได้ แต่เซอร์โวมอเตอร์แบบนี้มีราคาแพงกว่า

- สำหรับระบบ control แบบ P จะลดค่าความผิดพลาด ได้โดยการเพิ่มเกน(gain)

ของระบบ การเพิ่มค่าเกนของเซอร์โวมอเตอร์ทำได้โดย

- การเพิ่มค่าโวลท์ (Volt) ที่ต้องจ่ายให้กับเซอร์โวมอเตอร์
- เพิ่มความถี่พัลส์ที่ต้องจ่ายให้กับเซอร์โวมอเตอร์

เอกสารอ้างอิง

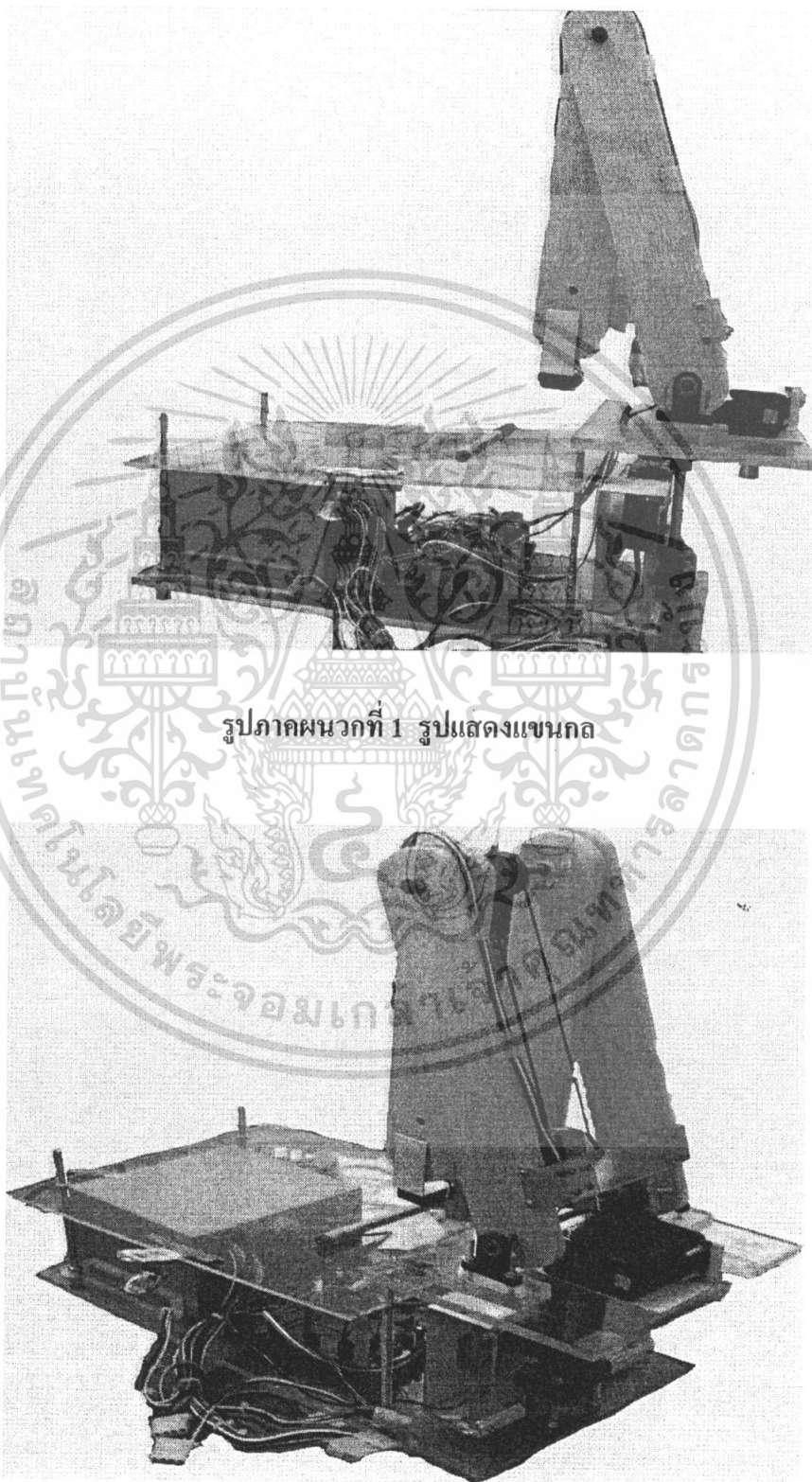
- 1.) Visual Basic 6 ฉบับโปรแกรมเมอร์ , กิตติ ภัคดีวิณะกุล – จำลอง ทรูอุตสาหะ , พิมพ์ครั้งที่ 12 , พฤศจิกายน 2547
- 2.) คู่มือการทดลอง dsPIC Microcontroller เบื้องต้น ด้วยโปรแกรมภาษา C กับ MPLAB C30 , นคร ภัคดีชาติ – ชัยวัฒน์ ลิ้มพรจิตรวิไล
- 3.) เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic , อภิชาติ ภู่พลับ , พิมพ์ครั้งที่ 1 , กันยายน 2546



