สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้ FPGA เพื่อสั่งการควบคุมแขนกลในอุตสาหกรรม

APPLICATION FPGA WITH ROBOT ARM CONTROL FOR INDUSTRY

T104107

นายธงไชย เจริญเสถียรโชค
นายธราพงษ์ คำชาลี
นายสุรศักดิ์ บุตรบำรุง

เลขหมู่.....
เลขทะเบียน 104107
วัน,เดือน,ปี 30 มี.ค. 2552
.b 21102556
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION FPGA WITH ROBOT ARM CONTROL FOR INDUSTRY

THONGCHAI JAROENSATHIANCHOK

THARAPONG KHAMCHALEE

SURASAK BUTTBUMRUNG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENT ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUE OF TECHNOLOGY LADKRABANG

2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	การประยุกต์ใช้ FPGA เพื่อสั่งการควบคุมแขนกลในอุตสาหกรรม		
	APPLICATION FPGA WITH ROBOT ARM CONTROL FOR INDUSTRY		
นักศึกษาผู้จัดทำ	นายธงไชย	เจริญเสถียรโชค	รหัสนักศึกษา 48010336
	นายธราพงษ์	คำชาลี	รหัสนักศึกษา 48010373
	นายสุรศักดิ์	บุตรบำรุง	รหัสนักศึกษา 48011023
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมการวัดคุม		
ปีการศึกษา	2551		

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ.ไสว พงศ์สวัสดิ์	
อ.ธีรวัฒน์ เทพมณี	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การประยุกต์ใช้ FPGA เพื่อสั่งการควบคุมแขนกลในอุตสาหกรรม		
	APPLICATION FPGA WITH ROBOT ARM CONTROL FOR INDUSTRY		
นักศึกษาผู้จัดทำ	นายธงไชย	เจริญเสถียรโชค	รหัสนักศึกษา 48010336
	นายธราพงษ์	คำชาลี	รหัสนักศึกษา 48010373
	นายสุรศักดิ์	บุตรบำรุง	รหัสนักศึกษา 48011023
อาจารย์ที่ปรึกษา	รศ.ไสว	พงศ์สวัสดิ์	
	อ.ธีรวัฒน์	เทพมณี	
ปีการศึกษา	2551		

บทคัดย่อ

โครงงานนี้นำเสนอการศึกษาการสร้างและพัฒนาสัญญาณขับเคลื่อนสั่งการแขนกล 4 แกน พร้อมกับชุดมือจับ เพื่อให้สามารถเคลื่อนที่และทำงานได้ตามที่ต้องการ การสร้างสัญญาณ-ขับเคลื่อนจะใช้สัญญาณ PWM แบบ 5 ช่องสัญญาณ ที่สร้างจากบอร์ด FPGA สำหรับการควบคุมตำแหน่ง, ความเร็วดีซีเซอร์โวมอเตอร์ในแต่ละแกน และคลิปที่ปลายแขน การสร้างสัญญาณ PWM จะพัฒนาด้วยโปรแกรม Verilog HDL และบอร์ด FPGA ของ ALTERA ส่วนการสั่งงานในการเคลื่อนที่สามารถทำได้โดยการใช้คีย์ปุ่มกด และคอมพิวเตอร์ที่มีการพัฒนาโปรแกรมด้วย Visual Basic 6 การทดลองการทำงานปรากฏว่าแขนกลสามารถเคลื่อนที่และทำงานโดยการใช้คลิปหยิบชิ้นงานที่กำหนดได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Application FPGA with Robot Arm Control for Industry	
Authors	Mr. Thongchai	Jaroensathianchok
	Mr. Tharapong	Khamchalee
	Mr. Surasak	Buttbumrung
Thesis Advisor	Assoc.Prof.Sawai	Pongswatd
	Mr.Teerawat	Thepmanee
Year	2008	

ABSTRACT

The thesis is presented about education, generated and the development of a Pulse Width Modulation (PWM) signal to control the 4-axis robot arm with grip. Principle of PWM with Five-channel PWM signal which is generated from Field Programmable Gate Arrays (FPGA) to control the position and speed of DC Servo Motor at each axis and its grip. To generate the PWM signal, using with Verilog HDL (Verilog Hardware Description Language program) and FPGA of ALTERA, then make the control of robot arm by keypad. The computer that has developed program with Visual Basic 6. From experiment, we know that robot arm can move and work by using grip to pick up desired object.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องด้วยความอนุเคราะห์จากปัจจัยหลาย ๆ อย่าง ทางคณะผู้จัดทำขอขอบคุณคณะวิศวกรรมศาสตร์ และภาควิชาวิศวกรรมการวัดคุม ที่ได้ให้ประสบการณ์ในการใช้ชีวิตในมหาวิทยาลัย การทำปริญญานิพนธ์ฉบับนี้ ขอขอบพระคุณเป็นอย่างสูงสำหรับ รองศาสตร์ตราจารย์ ไสว พงศ์สวัสดิ์, อาจารย์ ธีรวัฒน์ เทพมณี, อาจารย์ กฤษณ์ เสมอพิทักษ์ และอาจารย์ท่านอื่น ๆ กับคำแนะนำคำปรึกษาแก่คณะผู้จัดทำตลอดมา อีกทั้งเอื้อเฟื้ออุปกรณ์ ห้องทำโครงงาน เครื่องมืออุปกรณ์ต่าง ๆ ตลอดจนความช่วยเหลือทั้งในด้านทุนทรัพย์และแรงงาน ทางคณะผู้จัดทำรู้สึกซาบซึ้งขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ภายในภาควิชาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยให้กำลังใจ และให้คำแนะนำที่ดีเสมอมา

สุดท้ายนี้ขอกราบขอบพระคุณ บิดา มารดา ที่สนับสนุนและเป็นกำลังใจเสมอมา และให้การสนับสนุนช่วยเหลือในทุก ๆ เรื่อง

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าปริญญานิพนธ์ฉบับนี้ จะก่อประโยชน์แก่ผู้ศึกษา และผู้ที่มีความสนใจในเรื่อง การประยุกต์ใช้ FPGA เพื่อสั่งการควบคุมแขนกลในอุตสาหกรรม และสามารถนำไปศึกษาและพัฒนาให้เกิดประโยชน์ต่อไป

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VIII

สารบัญรูป IX

บทที่ 1 บทนำ 1

1.1 ความสำคัญของปริญญานิพนธ์ 1

1.2 ที่มาของปริญญานิพนธ์ 1

1.3 วัตถุประสงค์ของปริญญานิพนธ์ 2

1.4 ขอบเขตของปริญญานิพนธ์ 2

1.5 ขั้นตอนการดำเนินงาน 2

1.6 ผลที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีพื้นฐานและการควบคุมแขนกล 3

2.1 แนวความคิด 3

2.2 หลักการ 3

2..2.1 การกำเนิดสัญญาณ PWM (Pulse Width Modulation) 4

2..2.2 การออกแบบการสร้างสัญญาณ PWM 5

2.3 มอเตอร์ไฟฟ้ากระแสตรงแบบดีซีเซอร์โวมอเตอร์ (DC Servo Motor) 8

2.3.1 ดีซีเซอร์โวมอเตอร์ (DC Servo Motor) 8

2.3.2 หลักการทำงานของดีซีเซอร์โวมอเตอร์ 10

2.4 อุปกรณ์ FPGA (Field Programmable Gate Array) 12

2.4.1 การออกแบบด้วย FPGA 13

2.4.2 คุณสมบัติของ FPGA 14

2.5 การทำงานของวงจร Opto Electronics ในการขยายสัญญาณขับมอเตอร์ 14

2.6 การออกแบบวงจรรับ – ส่งข้อมูลแบบอนุกรมด้วย FPGA 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.6.1 การสื่อสารแบบขนาน (Parallel Communication) 15
2.6.2 การสื่อสารแบบอนุกรม (Serial Communication) 15
2.6.3 ทฤษฎีพื้นฐานการสื่อสารแบบอนุกรม 16
2.6.3.1 รูปแบบของการสื่อสารแบบอนุกรม 16
2.6.3.2 ประเภทของการสื่อสารแบบอนุกรม 16
2.6.3.2.1 การสื่อสารแบบซิงโครนัส (Synchronous) 16
2.6.3.2.2 การสื่อสารแบบอะซิงโครนัส (Asynchronous) 16
2.6.4 การอินเตอร์เฟสตามมาตรฐาน RS-232 17
2.6.5 ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟส 17
2.6.6 การออกแบบตัวแปลงสัญญาณ 18
2.6.6.1 หน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS232 18
2.6.7 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม 20
2.6.8 วงจรอะแดปเตอร์ 21
2.6.9 รูปแบบของการส่งข้อมูลแบบอนุกรมและอัตราบอดในการสื่อสาร 22
2.6.10 หลักการออกแบบวงจรรับ – ส่งข้อมูลแบบอนุกรมด้วย FPGA 23
2.6.10.1 ภาคส่งข้อมูลแบบอนุกรม 23
2.6.10.2 ภาครับข้อมูลแบบอนุกรม 24

บทที่ 3 โครงสร้างการทำงานของแขนกล 27
3.1 บทนำ 27
3.2 การออกแบบระบบ 27
3.3 ส่วนประกอบของแขนกลและระบบควบคุมแขนกล 28
3.3.1 ส่วนปุ่มกดควบคุม 29
3.3.2 ส่วนประมวลผล FPGA (Field Programmable Gate Array) 30
3.3.3 ส่วนเซอร์โวมอเตอร์ในแขนกล 31
3.2.4 ส่วนซอฟต์แวร์ (โปรแกรม) และส่วนอินเตอร์เฟสบนคอมพิวเตอร์ 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 โปรแกรมควบคุมแขนกล **32**

4.1 การออกแบบโดยใช้ภาษาระดับสูง (HDL) 32

4.2 ลักษณะภาษา Verilog 32

4.2.1 ขั้นตอนออกแบบบน FPGA 32

4.3 การใช้โปรแกรม Max+Plus II และเขียนโปรแกรม
ด้วยภาษา Verilog ในการคอนโทรล DC Servo Motor 33

4.4 การออกแบบขับแขนกลด้วยโปรแกรม Verilog เชื่อมต่อ
กับคอมพิวเตอร์ผ่านพอร์ตอนุกรมสร้าง PWM 34

บทที่ 5 โปรแกรม Visual Basic **40**

5.1 การออกแบบโปรแกรมอินเตอร์เฟสด้วย Visual Basic 40

5.1.1 จุดเด่นของ Visual Basic 40

5.1.2 สร้างแอพลิเคชั่นด้วย Visual Basic 41

5.2 องค์ประกอบของวิชวลเบสิก 42

5.2.1 ส่วนประกอบของวิชวลเบสิก 42

5.2.2 ขั้นตอนการสร้างโปรแกรมประยุกต์ 42

5.3 อินเตอร์เฟสบนคอมพิวเตอร์ 42

5.3.1 คอนโทรล MSComm 42

5.4 โปรแกรม Visual Basic 6 ที่ได้ออกแบบในการควบคุมแขนกล 44

5.4.1 การใช้งานของโปรแกรมควบคุมแขนกล 44

บทที่ 6 การทดลองและผลการทดลอง **52**

6.1 การสั่งงานโดยการป้อนคำสั่งผ่านคอมพิวเตอร์ครั้งละ 1 ชุดคำสั่ง
เพื่อแสดงถึงค่า PWM ที่ส่งออกไปยังแขนกล 52

6.2 การสั่งงานโดยการป้อนคำสั่งผ่านคอมพิวเตอร์ครั้งละ 1 ชุดคำสั่ง
เพื่อตรวจสอบระยะและทิศทางการหมุนของแขนกล 56

6.3 การสั่งงานแบบหลาย ๆ ชุดคำสั่งต่อเนื่องกัน (แบบอัตโนมัติ) 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 7 สรุปผลการวิจัยและข้อเสนอแนะ 61

บรรณานุกรม 63

ภาคผนวก 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันนี้เทคโนโลยีทางด้านอิเล็กทรอนิกส์ ได้เข้ามามีบทบาทในชีวิตประจำวันของเราอย่างมากไม่ว่าจะเป็นเครื่องใช้ต่าง ๆ อาทิเช่น รถยนต์ โทรศัพท์ วิทยุ คอมพิวเตอร์ หุ่นยนต์ ฯลฯ ซึ่งนับวันก็ยิ่งมีการพัฒนาอุปกรณ์เหล่านี้ ให้สะดวกต่อการใช้งานของเราด้วยการเพิ่มฟังก์ชั่นต่าง ๆ เข้าไป ไม่ว่าจะเป็นหุ่นยนต์ที่มีการทำงานที่ซับซ้อน มีสมองกลสามารถคิดตัดสินใจด้วยตัวเอง โทรศัพท์ที่สามารถส่งข้อความหรือแม้แต่ถ่ายรูปภาพได้ ซึ่งการพัฒนาเทคโนโลยีทางด้านอิเล็กทรอนิกส์เหล่านี้ ล้วนแต่ต้องอาศัยบุคลากรผู้ซึ่งมีความสามารถที่จะคิดค้นและผลิตนวัตกรรมด้านอิเล็กทรอนิกส์ขึ้นมา ซึ่งในขณะที่ประเทศไทยยังขาดแคลนทรัพยากรบุคคลที่จะคอยผลักดันให้ความก้าวหน้าทางเทคโนโลยีอิเล็กทรอนิกส์พัฒนาไปได้อย่างรวดเร็ว และมั่นคงให้ทัดเทียมกับนานาประเทศ

ด้านอุตสาหกรรมต่าง ๆ มีการแข่งขันรุนแรงขึ้นเป็นลำดับ ส่วนใหญ่ต้องพึ่งพาเทคโนโลยีจากต่างประเทศที่มีราคาแพงและมีค่าใช้จ่ายด้านการบำรุงรักษาสูง ทำให้ต้นทุนการผลิตจึงสูงตามไปด้วย การนำแขนกลเข้ามาช่วยในงานอุตสาหกรรมจึงมีความสำคัญเป็นอย่างยิ่งต่อการพัฒนาอุตสาหกรรมภายในประเทศ และการขยายการศึกษาและการค้นคว้าเกี่ยวกับแขนกล ซึ่งแขนกลที่ได้มีการออกแบบสามารถเคลื่อนที่ได้ทั้งหมด 4 แกนและชุดมือจับวัตถุ ลักษณะของแขนกลจะประกอบด้วยดีซีเซอร์โวมอเตอร์ (DC Servo Motor) 5 ตัว ซึ่งข้อดีของดีซีเซอร์โวมอเตอร์ คือ สามารถควบคุมตำแหน่งทิศทางที่สั่งออกเป็นสัญญาณ PWM (Pulse Width Modulation) และออกแบบจัดสร้างสามารถนำไปประยุกต์ใช้ในงานการหยิบจับชิ้นงานแทนผู้ปฏิบัติ หรือพัฒนาแขนกลให้สามารถทำงานด้านอื่น ๆ ตามที่ต้องการได้

1.2 ที่มาของปริญญานิพนธ์

การประยุกต์ใช้หลักการดิจิตอลมาสร้างสัญญาณ PWM (Pulse Width Modulation) ด้วยบอร์ด FPGA (Field Programmable Gate Arrays) ทำให้สามารถสร้างสัญญาณ PWM ได้หลายช่องสัญญาณพร้อมกันและสามารถปรับช่วงความกว้างพัลส์ได้อย่างอิสระต่อกัน ดังนั้นเมื่อนำสัญญาณดังกล่าวไปขับดีซีเซอร์โวมอเตอร์ที่ติดตั้งในแกนหมุนต่างๆ ของแขนกล ก็สามารถกำหนดตำแหน่งของแขนกลได้ตามช่วงความกว้างพัลส์หรือดิวตี้ไซเคิลที่กำหนด ความเร็วในการปรับความกว้างพัลส์จะมีผลต่อความเร็วในการเคลื่อนที่ของแขนกล การสั่งงานจะใช้โปรแกรม Visual Basic 6 ที่พัฒนา เพื่อเป็นการเชื่อมต่อกับบอร์ด FPGA ในการควบคุมแขนกลไปยังตำแหน่งที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาระบบกลไกการทำงานของแขนกล และสามารถนำไปประยุกต์ใช้ในอุตสาหกรรมได้
2. ศึกษาและสร้างสัญญาณ PWM เพื่อนำมาขับดีซีเซอร์โวมอเตอร์
3. ศึกษาโครงสร้างหลักการทำงานของดีซีเซอร์โวมอเตอร์
4. ศึกษาการใช้งานโปรแกรม Max+plus II 10.2 Baseline และโปรแกรม Quartus II 7.2 sp2 Programmer เพื่อใช้ในการสร้างสัญญาณ PWM ร่วมกับบอร์ด FPGA
5. ศึกษาคำสั่งต่าง ๆ ในภาษา Verilog HDL เพื่อใช้ในการเขียนโปรแกรมใน Max+plus II 10.2 Baseline
6. สร้างโปรแกรมแสดงผลการอินเตอร์เฟสระหว่างคอมพิวเตอร์กับบอร์ด FPGA เพื่อการสั่งงานแขนกลผ่านโปรแกรม Visual Basic 6

1.4 ขอบเขตของปริญญานิพนธ์

1. ออกแบบ ควบคุม และจัดสร้างอุปกรณ์ชุดควบคุมดีซีเซอร์โวมอเตอร์ที่ใช้ในแขนกล
2. เขียนโปรแกรมเพื่อใช้ควบคุมดีซีเซอร์โวมอเตอร์
3. วิเคราะห์และแก้ไขปัญหาที่เกิดขึ้นจากการทดลอง

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาวัตถุประสงค์และขอบเขตการทำงานของโครงงาน
2. ศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องกับโครงงาน
3. ศึกษาโครงสร้างของหลักการของดีซีเซอร์โวมอเตอร์และหลักการสร้างสัญญาณ PWM
4. ศึกษาและสร้างวงจรรับ-ส่งข้อมูลอนุกรม และโปรแกรมอินเตอร์เฟส Visual Basic 6
5. ทำการทดลองและศึกษาผลทดลองที่ได้และนำผลทดลองที่ได้มาวิเคราะห์
6. สรุปผลการทดลองรวมถึงปัญหาและอุปสรรคในการทำงาน

1.6 ผลที่คาดว่าจะได้รับ

เข้าใจในระบบกลไกและการทำงานของแขนกล พัฒนาแขนกลในการทำงานให้ดีขึ้นเพื่อนำไปประยุกต์ใช้ต่อไปได้ เข้าใจการเขียนโปรแกรมเพื่อการพัฒนางานด้วยบอร์ด FPGA และการเชื่อมต่อผ่านโปรแกรม Visual Basic 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐานและการควบคุมแขนกล

2.1 แนวความคิด

การประยุกต์ใช้หลักการดิจิตอลมาสร้างสัญญาณ PWM (Pulse Width Modulation) ด้วย FPGA (Field Programmable Gate Arrays) ทำให้สามารถสร้างสัญญาณ PWM ได้หลายช่องสัญญาณพร้อมกันและสามารถปรับช่วงความกว้างพัลส์ได้อย่างอิสระต่อกัน ดังนั้นเมื่อนำสัญญาณดังกล่าวไปขับดีซีเซอร์โวมอเตอร์ที่ติดตั้งภายในแกนหมุนต่าง ๆ ของแขนกล ซึ่งจะทำให้สามารถกำหนดตำแหน่งของแขนกลได้ตามช่วงความกว้างพัลส์หรือดิวตี้ไซเคิลที่กำหนด ความเร็วในการปรับความกว้างพัลส์จะมีผลต่อความเร็วในการเคลื่อนที่ของแขนกล

2.2 หลักการ

การควบคุมการขับเคลื่อนดีซีเซอร์โวมอเตอร์สามารถทำได้ โดยการป้อนสัญญาณพัลส์ที่มีความกว้างแตกต่างกัน เพื่อกำหนดทิศทางและตำแหน่งในการหมุน ส่วนความเร็วในการหมุนของดีซีเซอร์โวมอเตอร์จะแปรผันตามอัตราการเปลี่ยนแปลงของช่วงความกว้างพัลส์

สร้างสัญญาณมอดูเลตตามความกว้างพัลส์ (Pulse Width Modulation PWM) ด้วยเทคนิคทางดิจิตอลบน FPGA (Field Programmable Gate Array) โดยการใช้วงจรนับและวงจรเปรียบเทียบทางดิจิตอล ที่พัฒนาในรูปของซอฟต์แวร์ Verilog HDL (Verilog Hardware Description Language) โดยการออกแบบให้วงจรนับสัญญาณนาฬิกาตามเวลาที่เปลี่ยนแปลงไป และผลของวงจรนับดังกล่าว ซึ่งจะทำให้เกิดค่าของสัญญาณดิจิตอลที่มีลักษณะเป็นขั้นบันไดเปลี่ยนแปลงตามเวลาค่าทางดิจิตอลที่เป็นลักษณะขั้นบันได และจะถูกนำไปเปรียบเทียบกับระดับสัญญาณอ้างอิงที่กำหนดมาจากวงจรนับอีกวงจรหนึ่ง ผลของการเปรียบเทียบของสัญญาณทั้งสองสัญญาณนี้จะทำให้ได้สัญญาณพัลส์ที่มีค่าดิวตี้ไซเคิลตามระดับสัญญาณอ้างอิง ซึ่งสามารถปรับเปลี่ยนค่าได้ในช่วง 0-100%

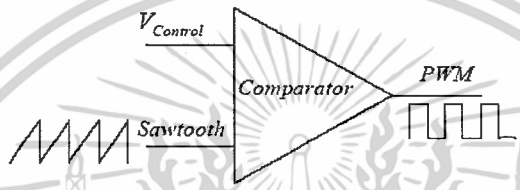
การสร้างสัญญาณ PWM ที่สามารถปรับค่าดิวตี้ไซเคิลของสัญญาณสามารถทำได้ทั้งวงจรอนาล็อกและวงจรดิจิตอล ซึ่งการใช้วิธีทางดิจิตอลมาสร้างสัญญาณที่เรียกว่า Digital PWM จะช่วยลดสัญญาณรบกวนที่เกิดจากตัววงจรอนาล็อก แต่การใช้วงจรดิจิตอลเพื่อสร้างสัญญาณที่ความละเอียดสูง ๆ จะพบปัญหาเกี่ยวกับจำนวนบิตของอุปกรณ์ที่ต้องใช้จำนวนบิตที่สูงวงจรมีขนาดใหญ่ และการสร้างสัญญาณ PWM ด้วยเทคนิคทางดิจิตอลโดยการประยุกต์ใช้บอร์ด FPGA มาเป็นตัวประมวลผลและสร้างสัญญาณผ่านซอฟต์แวร์ Verilog HDL เนื่องจากใช้พื้นที่หน่วยความจำน้อย-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กว่าการเขียนวงจรแบบกราฟิกง่ายต่อการปรับปรุงเปลี่ยนแปลงค่าสัญญาณ หรือขยายช่องสัญญาณ และสร้างสัญญาณได้หลายช่องสัญญาณพร้อมกันได้

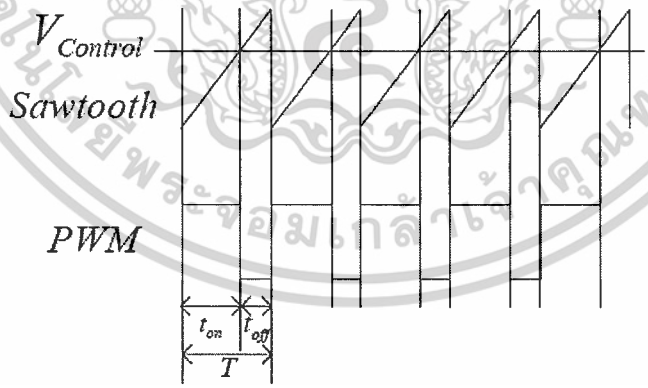
2..2.1 การกำเนิดสัญญาณ PWM (Pulse Width Modulation)

การสร้างสัญญาณ PWM ด้วยวงจรอนาล็อกจะใช้การสร้างสัญญาณฟันเลื่อยหรือสัญญาณแบบขั้นบันไดมาเปรียบเทียบกับระดับอ้างอิงที่เป็นระดับสัญญาณกระแสตรง หรือสัญญาณซายน์ตามรูปแบบของสัญญาณ PWM ที่ต้องการ การสร้างสัญญาณ PWM แบบอนาล็อกแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 วงจร PWM แบบอนาล็อก

จากรูปสัญญาณ Sawtooth จะถูกเปรียบเทียบกับสัญญาณ Vcontrol ผลของการเปรียบเทียบทำให้ได้สัญญาณ ตามรูปที่ 2.2

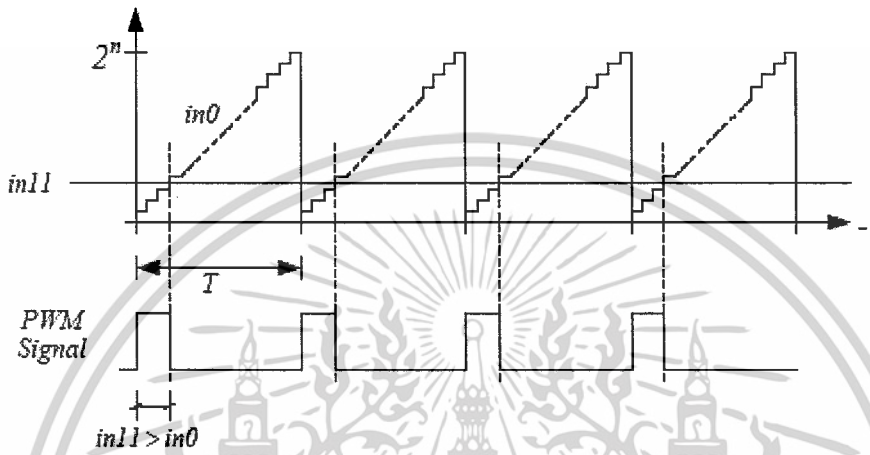


รูปที่ 2.2 การสร้างสัญญาณ PWM

รูปที่ 2.2 แสดงสัญญาณ PWM ที่ได้จาก สัญญาณ Sawtooth กับ Vcontrol ที่เปรียบเทียบกัน ผลของการเปรียบเทียบ เมื่อระดับของสัญญาณ Vcontrol สูงกว่าระดับของสัญญาณ Sawtooth จะทำให้ได้ระดับของ PWM เป็น High แต่เมื่อระดับของสัญญาณ Vcontrol ต่ำกว่าระดับของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

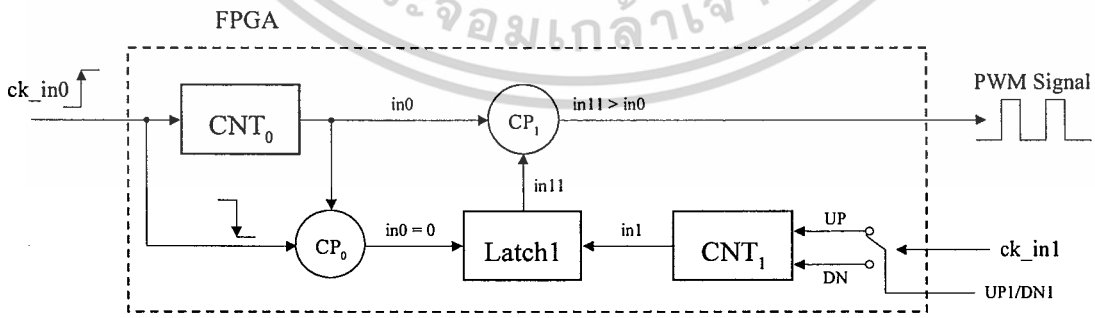
Sawtooth จะทำให้ได้ระดับของ PWM เป็น Low การสร้างสัญญาณ PWM ด้วยวงจรดิจิตอลจะใช้วงจรนับเป็นตัวนับสัญญาณนาฬิกา in0 ที่ เข้ามาซึ่งค่าที่นับได้จะมีค่าเพิ่มขึ้นเป็นรูปขั้นบันไดตามเวลาที่เปลี่ยนแปลงไป และนำสัญญาณขั้นบันไดดังกล่าว มาเปรียบเทียบกับระดับสัญญาณอ้างอิง in11 ที่กำหนด ผลของการเปรียบเทียบทำให้ได้สัญญาณ PWM เป็น ระดับลอจิก High และ Low ตามรูปที่ 2.3