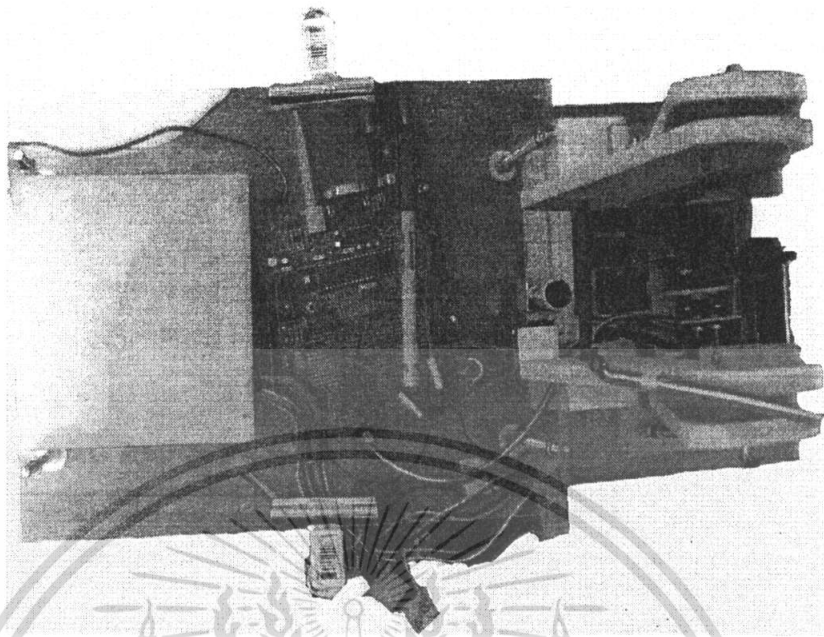
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