รูปที่ 2.3 การสร้างสัญญาณ PWM ด้วยหลักการทางดิจิตอล

2..2.2 การออกแบบการสร้างสัญญาณ PWM

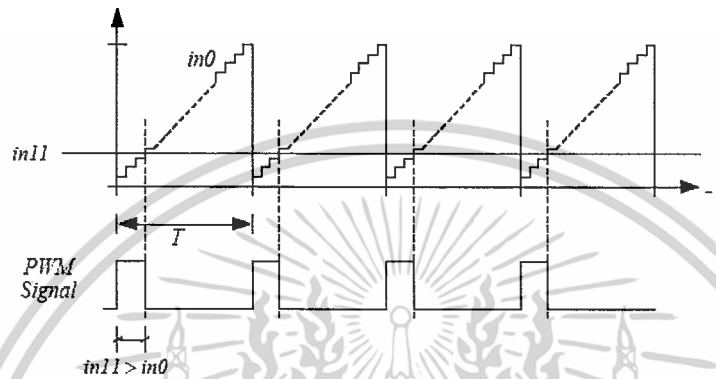
ระบบที่ใช้จะออกแบบให้สัญญาณนาฬิกา ck_in0 ป้อนเข้าสู่วงจรนับ CNT_0 ซึ่งภายในโครงงานนี้ใช้เป็นวงจรนับขึ้นขนาด 14 บิตสัญญาณเอาท์พุต in0 จะถูกนำไปเปรียบเทียบที่ CP_1 กับค่า in11 ที่ได้จากวงจรนับ CNT_1 ทุก ๆ รอบการนับที่ in0 = 0



รูปที่ 2.4 ระบบที่ได้ออกแบบไว้

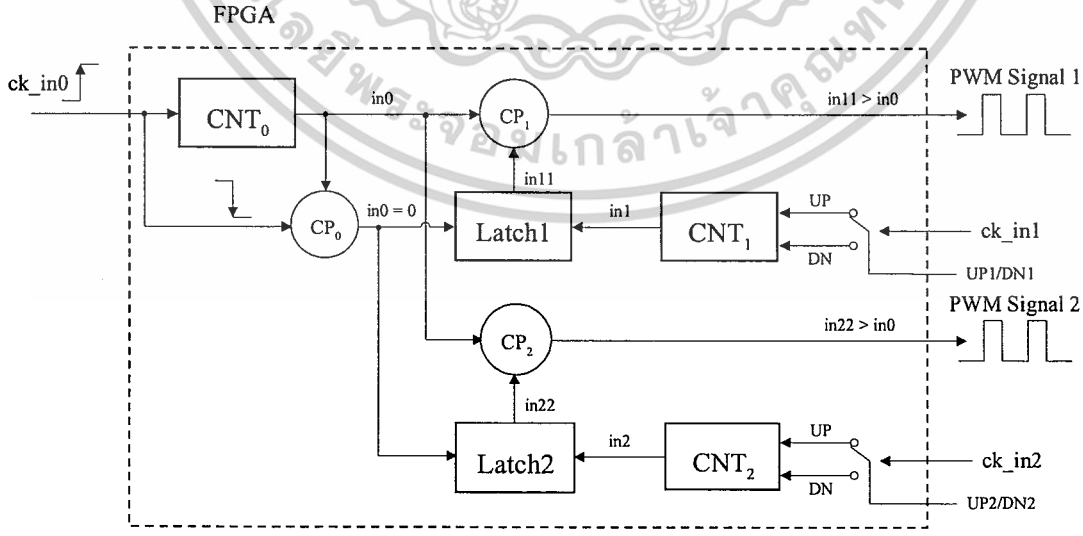
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรนับ CNT_1 สามารถเลือกให้นับขึ้นหรือลงได้ เพื่อกำหนดระดับสัญญาณอ้างอิงไปยังวงจร Latch1 เมื่อ in0 = 0 เพื่อรอการเปรียบเทียบกับค่าที่ได้จากวงจรนับ CNT_0 ซึ่งผลที่ได้จากการนำค่าของวงจรนับทั้งสองมาเปรียบเทียบกันทุก ๆ รอบ in0 = 0 นั้นจะได้ผลของการเปรียบเทียบ 3 ค่าคือ มากกว่า เท่ากับ และน้อยกว่า ในโครงงานนี้จะนำผลของสัญญาณที่ได้จากการเปรียบเทียบที่ค่า in11 มากกว่า in0 มาเป็นเอาท์พุตที่เป็นลอจิก High ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 สัญญาณ PWM ที่ได้จากระบบที่ออกแบบไว้

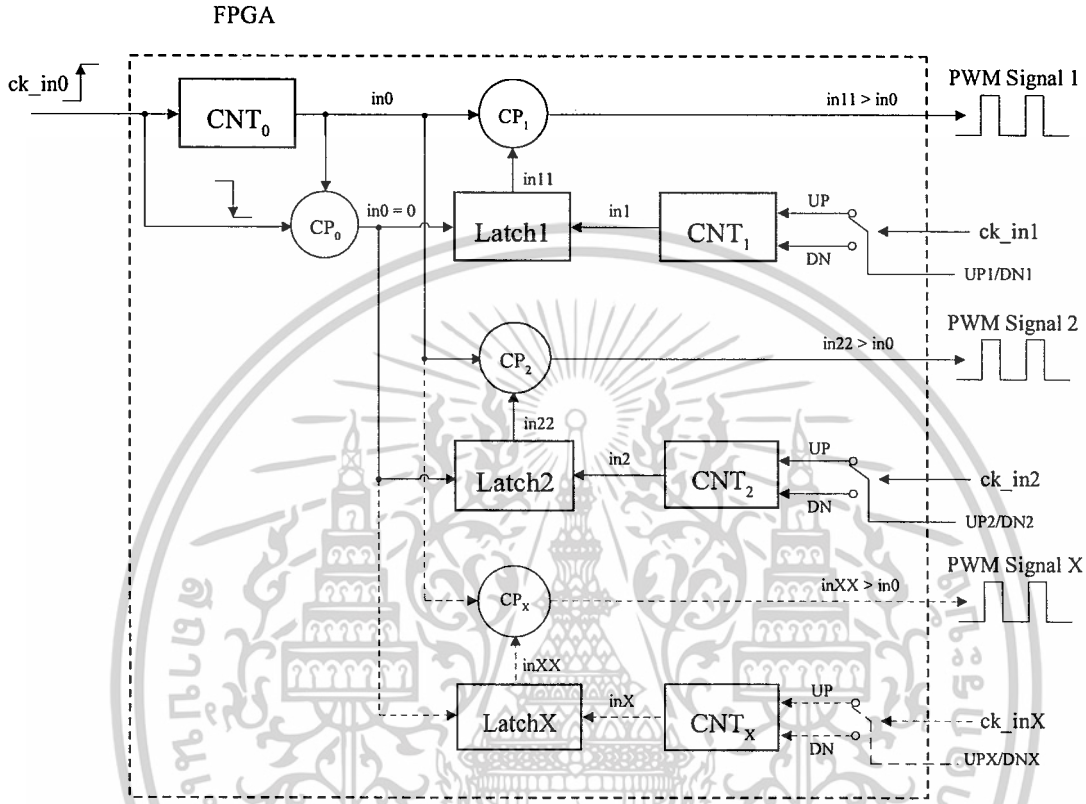
จากรูปที่ 2.5 สามารถปรับความกว้างของพัลส์ หรือดิวตี้ไซเคิลได้โดยการปรับระดับของค่า in11 และสามารถปรับความถี่ของสัญญาณ PWM ได้ด้วยการปรับค่าความถี่ของสัญญาณนาฬิกา ck_in0 เพื่อให้มีรอบการนับต่อคาบเวลาน้อยลง ในกรณีที่ต้องการเพิ่มช่องสัญญาณ PWM เป็น 2 ช่องสัญญาณสามารถทำได้ดังรูปที่ 2.6



รูปที่ 2.6 ระบบสำหรับกำเนิดสัญญาณ PWM พร้อมกัน 2 ช่องสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะใช้วงจรนับ CNT_0 เป็นตัวกำหนดความถี่ของสัญญาณ PWM และใช้ ck_in1 ปรับค่าดิวตี้ไซเคิลของ PWM Signal1 ใช้ ck_in2 ปรับค่าดิวตี้ไซเคิลของ PWM Signal2 การขยายช่องสัญญาณหรือเพิ่มจำนวนช่องสัญญาณเป็นหลายช่องสัญญาณสามารถทำได้ดังรูปที่ 2.7



รูปที่ 2.7 ระบบการสร้างสัญญาณ PWM แบบหลายช่องสัญญาณ

สัญญาณ PWM ที่ส่งไปขับมอเตอร์ในแต่ละแกนจะถูกกำหนดช่วงความกว้างพัลส์ตามขีดจำกัดในการเคลื่อนที่ของแต่ละแกนจากบอร์ด FPGA ซึ่งในตำแหน่งการเคลื่อนที่ของแขนกลแต่ละแกนหมุนนั้นจะขึ้นอยู่กับช่วงความกว้างของพัลส์ที่กำหนด อัตราในการเปลี่ยนแปลงช่วงความกว้างพัลส์จะกำหนดความเร็วในการเคลื่อนที่ เพื่อเป็นการเปลี่ยนตำแหน่งของแขนกลการกำหนดหรือการเปลี่ยนแปลงค่า ซึ่งจะใช้การส่งสัญญาณพัลส์เพื่อเป็นสัญญาณนาฬิกาป้อนให้กับบอร์ด FPGA ในการปรับเปลี่ยนช่วงความกว้างพัลส์ โดยที่จำนวนพัลส์นั้นจะเป็นตัวกำหนดค่าความกว้างพัลส์ (ตำแหน่งของแขนกล) และความถี่ของพัลส์จะเป็นตัวที่สามารถกำหนดความเร็วในการเปลี่ยนช่วงความกว้างพัลส์ (ความเร็วในการเคลื่อนที่ของแขน) การตั้งค่าและเปลี่ยนแปลงค่าจะใช้การแก้ไขจากตัวโปรแกรมเพื่อสร้างสัญญาณพัลส์ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 มอเตอร์ไฟฟ้ากระแสตรงแบบดีซีเซอร์โวมอเตอร์ (DC Servo Motor)

2.3.1 ดีซีเซอร์โวมอเตอร์ (DC Servo Motor)

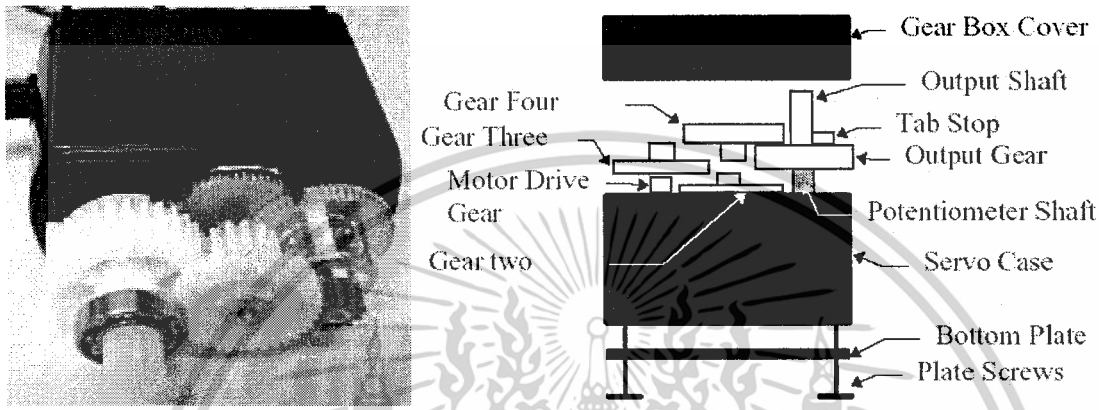


รูปที่ 2.8 ดีซีเซอร์โวมอเตอร์

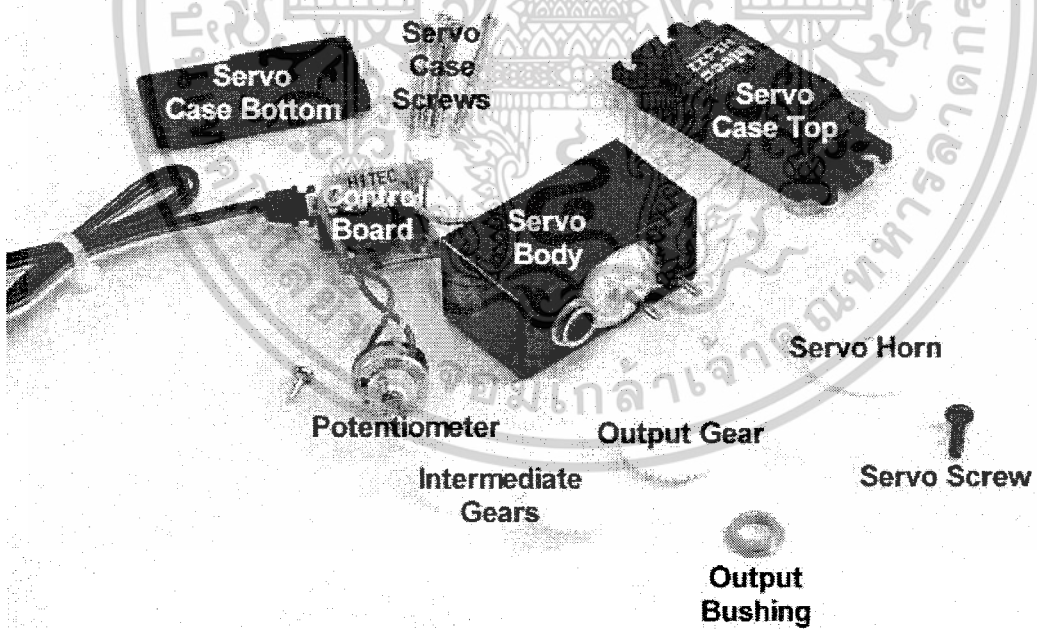
DC Servo Motor คือมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) ที่ภายในถูกประกอบด้วยชุดเกียร์ และส่วนควบคุมต่างๆไว้ภายในโมดูลเดียวกัน โดยมีสัญญาณในการควบคุมการหมุนของดีซี-เซอร์โวมอเตอร์ (Control Line) 1 เส้น และอีก 2 เส้นจะเป็น VCC และ GND เท่านั้น ซึ่งสามารถควบคุมให้ดีซีเซอร์โวมอเตอร์หมุนซ้ายหรือหมุนขวาได้ +90 องศา ถึง -90 องศา (กรณีที่ดีซีเซอร์โวมอเตอร์หมุนได้ 180 องศา) ซึ่งสามารถสั่งงานในการหมุนให้หมุนไปได้ตามองศาต่าง ๆ ที่ต้องการ สัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณ PWM แบบ TTL Level ระดับแรงดันที่จ่ายให้กับมอเตอร์นี้จะอยู่ภายในช่วงประมาณ 4 - 6 โวลท์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้ก็คือ จะมีขนาดเล็กน้ำหนักเบา, ให้แรงบิดสูง, กินพลังงานน้อย และสามารถควบคุม ด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรงไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่น ๆ เพราะตัวมอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุนไปในตำแหน่งหรือทิศทางองศาที่ต้องการได้ โดยอาศัยสัญญาณความกว้างพัลส์ที่ป้อนให้กับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ดีซีเซอร์โวมอเตอร์นี้จะหมุนได้แค่เพียงในช่วงประมาณ 180 องศาหรือครึ่งรอบเท่านั้น บางรุ่นอาจหมุนได้ถึง 210 องศา แต่จะไม่สามารถหมุนเป็นวงรอบได้ เนื่องจากโครงสร้างภายในนั้นจะประกอบด้วยตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของมอเตอร์ โดยตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ และการกำหนดด้วยชุดแมคคา-นิกส์ที่แกนเพลา ดังนั้นเซอร์โวมอเตอร์ที่เลือกใช้นี้จึงหมุนได้ 180 องศาหรือครึ่งรอบเท่านั้น



รูปที่ 2.9 ชุดเฟืองเกียร์ของดีซีเซอร์โวมอเตอร์



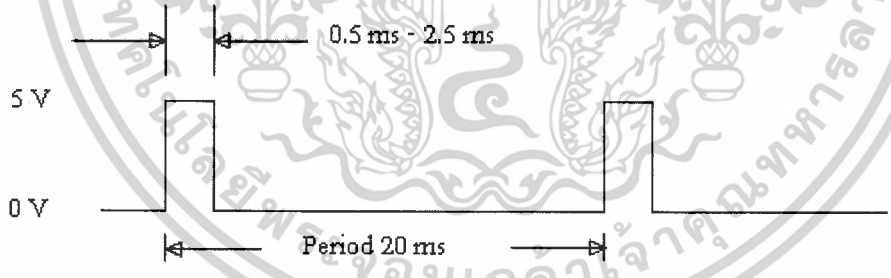
รูปที่ 2.10 ส่วนประกอบต่าง ๆ ของดีซีเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 หลักการทำงานของดีซีเซอร์โวมอเตอร์

การควบคุมการทำงานของดีซีเซอร์โวมอเตอร์นั้น จะถูกควบคุมด้วยวงจรอิเล็กทรอนิกส์และตัววัดระดับไฟฟ้า (Potentiometer) ซึ่งปลายของมอเตอร์จะเชื่อมต่อกับแกนที่ยื่นออกมาผ่านชุดเฟือง และส่วนหนึ่งของชุดเฟืองก็จะทำการหมุนแกนของ Potentiometer ซึ่ง Potentiometer จะคอยตรวจสอบตำแหน่งเชิงมุมของเซอร์โวในขณะนั้น หรือตำแหน่งการทำงานของมอเตอร์นั่นเอง เมื่อแกนของเซอร์โวหมุนมาอยู่ในตำแหน่งที่ถูกต้อง มอเตอร์ของเซอร์โวก็จะถูกตัดการทำงาน และถ้าวงจรตรวจพบว่ามุมของแกนเซอร์โวไม่ถูกต้อง วงจรจะสั่งงานให้มอเตอร์หมุนเรื่อย ๆ จนมุมของ เซอร์โวอยู่ในตำแหน่งที่ถูกต้อง แกนของเซอร์โวนั้นสามารถหมุนได้เป็นมุมมากถึง 180 องศา และ เซอร์โวบางรุ่นสามารถหมุนได้เป็นมุมกว้างถึง 210 องศา ซึ่งจะแตกต่างกันไปตามผู้ประดิษฐ์ได้ออกแบบมา แต่โดยทั่วไปแล้วเซอร์โวที่ถูกเลือกใช้สำหรับควบคุมการเคลื่อนไหวจะอยู่ระหว่าง 0 – 180 องศา ซึ่งเซอร์โวมอเตอร์ทั่วไปจะไม่สามารถหมุนทำมุมได้มากกว่านี้ เนื่องจากถูกควบคุมที่ชุดเฟืองเกียร์

เราสามารถควบคุมการทำงานของดีซีเซอร์โวมอเตอร์ได้ โดยป้อนการป้อนสัญญาณความ-กว้างของพัลส์ให้กับมอเตอร์ ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้น ๆ ความกว้างของสัญญาณพัลส์นี้จะไปกำหนดตำแหน่งของ Servo Motor ว่าหมุนไปอยู่ตำแหน่งใด ซึ่งเราสามารถป้อนความกว้างพัลส์ ได้ตั้งแต่ 0.5 ms – 2.5 ms โดยที่คาบเวลา (Period) ในการส่งสัญญาณความกว้างพัลส์ เท่ากับ 20 ms ดังรูปที่ 2.11



รูปที่ 2.11 สัญญาณ PWM ในการควบคุมตำแหน่งดีซีเซอร์โวมอเตอร์

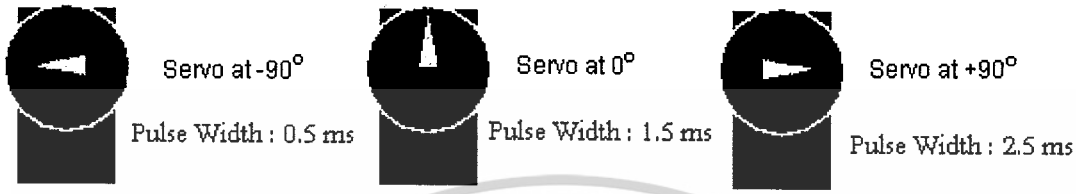
การควบคุมการทำงานใช้ความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังรูปที่ 2.12 และสามารถอธิบายขั้นตอนการทำงานดังต่อไปนี้

1. ป้อนสัญญาณความกว้างพัลส์ขนาด 0.5 ms จะควบคุมให้ดีซีเซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม - 90 องศา หรือในทิศทางทวนเข็มนาฬิกา (ซ้ายสุด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ป้อนสัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้ดีซีเซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือจุดกึ่งกลางของมอเตอร์ (ตรงกลาง)

3. ป้อนสัญญาณความกว้างพัลส์ขนาด 2.5 ms จะควบคุมให้ดีซีเซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 90 องศา หรือในทิศทางตามเข็มนาฬิกา (ขวาสุด)

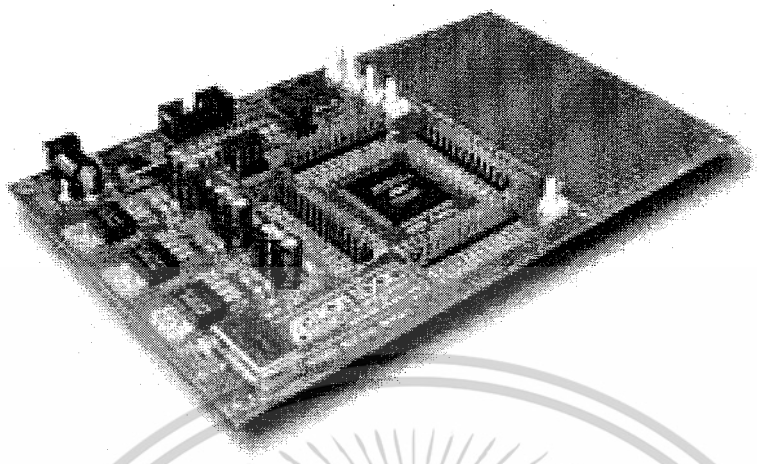


รูปที่ 2.12 ตำแหน่งการหมุนของดีซีเซอร์โวมอเตอร์ที่ -90 องศา ถึง 90 องศา

ส่วนการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมอื่น ๆ นั้นก็สามารถทำได้ โดยสามารถทำการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่าง ๆ โดยอ้างอิงจากจุด ทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่างเช่น ถ้าต้องการให้มอเตอร์หมุนไปที่มุม -45 องศา เราก็จะต้องป้อนสัญญาณพัลส์ที่มีความกว้างขนาด [(0.5 + 1.5) / 2] = 1.0 ms เป็นต้น และสัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุก ๆ 20 ms (Period) เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้ โดยหลักการก็คือจะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างของพัลส์ที่จ่ายให้กับตัวมอเตอร์ทางขาสัญญาณควบคุมกับค่าเวลาของวงจร RC ภายในบอร์ดควบคุมในตัวของมอเตอร์ ซึ่งค่าเวลาของวงจร RC นี้จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์เนื่องจากตัวต้านทานปรับค่าจะถูกยึดติดอยู่กับแกนหมุนของมอเตอร์ ซึ่งในการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่า (VR) เปลี่ยนแปลงเป็นผลทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงตามไปด้วย โดยขณะที่เราป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ทางขาสัญญาณควบคุม สัญญาณนี้จะถูกนำไปเปรียบเทียบกับค่าเวลาของวงจร RC หากค่าทั้ง 2 ไม่เท่ากัน มอเตอร์ก็จะหมุนทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของวงจร RC เปลี่ยนแปลงจนเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control line) มอเตอร์จึงจะหยุดหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 อุปกรณ์ FPGA (Field Programmable Gate Array)



รูปที่ 2.13 บอร์ด FPGA

อุปกรณ์ FPGA (Field Programmable Gate Array) เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรที่ได้ออกแบบลงไป เพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามที่ผู้ออกแบบต้องการ ในการใช้งาน FPGA เมื่อเทียบกับการทำ ASICs (Application Specific Integrated Circuits) แล้วนั้นก็มีทั้งข้อดีและข้อเสีย คือ การใช้งาน FPGA จะมีข้อจำกัดในด้านขนาดของวงจร เพราะภายในอุปกรณ์ FPGA จะมีจำนวน gate ให้ใช้จำนวนจำกัด และการใช้งาน FPGA ก็เหมาะกับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการใช้งาน FPGA ก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียน code อธิบายฮาร์ดแวร์ จนกระทั่ง download นั้นน้อยกว่าการทำ ASIC มาก และการตรวจสอบหรือแก้ไข design ก็ทำได้สะดวก การใช้งาน FPGA ในปัจจุบันมีประสิทธิภาพมากขึ้นและสะดวกขึ้น ทั้งนี้ก็เนื่องจากทางบริษัทผู้ผลิตอุปกรณ์ FPGA ได้เพิ่มความสามารถของอุปกรณ์ FPGA โดยเพิ่มจำนวนองค์ประกอบภายใน หรือปรับปรุงโครงสร้างสถาปัตยกรรมภายใน และยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้ทำ PPR (Partitioning, Placement and Routing) สำหรับอุปกรณ์นั้น ๆ ด้วย

โครงสร้างภายในของ FPGA นั้นสามารถโปรแกรมให้มีหน้าที่การทำงานเหมือนลอจิกเกตพื้นฐาน เช่น AND, OR, XOR, NOT หรือรวมกันหลาย ๆ ชนิด (combinational logic) เพื่อให้ทำหน้าที่ที่มีความซับซ้อนเพิ่มมากขึ้น เช่น decoders หรือฟังก์ชันทางตณิตศาสตร์ใน FPGAs ทั่วไปนอกจากจะประกอบด้วยส่วนของวงจรลอจิกแบบโปรแกรมได้แล้วจะยังมีบล็อกของหน่วยความจำซึ่งอาจจะสร้างด้วยฟลิบฟลอปอย่างง่าย หรือใช้พื้นที่ของสารกึ่งตัวนำสร้างเป็นหน่วยความจำจริง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ภายในก็ได้ การออกแบบวงจรดิจิตอลอิเล็กทรอนิกส์ที่มี FPGA อยู่บนแผงวงจรด้วยนั้นจะช่วยให้ผู้ออกแบบสามารถลดขนาดของแผงวงจร รวมทั้งสามารถออกแบบได้รวดเร็วไม่ต้องทดสอบรายละเอียดภายในให้เสร็จสมบูรณ์ 100 % ก็สามารถที่จะออกแบบแผงวงจรได้ เมื่อได้รับแผงวงจรและประกอบอุปกรณ์ต่าง ๆ เสร็จแล้ว จึงค่อยกำหนดหน้าที่การทำงานของ FPGA ได้ในภายหลัง ต่างจากการออกแบบด้วยลอจิกเกตขนาดเล็ก ที่จะต้องออกแบบทางเดินของลายทองแดงให้เสร็จสมบูรณ์ก่อนและไม่สามารถแก้ไขได้ในภายหลัง นอกจากนี้ภายในการใช้งานของ FPGA สามารถโปรแกรมการทำงานได้ในทุกขณะแม้แต่ขณะที่ส่งมอบงานแล้ว ก็ยังสามารถเข้าไปแก้ไขวงจรได้โดยง่ายดาย จึงเป็นที่มาของคำว่า "field programmable" ซึ่งก็หมายถึงโปรแกรมได้ในภาคสนามหรือที่หน้างานนั่นเอง อย่างไรก็ตามข้อกำหนด (Configuration) ของ FPGA จะหายไปหลังจากปิดไฟเลี้ยง ดังนั้นจะต้องมีหน่วยความจำภายนอก (Flash) มาคอยรักษาข้อกำหนดของ FPGA ไว้ ซึ่ง FPGA จะมีกระบวนการอ่านข้อกำหนดนั้นโดยอัตโนมัติหลังจากได้รับไฟเลี้ยง

2.4.1 การออกแบบด้วย FPGA

การออกแบบวงจรดิจิตอลด้วย FPGA โดยทั่วไปมีองค์ประกอบ 3 ส่วน

1. ซอฟต์แวร์ Design Entry

- โดยใช้ Schematic Design Entry ใช้ไลบารีของ FPGA
- ใช้ภาษา HDL เช่น VHDL, Verilog HDL

การออกแบบโดยใช้ภาษาชั้นสูงนั้น การทดสอบของวงจรยังไม่ขึ้นกับเทคโนโลยีเป้าหมาย (Technology independent) ผู้ออกแบบไม่จำเป็นต้องกังวลถึงค่าความหน่วงทางเวลาของอุปกรณ์ที่มากับเทคโนโลยีนั้น การทดสอบความถูกต้องเป็นในลักษณะการตรวจสอบระดับฟังก์ชันการทำงานโดยใช้ซอฟต์แวร์สำหรับจำลองการทำงาน (Simulation)

2. Design Implementation

ขั้นตอนนี้ต่อเนื่องจากขั้นตอนที่ 1 ซึ่งจะเกี่ยวข้องกับการแปลงแบบที่ได้ออกแบบจาก Schematic หรือ HDL ให้เป็นลอจิก ซึ่งอาจจะใช้ซอร์ฟแวร์สำหรับสังเคราะห์วงจร (Logic synthesis) แล้วทำการแบ่งลอจิกเป็นส่วนๆ (Partitioning) และวางตำแหน่ง (Placement) ของลอจิกทำการเชื่อมต่อสายสัญญาณ (routing) สุดท้ายเป็นการสร้างไฟล์สำหรับโปรแกรมลงชิป (bit file)

3. Device Programming

การโปรแกรมอุปกรณ์หรือชิป FPGA นั้น มีเทคนิคหรือวิธีใหญ่ ๆ 3 ลักษณะ ทั้งนี้ตัวชิปจะต้องสนับสนุนการทำงานในโหมดของการโปรแกรมเหล่านี้ด้วย

1. การโปรแกรมโดยผ่านสายดาวน์โหลด หรือผ่าน JTAG หรือผ่าน ISP
2. การโปรแกรมโดยใช้ ตัวเก็บข้อมูลไฟล์บิต
3. การโปรแกรมโดยใช้เครื่องโปรแกรมไอซี

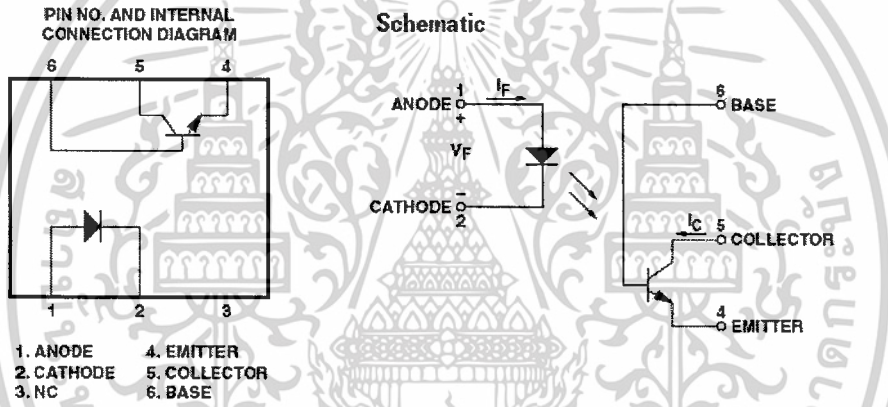
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.8 คุณสมบัติของ FPGA

คุณสมบัติของ FPGA ภายในจะประกอบไปด้วย

1. ลอจิกเซลล์ที่สามารถโปรแกรมได้
2. หน่วยความจำ RAM
3. DSP block
4. พอร์ต I/O จำนวนมาก
5. สามารถส่งข้อมูลแบบอนุกรมที่ความสูงมากกว่า 600Mb/s (Multi-gigabit transceiver)

2.5 การทำงานของวงจร Opto Electronics ในการขยายสัญญาณขับมอเตอร์



รูปที่ 2.14 รูปแบบภายในของไอซีเบอร์ 4N35

อุปกรณ์กำเนิดแสง LED (Light Emitting Diode)

LED เป็นไดโอดที่สามารถเปล่งแสงได้ในขณะที่มันนำกระแส เนื่องจาก LED เป็นไดโอดแบบหนึ่ง ดังนั้นการทำให้มันนำกระแสได้ก็ต้องให้ไบแอสเช่นเดียวกับไดโอดคือ ต่อไฟบวกเข้าที่แอโนดและต่อไฟลบเข้าที่แคโทด แต่จะต้องต่อตัวต้านทานอนุกรมให้กับ LED ด้วย เพื่อจำกัดกระแสไม่ให้ไหลผ่าน LED มากเกินไป

การใช้ IC เบอร์ 4N35 ในการขยายกระแสสัญญาณ โดยใช้ความต้านทาน 200 โอห์ม เพื่อทำให้ค่ากระแสที่ไหลผ่านไดโอดอยู่ในช่วง 5-30 mA แล้วนำสัญญาณที่ได้จากขา Emitter ไปใช้ในการขับเซอร์โวมอเตอร์ โดยต่อแรงดันไฟฟ้า 5V ที่ขา 5 ของไอซี

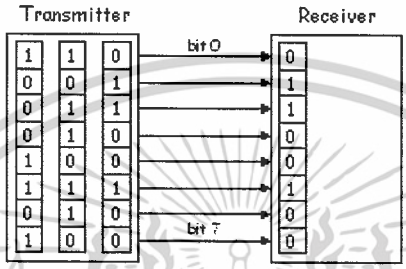
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การออกแบบวงจรรับ – ส่งข้อมูลแบบอนุกรมด้วย FPGA

การสื่อสารหรือการส่งข้อมูลในระบบดิจิตอลนั้นมีรูปแบบในการสื่อสารที่สำคัญอยู่ 2 แบบ คือ

2.6.1 การสื่อสารแบบขนาน (Parallel Communication)

การสื่อสารในรูปแบบที่ส่งข้อมูลออกไปทีเดียวพร้อม ๆ กันทั้ง 8 bits ผ่านสายสัญญาณทั้ง 8 เส้น การสื่อสารและการส่งข้อมูลแบบขนานจะแสดงให้เห็นดังรูปที่ 2.15

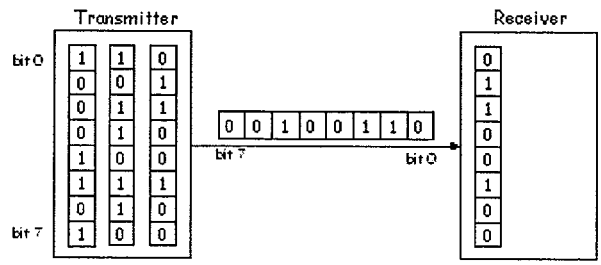


รูปที่ 2.15 ลักษณะการสื่อสารแบบขนาน

จะเห็นได้ว่าการสื่อสารแบบขนานมีข้อดีคือทำให้สามารถส่งข้อมูลทีละมาก ๆ และรวดเร็วกว่าการส่งแบบอนุกรม แต่การสื่อสารแบบขนานมีข้อจำกัดคือ ไม่สามารถส่งข้อมูลในระยะไกล ๆ ได้และยังต้องใช้สายสัญญาณหลายเส้นในการส่งข้อมูลทำให้สิ้นเปลืองกว่าการสื่อสารแบบอนุกรม ทำให้ไม่สะดวกในการใช้งาน ตัวอย่างของการสื่อสารแบบขนาน เช่น เครื่องพิมพ์ (Printer) และการสื่อสารทางพอร์ตขนาน (ECP Printer Port) เป็นต้น

2.6.2 การสื่อสารแบบอนุกรม (Serial Communication)

เป็นการสื่อสารโดยการส่งข้อมูลทีละบิท ผ่านสายสัญญาณเส้นเดียว จนครบทั้ง 8 bits หรือไบท์ โดยจะส่งบิทต่ำ (LSB) ออกไปก่อน สามารถแสดงให้เห็นหลักการส่งข้อมูล แบบอนุกรมได้ดังรูปที่ 2.16



รูปที่ 2.16 ลักษณะการส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3 ทฤษฎีพื้นฐานการสื่อสารแบบอนุกรม

ในการสื่อสารแบบอนุกรมในเครื่องคอมพิวเตอร์นั้น จะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน เพราะว่าการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้น เป็นการส่งข้อมูลทีละ 1 บิต แต่พอร์ตขนานนั้นสามารถส่งข้อมูลได้ครั้งละหลาย ๆ บิตพร้อมกัน ดังนั้นจึงทำให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน แต่ว่าข้อดีของการส่งข้อมูลแบบอนุกรมคือ สามารถส่งข้อมูลได้ระยะทางไกลกว่าแบบขนาน และอีกทั้งสายสัญญาณก็มีน้อยกว่าการส่งข้อมูลแบบขนานอีกด้วย

2.6.3.1 รูปแบบของการสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมสามารถแบ่งออกได้เป็น 3 รูปแบบ ดังนี้

1. Simplex สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว
2. Half–Duplex สามารถส่งข้อมูลไปยังปลายทาง และสามารถรับข้อมูลจากปลายทางได้แต่ไม่สามารถทำการส่งและรับข้อมูลในเวลาเดียวกันได้
3. Full–Duplex สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

2.6.3.2 ประเภทของการสื่อสารแบบอนุกรม

ประเภทของการสื่อสารแบบอนุกรมนั้นสามารถแบ่งประเภทของการสื่อสารตามลักษณะสัญญาณในการสื่อสารได้ 2 แบบ คือ

2.6.3.2.1 การสื่อสารแบบซิงโครนัส (Synchronous)

สำหรับการสื่อสารแบบซิงโครนัสนั้น จะใช้สัญญาณนาฬิกาควบคุมการรับส่งสัญญาณ เช่น สายคีย์บอร์ดคอมพิวเตอร์ โดยจะมีสายสัญญาณเส้นหนึ่งเป็นสายสัญญาณ-นาฬิกา ส่วนอีกเส้นหนึ่งเป็นสายของข้อมูล (และมักจะมีสายกราวน์ด้วย) สำหรับการสื่อสารแบบซิงโครนัสนี้เหมาะสำหรับการทำงานในระยะใกล้ข้อมูลที่จะส่งมีไม่มากนัก เพราะถ้าระยะทางไกลขึ้นจะทำให้สัญญาณนาฬิกามีปัญหาอีกทั้งต้องมีสายหลายเส้นทำให้สิ้นเปลืองมาก

2.6.3.2.2 การสื่อสารแบบอะซิงโครนัส(Asynchronous)

สำหรับการสื่อสารแบบอะซิงโครนัสคือ การรับและส่งข้อมูลไปในสายโดยไม่จำเป็นที่ต้องมีสายสัญญาณนาฬิการ่วมด้วยนั้น ซึ่งจะใช้สายสัญญาณเพียงตัวเดียวแต่จะใช้รูปแบบการส่งข้อมูล หรือ Bit Pattern เป็นตัวกำหนดว่าส่วนไหนเป็นตัวเริ่มต้นข้อมูล ส่วนไหนเป็นตัวข้อมูล ส่วนไหนจะเป็นตัวตรวจสอบความถูกต้องของข้อมูล และส่วนไหนเป็นส่วนปิดท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาครับและภาคส่ง ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ทั้งภาครับและภาคส่งว่า อัตราการถ่ายทอดข้อมูล หรือบอดเรต (baudrate) มีหน่วยเป็นบิตต่อวินาที (Bit Per Second : bps) ซึ่งจะมีอุปกรณ์พิเศษที่เรียกว่า UART หรือ Universal Asynchronous Reciever/Trasmitter คอยควบคุมการรับ และการส่งข้อมูลที่ได้รับความนิยมอย่างสูงตั้งแต่อดีตจนถึงปัจจุบันนี้ก็คือ มาตรฐาน RS -232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.6.4 การอินเตอร์เฟสตามมาตรฐาน RS-232

มาตรฐาน RS-232 เป็นมาตรฐานที่ได้รับการพัฒนามายาวนาน และถูกใช้งานกันอย่างแพร่หลาย เราใช้ RS-232 ในยุคแรก ๆ เชื่อมต่อ DTE (Data Terminal Equipment) เป็นอุปกรณ์สำหรับส่งข้อมูล (output) เข้ากับ DCE (Data Communication Equipment) เป็นอุปกรณ์สำหรับรับข้อมูล (Input) เช่น การต่อเทอร์มินัลเข้ากับโมเด็มมาตรฐาน RS-232 ซึ่งเป็นมาตรฐานที่ได้รับการออกแบบมา เพื่อที่จะทำให้อุปกรณ์ต่อพ่วงจากผู้ผลิตต่างกันสามารถทำงานร่วมกันได้ ทั้งนี้เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลบนสายเดียวกัน และกล่าวถึงลักษณะทางกล, ลักษณะของสัญญาณ-ไฟฟ้า และลักษณะการทำงานที่ใช้การอินเทอร์เฟส ตัวอย่างของอุปกรณ์ที่ใช้ในการอินเทอร์เฟสตามมาตรฐาน RS-232 ได้แก่ เทอร์มินัล, พล็อตเตอร์, ลอจิกอนาไลเซอร์ (Logic Analyzer) และเครื่องพิมพ์ ถ้าในการประยุกต์ใช้งานของเราต้องการทำอินเทอร์เฟสอุปกรณ์เข้ากับอินเทอร์เฟสมาตรฐาน RS-232 เราจำเป็นต้องแปลงระดับสัญญาณ TTL ให้เป็นระดับสัญญาณแบบอื่นมาตรฐานของ RS-232 ที่มีการปรับปรุงแก้ไขจากมาตรฐานเดิม ซึ่งเราอาจคุ้นเคยกับชื่อนี้มากกว่า RS-232-A หรือ RS-232-B ในอันที่จริงแล้วยังมีมาตรฐาน RS-232-D ที่ใหม่กว่า RS-232-C โดยที่มีการเพิ่มข้อกำหนดของคอนเน็กเตอร์แบบ DB เข้าไปด้วย เช่น DB-25 ซึ่งในขณะนั้นสิทธิบัตรของตัวคอนเน็กเตอร์แบบนี้ได้หมดอายุลงพอดีจึงสามารถรวมข้อกำหนดเข้าไว้ได้ ลักษณะโดยทั่วไปของการเชื่อมต่อข้อมูลแบบอนุกรมตามมาตรฐาน RS-232 คือเป็นการสื่อสารข้อมูลแบบจุดต่อจุด ซึ่งในเดิมทีเป็นการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับโมเด็ม ซึ่งจริง ๆ แล้วทั้งสองฝั่งจะเป็นอะไรก็ได้การสื่อสารเป็นแบบสองทางพร้อมกัน (Full-duplex) โดยอาจจะใช้สายสัญญาณอื่นร่วมเพื่อทำแฮนด์เชค (Hand-shake) หรือไม่ก็ได้ มาตรฐาน RS-232 จำกัดความยาวสายไว้ที่ 50 ฟุต (หรือประมาณ 15 เมตร) สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิทต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้น และถ้ามีสัญญาณรบกวนมาก ๆ เช่น ในโรงงานหรือบริเวณที่ใกล้เครื่องจักรที่เป็นแบบมีการสวิทช์สัญญาณไฟฟ้าที่กระแสสูง ๆ ก็จะทำให้ต้องมีการลดความเร็วในการส่งสัญญาณลงหรือใช้สายที่สั้นลง

2.6.5 ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟส

มาตรฐาน RS-232 ใช้สัญญาณเพียงเส้นเดียวในการส่งสัญญาณ โดยที่สัญญาณจะส่งไปในทิศทางเดียวกัน ในกรณีที่อัตราเร็วในการส่งข้อมูลมีค่าเท่ากับ 20 kbps (กิโลบิทต่อวินาที) ซึ่งค่านี้เป็นค่าสูงสุดที่ใช้ในการสื่อสารข้อมูล (ซึ่งภายในปัจจุบันพัฒนาให้สามารถส่งข้อมูลได้มากกว่านี้) ระยะทางในการส่งข้อมูลไม่ควรเกิน 50 ฟุต (ตามข้อกำหนดในมาตรฐาน) สำหรับการแทนแรงดันของระดับสัญญาณ จะแทนระดับสัญญาณของลอจิก “0” ด้วยค่าแรงดัน +3 โวลท์ ถึง +12 โวลท์ ส่วนลอจิก “1” จะแทนระดับสัญญาณด้วยค่าแรงดันระหว่าง –3 โวลท์ ถึง –12 โวลท์

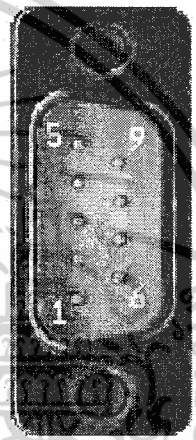
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
104107

2.6.6 การออกแบบตัวแปลงสัญญาณ

การเชื่อมต่อกับพอร์ตสื่อสารของคอมพิวเตอร์ส่วนบุคคลนั้น จะเลือกใช้พอร์ตสื่อสารแบบอนุกรม 9 ขา (DB-9) ซึ่งสามารถจะทำการรับส่งข้อมูลได้แบบอนุกรม โดยลักษณะของสัญญาณเป็นไปตามมาตรฐาน RS-232 โดยลักษณะของการเชื่อมต่อของพอร์ตสื่อสารสำหรับคอนเน็คเตอร์แบบ DB-9 สามารถแสดงให้เห็นได้ดังรูปที่ 2.17 และรูปที่ 2.18

ตารางที่ 2.1 ตำแหน่งขาของพอร์ต DB-9

ตำแหน่งขา DB-9	สัญญาณ
1	Data Carrier Detect : DCD
2	Received Data : RxD
3	Transmitted Data : TxD
4	Data Terminal Ready : DTR
5	Signal Ground : GND
6	Data Set Ready : DSR
7	Request To Send : RTS
8	Clear To Send : CTS
9	Ring Indicator : RI



รูปที่ 2.17 การจัดขาของคอนเน็คเตอร์พอร์ตอนุกรมแบบ DB-9

2.6.6.1 หน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS232

1. Data Carrier Detect: DCD หรืออาจจะเรียกว่า Carrier Detect: CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่นโมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ถูกใช้งานมากนัก

2. Receive Data: RD หรือ RXD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

3. Transmitted Data: TD หรือ TXD ใช้ส่งข้อมูลออกจากคอมพิวเตอร์โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

4. Data Terminal Ready: DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วยโดยที่ขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาร์

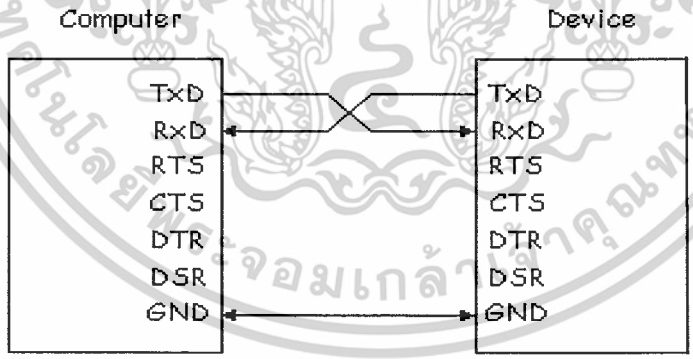
5. Signal Ground: GND กราวด์ระบบ

6. Data Set Ready: DSR ขานี้จะใช้คู่กับขา DTR เพื่อใช้ในการตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

7. Request To Send: RTS เป็นขาสำหรับการส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมาทางคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สายจะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกันเพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

8. Clear To Send: CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อได้รับสัญญาณได้ข้อมูลที่ขา TXD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

9. Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากโทรศัพท์ ปกติ-ในการสื่อสารโดยทั่วไป สายนี้นั้นจะไม่ถูกใช้งานจะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็ม และโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น



รูปที่ 2.18 การต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์โดยใช้สัญญาณเพียง 3 เส้น

2.6.7 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม

1. Start Bit (ขนาด 1 บิต) จะใส่ที่จุดเริ่มต้นเสมอเพื่อเตือนอุปกรณ์ฝ่ายรับว่าข้อมูลกำลังจะมาถึง

2. Data Character (ขนาด 7 บิต หรือ 8 บิต) การส่งบิตข้อมูลจะส่งเป็นกลุ่ม ๆ โดยทั่วไปจะส่งเป็น 7 บิต หรือ 8 บิต ซึ่งเพียงพอสำหรับการส่ง Ascii Word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

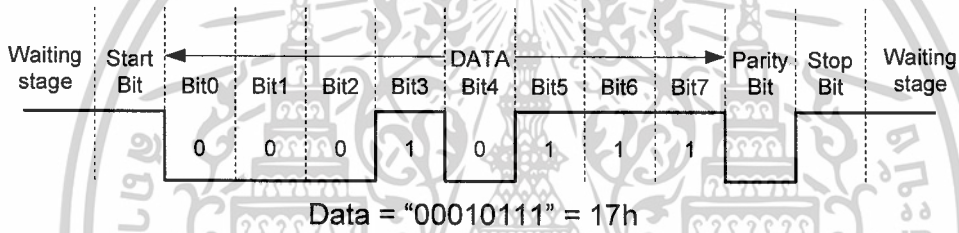
3. Parity Bit (ขนาด 1 บิต) ใช้สำหรับตรวจสอบความถูกต้องของข้อมูลที่ส่ง เราจะใส่บิตพาริตี้เข้าไปบิตพาริตี้มีหลายแบบดังนี้

- พาริตี้คู่ (Even Parity) ซึ่งค่าของบิตพาริตี้นี้เมื่อรวมกันทุก ๆ บิตของข้อมูลแล้วจะต้องมีจำนวนบิตที่เป็นเลข 1 เป็นเลขคู่ตัวอย่างเช่นข้อมูล 1000111 มีเลข 1 ทั้งหมด 3 ตัว ดังนั้นบิตพาริตี้จะเป็น 0

- พาริตี้คี่ (odd Parity) ซึ่งค่าของบิตพาริตี้นี้เมื่อรวมกันทุก ๆ บิตของข้อมูลแล้วจะต้องมีจำนวนบิตที่เป็นเลข 1 เป็นเลขคี่ตัวอย่างเช่นข้อมูล 1000101 มีเลข 1 ทั้งหมด 3 ตัว ดังนั้นบิตพาริตี้จะเป็น 1

- ไม่มีพาริตี้ (None) ถ้าตั้งค่าบิตพาริตี้เป็น None ทั้งภาครับและภาคส่งจะไม่มีการตรวจสอบบิตพาริตี้

4. Stop Bit (ขนาด 1 บิต หรือ 2 บิต) เป็นบิตที่ส่งมาปิดท้ายข้อมูล



รูปที่ 2.19 รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัส

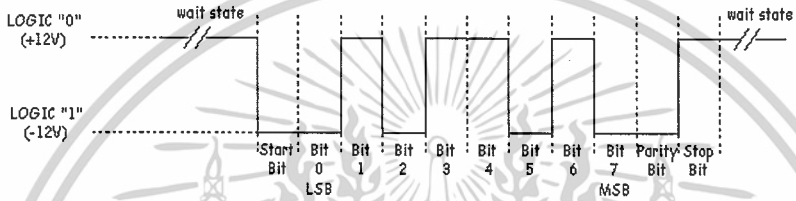
หลักการทำงานของไดอะแกรมในรูปที่ 2.19 สามารถอธิบายได้คือขณะที่ไม่มีข้อมูลส่งมานั้น ขา Data จะมีสถานะลอจิก "1" (High) ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (Waiting Stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา Data มีลอจิก "0" (Low) ด้วยช่วงขนาด 1 บิต เราเรียกบิตนี้ว่า บิตเริ่มต้น (Start Bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มส่งจากบิต LSB ก่อนจนถึงบิต MSB จากนั้นจะตามด้วยบิตพาริตี้ (Parity Bit) ซึ่งจะใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือ Stop Bit ซึ่งจะให้ขา Data มีสถานะลอจิก "1" (High) อีกครั้งเพื่อทำให้ยืนยันสถานะหยุดรอ (Waiting Stage) ด้วยขนาด1 บิต และเป็นการแสดงว่าสิ้นสุดข้อมูล

อัตราเร็วในการรับส่งข้อมูลแบบอนุกรมการที่อุปกรณ์ 2 อย่างจะติดต่อสื่อสารกันได้นั้นจะต้องทำงานด้วยอัตราเร็วเท่ากัน ซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัส คือค่าบอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที ซึ่งค่าอัตราเร็วในการสื่อสารแบบอนุกรมสำหรับมาตรฐาน RS-232c นั้นการใช้ ดังนี้ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที

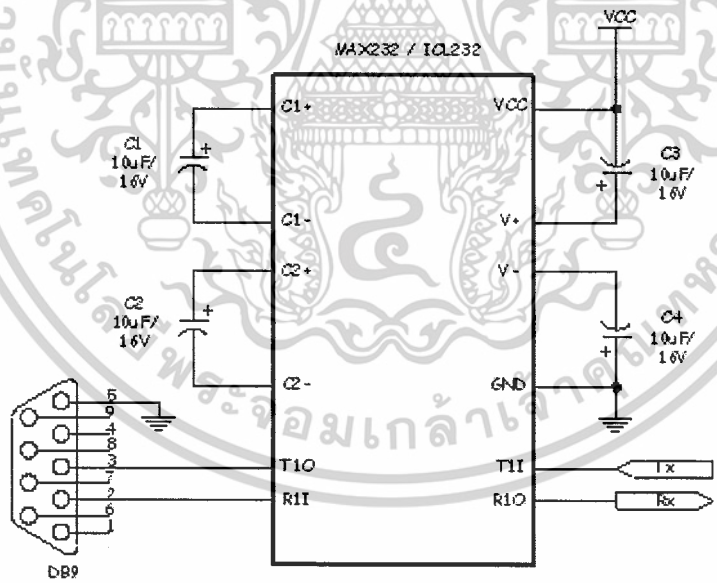
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.8 วงจรอะแดปเตอร์

จากหัวข้อที่ผ่านมาตอนนี้เราทราบแล้วว่าในการสื่อสารแบบอนุกรมนั้นในระดับของลอจิก “0” จะแทนระดับลอจิกด้วยระดับแรงดันระหว่าง +3 โวลท์ ถึง +12 โวลท์และในระดับของลอจิก “1” จะแทนระดับลอจิกด้วยระดับแรงดันระหว่าง –3 โวลท์ ถึง –12 โวลท์ เพราะฉะนั้นต้องทำการแปลงระดับของลอจิก “1” และ “0” ให้เป็นระดับแรงดันดังกล่าว ซึ่งจะต้องใช้วงจรในการแปลง-ระดับแรงดัน สามารถแสดงลักษณะการส่งข้อมูลแบบอนุกรมที่ผ่านตัวแปลงแรงดันได้ดังรูปที่ 2.20 ในส่วนของวงจรแปลงระดับแรงดันสำหรับการสื่อสาร RS-232 นั้นจะใช้ไอซีเบอร์ MAX232 หรือ ICL232 ดังรูปที่ 2.21



รูปที่ 2.20 รูปแบบของการสื่อสารแบบอนุกรม



รูปที่ 2.21 วงจรแปลงระดับแรงดันของการสื่อสารแบบอนุกรม

2.6.9 รูปแบบของการส่งข้อมูลแบบอนุกรมและอัตราบอดในการสื่อสาร

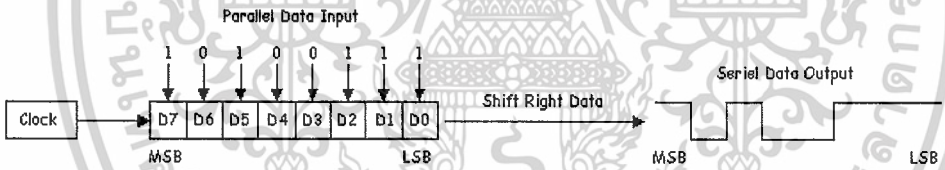
อัตราบอด (Baud Rate) คือความเร็วในการรับ - ส่งข้อมูลอนุกรมมีหน่วยเป็นบิตต่อวินาที ซึ่งจะบอกจำนวนบิตที่รับ - ส่งในเวลา 1 วินาที เช่น ส่งข้อมูลด้วยอัตรา 9600 บิตต่อวินาที ก็คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลตัวอักขระขนาด 10 บิต (บิต Start 1 บิต บิตข้อมูล 8 บิต บิต Stop 1 บิต) ได้ 960 ตัวอักขระภายใน 1 วินาที ตารางที่ 2.2 เป็นการแสดงอัตราบอดทั่วไปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม

ตารางที่ 2.2 อัตราบอดที่ใช้กันทั่วไป

อัตราบอด	ช่วงเวลาของแต่ละบิต
110	9.91 ms
150	6.67 ms
300	3.33 ms
600	1.67 ms
1200	0.833 ms
2400	0.417 ms
4800	0.208 ms
9600	0.104 ms
19200	0.052 ms



รูปที่ 2.22 การแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม

ในรูปที่ 2.22 เป็นการแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม โดยที่เริ่มแรกข้อมูลแบบขนานจะถูกนำไปเก็บไว้ใน Shift Register หลังจากนั้นจะใช้สัญญาณนาฬิกาในการเลื่อนค่าในรีจิสเตอร์ออกมาทีละบิต (โดยการเลื่อนค่าไปทางขวามือ) โดยบิตแรกที่ถูกเลื่อนออกมาคือบิต LSB ของข้อมูลและบิตที่สองที่ถูกเลื่อนออกมาก็คือบิตที่อยู่ถัดจากบิต LSB และบิตต่อ ๆ ไป สำหรับบิตสุดท้ายที่ถูกเลื่อนออกมาก็คือบิต MSB ของข้อมูล

การแปลงข้อมูลแบบอนุกรมเป็นข้อมูลแบบขนานนั้นจะมีขั้นตอนตรงกันข้ามกับที่กล่าวมา นั่นคือข้อมูลแบบอนุกรมจะถูกเคลื่อนเข้าไปใน Shift Register โดยที่จะใช้สัญญาณนาฬิกาเป็นตัวควบคุมและหลังจากที่มีการเคลื่อนข้อมูลทุกบิตเข้าไปใน Shift Register ได้หมดแล้ว ข้อมูลในรีจิสเตอร์นี้จะถูกนำออกมาแบบขนานเพื่อนำไปใช้งานต่อไป