1. รูปแขนกล



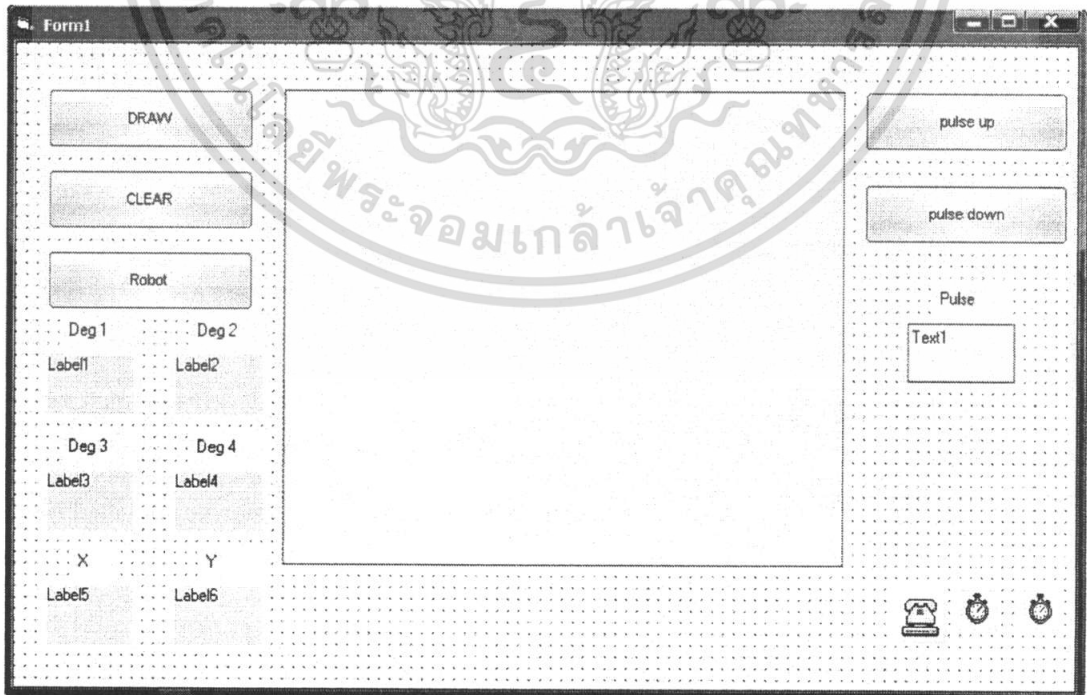
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปภาคผนวกที่ 2 รูปแสดงแขนกล
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาคผนวกที่ 3 รูปแสดงแขนกล

2. โปรแกรมวิชวลเบสิก 6 (Visual Basic 6)

1.1 ฟอรัม (Form)



รูปภาคผนวกที่ 4 หน้าต่างฟอรัม(Form)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพียงอย่างเดียว (ลิขสิทธิ์) เพื่อให้ผู้ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 โค้ด(Code)

```
Dim XX(10000) As Single, YY(10000) As Single, no As Integer, i As Integer
Dim cc As Integer, AA As Integer
Dim st1 As Integer, st2 As Integer, st3 As Integer
Dim pulse As Integer
Dim x1 As Single, x2 As Single, a1 As Single, a2 As Single, a3 As Single,
    a4 As Single, a5 As Single, b1 As Single, b2 As Single, c1 As Single,
    c2 As Single, c3 As Single, c4 As Single
Dim rada As Single, radb As Single, radc As Single, radd As Single
Dim dega(10000) As Integer, degb(10000) As Integer, degc(10000) As Integer,
    degd(10000) As Integer
```

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Dim deg1 As Single, deg2 As Single, deg3 As Single, deg4 As Single
Private Sub cmdDeg_Click()
```

```
    For i = 0 To no - 1
        Form3.Show vbModeless
        Form3.Print dega(i); degb(i); degd(i); degc(i); no
    Next i
End Sub
```

```
Private Sub cmdDraw_Click()
    cc = 0
End Sub
```

```
Private Sub cmdPoint_Click()
    For i = 0 To no - 1
        Form2.Show vbModeless
        Form2.Print XX(i); YY(i)
    Next i
End Sub
```

```
Private Sub Command1_Click()
Dim dd As Integer
dd = 90
    mscsend.Output = Chr(dd)
    Sleep (1)
    mscsend.Output = Chr(dd)
    Sleep (1)
    mscsend.Output = Chr(dd)
    Sleep (1)
    mscsend.Output = Chr(dd)
    Sleep (100)
    pulse = pulse + 10
    Text1.Text = pulse
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Command2_Click()
```

```
For i = 0 To no - 1
    mscsend.Output = Chr(dega(i))
    Sleep (1)
    mscsend.Output = Chr(degbi(i))
    Sleep (1)
    mscsend.Output = Chr(degci(i))
    Sleep (1)
    mscsend.Output = Chr(degdi(i))
    Sleep (200)
Next i
Sleep (1000)
mscsend.Output = Chr(90)
Sleep (1)
mscsend.Output = Chr(90)
Sleep (1)
mscsend.Output = Chr(90)
Sleep (1)
mscsend.Output = Chr(90)
Sleep (50)
End Sub
```