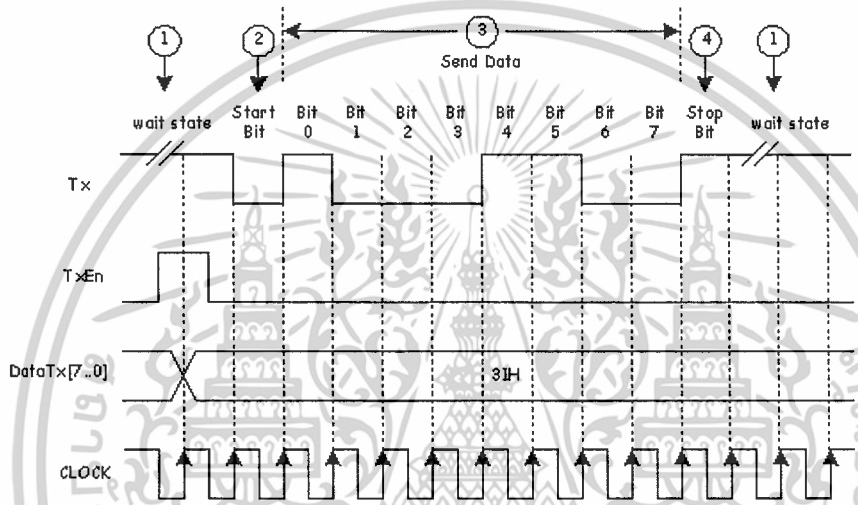
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.10 หลักการออกแบบวงจรรับ – ส่งข้อมูลแบบอนุกรมด้วย FPGA

สำหรับวงจรสื่อสารข้อมูลแบบอนุกรมที่เรากำลังจะออกแบบนั้น จะมีคุณสมบัติดังนี้ Baud Rate = 9600 Bits/Sec, Data = 8 Bits, Start Bit = 1 Bit, Stop Bit = 1 Bit, Parity = none

2.6.10.1 ภาคส่งข้อมูลแบบอนุกรม

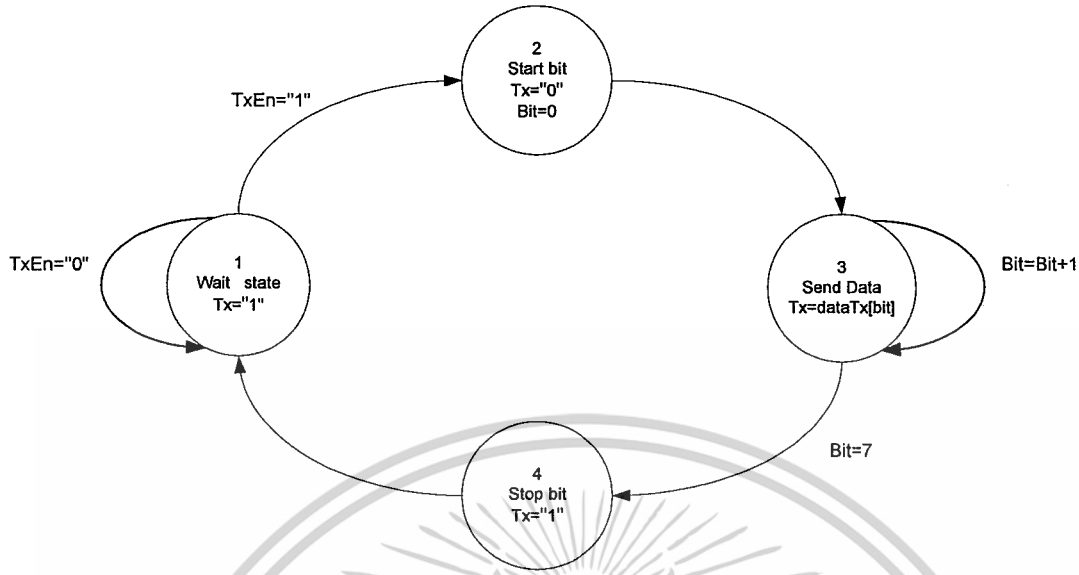
ในส่วนนี้เป็นการออกแบบ FPGA ให้ทำหน้าที่เป็นตัวส่งข้อมูลแบบอนุกรมให้แก่อุปกรณ์ต่าง ๆ เช่น คอมพิวเตอร์, ไมโครคอนโทรลเลอร์ เป็นต้น รูปแบบการส่งข้อมูลแบบอนุกรมแสดงดังรูปที่ 2.23



รูปที่ 2.23 รูปแบบการส่งข้อมูลแบบอนุกรม

จากรูปที่ 2.23 สัญญาณ Tx เป็นสัญญาณขนาด 1 บิต ใช้สำหรับส่งข้อมูลแบบอนุกรมออกไป ในการออกแบบวงจรส่งข้อมูลแบบอนุกรมจะต้องมีสัญญาณ CLOCK เป็นสัญญาณอ้างอิง เพื่อกำหนดความเร็วในการส่งข้อมูลซึ่งจะต้องตรงกับทางภาครับด้วย ความเร็วในการส่งข้อมูลนี้ก็คือ Baud Rate นั้นเอง เช่น ต้องการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที ก็จะต้องกำหนดให้สัญญาณ CLOCK มีค่าเท่ากับ 9600 Hz เมื่อสัญญาณ TxEn มีค่าเป็นลอจิก “0” แสดงว่ายังไม่ต้องการส่งข้อมูล จะทำให้สัญญาณ Tx มีสถานะเป็นลอจิก “1” จนกว่าสัญญาณ TxEn จะมีสถานะเป็นลอจิก “1” แสดงว่าต้องการส่งข้อมูลออกไป จะทำให้สัญญาณ Tx มีค่าเป็นลอจิก “0” เพื่อเป็นการบอกทางภาครับว่าจะเริ่มต้นส่งข้อมูลออกไป หลังจากนั้นจะทำการส่งข้อมูลออกไป DataTx [7..0] เป็นข้อมูลที่จะต้องการส่งแบบอนุกรม โดยที่จะส่งบิตที่มีความสำคัญต่ำสุด (Bit 0) ออกไปก่อน จะส่งข้อมูลออกไปที่ละบิตจนครบทั้ง 8 บิต จากนั้นสัญญาณ Tx จึงมีสถานะเป็นลอจิก “1” เพื่อบอกทางภาครับว่าเป็นการสิ้นสุดการส่งข้อมูลแล้ว เราสามารถจะนำวิธีการส่งข้อมูลแบบอนุกรมที่ได้กล่าวมาแล้วมาเขียนเป็น State Diagram ได้ดังรูปที่ 2.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



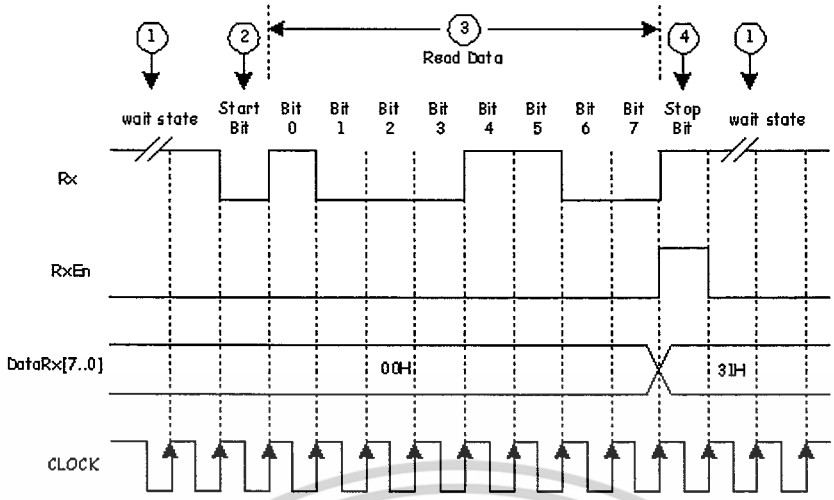
รูปที่ 2.24 State Diagram ของการส่งข้อมูลแบบอนุกรม

จาก State Diagram ในรูปที่ 2.24 เมื่อ TxEn มีค่าเป็นลอจิก “0” จะทำให้ Tx มีค่าเท่ากับลอจิก “1” และจะยังอยู่ใน State 1 จนกว่า TxEn จะมีค่าเป็นลอจิก “1” จึงจะเข้าสู่ State 2 ใน State 2 สัญญาณ Tx จะมีค่าเป็นลอจิก “0” เป็นการกำหนดบิตเริ่มต้นการส่งข้อมูล หลังจากนั้นจะเข้าสู่ State 3 ซึ่งจะเป็นการส่งข้อมูลออกไปที่ละบิต โดยที่เริ่มต้นจะส่งข้อมูลบิตที่ 0 ออกไปก่อนและวนส่งข้อมูลออกไปจนครบ 8 บิต จึงจะหลุดเข้าสู่ State 4 จะกำหนดให้ Tx มีค่าเป็นลอจิก “1” เป็นการกำหนดบิตสิ้นสุดการส่งข้อมูล หลังจากนั้นจึงกลับเข้าสู่ State 1 ใหม่อีกครั้ง

2.6.10.2 ภาครับข้อมูลแบบอนุกรม

สำหรับภาครับข้อมูลแบบอนุกรม การทำงานจะคล้ายๆกับทางภาคส่งข้อมูลแบบอนุกรมจะถูกส่งมาจากตัวส่ง เช่นคอมพิวเตอร์, ไมโครคอนโทรลเลอร์ เป็นต้น ข้อมูลแบบอนุกรมที่ส่งมาจะมีลักษณะดังรูปที่ 2.25

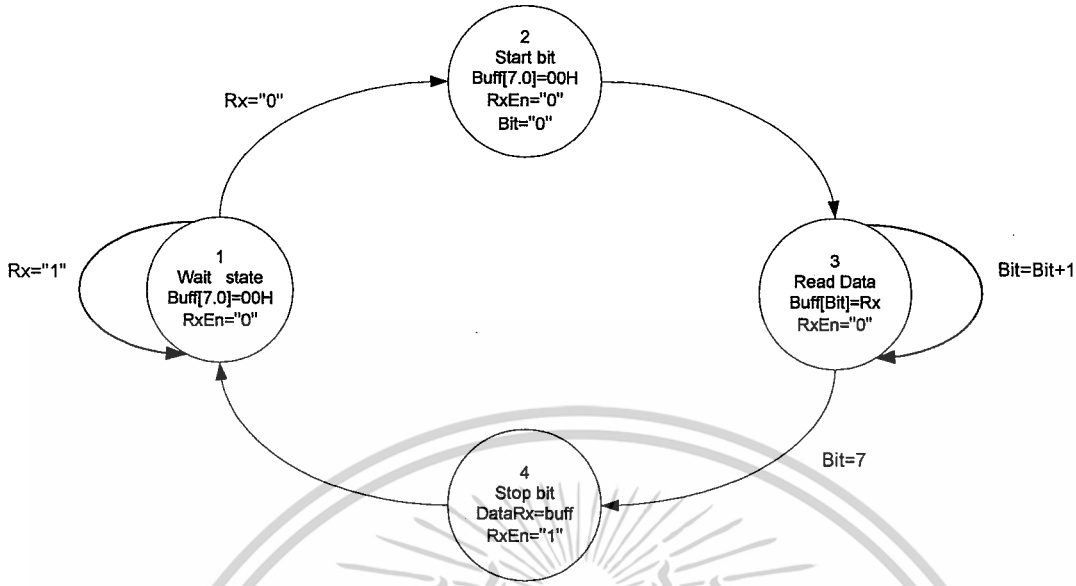
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 รูปแบบการรับข้อมูลแบบอนุกรม

จากรูปที่ 2.25 สัญญาณ Rx เป็นสัญญาณขนาด 1 บิตใช้สำหรับรับข้อมูลแบบอนุกรมที่ส่งมาจากตัวส่ง ในการออกแบบภาครับข้อมูลจะต้องมีสัญญาณ CLOCK เป็นสัญญาณอ้างอิง เพื่อกำหนดความเร็วในการรับข้อมูล ซึ่งจะต้องสอดคล้องกับทางภาคส่ง เช่น ภาคส่งใช้อัตราเร็วในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ในทางภาครับก็จะต้องกำหนดให้มีอัตราเร็วในการรับข้อมูลเป็น 9600 บิตต่อวินาทีเช่นกัน เพราะฉะนั้นจะต้องกำหนดให้สัญญาณ CLOCK มีความถี่เท่ากับ 9600 Hz เมื่อสัญญาณ Rx มีค่าเป็นลอจิก “1” แสดงว่ายังไม่มีการส่งข้อมูลออกมา จะรอจนกว่า Rx มีค่าเป็นลอจิก “0” แสดงว่าทางภาคส่งจะเริ่มส่งข้อมูลมาแล้ว หลังจากนั้นจะทำการอ่านข้อมูลเข้ามาเก็บไว้ที่ละบิตจนครบทั้ง 8 บิต และตรวจสอบสัญญาณ Rx ว่าเป็นลอจิก “1” หรือไม่ หากเป็นลอจิก “1” แสดงว่าสิ้นสุดการส่งข้อมูล และกำหนดให้สัญญาณ DataRx[7..0] มีค่าเท่ากับสัญญาณที่รับมาได้ และกำหนดให้สัญญาณ RxEn มีค่าเป็นลอจิก “1” เพื่อเป็นสัญญาณกระตุ้นให้วงจรต่อไปนำข้อมูล DataRx[7..0] ไปใช้งาน เราสามารถนำวิธีการรับข้อมูลแบบอนุกรมที่ได้กล่าวมาแล้วมาเขียนเป็น State Diagram ได้ดังรูปที่ 2.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 State Diagram ของการรับข้อมูลแบบอนุกรม

จาก State Diagram ในรูปที่ 2.26 เริ่มต้นจะอยู่ใน State 1 หากสัญญาณ Rx มีค่าเป็นลอจิก "1" จะไม่มีการเปลี่ยน Stateจะให้ค่า RxEn เป็นลอจิก "0" และ Buff[7..0] ซึ่งเป็น Buffer สำหรับเก็บข้อมูลที่อ่านมาได้จากสัญญาณ Rx มีค่าเป็น "00000000" เมื่อสัญญาณ Rx มีค่าเป็นลอจิก "0" แสดงว่าเป็นบิตเริ่มต้นของข้อมูลจะเปลี่ยนเป็น State 2 เมื่อเข้ามาที่ State 2 แสดงว่าได้มีการส่งบิตเริ่มต้นข้อมูลมาแล้ว ใน State นี้จะยังคงกำหนดให้ Buff[7..0] มีค่าเท่ากับ "00000000", RxEn เป็นลอจิก "0" และ Bit มีค่าเท่ากับ 0 หลังจากนั้นจะเข้าสู่ State 3 โดยอัตโนมัติ ภายใน State 3 จะมีการอ่านข้อมูลจาก Rx เข้ามาเก็บไว้ใน Buff จะอ่านข้อมูลจนครบทั้ง 8 บิต นำข้อมูลที่ได้ทั้งหมดเก็บไว้ใน Buff และกำหนดให้สัญญาณ RxEn มีค่าเป็นลอจิก "0" เมื่ออ่านข้อมูลจนครบทั้ง 8 บิต แล้วก็เปลี่ยน State เป็น State 4 ภายใน State 4 จะมีการโอนย้ายข้อมูลจาก Buff มาเก็บไว้ใน DataRx ซึ่งเป็นข้อมูลขนาด 8 บิตทั้งหมดที่รับมาได้ และกำหนดให้ RxEn มีค่าเป็นลอจิก "1" เพื่อเป็นสัญญาณกระตุ้นให้ภาคอื่น ๆ รับทราบว่าได้รับข้อมูลมาครบ 8 บิตแล้ว หลังจากนั้นก็กลับมา State 1 เพื่อรอรับข้อมูลอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

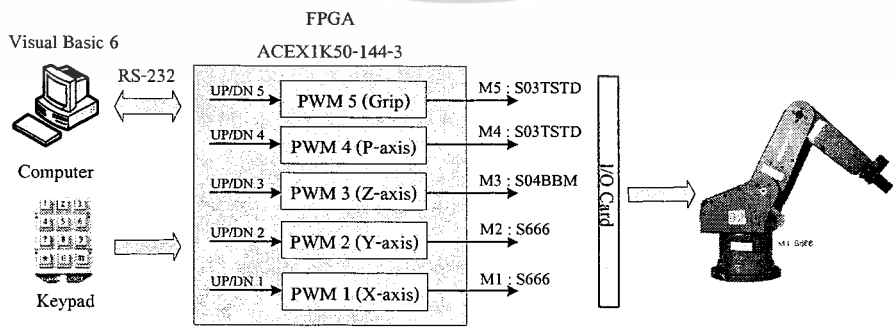
บทที่ 3
โครงสร้างการทำงานของแขนกล

3.1 บทนำ

แขนกลอุตสาหกรรมนั้นหากจะนิยามสั้น ๆ ก็คงจะกล่าวได้ว่า คือเครื่องมือชนิดหนึ่ง ซึ่งสามารถทำงานได้ตามคำสั่งโดยสามารถทำงานซ้ำ ๆ ได้หลายครั้ง โดยไม่มีความผิดพลาดจนกว่าจะมีการเปลี่ยนแปลงคำสั่งการทำงานของแขนกลอุตสาหกรรมนี้ หลักการของแขนกลคล้ายกับการทำงานของอุปกรณ์ที่ควบคุมด้วยคำสั่งเชิงตัวเลข (numerial control machine) ในลักษณะของการทำงานการป้อนกลับ (feedback) ฯลฯ แต่จุดที่ต่างกันก็คือ การสั่งงาน หรือการโปรแกรม-แขนกลนี้จะทำโดยตรงมีแนวโน้มว่าจะมีการนำไปใช้ภายในอุตสาหกรรมขนาดเล็กเพิ่มขึ้น เพราะสามารถควบคุมได้ด้วยไมโครคอมพิวเตอร์ทำให้มีความยืดหยุ่นสูง ในการออกแบบโครงสร้างการทำงานของโครงงานนี้เป็นการประยุกต์ และเลือกใช้ฮาร์ดแวร์ให้น้อยที่สุดเพื่อเป็นการประหยัดค่าใช้จ่ายแต่มีประสิทธิภาพที่ดี โดยใช้ปุ่มกดและโปรแกรม Visual Basic 6 เชื่อมต่อรับ-ส่งค่าผ่านบอร์ด FPGA ที่เป็นตัวสร้างสัญญาณ PWM ในการขับมอเตอร์ เพื่อการควบคุมให้ดีซีเซอร์โวมอเตอร์หมุนตามความเร็วและไปยังตำแหน่งที่ต้องการนั้น

3.2 การออกแบบระบบ

ระบบที่ออกแบบนั้นจะประกอบด้วยคอมพิวเตอร์รับ-ส่งข้อมูลผ่านพอร์ตอนุกรมติดต่อกับ FPGA เพื่อควบคุมตัวสร้างสัญญาณ PWM จำนวน 5 ช่องสัญญาณที่อิสระต่อกัน และสามารถเปลี่ยนแปลงช่วงความกว้างของพัลส์ตามสัญญาณนาฬิกา (ck_inX) และสัญญาณการนับขึ้น-ลง (UPX/DNX) โดยที่มีการกำหนดการเคลื่อนที่จากคอมพิวเตอร์ ค่าช่วงความกว้างสูงสุดและต่ำสุดตามดีซีเซอร์โวในแต่ละแกนหมุน



รูปที่ 3.1 ระบบโดยรวมและการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 จะมีการควบคุมอยู่ 2 ส่วนคือส่วนแรกใช้คอมพิวเตอร์ควบคุมการเคลื่อนที่ของแขนกลโดยจะส่งค่าช่วงความกว้างพัลส์ในแต่ละแกนให้กับ FPGA และส่งค่าความถี่ของพัลส์เพื่อควบคุมความเร็วในการเคลื่อนที่ของแขนกล การตั้งค่าและการเปลี่ยนแปลงค่าด้วยโปรแกรมวิชวลเบสิค (Visual Basic) ส่วนที่สองใช้ปุ่มกด (Keypad) เพื่อควบคุมการเคลื่อนที่แขนกลในแต่ละแกน โดยจะเคลื่อนที่ตามการกดปุ่มค้างไว้และจะหยุดเมื่อปล่อยปุ่มกด

3.3 ส่วนประกอบของแขนกลและระบบควบคุมแขนกล

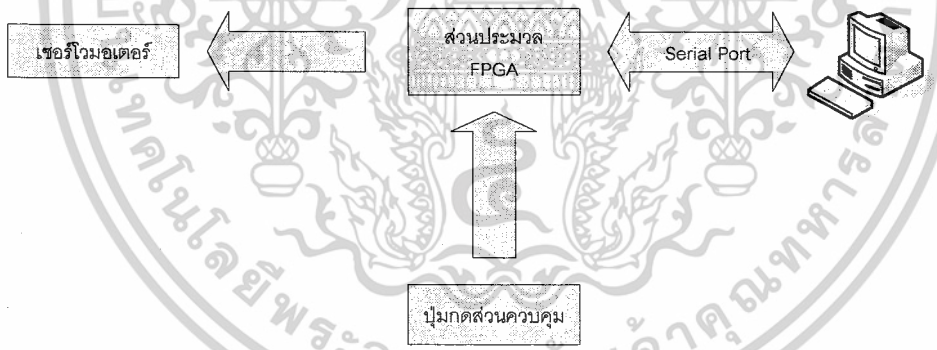
ในส่วนของแขนกลนี้ ซึ่งประกอบได้ด้วย 4 แกน คือ X, Y, Z, P และ 1 Grip ซึ่งแขนกลนั้นมีดีซีเซอร์โวมอเตอร์ในการขับเคลื่อน 5 ตัวที่รับสัญญาณ PWM จากบอร์ด FPGA ที่เชื่อมต่อกับปุ่มกดและเชื่อมต่อผ่านคอมพิวเตอร์ ดังรูปที่ 3.2 ส่วนประกอบหลัก ๆ ของแขนกลจะแบ่งออกเป็น 4 ส่วน คือ

3.2.1 ส่วนปุ่มกดควบคุม

3.2.2 ส่วนประมวลผล (FPGA) และสร้างสัญญาณ PWM

3.2.3 ส่วนเซอร์โวมอเตอร์ ในแขนกล

3.2.4 ส่วนซอฟต์แวร์ (โปรแกรม) และส่วนอินเตอร์เฟส บนคอมพิวเตอร์

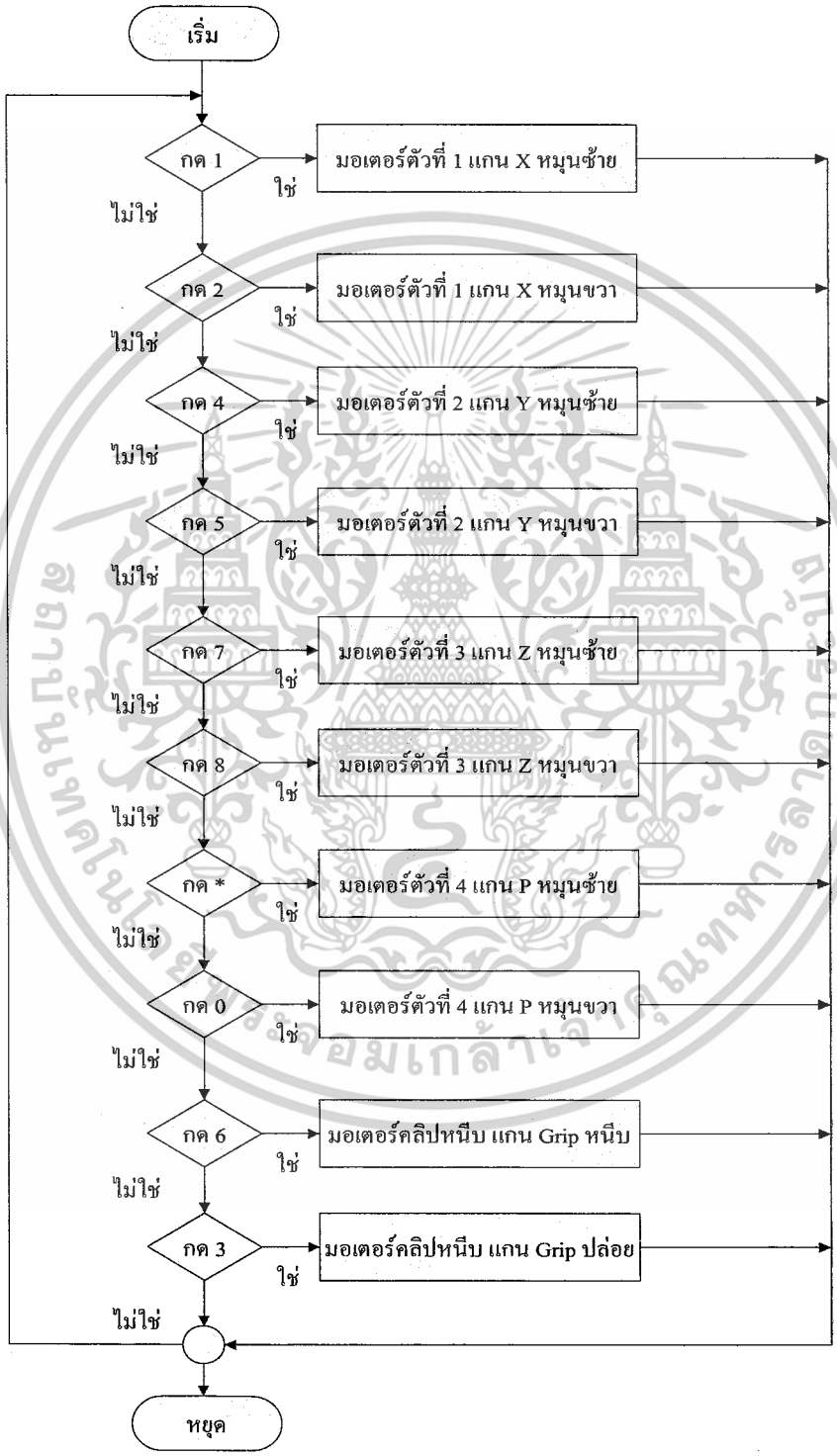


รูปที่ 3.2 ไดอะแกรมการทำงานของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 ส่วนปุ่มกดควบคุม

ในส่วนปุ่มกดควบคุมจะใช้ปุ่มกดคีย์ของโทรศัพท์ขนาด 4X3 MATRIX KEY ออกแบบโดยให้แต่ละปุ่มกดเป็นดังนี้

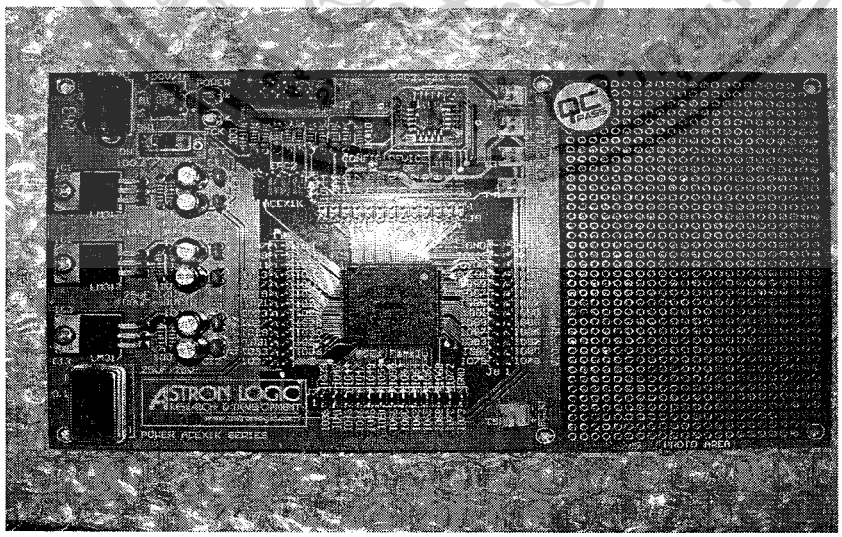


รูปที่ 3.3 Flowchart ปุ่มกดควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ส่วนประมวลผล FPGA (Field Programmable Gate Array)

เอฟพีจีเอ (FPGA) คือวงจรรวมชนิดหนึ่ง ซึ่งภายในประกอบไปด้วยเกตจำนวนมากเป็นฮาร์ดแวร์ที่สามารถโปรแกรมได้ ในโครงงานนี้ได้เลือกใช้บอร์ด FPGA ของ Astron Logic เพื่อทำการสร้างสัญญาณ PWM ไปควบคุมตำแหน่งการทำงานของดีซีเซอร์โวมอเตอร์ ซึ่งเลือกใช้ตระกูล ALTERA รุ่น POWER ACEX 1K-50 เป็นชิป FPGA ที่มีโครงสร้างภายในเป็นแบบ SRAM-Based FPGA ใช้เทคโนโลยีการโปรแกรมเหมือนกับหน่วยความจำแบบ SRAM (Static RAM) ทำให้การโปรแกรมสามารถทำซ้ำได้ง่าย แต่ข้อเสียที่ต้องการไฟเลี้ยงในการเก็บวงจรที่ได้ออกแบบไว้ภายในบอร์ดจะใช้ชิป FPGA เบอร์ EP1K50TC144-3 มีความจุของเกตประมาณ 50,000 เกต มี Logic Elements (LEs) 2,880 LEs มีหน่วยความจำภายใน (Internal Memory) 40,960 Bits และมีขา I/O ให้สามารถใช้งานได้อย่างอิสระถึง 102 ขา สามารถใช้งานกับ I/O แบบ 5V, 3.3V หรือ 2.5V ด้วยความจุเกตของ FPGA ในกลุ่ม POWER ACEX1K SERIES นี้จึงจะทำให้เราสามารถนำไปใช้ในการออกแบบระบบดิจิตอลที่มีขนาดใหญ่หรือสลับซับซ้อนได้เป็นอย่างดี เช่น การออกแบบระบบไมโครคอนโทรลเลอร์, การออกแบบระบบ CPU (Central Processing Unit), ระบบ DSP (Digital Signal Processing) หรือนำไปใช้งานในระบบโทรคมนาคม และในแต่ละรุ่นจะมี Socket สำหรับใส่ Configuration Device เพื่อเก็บวงจรที่ได้ออกแบบไว้สำหรับโหลดวงจรอัตโนมัติลงใน FGPA เมื่อเริ่มจ่ายไฟเลี้ยงเข้าบอร์ด มีการจัดเรียง I/O ของ FPGA เป็นชุด ๆ ให้สามารถใช้งานได้อย่างสะดวกและอิสระ สามารถเชื่อมต่อได้ด้วยสายแพร์พร้อมทั้งโมดูลออสซิเลเตอร์ความถี่ 1 MHz สำหรับการออกแบบวงจรประเภท Sequential Circuit และมีพื้นที่จุดบัดกรีขนาด 101.6 มม. คูณ 66.04 มม. จำนวนจุดบัดกรี 868 จุด สำหรับจัดวางอุปกรณ์ต่าง ๆ หรือเลือกติดโปรโตบอร์ดเพื่อความสะดวกในการพัฒนา



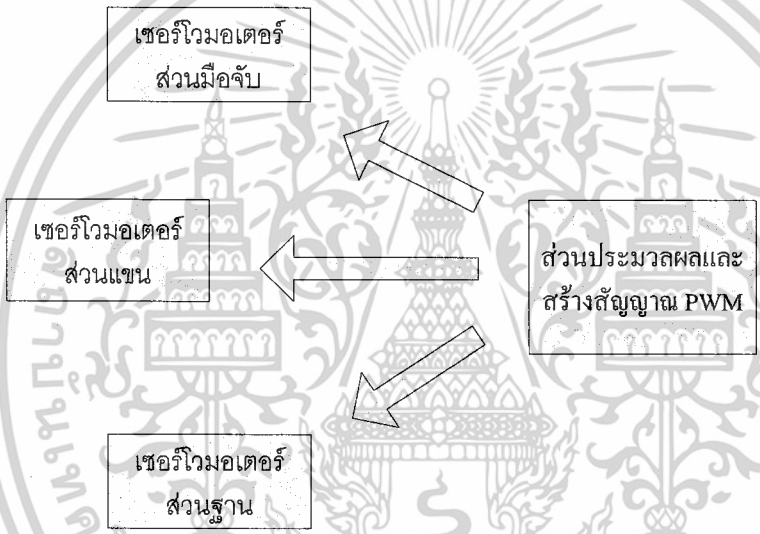
รูปที่ 3.4 บอร์ด FPGA ที่ใช้ในการทำโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ส่วนเซอร์โวมอเตอร์ในแขนกล

ในส่วนของแขนกลที่ขับเคลื่อนได้โดยใช้ดีซีเซอร์โวมอเตอร์ เป็นส่วนของมอเตอร์ที่ใช้ในการหมุนของแต่ละแกนขับเคลื่อนไปตามองศาที่เราสั่งการ ซึ่งในโครงงานนี้กำหนดใช้ดีซีเซอร์โวมอเตอร์สำหรับแกนต่าง ๆ ดังนี้

- แกน X ใช้ดีซีเซอร์โวมอเตอร์ของ GWS รุ่น S666 – N
- แกน Y ใช้ดีซีเซอร์โวมอเตอร์ของ GWS รุ่น S666 – N
- แกน Z ใช้ดีซีเซอร์โวมอเตอร์ของ GWS รุ่น S04 – BBM
- แกน P ใช้ดีซีเซอร์โวมอเตอร์ของ GWS รุ่น S03T – STD
- แกน Gripper ใช้ดีซีเซอร์โวมอเตอร์ของ GWS รุ่น S03T – STD



รูปที่ 3.5 ผังการทำงานของดีซีเซอร์โวมอเตอร์

จากรูปที่ 3.5 แสดงถึงส่วนของดีซีเซอร์โวมอเตอร์ที่รับสัญญาณ PWM จากบอร์ด FPGA ที่สามารถสร้างสัญญาณออกมา 5 ช่องสัญญาณ ในการขับเคลื่อนดีซีเซอร์โวมอเตอร์ 5 ตัวที่อยู่ตามส่วนต่าง ๆ ของแขนกล

3.2.4 ส่วนซอฟต์แวร์ (โปรแกรม) และส่วนอินเตอร์เฟส บนคอมพิวเตอร์

ในส่วนของโปรแกรมที่ใช้จัดการสร้างสัญญาณ PWM และใช้ควบคุมแขนกลโดยเชื่อมต่อผ่านคอมพิวเตอร์จะกล่าวในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โปรแกรมควบคุมแขนกล

4.1 การออกแบบโดยใช้ภาษาระดับสูง (HDL)

ในปัจจุบันเทคโนโลยีของการออกแบบดิจิตอลได้พัฒนาสูงขึ้น และได้มีการพัฒนาภาษาที่ใช้สำหรับการออกแบบวงจรดิจิตอล พร้อมกระบวนการออกแบบแนวใหม่ที่จะทำให้ผู้ออกแบบไม่ถูกจำกัดในการออกแบบวงจรอีกต่อไป ซึ่งการออกแบบวงจรดิจิตอลแนวใหม่นี้จะใช้การเขียนภาษาบรรยายพฤติกรรมฮาร์ดแวร์ (HDL : Hardware Description Language) มาใช้ในการออกแบบวงจร โดยภาษาจะบรรยายพฤติกรรมฮาร์ดแวร์ที่เป็นมาตรฐานซึ่งมีอยู่ 2 ภาษาที่นิยมใช้ด้วยกัน คือ ภาษา Verilog HDL และ VHDL โดยถ้านำมาเปรียบเทียบในโลกของการออกแบบซอฟต์แวร์ก็สามารถเปรียบได้กับภาษา C และภาษา Pascal นั่นเอง โดยที่ภาษา Verilog HDL จะมีโครงสร้างคล้ายกับภาษา C ส่วนภาษา VHDL จะมีโครงสร้างคล้ายกับภาษา Pascal

4.2 ลักษณะภาษา Verilog HDL

ภาษา Verilog HDL เป็นภาษาที่ใช้การอธิบายการทำงานของวงจรดิจิตอล ซึ่งมีการทำงานพร้อม ๆ กันหลาย ๆ วงจรย่อยเป็นลักษณะแบบขนาน (Concurrency) แนวคิดของการสร้างระบบโดยใช้ภาษา Verilog HDL ที่มองระบบเหมือนกับกล่องดำ (Black box) อันหนึ่งที่มีส่วนที่ต่อเชื่อมกับภายนอกอยู่รอบ ๆ ซึ่งตัวกล่องดำอันนี้ใน Verilog HDL เรียกว่าโมดูล (Module) การบรรยายวงจรทำได้ 2 แบบคือ

1. Structural Model : กำหนดการเชื่อมต่อระหว่าง Primitive Element (Gates) เป็นการเขียนบรรยายการเชื่อมต่อของ Gates ตามวงจรที่กำหนด

2. Behavioral Model : กำหนดรูปแบบการทำงาน ความสัมพันธ์ ระหว่าง Input-Output เป็นการเขียนการทำงานของวงจร โดยมีลักษณะเหมือนการเขียนซอฟต์แวร์ แต่ HDL (Verilog หรือภาษาอื่นก็ตาม) ไม่ใช่การเขียนซอฟต์แวร์เพื่อไปสร้างฮาร์ดแวร์

4.2.1 ขั้นตอนออกแบบบน FPGA

1. Design Entry

เป็นขั้นตอนของการรับข้อมูลของการออกแบบเข้าไปในระบบ โดยทั่วไปสามารถทำได้หลายทาง เช่น โดยใช้ Schematic design entry ดึงอุปกรณ์มาจากไลบารีของ FPGA หรือโดยใช้ภาษา HDL อย่างเช่น Verilog HDL หรือ VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Synthesis

ขั้นตอนนี้นั้นจะใช้ซอฟต์แวร์ในการสังเคราะห์วงจร ซึ่งจะเกี่ยวข้องกับการแปลงแบบที่ได้ออกแบบจาก Schematic หรือ HDL ให้เป็นวงจรลอจิกที่เหมาะสมกับอุปกรณ์โดยที่จะมีขั้นตอนย่อย ๆ คือ การสังเคราะห์วงจร (Logic synthesis) ซึ่งเป็นไฟล์ที่จะอธิบายการเชื่อมต่อ (Netlist) ในขั้นตอนการสังเคราะห์วงจรผู้ออกแบบสามารถที่จะกำหนดข้อบังคับในเรื่องของพื้นที่ (Area) หรือกำหนดชนิด และตำแหน่งของ I/O

3. Implementation

หลังจากได้ Netlist file มาแล้วจะเป็นการแมป (MAP) ให้เข้ากับเทคโนโลยีหรือตัวชิฟที่จะใช้งาน (Technology mapping) เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างสถาปัตยกรรมภายใน โดยส่วนมากแล้ว FPGA จะเป็นเทคโนโลยีแบบ LUT (Look-up table) และหลังจากนั้นก็ทำการวางตำแหน่ง (Placement) ของลอจิกต่าง ๆ ซึ่งจะใช้ซอฟต์แวร์วางตำแหน่งขา I/O (I/O Pin) ตามตำแหน่งขา I/O ของ FPGA แล้วทำการเชื่อมต่อสายสัญญาณ (Routing) ระหว่างองค์ประกอบต่าง ๆ ภายในอุปกรณ์ FPGA เช่น ระหว่าง CLBs หรือ CLBs กับ IOBs ซึ่งจะทำต่อจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ก็สามารถจะทำการเชื่อมต่อสัญญาณได้ไม่หมดหรือเกิดความหน่วงเกินค่าที่กำหนด ดังนั้นควรใช้ซอฟต์แวร์ในการค้นหาเส้นทาซึ่งจะช่วยให้ผลที่ได้จากการทำการเชื่อมต่อสัญญาณดีขึ้นได้

4. Hardware Program (Configuration)

การโปรแกรมอุปกรณ์หรือชิฟ FPGA หลังจากวงจรผ่านได้ขั้นตอนต่าง ๆ มาแล้ว ถึงตอนนี้ก็สามารถที่จะดาวน์โหลด (download) ลงในอุปกรณ์ FPGA ได้แล้ว เราสามารถทำเองได้ทันทีรวมทั้งสามารถปรับเปลี่ยนแก้ไขได้ตลอดเวลา โดยสามารถกลับไปทำตั้งแต่ต้นก่อนที่จะทำการโปรแกรมทดสอบการทำงานจริงๆได้โดย ไม่มีค่าใช้จ่าย หรือความเสียหายเกิดขึ้น กรณีที่เป็นอุปกรณ์ FPGA ชนิดที่ต้องโปรแกรมโดยวิธี SRAM นั้น การใช้งานผู้ออกแบบจะต้องเก็บข้อมูลวงจรไว้ในหน่วยความจำประเภท EPROM หรือ serial PROM เพื่อจะใช้งานสะดวกขึ้น คือในการใช้งานครั้งต่อไปไม่ต้องดาวน์โหลดข้อมูลวงจรจากเครื่องคอมพิวเตอร์อีก เพราะมีข้อมูลวงจรเก็บอยู่ในหน่วยความจำอยู่แล้ว

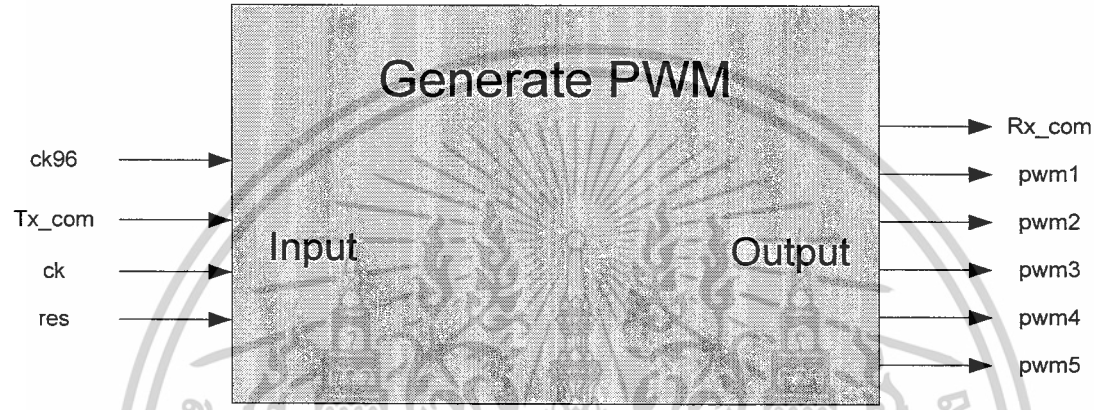
4.3 การใช้โปรแกรม Max+Plus II และ เขียนโปรแกรมด้วยภาษา Verilog HDL ในการคอนโทรลดีซีเซอร์โวมอเตอร์

Max+Plus II เป็นโปรแกรมซึ่งถูกสร้างและพัฒนาขึ้นโดยบริษัท ALTERA เพื่อใช้สำหรับออกแบบสังเคราะห์วงจรทางดิจิตอลและโปรแกรมวงจรลอจิกที่สร้างขึ้นลงในชิป FPGA ซึ่งในการออกแบบวงจรด้วยโปรแกรม Max+Plus II สามารถทำได้หลายลักษณะ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. วาดวงจรโดยการนำเกตหรือสัญลักษณ์ต่าง ๆ มา เชื่อมต่อกัน
2. ใช้ภาษา Verilog HDL อธิบายพฤติกรรมของวงจรที่ต้องการ
3. ใช้การกำหนดรูปแบบ Waveform ทางอินพุตและเอาท์พุตของวงจรที่ต้องการ

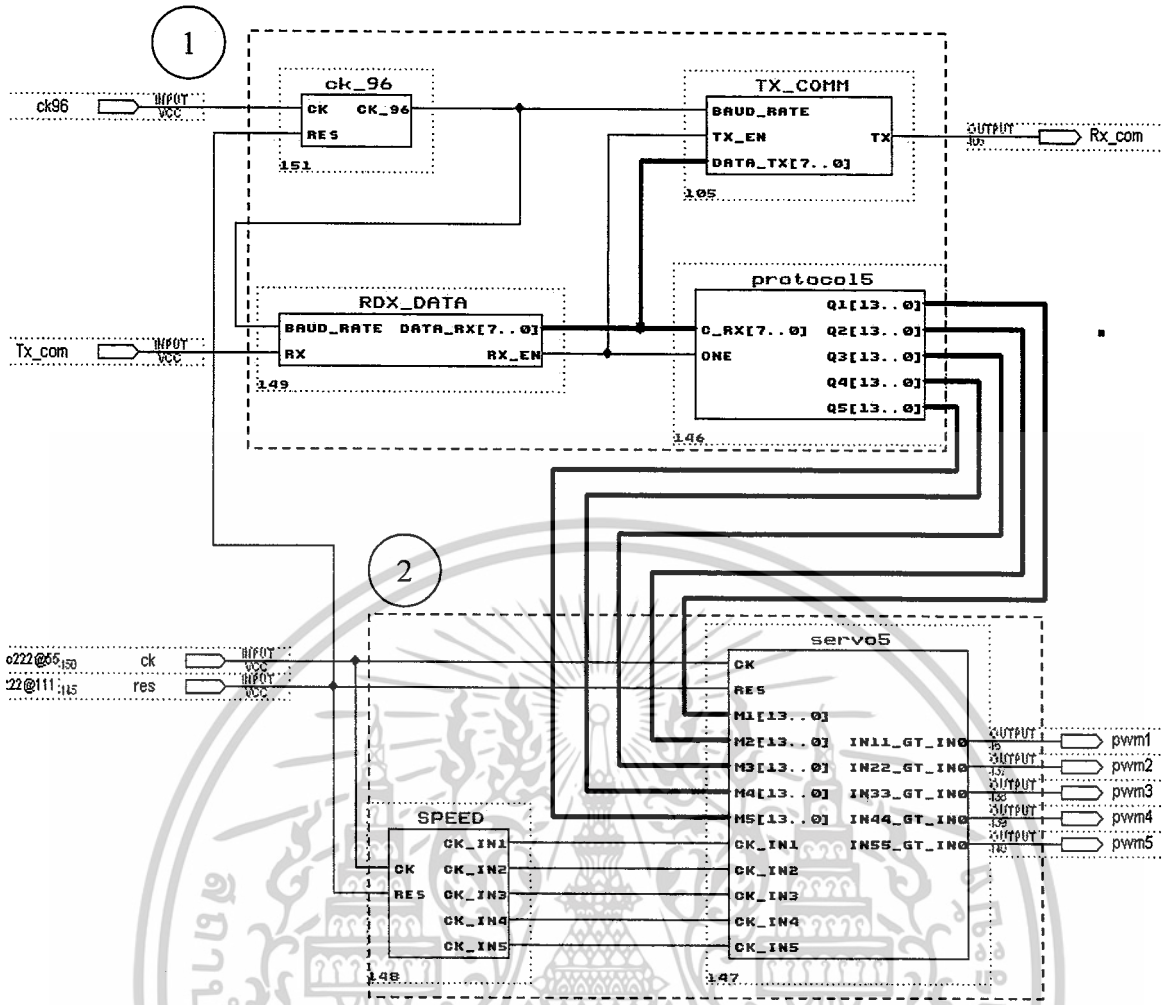
4.4 การออกแบบขับแขนกลด้วยโปรแกรม Verilog HDL เชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรมสำหรับสร้าง PWM



รูปที่ 4.1 Block รวมของโปรแกรมควบคุมแขนกลเชื่อมต่อกับคอมพิวเตอร์

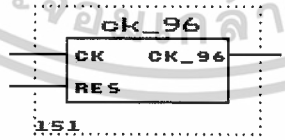
โปรแกรมที่ออกแบบมานี้แบ่งออกเป็น 2 ส่วนด้วยกันคือ ส่วนแรกจะเป็นส่วนวงจรรับส่งข้อมูลแบบอนุกรมที่ใช้ติดต่อระหว่างคอมพิวเตอร์กับบอร์ด FPGA และส่วนที่สองจะเป็นการสร้างสัญญาณ PWM ไปขับเคลื่อนดีซีเซอร์โวมอเตอร์ในแขนกล ซึ่งโปรแกรมแต่ละส่วนจะประกอบไปด้วยโมดูลต่าง ๆ ดังรูปที่ 4.2 ซึ่งแต่ละโมดูลมีหน้าที่และหลักการออกแบบดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การเชื่อมต่อโมดูลของโปรแกรมควบคุมแขนกลติดต่อกับคอมพิวเตอร์

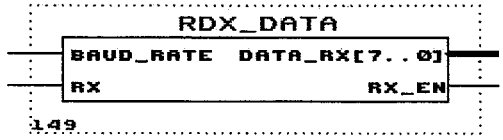
จากรูปที่ 4.2 แสดงรายละเอียดของการเชื่อมต่อโมดูลของโปรแกรมควบคุมแขนกลติดต่อกับคอมพิวเตอร์



รูปที่ 4.3 โมดูล ck_96

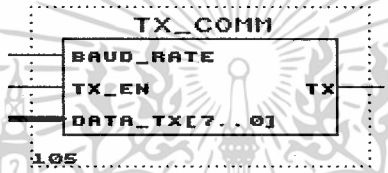
จากรูปที่ 4.3 ck_96 เป็นโมดูลของการปรับอัตราบอดเรตให้เท่ากับความถี่คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



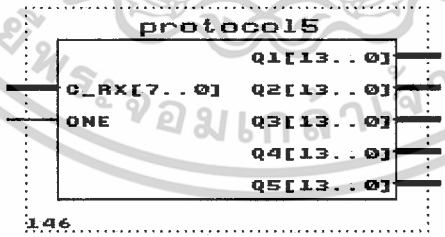
รูปที่ 4.4 โมดูล RDX_DATA

จากรูปที่ 4.4 RDX_DATA เป็นโมดูลทำหน้าที่รับข้อมูลแบบอนุกรมจากคอมพิวเตอร์ แล้วทำการแปลงข้อมูลจากแบบอนุกรมเก็บเป็นข้อมูลแบบขนาน กำหนดจังหวะการรับ-ส่งข้อมูลด้วยบอดเรต



รูปที่ 4.5 โมดูล TX_COMM

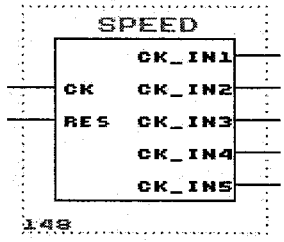
จากรูปที่ 4.5 TX_COMM เป็นโมดูลทำหน้าที่รับข้อมูลจากส่วน RDX_DATA เพื่อส่งค่าข้อมูลนั้นไปยังคอมพิวเตอร์ ในการเช็คค่าความถูกต้องของข้อมูล กำหนดจังหวะการรับ-ส่งข้อมูลด้วยบอดเรต



รูปที่ 4.6 โมดูล Protocol 5

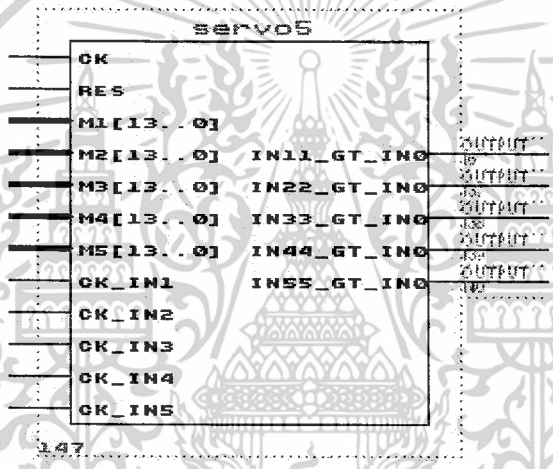
จากรูปที่ 4.6 Protocol 5 เป็นโมดูลทำหน้าที่ในการตรวจสอบข้อมูลที่ได้รับเข้ามา ว่าเป็นข้อมูลอะไร แล้วนำข้อมูลที่ได้ไปสร้างเป็นสัญญาณ PWM เพื่อใช้ในการขับดีซีเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 โมดูล SPEED

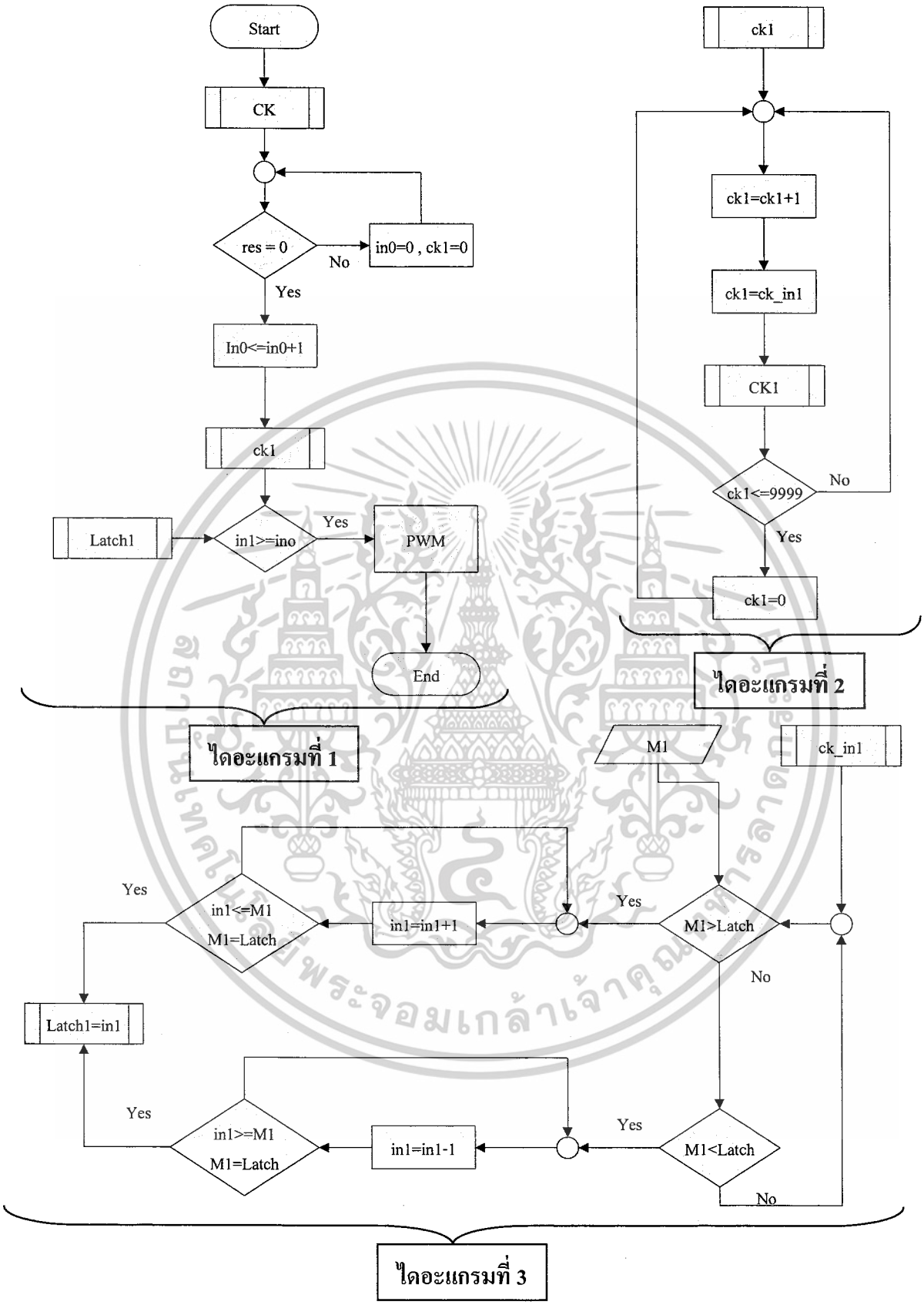
จากรูปที่ 4.7 SPEED เป็นโมดูลที่ใช้หารความถี่จาก oscillator ภายในบอร์ด FPGA เพื่อใช้ในการปรับความเร็วของดีซีเซอร์โวมอเตอร์



รูปที่ 4.8 โมดูล Servo 5

จากรูปที่ 4.8 Servo 5 เป็นโมดูลที่ทำหน้าที่รับข้อมูลจากส่วนของ Protocol 5 รับเข้ามาเปรียบเทียบกับสัญญาณ Sawtooth เพื่อสร้างเป็นสัญญาณ PWM 5 ช่องสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 Flowchart การสร้าง PWM ที่ FPGA เชื่อมต่อกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากไดอะแกรมที่ 1 ของรูปที่ 4.9 เป็นการสร้างสัญญาณ sawtooth เก็บค่าไว้ในรีจิสเตอร์ in0 แล้วนำมาเปรียบเทียบกับค่าสัญญาณ Vcontrol ที่จะเก็บค่าไว้ในรีจิสเตอร์ in1 เพื่อสร้างออกเป็นสัญญาณ PWM

จากไดอะแกรมที่ 2 ของรูปที่ 4.9 เป็นการหารสัญญาณความถี่จาก oscillator ภายในบอร์ด FPGA เพื่อปรับความเร็วของดีซีเซอร์โวมอเตอร์

จากไดอะแกรมที่ 3 ของรูปที่ 4.9 เป็นการสร้างสัญญาณ Vcontrol ซึ่ง M1 เป็นค่าข้อมูลที่ส่งมาจากคอมพิวเตอร์แล้วนำมาเข้าโปรแกรมเพื่อสร้างเป็นสัญญาณ Vcontrol โดยเก็บค่าสัญญาณ Vcontrol ไว้ในรีจิสเตอร์ in1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โปรแกรม Visual Basic

5.1 การออกแบบโปรแกรมอินเตอร์เฟสด้วย Visual Basic 6

วิชวลเบสิก (Visual Basic) เป็นโปรแกรมที่ใช้สร้างโปรแกรมประยุกต์สำหรับระบบปฏิบัติการวินโดว์ (Windows) เนื่องจากความง่ายของภาษาที่ใช้ในการเขียน และวิชวลเบสิกเป็นโปรแกรมที่เน้นในการสร้างแอพพลิเคชั่นเพื่อติดต่อกับผู้ใช้งานเป็นหลัก ซึ่งรูปแบบนี้คือการเขียนโปรแกรมไปพร้อม ๆ กับ การเห็นผลลัพธ์ที่เกิดขึ้น มีเครื่องมือคอยอำนวยความสะดวกในการเขียนโปรแกรมมากขึ้นลดความซับซ้อนยุ่งยากต่าง ๆ ลง ทำให้ได้ผลงานออกมารวดเร็วสวยงาม และนอกจากนี้วิชวลเบสิกยังมีเครื่องมือที่ใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมซึ่งนำไปใช้ในการควบคุมเครื่องอีกด้วย