```
Private Sub Command3_Click()
```

```
Dim dd As Integer
```

```
ddd = 80
mscsend.Output = Chr(ddd)
Sleep (1)
mscsend.Output = Chr(ddd)
Sleep (1)
mscsend.Output = Chr(ddd)
Sleep (1)
mscsend.Output = Chr(ddd)
Sleep (100)
pulse = pulse - 10
Text1.Text = pulse
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim i As Integer
```

```
mscsend.CommPort = 5
```

```
mscsend.Settings = "9600,n,8,1"
```

```
mscsend.PortOpen = True
```

```
pulse = 12000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

```
Private Sub Frame1_DragDrop(Source As Control, X As Single, Y As Single)
```

End Sub

```
Private Sub picture1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
Picture1.CurrentX = X
```

```
Picture1.CurrentY = Y
```

End Sub

```
Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
Picture1.ScaleMode = 0
```

```
Picture1.Scale (0, 100)-(287, -100)
```

```
AA = 1
```

```
If Button = vbLeftButton Then
```

```
  If cc = 0 Then
```

```
    Picture1.Line (Picture1.CurrentX, Picture1.CurrentY)-(X, Y)
```

```
    XX(no) = X
```

```
    YY(no) = Y
```

```
    x1 = XX(no)
```

```
    x2 = YY(no)
```

```
    R = 250
```

```
    a3 = (x1 + 87) * (x1 + 87)
```

```
    a4 = x2 * x2
```

```
    a5 = a3 + a4
```

```
    a1 = Sqr(a5)
```

```
    a2 = x2 / a1
```

```
    rada = Atn(a2)
```

```
    deg1 = Abs((rada * 57.3) - 90)
```

```
    dega(no) = Abs((rada * 57.3) - 90) 'base
```

```
    b1 = 2 * R
```

```
    b2 = a1 / b1
```

```
    radb = Atn(-b2 / Sqr(-b2 * b2 + 1)) + 2 * Atn(1)
```

```
    deg2 = (radb * 57.3) + 10
```

```
    degb(no) = (radb * 57.3) + 10 'big
```

```
    radd = radb
```

```
    deg4 = (radb * 57.3) + 10
```

```
    degd(no) = (radb * 57.3) + 10 'hand
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

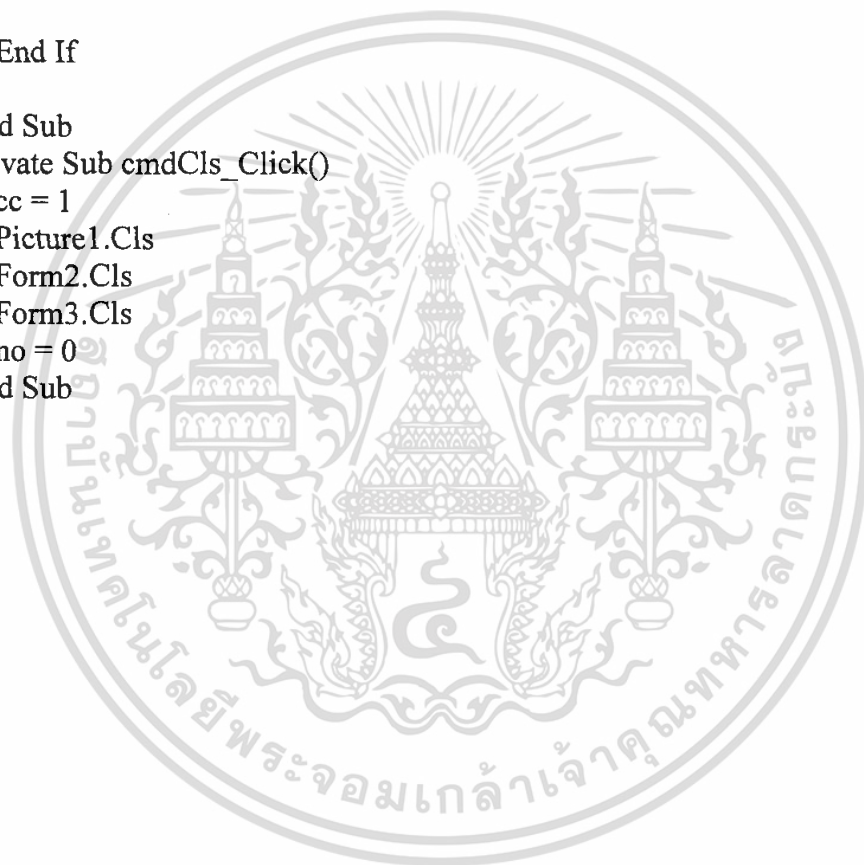
```