5.1.1 จุดเด่นของ Visual Basic

1. สร้างแอพลิเคชั่นได้ง่ายและรวดเร็ว

Visual Basic ได้รับการวางตัวให้เป็นเครื่องมือที่ช่วยให้สร้างแอพลิเคชั่นได้อย่างรวดเร็วและง่ายดาย เพื่อลดเวลาในการสร้างแอพลิเคชั่นให้สั้นลง ซึ่งเรียกรูปแบบนี้ว่า Rapid Application Development หรือ RAC ทั้งนี้เพราะมีการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำ ๆ ซาก ๆ ออกไป เหลือเฉพาะที่ต้องโฟกัสเกี่ยวกับปัญหาของงานจริง ๆ แล้วเขียนโปรแกรมจัดการปัญหานั้น ๆ ส่วนเรื่องอื่น ๆ เหลือให้ Visual Basic จัดการ

2. การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้

ถ้ามีโอกาสเขียนโปรแกรมด้วย Visual Basic แล้วจะเห็นว่าภาษา Basic ที่ใช้ใน Visual Basic นั้นสามารถเข้าใจง่าย คือใกล้เคียงกับภาษาที่เราใช้งานปกติสื่อความหมายและเข้าใจง่ายกว่าภาษาโปรแกรมอื่น ๆ ทำให้ผู้เริ่มต้นเขียนโปรแกรมสามารถทำความเข้าใจกับการเขียนโปรแกรมได้อย่างรวดเร็ว

3. รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

นอกจากจะง่ายต่อการเรียนรู้แล้ว Visual Basic ยังมีเครื่องมือที่ช่วยในการเขียนโปรแกรมเป็นเรื่องที่ไม่ยุ่งยาก เพราะจะมีเครื่องมือที่ช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยากวุ่นวาย มีการตรวจสอบอัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่ ซึ่งจะมีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบทำให้งานของโปรแกรมเมอร์นั้นลดลงได้มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากจะมีเครื่องมือช่วยในการเขียนโปรแกรมแล้ว ยังมีเครื่องมือที่ใช้ในการทดสอบแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นออกมาว่าสามารถทำงานได้ถูกต้องหรือไม่และมีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิงสามารถขอความช่วยเหลือในจุดที่เราสงสัยข้องใจได้

เครื่องมือทั้งหมดที่กล่าวมาถูกจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกัน (เรียกย่อว่า ๆ IDE ซึ่งย่อมาก Integral Development Environment) ทำให้เรียกใช้งานได้สะดวกตั้งแต่เขียนโปรแกรม ทดสอบ แก้ไข สร้างชุดติดตั้ง รวมทั้งระบบขอความช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือใหม่ ๆ เข้าไปได้เรื่อย ๆ หรือถอดเครื่องมือที่ไม่จำเป็นต้องใช้เพื่อประหยัดพื้นที่ฮาร์ดดิสก์ก็ได้เช่นกัน

5.1.2 สร้างแอพลิเคชั่นด้วย Visual Basic

1. ออกแบบแอพลิเคชั่น

ต้องทราบให้แน่ชัดก่อนว่าแอพลิเคชั่นที่เราจะทำการสร้างนั้น จะนำไปใช้ประโยชน์อะไร มีความสำคัญอะไรบ้าง และต้องมีรูปร่างหน้าตาเป็นอย่างไร ซึ่งจำเป็นอย่างยิ่งที่ต้องคิดให้รอบคอบและเขียนออกมาให้ชัดเจนโดยอาจจะร่างง่าย ๆ ในกระดาษ

2. ตกแต่งหน้าตาแอพลิเคชั่น

เป็นการตกแต่งรูปร่างของแอพลิเคชั่นที่ได้ออกแบบไว้แล้วพร้อม ๆ กับการคำนวณค่าของพ็อพเพอร์ตี้ต่าง ๆ ให้กับคอนโครลแต่ละตัวในแอพลิเคชั่น

3. เขียนโค้ดกำกับแอพลิเคชั่น

หลังจากตกแต่งหน้าตาเสร็จแล้วขั้นตอนต่อไป คือการเขียนโค้ด หรือการเขียนโปรแกรมเพื่อควบคุมการทำงานต่าง ๆ ซึ่งเราจะใช้การเขียนโปรแกรมแบบ Event Driven Programming ซึ่งจะเป็นการเขียนโค้ดเพื่อรองรับกับเหตุการณ์ต่าง ๆ ที่จะเกิดขึ้นกับคอนโทรลในแอพลิเคชั่นของเรา

4. ทดสอบการทำงานของแอพลิเคชั่น

เมื่อเขียนโค้ดเสร็จแล้วก็ถึงเวลาที่จะทดสอบการทำงานของแอพลิเคชั่นที่เราสร้างขึ้นมา ซึ่งประกอบไปด้วยคอนโทรลต่าง ๆ

5. บันทึกเก็บไว้ในคอมพิวเตอร์

หลังจากทดสอบจนแน่ใจว่าแอพลิเคชั่นที่สร้างมานั้นทำงานได้ถูกต้อง เราจึงบันทึกเก็บไว้ซึ่งสามารถแก้ไขและเพิ่มเติมความสามารถอื่น ๆ ได้ภายหลัง

6. สร้างไฟล์ .EXE (Make)

เมื่อเราสร้างแอพลิเคชั่นเสร็จแล้ว หากเราต้องการนำแอพลิเคชั่นที่สร้างขึ้นให้เรียกใช้งานได้เองโดยไม่ต้องเรียกผ่าน Visual Basic หรือที่จะต้องการนำไปใช้งานในเครื่องคอมพิวเตอร์เครื่องอื่น ๆ เราสามารถทำได้โดยสร้างไฟล์เอ็กซ์คิวต์ (ไฟลที่มีนามสกุลเป็น .EXE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 องค์ประกอบของ Visual Basic

5.2.1 ส่วนประกอบของ Visual Basic

โปรแกรมวิชวลเบสิกแบ่งออกเป็นสองส่วนคือ

1. ส่วนที่มองเห็น (Visual Component) คือส่วนที่ผู้ใช้โปรแกรมสามารถมองเห็นได้บนจอภาพขณะที่โปรแกรมกำลังทำงาน ส่วนนี้เรียกว่าส่วนประสานกับผู้ใช้โปรแกรม (User Interface) ได้แก่ส่วนที่เป็นแบบฟอร์ม และวัตถุควบคุมต่าง ๆ

2. ส่วนที่มองไม่เห็น (Code Component) คือส่วนที่เป็นคำสั่งต่าง ๆ ของภาษาวิชวลเบสิกที่ผู้พัฒนาโปรแกรมเขียนขึ้นมาแล้วคีย์ลงไปในพื้นที่สำหรับพัฒนาโปรแกรมของแต่ละฟอร์ม ซึ่งผู้ใช้โปรแกรมไม่สามารถมองเห็นขณะที่โปรแกรมกำลังทำงาน ได้แก่คำสั่งประกาศชื่อตัวแปรและส่วนโปรแกรม (Sub) ต่าง ๆ

5.2.2 ขั้นตอนการสร้างโปรแกรมประยุกต์

การสร้างโปรแกรมประยุกต์วิชวลเบสิก ประกอบด้วยขั้นตอนหลัก 3 ขั้นตอน คือ

1. การสร้างอินเตอร์เฟส (Interface) โดยมีฟอร์ม (Form) เป็นอ๊อปเจ็ค (Object) พื้นฐานและเป็นที่วางตัวคอนโทรล (Control) สำหรับการติดต่อกับผู้ใช้

2. ตั้งค่าคุณสมบัติ เป็นการกำหนดพฤติกรรม และการทำงานให้กับอ๊อปเจ็คต่างๆ

3. การเขียนคำสั่งเป็นการควบคุมการประมวลผลผ่านโพรซีเจอร์ (Proceduce) ที่กำหนด

5.3 อินเตอร์เฟสบนคอมพิวเตอร์

ในส่วนนี้จะประกอบด้วยโปรแกรมอินเตอร์เฟส Visual Basic 6 สำหรับการรับและส่งค่าทิศทาง ความเร็ว โดยติดต่อกันทางพอร์ตอนุกรมของคอมพิวเตอร์กับตำแหน่งขารับ-ส่งบนบอร์ด FPGA ในการควบคุมแขนกลผ่านคอมพิวเตอร์

5.3.1 คอนโทรล MSComm

สำหรับการใช้งาน Visual Basic จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้ ซึ่งก็คือ MSComm ที่มีทางเลือก 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูลทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communication) เป็นรูปแบบของการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier (DCD) หรือขา Request To Send (RTS) เหตุการณ์ Oncomm ของ MSComm จะสามารถตรวจจับสัญญาณนั้นได้ทันที ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าของเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วย การดูค่าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานภายในฟังก์ชั่นต่าง ๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ต นั้นจะต้องใช้งานคอนโทรล MSComm มากกว่า 1 ตัว เพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ตแอดเดรสของพอร์ตอนุกรม และแอดเดรสของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

การกำหนดคุณสมบัติ (Property) ของ MSComm Control ให้สามารถติดต่อกับ FPGA ทาง Serial Port ซึ่งก็คือการกำหนดอัตราการรับส่งข้อมูลบอตเรต, ชนิดของพาริตี้บิตที่ใช้, จำนวนบิต-ข้อมูลที่ต้องการส่ง, และจำนวนบิตปิดท้าย



รูปที่ 5.1 MSComm Control

1. Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งานในการกำหนดและอ่านค่าอนุกรมที่ติดต่ออยู่ สามารถเขียนโปรแกรมได้ดังนี้

- ตัวอย่าง MSComm1.CommPort = 1 ในที่นี้เลือกจะใช้ Com1 อยู่ที่ด้านหลังเครื่องคอมพิวเตอร์

2. Property ชื่อ Settings คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอดเรต (การรับส่งข้อมูลของอุปกรณ์) ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ค่าต่างๆ ดังนี้ MSComm1.Settings = "Baud (อัตราการรับส่งข้อมูล), Parity ถ้าไม่ใช้ใส่ N, จำนวนบิตข้อมูล, บิต-สต๊อบ" สามารถเขียนโปรแกรมได้ดังนี้

- ตัวอย่าง MSComm1.Settings = "9600, N, 8, 1"

3. Property ชื่อ InputLen คือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัพเฟอร์ สามารถเขียนโปรแกรมได้ดังนี้

- ตัวอย่าง MSComm1.InputLen = 1

4. Property ชื่อ PortOpen คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด = True ถ้าปิด = False สามารถเขียนโปรแกรมได้ดังนี้

- ตัวอย่าง MSComm1.PortOpen = True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

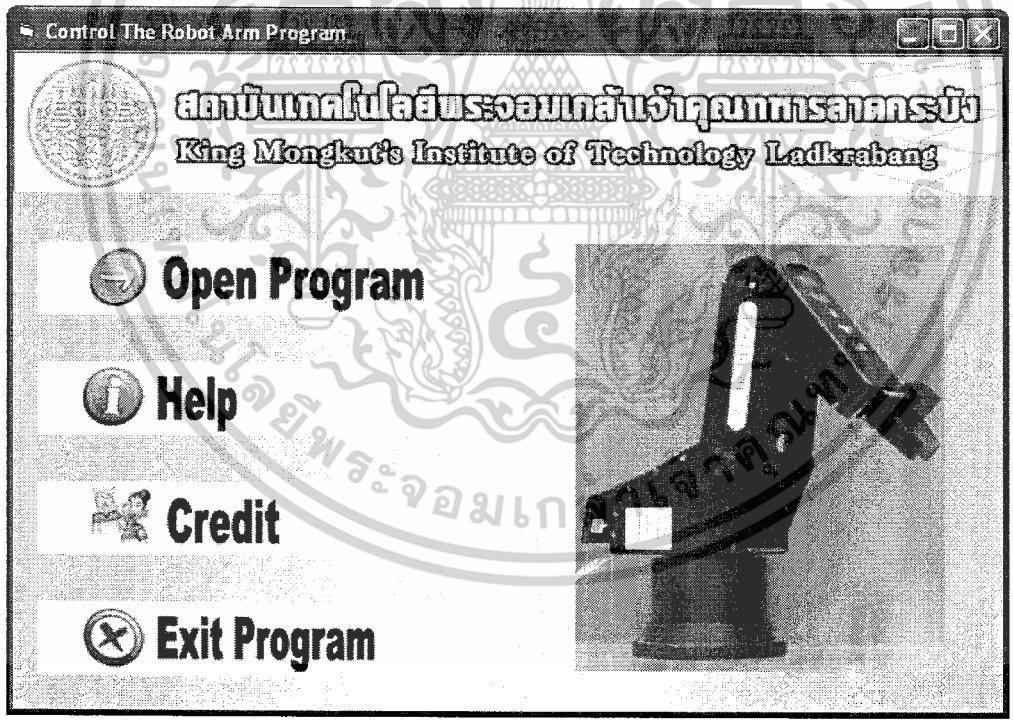
5.4 โปรแกรม Visual Basic 6 ที่ได้ออกแบบในการควบคุมแขนกล

ใช้โปรแกรม Visual Basic 6 เป็นตัวสร้างส่วนควบคุมระหว่างแขนกลกับคอมพิวเตอร์ ซึ่งออกแบบโปรแกรมให้ติดต่อรับ-ส่งค่าจากหน้าอินเตอร์เฟสในโปรแกรมรับ-ส่งข้อมูลอนุกรมติดต่อกับบอร์ด FPGA ในการเลือกกำหนดค่าเพื่อให้แขนกลที่ขับเคลื่อนโดยดีซีเซอร์โวมอเตอร์เคลื่อนที่ตามที่ต้องการ

5.4.1 การใช้งานของโปรแกรมควบคุมแขนกล

ซึ่งเมื่อทำการเปิดโปรแกรมเพื่อทำการควบคุมแขนกล (Control The Robot Arm Program) จะปรากฏหน้าต่างของโปรแกรมแสดงขึ้นมา ดังรูปที่ 5.2 โดยจะแสดงเมนูการใช้งานต่าง ๆ ดังนี้

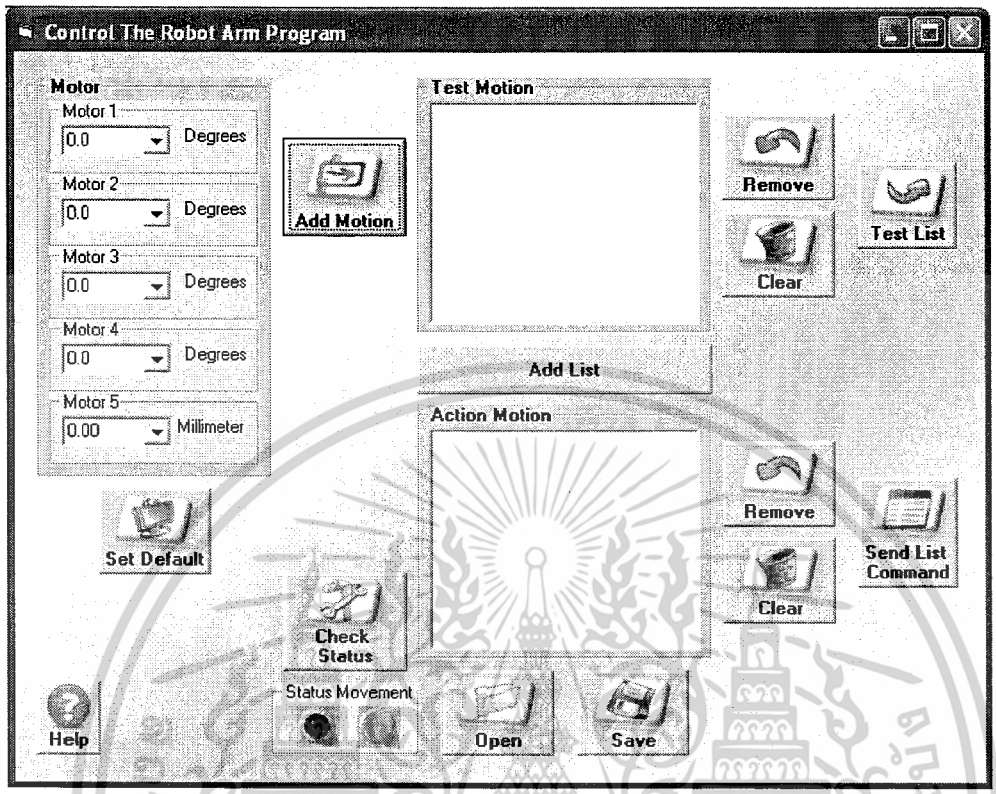
1. Open Program (เปิดโปรแกรม)
2. Help (ตัวช่วยการใช้งาน)
3. Credit (คณะผู้จัดทำ)
4. Exit Program (ออกจากโปรแกรม)



รูปที่ 5.2 แสดงหน้าต่างแรกของโปรแกรมควบคุมแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Open Program (เปิดโปรแกรม)



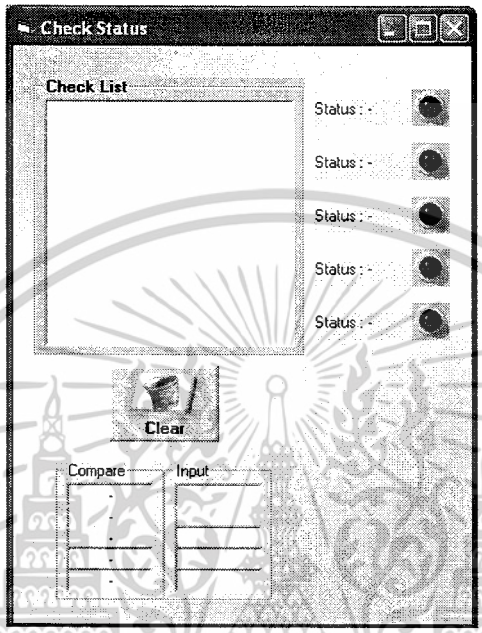
รูปที่ 5.3 แสดงหน้าต่างภายในของโปรแกรมควบคุมแขนกล

จากรูปที่ 5.3 แสดงหน้าตาของโปรแกรมที่ใช้ควบคุมแขนกล โดยกรอบด้านซ้ายที่ปรากฏขึ้นมาจะมีกรอบข้อความที่เขียนว่า "Motor" ซึ่งภายในนั้นจะเป็นการเลือกตำแหน่งของมอเตอร์ทั้ง 5 แกน ว่าต้องการให้แขนกลเคลื่อนที่ไปในตำแหน่งใดตามตำแหน่งที่ผู้ใช้ต้องการเลือกใช้ ซึ่งในตำแหน่งของมอเตอร์ที่ใช้จะเป็นหน่วยองศา (Degree) และเมื่อผู้ใช้ได้ทำการกดปุ่ม Add Motion ตำแหน่งของมอเตอร์ที่ผู้ใช้ได้ทำการเลือกไว้ในตอนแรกนั้น ก็จะไปปรากฏอยู่ภายในกรอบของ Test Motion

เมื่อตำแหน่งที่ได้เลือกไว้มาปรากฏอยู่ในกรอบนี้แล้ว และเมื่อผู้ใช้ได้ทำการกดปุ่ม (Test List) จะเป็นการส่งการเคลื่อนที่ไปให้กับแขนกลทั้งหมด 1 ชุดคำสั่ง (ส่งค่าตำแหน่งของมอเตอร์แต่ละตัวไปยังแขนกลทั้ง 5 แกน พร้อม ๆ กัน) ซึ่งภายในกรอบของ Test Motion นี้นั้นจะเป็นการทดสอบการหาตำแหน่งของวัตถุว่าตำแหน่งที่เคลื่อนที่ไปนั้น แขนกลนั้นได้ไปอยู่ในตำแหน่งที่ต้องการหยิบจับหรือไม่ โดยถ้าแขนกลอยู่ในตำแหน่งของวัตถุนั้น ๆ แล้ว เราสามารถนำตำแหน่งการเคลื่อนที่มาให้ทำให้แขนกลนั้นสามารถเคลื่อนที่เป็นแบบ Automatic ได้ โดยการกด Add List ค่าที่เราต้องการใช้ก็จะมาปรากฏอยู่ในกรอบ "Action Motion" ที่จะเราสามารถเพิ่มจำนวนชุดคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้หลาย ๆ ชุดคำสั่ง ภายหลังจากที่เราได้ชุดคำสั่งในการเคลื่อนที่แล้ว เราสามารถทำการกดปุ่ม "Send List Command" แขนกลก็จะทำการเคลื่อนที่ตามชุดคำสั่งที่ผู้ใช้ได้ทำการเลือกไว้ โดยแต่ละคำสั่งนั้นจะมีการหน่วงเวลาประมาณ 2.5 วินาที เพื่อให้แขนกลนั้นสามารถเคลื่อนที่ได้เสร็จสิ้นก่อนที่จะทำการส่งคำสั่งถัดไปออกมา



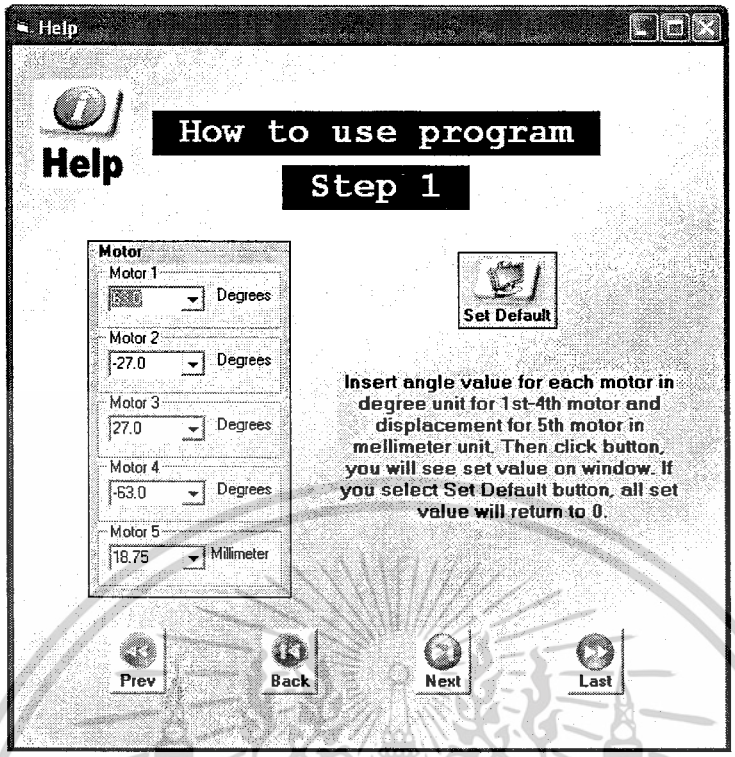
รูปที่ 5.4 แสดงหน้าต่างตรวจสอบข้อมูล

จากหน้าต่างที่แล้วรูปที่ 5.3 หลังจากที่เราได้ทำการส่งค่าต่าง ๆ ออกไปแล้ว เราสามารถนำค่าต่าง ๆ ที่เราส่งค่าออกไปมาทำการตรวจสอบได้ว่าค่าที่ส่งออกไปนั้นถูกต้องหรือเปล่า โดยทำการเปรียบเทียบระหว่างโปรโตคอลของการส่งค่าจากคอมพิวเตอร์ไปยังแขนกล และโปรโตคอลของการรับค่าจากแขนกลมายังคอมพิวเตอร์ว่ามีโปรโตคอลที่ตรงกันถูกต้องหรือไม่ โดยถ้าค่าที่รับเข้ามาตรงกันก็จะแสดงสัญญาณไฟสีเขียวและจะมีข้อความแสดงขึ้นมาว่า Status: OK ส่วนถ้ามีการส่งข้อมูลออกไปและค่าที่รับเข้ามาไม่ตรงกันก็จะแสดงสัญญาณไฟแดง และจะมีข้อความแสดงขึ้นมาว่า Status : Error ดังรูปที่ 5.4

2. Help (ตัวช่วยการใช้งาน)

เมื่อกดปุ่ม Help จะเป็นการเปิดหน้าต่างตัวช่วยการใช้งาน ซึ่งจะแสดงวิธีการใช้โปรแกรมเป็นขั้นตอนว่าตัวโปรแกรมทำงานอย่างไร ผู้ใช้จะต้องทำอย่างไรและผู้ใช้ควรรู้อะไรบ้างเกี่ยวกับการใช้โปรแกรมนี้ ดังรูปที่ 5.5-5.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

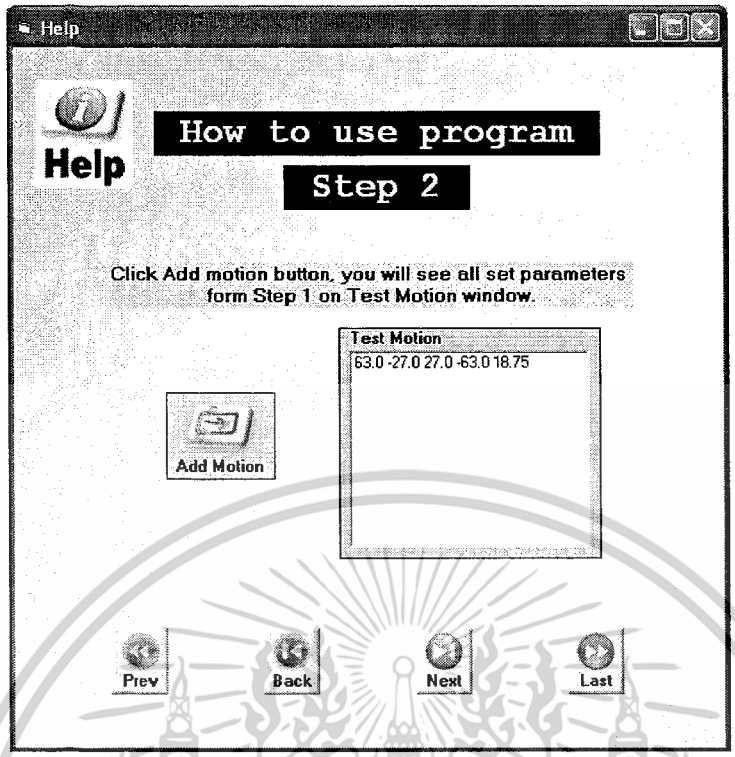


รูปที่ 5.5 แสดงหน้าต่างตัวช่วยหน้าที่ 1

จากรูปที่ 5.5 เป็นหน้าต่างแสดงตัวช่วยหน้าที่ 1 เพื่อบอกถึงวิธีการใช้โปรแกรมหน้าแรก คือเป็นการใส่ค่ามุมของมอเตอร์แต่ละตัว โดย 4 ค่าแรกจะเป็นค่าเกี่ยวกับระยะการหมุนตามแกนต่าง ๆ ของแขนกล ส่วนมอเตอร์ตัวที่ 5 จะเป็นค่าที่ใส่ให้กับตัวหนีบจับชิ้นงานของแขนกล

ส่วนปุ่ม Set Default เป็นการเซตตำแหน่งเริ่มต้นของแขนกล โดยเป็นการกำหนดให้แขนกลอยู่ ณ ตำแหน่งที่ 0 หรือเป็นค่าเริ่มต้นของแขนกล

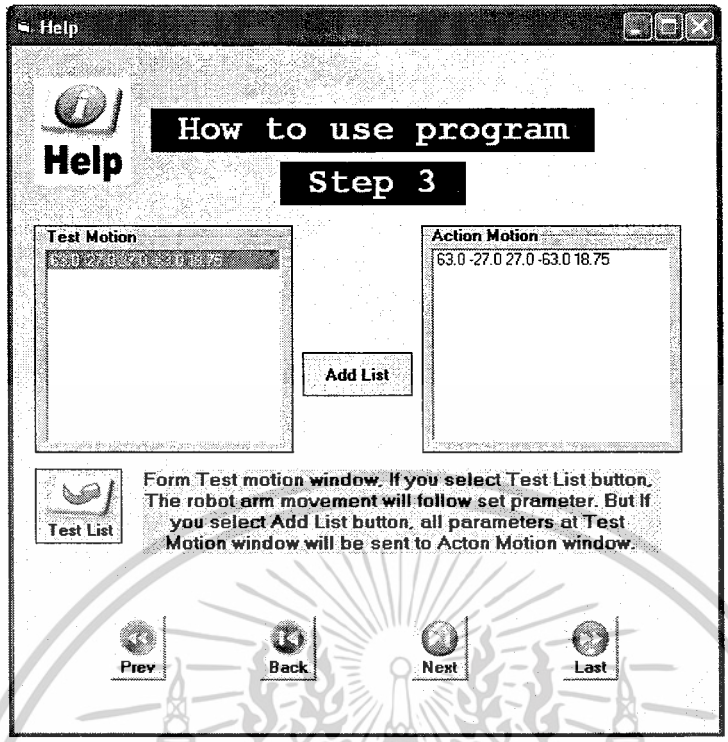
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 แสดงหน้าต่างตัวช่วยหน้าที่ 2

จากรูปที่ 5.6 เป็นหน้าต่างแสดงตัวช่วยหน้าที่ 2 เพื่อบอกถึงวิธีการใช้โปรแกรมหน้าที่ 2 คือ หลังจากหน้าแรกที่ได้ทำการเลือกค่ามุมต่าง ๆ ไว้แล้ว ก็จะเป็นการบันทึกค่าลงใน List เพื่อส่งค่าที่ผู้ใช้ต้องทำการส่งค่าไปยังแขนกล โดยทำการกดปุ่ม Add Motion โปรแกรมก็จะทำการบันทึกค่าลงใน List เพื่อเตรียมค่าต่าง ๆ นี้ เพื่อนำไปใช้สั่งงานแขนกล

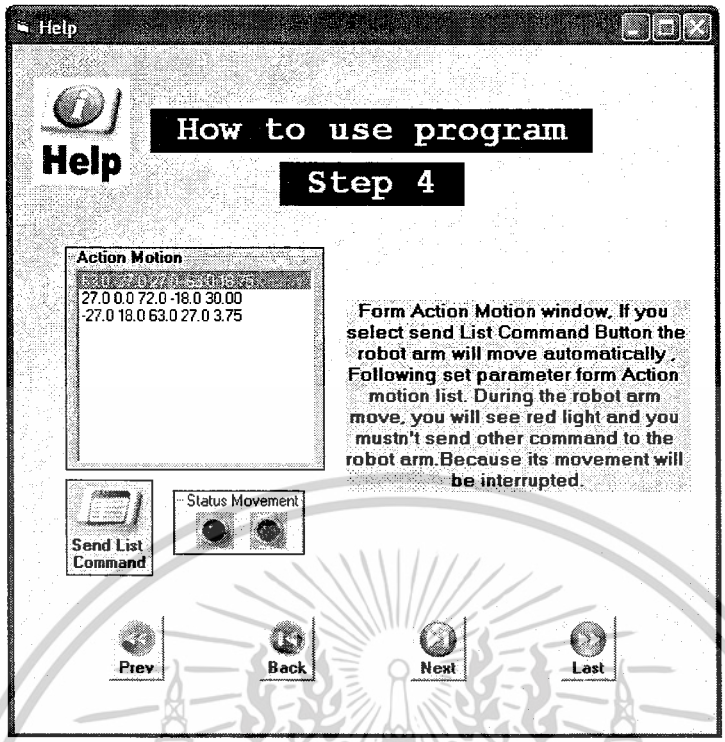
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แสดงหน้าต่างตัวช่วยหน้าที่ 3

จากรูปที่ 5.7 เป็นหน้าต่างแสดงตัวช่วยหน้าที่ 3 จากหน้าต่างที่ 2 ค่ามุมที่เราได้ทำการเลือกมุมไว้แล้ว เราสามารถสั่งงานแขนกลได้ โดยสั่งงานแขนกลเพียง 1 ชุดคำสั่ง ด้วยการกดปุ่ม (Test List) แขนกลก็จะเคลื่อนที่ไปเป็นระยะมุมที่ผู้ใช้ได้ทำการเลือกไว้ แต่ถ้าผู้ใช้ต้องการให้แขนกลสั่งงานเป็นแบบ Automatic ให้กดปุ่ม Add List โปรแกรมก็จะนำค่าต่าง ๆ ที่ผู้ใช้ได้ทำการเลือกไว้ไปอยู่ใน List Action Motion เพื่อให้แขนกลนั้นสั่งงานเป็นแบบ Automatic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



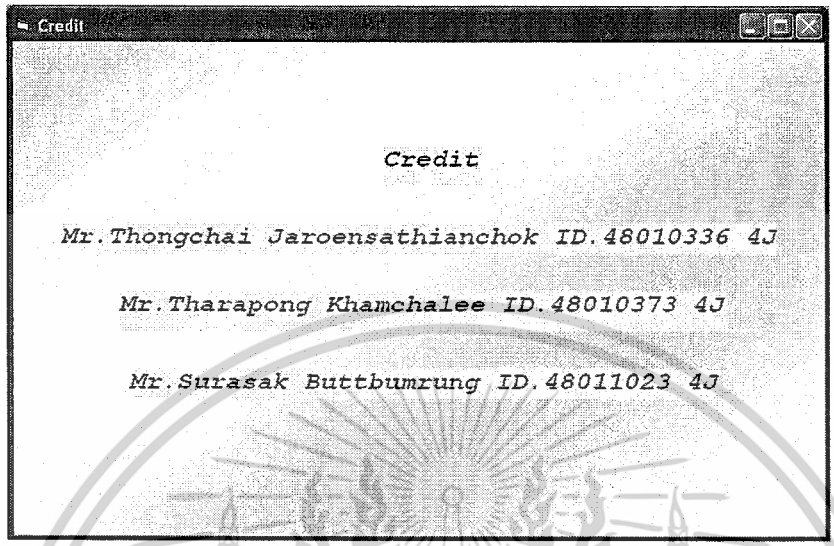
รูปที่ 5.8 แสดงหน้าต่างตัวช่วยหน้าที่ 4

จากรูปที่ 5.8 เป็นหน้าต่างแสดงตัวช่วยหน้าที่ 4 เมื่อผู้ใช้ได้ทำการเลือกให้ค่ามุมต่าง ๆ เข้าอยู่ใน List Action Motion ผู้ใช้ก็สามารถสั่งงานให้แขนกลสั่งงานเป็นแบบ Automatic ได้แล้ว โดยการกดปุ่ม Send List Command แต่เมื่อผู้ใช้งานได้สั่งงานออกไปแล้ว ผู้ใช้งานไม่ควรจะกดปุ่มอื่น ๆ เพราะจะเป็นการสั่งงานรบกวนการทำงานแขนกล ซึ่งแสดงเป็นรูปไฟสีเขียวกับสีแดง โดยถ้าสัญญาณไฟเป็นสีเขียว ผู้ใช้สามารถทำการสั่งงานแขนกลได้ แต่ถ้าเป็นสัญญาณไฟสีแดงผู้ใช้งานไม่ควรสั่งงานใด ๆ ให้แขนกลอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Credit (คณะผู้จัดทำ)

เมื่อกดปุ่ม Credit จะเป็นการเปิดหน้าต่างแสดงเกี่ยวกับคณะผู้จัดทำ



รูปที่ 5.9 แสดงหน้าต่างคณะผู้จัดทำ

4. Exit Program (ออกจากโปรแกรม)

เมื่อกดปุ่ม Exit จะเป็นการสั่งให้โปรแกรมปิดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

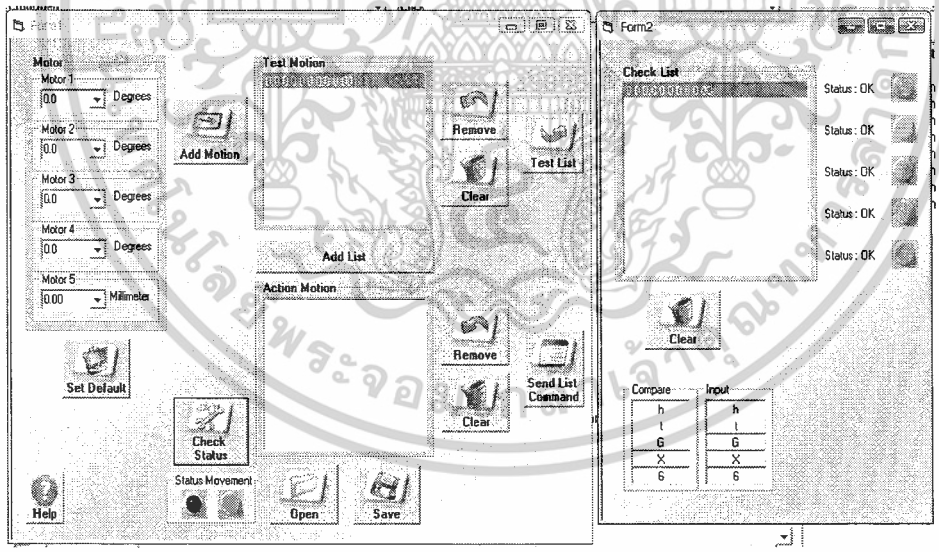
บทที่ 6

การทดลองและผลการทดลอง

การควบคุมการเคลื่อนที่ของแขนกลสามารถสั่งการผ่านคอมพิวเตอร์ โดยให้มีการเคลื่อนที่ของแขนกลตามการกำหนดในโปรแกรม ซึ่งสามารถสั่งการให้มีการทดสอบการเคลื่อนที่ครั้งละ 1 แกน หรือหลาย ๆ แกนพร้อมกัน และสามารถสั่งการเคลื่อนที่ครั้งละ 1 ชุดคำสั่ง หรือหลายชุดคำสั่งต่อเนื่องกันตามที่ผู้ใช้ต้องการจะกำหนดให้กับตัวแขนกล ในการทดลองนี้จะแสดงถึงการสั่งงานผ่านคอมพิวเตอร์ด้วยการสั่งชุดคำสั่ง 1 ชุดคำสั่งเพื่อแสดงถึงค่าของสัญญาณ PWM ที่ส่งออกไปยังแขนกล การสั่งงานแขนกลให้เคลื่อนที่ไปเพื่อตรวจสอบระยะและทิศทางการหมุนของแขนกล และการสั่งงานแบบหลาย ๆ ชุดคำสั่งต่อเนื่องกัน (แบบอัตโนมัติ) ดังนี้

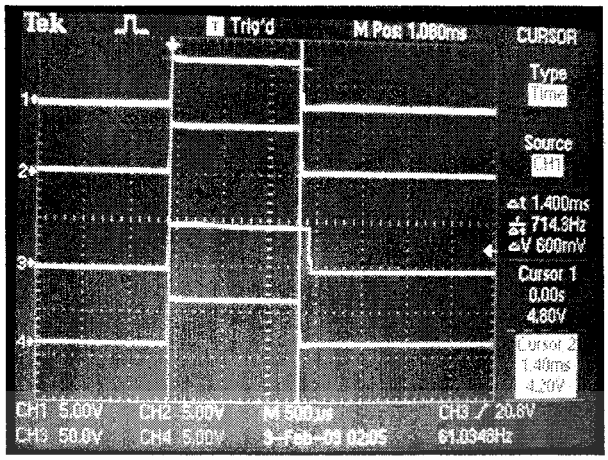
6. ผลการทดลอง

6.1 การสั่งงานโดยการป้อนคำสั่งผ่านคอมพิวเตอร์ครั้งละ 1 คำสั่ง เพื่อแสดงถึงค่า PWM ที่ส่งออกไปยังแขนกล



รูปที่ 6.1 แสดงหน้าต่างการส่งข้อมูลค่ามุมของมอเตอร์ 5 แกน ที่มุม 0 องศา (ตำแหน่งเริ่มต้น)

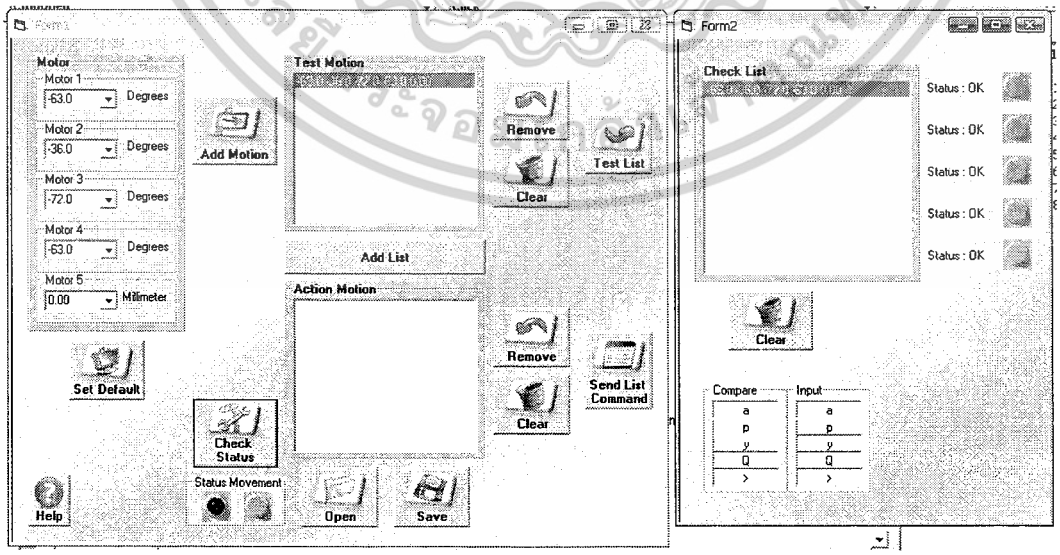
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 แสดงสัญญาณ PWM ของการส่งค่ามุมของมอเตอร์ที่มุม 0 องศา (ตำแหน่งเริ่มต้น)

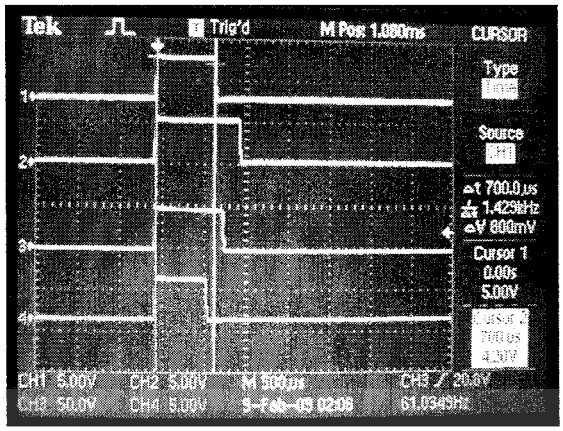
รูปที่ 6.1 แสดงถึงการสั่งงานที่ค่ากึ่งกลางมอเตอร์ทั้ง 5 แกน (มอเตอร์แกน X อยู่ที่มุม 0 องศา, มอเตอร์แกน Y อยู่ที่มุม 0 องศา, มอเตอร์แกน Z อยู่ที่มุม 0 องศา, มอเตอร์แกน P อยู่ที่มุม 0 องศา, มอเตอร์แกน Grip อยู่ที่มุม 0 องศา) เมื่อใช้ Oscilloscope จับสัญญาณของดีซีเซอร์โวมอเตอร์แกน X, Y, Z, และ P แสดงดังรูปที่ 6.2 ซึ่งเป็นสัญญาณ PWM ที่มีความกว้างพัลส์ออกมามีค่าที่แตกต่างกัน คือ (มอเตอร์แกน X สัญญาณ PWM เท่ากับ 1.4 ms, มอเตอร์แกน Y สัญญาณ PWM เท่ากับ 1.4 ms, มอเตอร์แกน Z สัญญาณ PWM เท่ากับ 1.5 ms, มอเตอร์แกน P สัญญาณ PWM เท่ากับ 1.4 ms)

* หมายเหตุ: เนื่องจาก Oscilloscope สามารถจับสัญญาณได้เพียงแค่ 4 ช่องสัญญาณเท่านั้น จึงไม่สามารถแสดงช่องสัญญาณที่ 5 ได้



รูปที่ 6.3 แสดงหน้าต่างการส่งข้อมูลค่ามุมของมอเตอร์ 5 แกน ที่ตำแหน่งต่ำสุด

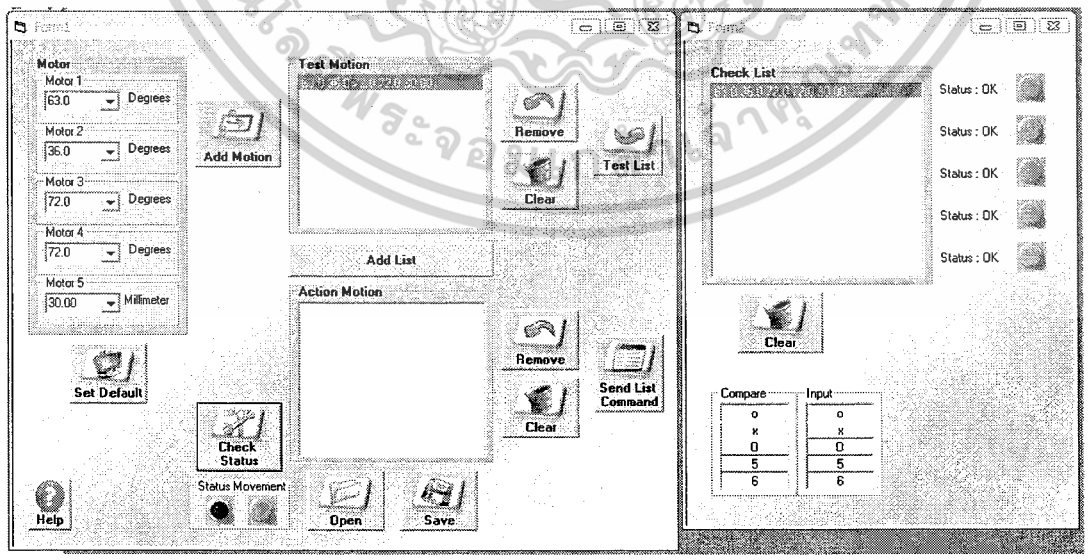
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 แสดงสัญญาณ PWM ของการส่งค่ามุมของมอเตอร์ที่ต่ำสุด

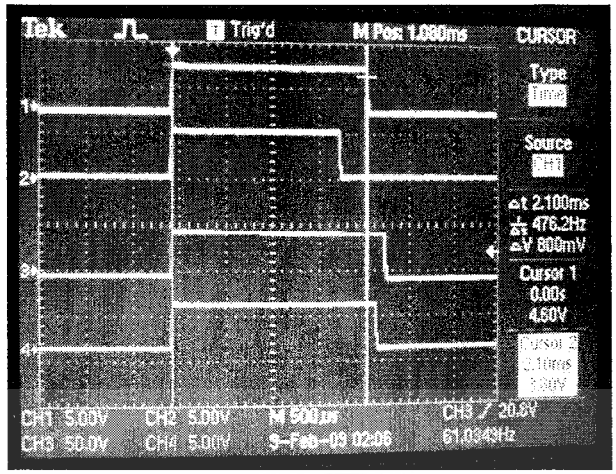
รูปที่ 6.3 แสดงถึงการสั่งงานที่จุดที่ต่ำที่สุดของมอเตอร์ทั้ง 5 แกน ที่สามารถเคลื่อนที่ไปได้ (มอเตอร์แกน X อยู่ที่มุม -63 องศา, มอเตอร์แกน Y อยู่ที่มุม -36 องศา, มอเตอร์แกน Z อยู่ที่มุม -72 องศา, มอเตอร์แกน P อยู่ที่ -63 องศา, มอเตอร์แกน Grip อยู่ที่ 0 องศา) เมื่อใช้ Oscilloscope จับสัญญาณของดีซีเซอร์โวมอเตอร์แกน X, Y, Z และ P ที่ออกมาจาก FPGA แสดงดังรูปที่ 6.4 ซึ่งเป็นสัญญาณ PWM ที่มีความกว้างพัลส์ออกมามีค่าที่แตกต่างกัน คือ (มอเตอร์แกน X สัญญาณ PWM เท่ากับ 0.7 ms, มอเตอร์แกน Y สัญญาณ PWM เท่ากับ 1 ms, มอเตอร์แกน Z สัญญาณ PWM เท่ากับ 0.8 ms, มอเตอร์แกนที่ P สัญญาณ PWM เท่ากับ 0.6 ms)

* หมายเหตุ: เนื่องจาก Oscilloscope สามารถจับสัญญาณได้เพียงแค่ 4 ช่องสัญญาณเท่านั้น จึงไม่สามารถแสดงช่องสัญญาณที่ 5 ได้



รูปที่ 6.5 แสดงหน้าต่างการส่งข้อมูลค่ามุมของมอเตอร์ 5 แกน ที่สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 แสดงสัญญาณ PWM ของการส่งค่ามุมของมอเตอร์สูงสุด

รูปที่ 6.5 แสดงถึงการสั่งงานที่จุดที่ต่ำที่สุดของมอเตอร์ทั้ง 5 แกน ที่สามารถเคลื่อนที่ไปได้ (มอเตอร์แกน X อยู่ที่มุม 63 องศา, มอเตอร์แกน Y อยู่ที่มุม 36 องศา, มอเตอร์แกน Z อยู่ที่มุม 72 องศา, มอเตอร์แกน P อยู่ที่มุม 72 องศา, มอเตอร์แกน Grip อยู่ที่มุม 30 องศา) เมื่อใช้ Oscilloscope จับสัญญาณของสัญญาณของดีซีเซอร์โวมอเตอร์แกน X, Y, Z และ P ที่ออกมาจาก FPGA แสดงดังรูปที่ 6.6 ซึ่งเป็นสัญญาณ PWM ที่มีความกว้างพัลส์ออกมามีค่าที่แตกต่างกัน คือ (มอเตอร์แกน X สัญญาณ PWM เท่ากับ 2.1 ms, มอเตอร์แกน Y สัญญาณ PWM เท่ากับ 1.8 ms, มอเตอร์แกน Z สัญญาณ PWM เท่ากับ 2.3 ms, มอเตอร์แกน P สัญญาณ PWM เท่ากับ 2.2 ms)

* หมายเหตุ: เนื่องจาก Oscilloscope สามารถจับสัญญาณได้เพียงแค่ 4 ช่องสัญญาณเท่านั้น จึงไม่สามารถแสดงช่องสัญญาณที่ 5 ได้

ตารางที่ 6.1 แสดงช่วงความกว้างพัลส์และระยะในการเคลื่อนที่ของมอเตอร์แต่ละแกน

มอเตอร์	รุ่นมอเตอร์	ช่วงความกว้างพัลส์ที่ใช้	ระยะในการเคลื่อนที่
M1	S666STD	0.7 ms-2.1 ms	126 องศา
M2	S666STD	1 ms-1.8 ms	72 องศา
M3	S04BBM	0.7 ms-2.3 ms	144 องศา
M4	S03TSTD	0.6 ms-2.2 ms	144 องศา
M5	S03TSTD	1.4 ms-2.2 ms	3 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

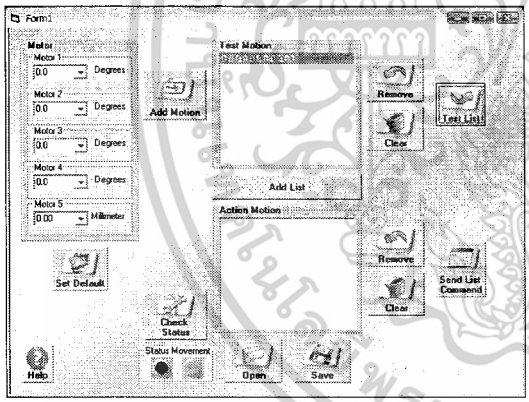
6.2 การสั่งงานโดยการป้อนคำสั่งผ่านคอมพิวเตอร์ครั้งละ 1 ชุดคำสั่ง เพื่อตรวจสอบระยะและทิศทางการหมุนของแขนกล

การทดลองสั่งงานแขนกลที่ละคำสั่งของแกน Z (มอเตอร์ M3) โดยกำหนดให้ตำแหน่งเริ่มต้นที่ช่วงความกว้างพัลส์ 0.7 ms ถึงช่วงความกว้างพัลส์ที่ 2.3 ms ดังรูปที่ 6.7-6.12 เพื่อเป็นการแสดงระยะการหมุนและการเคลื่อนที่ของแขนกลในแกน Z ว่ามีระยะในการเคลื่อนที่อยู่ที่เท่าใดเหมาะสมกับการใช้งานของผู้ใช้งานหรือไม่

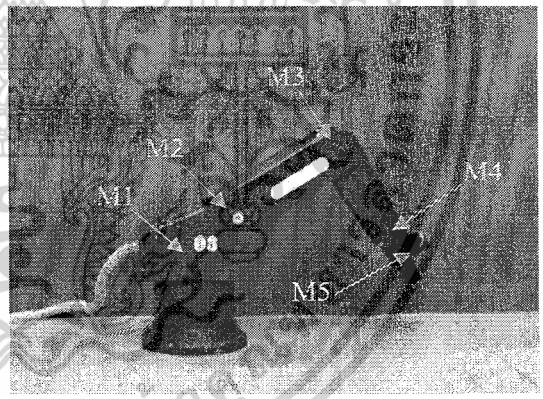
ตารางที่ 6.2 ช่วงความกว้างพัลส์และระยะการเคลื่อนที่ของมอเตอร์ (M3) แกน Z

ช่วงความกว้างพัลส์	ระยะในการเคลื่อนที่	หมายเหตุ
0.7 ms	-72 องศา	รูปที่ 6.8
1.5 ms	0 องศา	รูปที่ 6.10
2.3 ms	72 องศา	รูปที่ 6.12

แสดงรูปแขนกลที่ตำแหน่งเริ่มต้น (ตำแหน่งอ้างอิง)



รูปที่ 6.7 หน้าต่างการส่งข้อมูลค่ามุม 0 องศา

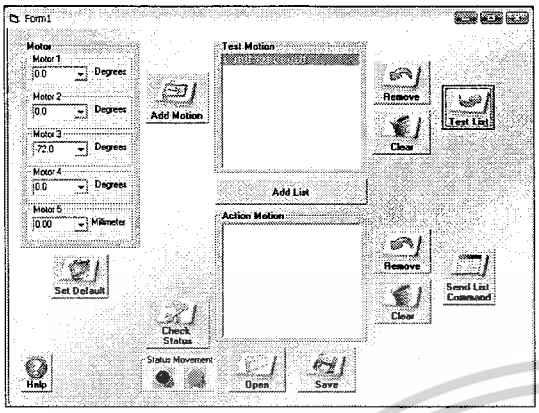


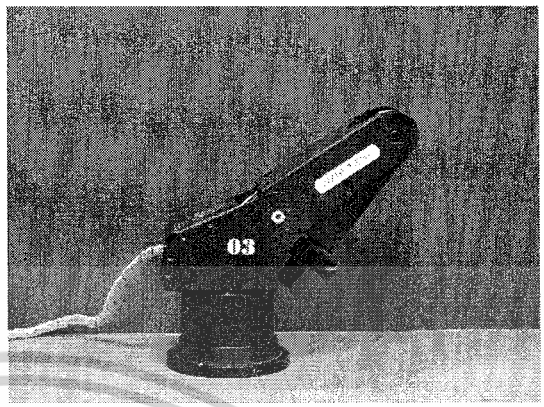
รูปที่ 6.8 แสดงการเคลื่อนที่ที่ค่ามุมที่ 0 องศา

จากรูปที่ 6.7-6.8 นั้นเป็นการสั่งงานจากคอมพิวเตอร์เพื่อสั่งงานแขนกลให้เคลื่อนที่ไปยังค่าที่กำหนด ซึ่งได้แสดงถึงระยะมุมที่แขนกลในระยะทางของแกน Z ซึ่งเป็นจุดกึ่งกลางระหว่างจุดที่ต่ำที่สุดของแขนกลที่สามารถเคลื่อนที่ไปได้ (-72 องศา) กับจุดที่แขนกลสามารถเคลื่อนที่ไปได้สูงที่สุดของแขนกลในแกน Z ที่สามารถเคลื่อนที่ไปได้ (72 องศา) ตำแหน่งอ้างอิงนี้คือแกน Z ที่มุม 0 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงรูปแขนกลที่ตำแหน่งต่ำสุด

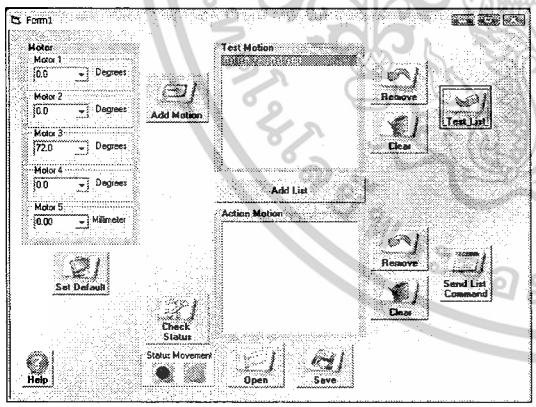


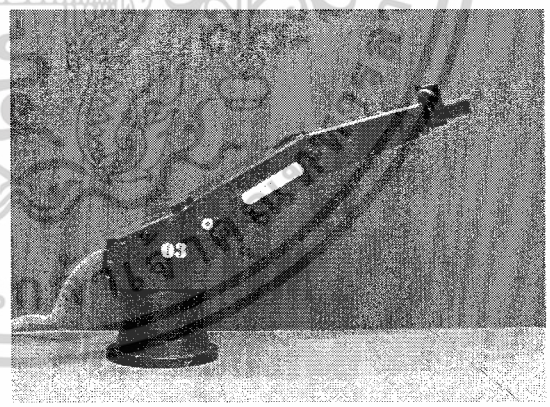


รูปที่ 6.9 หน้าต่างการส่งข้อมูลค่ามุม -72 องศา **รูปที่ 6.10** แสดงการเคลื่อนที่ที่ค่ามุม -72 องศา