c1 = 2 * R * R
c2 = a1 * a1
c3 = c1 - c2
c4 = c3 / c1
radc = Atn(-c4 / Sqr(-c4 * c4 + 1)) + 2 * Atn(1)
deg3 = (radc * 57.3)
degc(no) = (radc * 57.3) 'small
'degc(no) = (radc * 57.3)
no = no + 1
Label1.Caption = deg1
Label2.Caption = deg2
Label3.Caption = deg3
Label4.Caption = deg4
End If

End If

End Sub
Private Sub cmdCls_Click()
cc = 1
Picture1.Cls
Form2.Cls
Form3.Cls
no = 0
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

,9193,9263,9323,9377,9453,9557,9647,9733,9790,9860,9933,9993,10053,10137,10197,10263,10327,10390,10450

,10503,10583,10647,10723,10773,10827,10880,10940,10983,11033,11083,11140,11203,11247,11327,11377,11433

,11500,11563,11633,11740,11850,11947,12030};

```

void _ISR_U2TXInterrupt(void)
{
    IFS1bits.U2TXIF=0;
}
void _ISR_U2RXInterrupt(void)
{
    IFS1bits.U2RXIF=0;
}

void __attribute__((__interrupt__)) _T1Interrupt( void )
{
    T1CONbits.TON = 0;
    IFS0bits.T1IF = 0;

    if(aaa1==0) { L1=0;

        if(set1>D1) {PR1=D1; set1=set1-D1;} else {PR1=set1;aaa1=1;}

        T1CONbits.TON = 1;
    }
    else {L1=1;
        PR1=D1;aaa1=0;
        set1=160000-D1;
        T1CONbits.TON = 1;
    }
}

void __attribute__((__interrupt__)) _T2Interrupt( void )
{
    T2CONbits.TON = 0;
    IFS0bits.T2IF = 0;

    if(aaa2==0) { L2=0;

        if(set2>D2) {PR2=D2; set2=set2-D2;} else {PR2=set2;aaa2=1;}

        T2CONbits.TON = 1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else {L2=1;
    PR2=D2;aaa2=0;
    set2=160000-D2;
    T2CONbits.TON = 1;
}
}
void __attribute__((__interrupt__)) _T3Interrupt( void )
{
    T3CONbits.TON = 0;
    IFS0bits.T3IF = 0;

    if(aaa3==0) { L3=0;

        if(set3>D3) {PR3=D3; set3=set3-D3;} else {PR3=set3;aaa3=1;}

        T3CONbits.TON = 1;
    }
    else {L3=1;
        PR3=D3;aaa3=0;
        set3=160000-D3;
        T3CONbits.TON = 1;
    }
}
void __attribute__((__interrupt__)) _T4Interrupt( void )
{
    T4CONbits.TON = 0;
    IFS1bits.T4IF = 0;

    if(aaa4==0) { L4=0;

        if(set4>D4) {PR4=D4; set4=set4-D1;} else {PR4=set4;aaa4=1;}

        T4CONbits.TON = 1;
    }
    else {L4=1;
        PR4=D4;aaa4=0;
        set4=160000-D4;
        T4CONbits.TON = 1;
    }
}
}
void Init_Timer1(void)
{ T1CON = 0;
    IFS0bits.T1IF = 0;
    IPC0bits.T1IP = 1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    IEC0bits.T1IE = 1;

    PR1 = D1;
    T1CONbits.TCS = 0;

    T1CONbits.TON = 1;
}
void Init_Timer2( void )
{

    T2CON = 0;
    IFS0bits.T2IF = 0;
    IPC1bits.T2IP = 2;
    IEC0bits.T2IE = 1;

    PR2 = D2;
    T2CONbits.TCS = 0;

    T2CONbits.TON = 1;
}
void Init_Timer3(void)
{ T3CON = 0;
  IFS0bits.T3IF = 0;
  IPC1bits.T3IP = 3;
  IEC0bits.T3IE = 1;

  PR3 = D3;
  T3CONbits.TCS = 0;

  T3CONbits.TON = 1;
}
void Init_Timer4( void )
{

    T4CON = 0;
    IFS1bits.T4IF = 0;
    IPC5bits.T4IP = 4;
    IEC1bits.T4IE = 1;

    PR4 = D4;
    T4CONbits.TCS = 0;

    T4CONbits.TON = 1 ;

}

```

```
void delay(int t)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  int i,j;
  for(i=0;i<t;i++)
  for(j=0;j<250;j++);
}

```

```

void init()

```

```

{
  unsigned int baudvalue;
  unsigned int U2MODEvalue;
  unsigned int U2STAValue;

  CloseUART2();
  ConfigIntUART2(UART_RX_INT_EN
&UART_RX_INT_PR6&UART_TX_INT_EN&UART_TX_INT_PR2 );
  baudvalue=51;//
  U2MODEvalue=UART_EN &
  UART_IDLE_CON &
  UART_RX_TX &
  UART_DIS_WAKE &
  UART_DIS_LOOPBACK &
  UART_DIS_ABAUD &
  UART_NO_PAR_8BIT &
  UART_1STOPBIT;
  U2STAValue= UART_INT_TX_BUF_EMPTY &
  UART_TX_PIN_NORMAL &
  UART_TX_ENABLE &
  UART_INT_RX_3_4_FUL &
  UART_ADR_DETECT_DIS &
  UART_RX_OVERRUN_CLEAR ;
  OpenUART2(U2MODEvalue,U2STAValue,baudvalue);
}

```

```

main()

```

```

{
  TRISB=0;
  TRISC=0;
  TRISD=0;
  TRISE=0;
  init();
  Init_Timer1();
  Init_Timer2();
  Init_Timer3();
  Init_Timer4();

```

```

  putsUART2((unsigned int *)"tomtam");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(BusyUART2());
while(1)
{
    if(DataRdyUART2())
    {
        // dat = ReadUART2();

        //if(deg==0) {if(dat==90) {D1=D1+10;deg=1;} else {D1=D1-10;deg=1;}}
        //else if(deg==1) {if(dat==90) {D2=D2-10;deg=2;} else {D2=D2+10;deg=2;}}
        //{D2=(dat*77.78)+5000;deg=2;}
        //else if(deg==1) {D2=12000;deg=2;} //{D2=(dat*77.78)+5000;deg=2;}
        // else if(deg==2){if(dat==90) {D3=D3+10;deg=3;} else {D3=D3-10;deg=3;}}
        //{D3=(dat*77.78)+5000;deg=3;}
        //else {if(dat==90) {D4=D4+10;deg=0;} else {D4=D4-10;deg=0;}}
        //{D4=(dat*77.78)+5000;deg=0;}
        //WriteUART2(dat);
        // while(BusyUART2());
        dat = ReadUART2();
        if(deg==0) {D1=motorbaseuptodown[dat];deg=1;}
        else if(deg==1) {D2=motorarm1uptodown[dat];deg=2;}
        else if(deg==2) {D3=motorarm2downtoup[dat];deg=3;}
        // else {D4=(dat*77.78)+5000;deg=0;}
        else {D4=motorsmalluptodown[dat];deg=0;}
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้