จากรูปที่ 6.9-6.10 นั้นเป็นการสั่งงานจากคอมพิวเตอร์เพื่อสั่งงานแขนกลให้เคลื่อนที่ไปยังค่าที่กำหนด ซึ่งแสดงถึงระยะมุมที่แขนกลในระยะทางของแกน Z ซึ่งเป็นจุดที่ต่ำที่สุดของแขนกลในแกน Z ที่สามารถเคลื่อนที่ไปได้นั้นคือแกน Z ที่มุม -72 องศา

แสดงรูปแขนกลที่ตำแหน่งสูงสุด





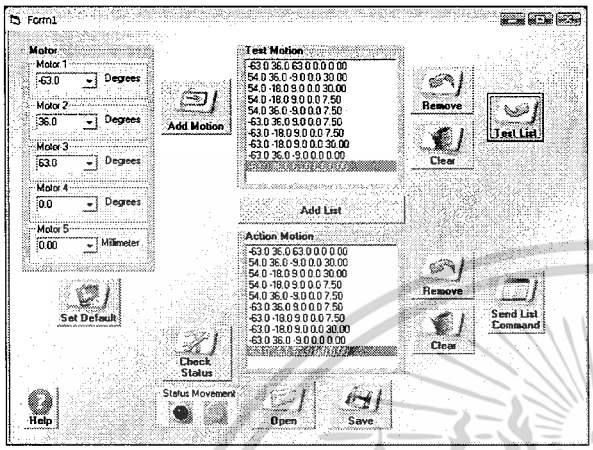
รูปที่ 6.11 หน้าต่างการส่งข้อมูลค่ามุม 72 องศา **รูปที่ 6.12** แสดงการเคลื่อนที่ที่ค่ามุม 72 องศา

จากรูปที่ 6.11-6.12 นั้นเป็นการสั่งงานจากคอมพิวเตอร์เพื่อสั่งงานแขนกลให้เคลื่อนที่ไปยังค่าที่กำหนด ซึ่งได้แสดงถึงระยะมุมที่แขนกลในระยะทางของแกน Z ซึ่งเป็นจุดที่สูงที่สุดของแขนกลในแกน Z ที่สามารถเคลื่อนที่ไปได้นั้นคือแกน Z ที่มุม 72 องศา

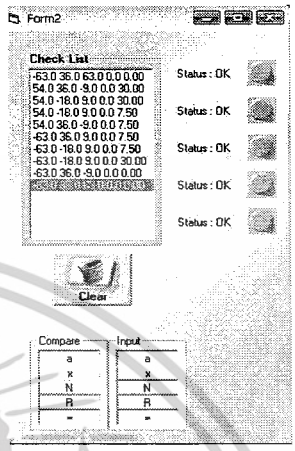
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การสั่งงานแบบหลาย ๆ ชุดคำสั่งต่อเนื่องกัน (แบบอัตโนมัติ)

จากรูปที่ 6.13-6.14 เป็นการสั่งงานจากคอมพิวเตอร์ให้แขนกลทำงาน โดยการนำชิ้นงานที่ต้องการหยิบนำไปวางไว้ในตำแหน่งที่ผู้ใช้งานต้องการ และแสดงการเคลื่อนที่ดังรูปที่ 6.15-6.22

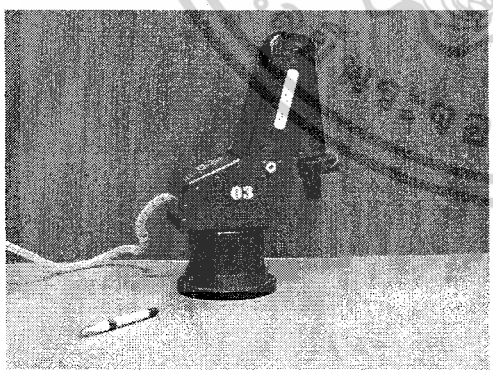


รูปที่ 6.13 หน้าต่างการส่งค่ามุมการเคลื่อนที่แบบอัตโนมัติ

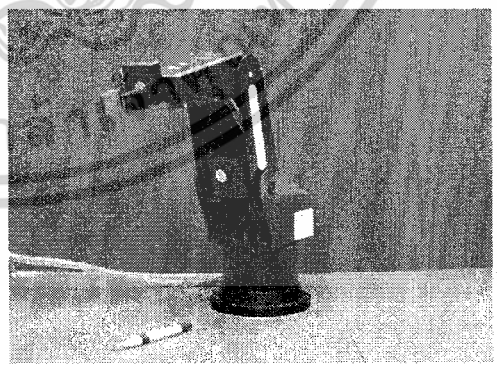


รูปที่ 6.14 หน้าต่างตรวจสอบข้อมูลที่ส่งไป FPGA

หลังจากได้ทำการส่งค่าเพื่อให้แขนกลหยิบวัตถุออกไปแล้ว ก็จะมีลักษณะของการเคลื่อนที่ตามรูปที่ 6.15-6.22 ซึ่งเป็นรูปของการสั่งงานเพื่อให้แขนกลทำการหยิบวัตถุโดยเป็นการสั่งงานของแขนกลเพียงครั้งเดียวแขนกลก็จะทำงานเป็นแบบอัตโนมัติ ซึ่งเราสามารถตรวจสอบความถูกต้องของแขนกลได้ตามรูปที่ 6.14

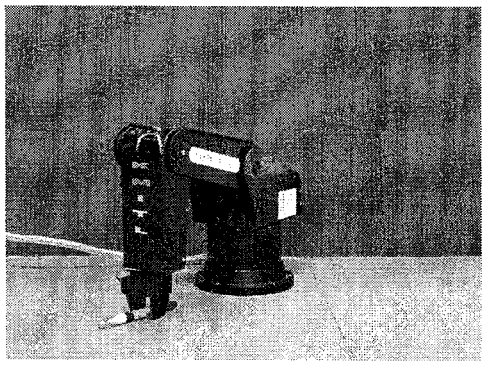


รูปที่ 6.15 การเคลื่อนที่แขนกลครั้งที่ 1

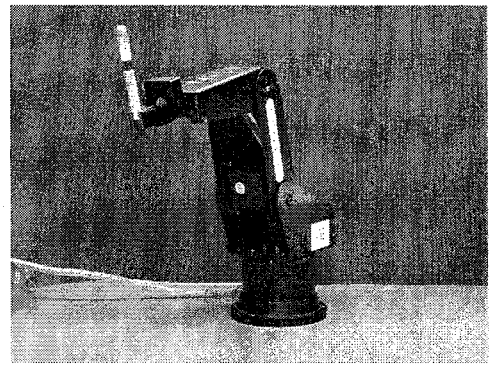


รูปที่ 6.16 การเคลื่อนที่แขนกลครั้งที่ 2

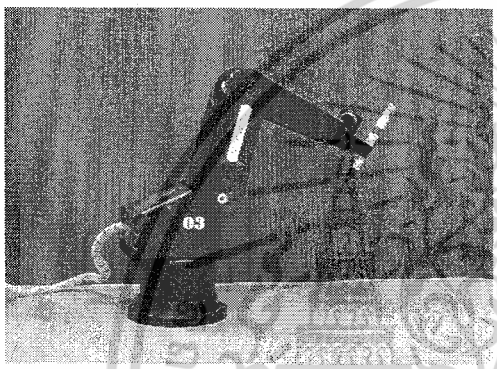
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



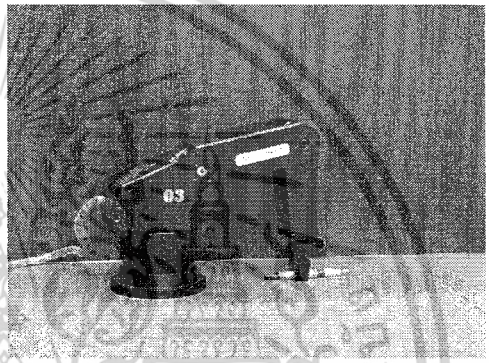
รูปที่ 6.17 การเคลื่อนที่แขนกลครั้งที่ 3



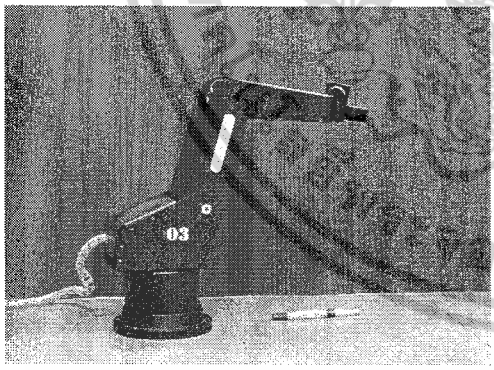
รูปที่ 6.18 การเคลื่อนที่แขนกลครั้งที่ 4



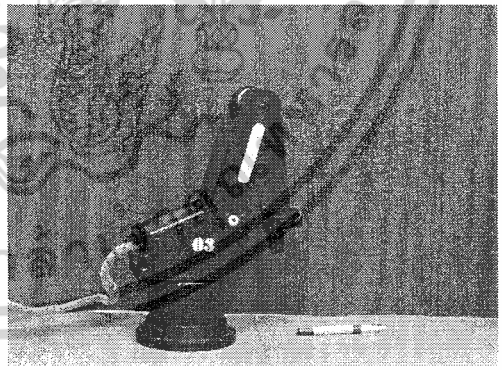
รูปที่ 6.19 การเคลื่อนที่แขนกลครั้งที่ 5



รูปที่ 6.20 การเคลื่อนที่แขนกลครั้งที่ 6



รูปที่ 6.21 การเคลื่อนที่แขนกลครั้งที่ 7



รูปที่ 6.22 การเคลื่อนที่แขนกลครั้งที่ 8

จากรูปที่ 6.13 นั้นจะเป็นการกำหนดการเคลื่อนที่ของแขนกลให้แขนกลสามารถเคลื่อนที่ไปได้อย่างอัตโนมัติ โดยการกำหนดระยะการเคลื่อนที่ มุมของการเคลื่อนที่ของมอเตอร์แต่ละแกนมาก่อนแล้วนำค่าต่าง ๆ มาสั่งงานภายในกรอบการเคลื่อนที่ Action Motion แล้วทำการสั่งงานเพียง 1 ครั้ง ก็จะสามารถทำการส่งค่าต่าง ๆ ออกไปเป็นชุดคำสั่งต่าง ๆ ทีละบรรทัดจนจบ เพื่อให้แขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนที่ตามชุดคำสั่งทีละชุด ซึ่งเป็นค่าของผู้ใช้งานที่ได้ทำการกำหนดทิศทางการเคลื่อนที่ของแขนกลไว้ล่วงหน้าแล้ว

การเคลื่อนที่หรือการเปลี่ยนแปลงตำแหน่งของค่ามุมอื่น ๆ นั้นสามารถทำได้โดยการเลือกค่ามุมของมอเตอร์แต่ละแกน กำหนดช่วงกว้างพัลส์ให้เพื่อให้สอดคล้องกับตำแหน่งที่ต้องการ และทุกแกนหมุนก็จะสามารถกำหนดการเคลื่อนที่ได้เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผลการวิจัยและข้อเสนอแนะ

7.1 สรุปผลการวิจัยและข้อเสนอแนะ

การออกแบบควบคุมการเคลื่อนที่ของดีซีเซอร์โวมอเตอร์หลายแกนในแขนกลด้วย FPGA ที่เป็นตัวสร้างสัญญาณขับเคลื่อนแบบหลายช่องสัญญาณพร้อมกัน สามารถที่จะสั่งงานและควบคุมการทำงานได้อย่างเที่ยงตรงแม่นยำตามที่กำหนดตำแหน่ง ความเร็ว และรูปแบบของการเคลื่อนที่ สามารถทำได้โดยการใช้ภาษา Verilog HDL ในการเขียนโปรแกรมควบคุมการทำงานลงในบอร์ด FPGA ซึ่งในการทดลองนี้นั้นจะใช้โปรแกรม Visual Basic 6 ติดต่อกับบอร์ด FPGA เพื่อควบคุมการเคลื่อนที่ของ แขนกลให้เคลื่อนที่หยิบจับชิ้นงานตามที่ต้องการได้

โครงงานนี้ประสบผลสำเร็จในระดับที่น่าพอใจ โดยเป็นไปตามจุดประสงค์และเป้าหมายของโครงการที่ได้วางแผนไว้ในตอนต้นของโครงการ จากการศึกษาการทำงานของแขนกลที่ใช้ดีซี-เซอร์โวมอเตอร์รวมถึงการออกแบบโปรแกรมที่ใช้ในการควบคุมให้แขนกลเคลื่อนที่ และส่วนของอินเตอร์เฟสบนคอมพิวเตอร์ที่ได้ใช้ Visual Basic 6 ในการออกแบบเพื่อให้แขนกลสามารถขยับได้ตามคำสั่ง ที่สามารถทำได้โดยเชื่อมต่อผ่านคอมพิวเตอร์ด้วยการสั่งงานครั้งละคำสั่งหรือการสั่งงานเป็นชุดคำสั่งโดยการเขียนเก็บไว้เป็น List ไฟล์ สามารถที่จะนำไปประยุกต์ใช้หรือสร้างเป็นแขนกลสำหรับงานทางด้านอุตสาหกรรมต่อไป

7.2 ปัญหาและอุปสรรคในการทำงาน

การใช้งานของแขนกลที่ออกแบบมีความละเอียดที่ยังไม่ดีพอ ซึ่งได้ออกแบบโปรแกรมการรับส่งข้อมูลระหว่าง FPGA กับ คอมพิวเตอร์ใช้การรับส่งข้อมูลอนุกรมขนาด 8 บิต และกำหนดความละเอียดในการเคลื่อนที่ของแขนกลไว้ที่ 9 องศา เพื่อความสะดวกในการเขียนโปรแกรม และทำให้ง่ายในการทดลอง ดังนั้น แขนกลจึงไม่สามารถใช้งานได้ในความละเอียดทุกย่านองศา และแขนกลไม่สามารถหยิบชิ้นงานที่มีน้ำหนักมาก ๆ ได้ เนื่องจากดีซีเซอร์โวมอเตอร์ที่ใช้ไม่สามารถรับโหลดที่สูงมากได้

7.3 แนวทางการแก้ไข

ในการปรับปรุงความละเอียด และความแม่นยำของแขนกล ควรจะเพิ่มบิตในการรับส่งข้อมูลเป็นขนาด 16 บิต หรือ 32 บิต เพื่อให้สามารถประยุกต์ใช้รหัส ASCII ได้มากขึ้น หรือปรับปรุงโปรแกรมใน FPGA เพื่อให้รับข้อมูลที่มีความกว้างของชุดข้อมูลมากขึ้น ซึ่งจะทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบโปรแกรมที่มีความละเอียดในการเคลื่อนที่ของแขนกลได้ และแนวทางแก้ไขการรับโหลดของดีซีเซอร์โวมอเตอร์นั้น ควรเลือกใช้ดีซีเซอร์โวมอเตอร์ที่มีค่าแรงบิดสูง ๆ (Torque) และมีขนาดที่เหมาะสมกับแขนกล เพื่อที่จะสามารถนำมาใช้ทดแทนดีซีเซอร์โวมอเตอร์ตัวเดิมได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Sawai Pongswatd, Ruedee Masuchun, Krit Smerpitak and Prapart Ukakimapurn **New Technique to Generate the PWM Signal** ICCAS2004 International Conference on Control, Automatic and Systems, Shangari-La Hotel, Bangkok, Thailand, August 25-27, 2004.

[2] Muhammad H. Rashid, **Power Electronics**, Prentice Hall International, Inc., 1988.

[3] ไสว พงศ์สวัสดิ์. **อิเล็กทรอนิกส์กำลัง.** ภาควิชาวิศวกรรมการวัดคุม. คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

[4] Astron Logic Research & Development Co. Ltd. (2549). **Power ACEX1K User's Manual.**

[5] ธีรยศ เวียงทอง. (2548). **เรียนรู้การออกแบบระบบดิจิตอลด้วยภาษา Verilog เบื้องต้น.** ภาควิชาอิเล็กทรอนิกส์. มหาวิทยาลัยเทคโนโลยีมหานคร.

[6] วัชรินทร์ เดารพ. (2546). **คู่มือการใช้งาน Servo Motor พร้อมตัวอย่างโปรแกรม.** บริษัท อีทีที จำกัด. กรุงเทพฯ. (www.ett.co.th).

[7] สัจจะ จรัสรุ่งรวีวร. (2548). **คู่มือมือเขียนโปรแกรม Visual Basic 6.** สำนักพิมพ์ DEV BOOK (พิมพ์ครั้งที่ 3). นนทบุรี.

[8] อภิชาติ ภู่พรับ. (2546). **การควบคุมฮาร์ดแวร์ด้วย Visual Basic.** สำนักพิมพ์ DEV BOOK (พิมพ์ครั้งที่ 1). นนทบุรี.

[9] ทีมวิชาการโรงเรียนแสงทองอิเล็คทรอนิคส์. (2547). **การออกแบบแผ่นวงจรพิมพ์ด้วยโปรแกรม Protel 99 SE.** หอรัตนชัยการพิมพ์. กรุงเทพฯ.

[10] http://www.etteam.com/download/SERVO_MOTOR/MANUAL.zip

[11] http://www.astronlogic.com/

[12] http://th.wikipedia.org/wiki/FPGA

[13] http://www.todaystep.com/wiki/doku.php?id=wiki:fpga

[14] http://www.es.co.th/Schemetic/PDF/AST-ACEX1K.PDF

[15] http://www.aldec.com/products/tutorials/

[16] Visual Basic Programming Interface Hardware (online) Available:http://www.thaiio.com/Programing/Vbpage.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

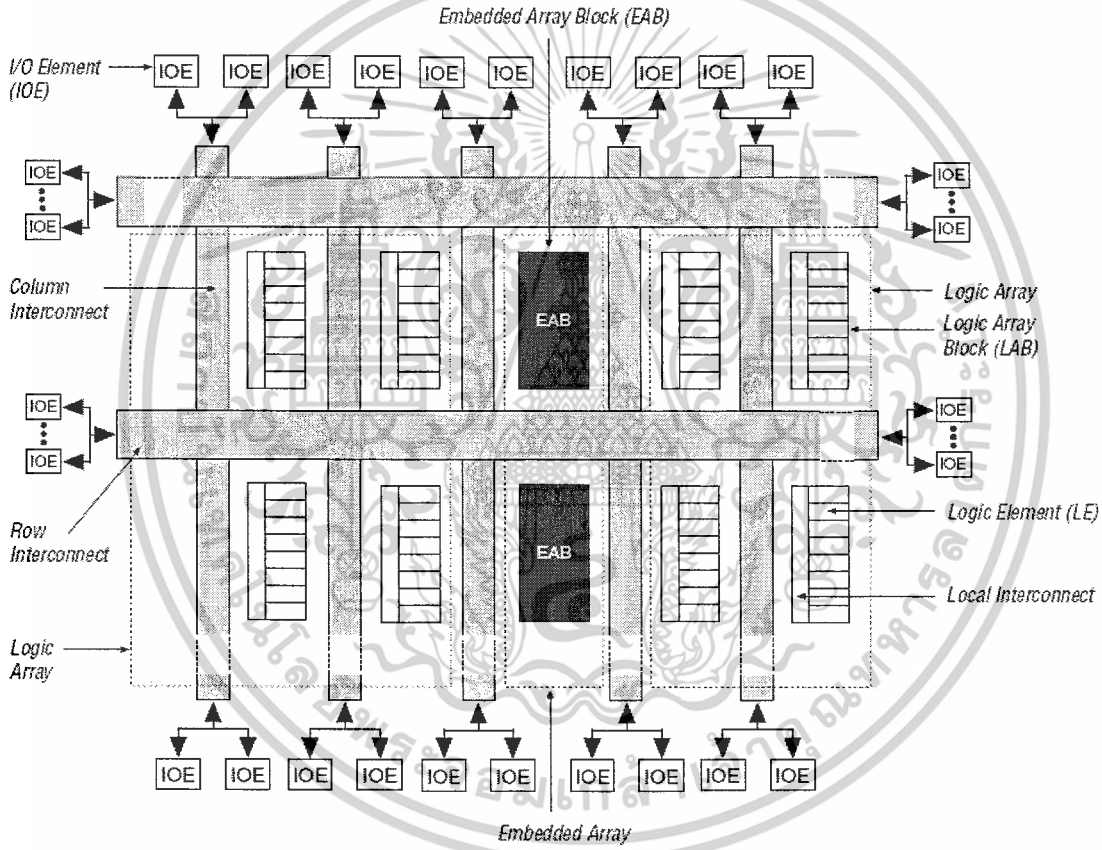
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.1 สถาปัตยกรรมภายในของ FPGA (Field Programmable Gate Arrays)

ในการทดลองนี้ได้ใช้ FPGA เบอร์ EP1k50TC144-3 ในตระกูล Power ACEX 1k-50

ก.1.1 รายละเอียดของ POWER ACEX1K SERIES

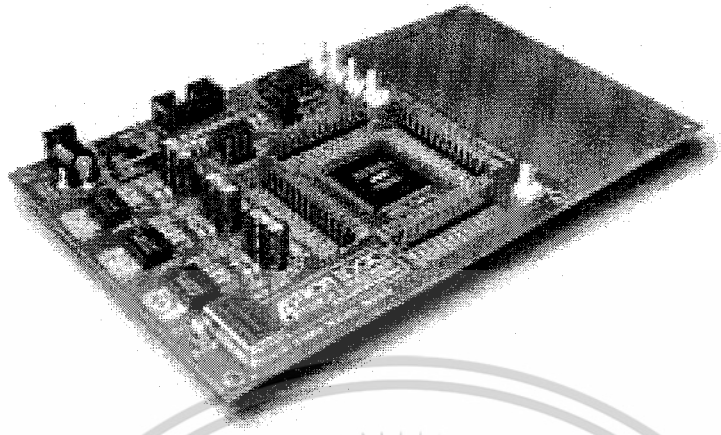
FPGA ตระกูล ACEX1K ของบริษัท ALTERA เป็น FPGA ที่มีโครงสร้างภายในเป็นแบบ SRAM – Based FPGA ใช้เทคโนโลยีในการโปรแกรมเหมือนกับหน่วยความจำแบบ SRAM (Static RAM) ทำให้การโปรแกรมสามารถทำซ้ำได้โดยไม่จำกัดจำนวนครั้ง และใช้เวลาในการโปรแกรมชิป FPGA น้อยมาก (ระดับ n Sec) การโปรแกรมทำได้ง่ายเช่นเดียวกับการเขียน SRAM ทั่วไป แต่มีข้อเสียคือ มันต้องการไฟเลี้ยงในการเก็บวงจรที่ได้ออกแบบไว้



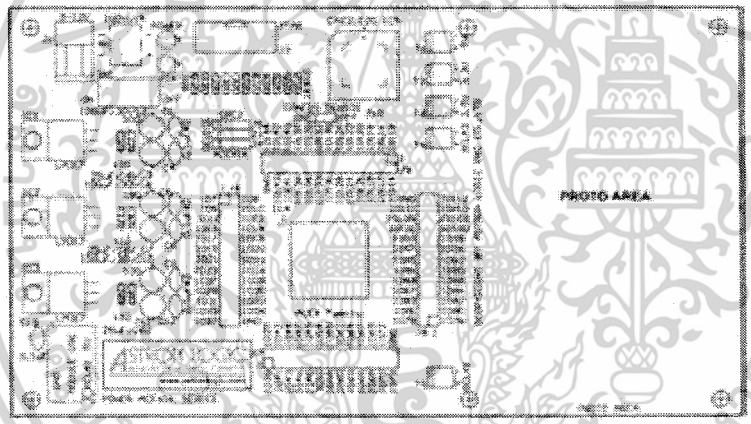
รูปที่ ก.1 โครงสร้างภายในของ FPGA ตระกูล ACEX1K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ POWER ACEX1K SERIES



รูปที่ ก.2 POWER ACEX1K SERIES



รูปที่ ก.3 ลักษณะการจัดวางอุปกรณ์ต่างๆ บน POWER ACEX1K SERIES

ตารางที่ 1 คุณลักษณะต่างๆของชิป FPGA ที่ใช้กับบอร์ดในกลุ่ม POWER ACEX1K SERIES

Feature	EP1K10	EP1K30	EP1K50	EP1K100
Typical gates	10,000	30,000	50,000	100,000
Maximum System gates	56,000	119,000	199,000	257,000
Logic element (LEs)	576	1,728	2,880	4,992
EABs	3	6	10	12
Total RAM bits	12,288	24,576	40,960	49,152
Maximum user I/O pins	136	171	249	333

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.1.2 JTAG CONNECTOR

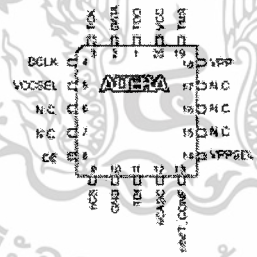
JTAG Connector เป็น Header ตัวผู้แบบ 5 X 2 ขา ใช้สำหรับต่อกับสาย Byte Blaster MV เพื่อสำหรับดาวน์โหลด ข้อมูลทางลอจิกของวงจรที่ได้ออกแบบไว้จากคอมพิวเตอร์ลงสู่ชิป FPGA

ในการโปรแกรมจะมี Jumper ให้เลือกว่าจะโปรแกรมลงในชิป FPGA หรือโปรแกรมลง Configuration Device

ในกรณีที่ต้องการโปรแกรมลงชิป FPGA ACEX 1K สามารถดูวิธีการ Complier และโปรแกรมลง Configuration Device ด้วยโปรแกรม MAX+PLUS II ให้ดูวิธีการใช้ MAX+PLUS II

ก.1.3 Configuration Device Socket

Socket แบบ PLCC 20 ขา สำหรับใส่ Configuration Device ซึ่งเป็นอุปกรณ์แบบ Flash Memory ใช้สำหรับเก็บข้อมูลวงจรที่เราได้ออกแบบไว้ เมื่อเริ่มต้นจ่ายไฟให้แก่บอร์ด POWER ACEX 1K SERIES วงจรที่อยู่ภายใน Configuration Device จะถูกโหลดลงไปเก็บไว้ในชิป FPGA โดยอัตโนมัติ Configuration Device ที่ใช้ร่วมกับ FPGA แบบ SRAM-BASE FPGA สำหรับเบอร์ของ Configuration Device ที่ใช้กับบอร์ด POWER ACEX 1K SERIES จะเป็นเบอร์ EPC2LI20 หรือ EPC2LC20 มีขนาด 1,695,680 X 1 Bit สามารถใช้กับแรงดัน 5.0 V หรือ 3.3 V ก็ได้ ในกรณีที่ไม่ต้องการให้ FPGA โหลดวงจรจาก Configuration Device ให้เลือก JUMPER J10 มาที่ตำแหน่ง OFF ลักษณะ และตำแหน่งขาของ EPC2LI20 และ EPC2LC20 แสดงดังรูปที่ ก.4

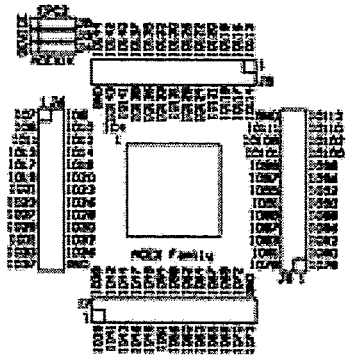


รูปที่ ก.4 ตำแหน่งขาของ EPC2LI20 และ EPC2LC20

ก.1.4 พอร์ตขยายช่องสัญญาณแบบ Header 13 X 2 ขา จำนวน 4 ตัว

บอร์ด POWER ACEX 1K SERIES แต่ละรุ่นจะมีพอร์ตขยายช่องสัญญาณ J6, J7, J8 และ J9 ให้ใช้งานได้อย่างอิสระ 4 พอร์ต แต่ละพอร์ตจะมีขนาด 13 X 2 ขา ต่อกับขา I/O ของ FPGA โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 ตำแหน่งการจัดวาง J6 – J9

ตารางที่ 2 ความสัมพันธ์ของ J6 กับตำแหน่ง I/O ของ ACEX1K FPGA

J6 Pin Number	ACEX1K Pin Number	J6 Pin Number	ACEX1K Pin Number
1	7 (I/O)	2	8 (I/O)
3	9 (I/O)	4	10 (I/O)
5	11 (I/O)	6	12 (I/O)
7	13 (I/O)	8	14 (I/O)
9	17 (I/O)	10	18 (I/O)
11	19 (I/O)	12	20 (I/O)
13	21 (I/O)	14	22 (I/O)
15	23 (I/O)	16	26 (I/O)
17	27 (I/O)	18	28 (I/O)
19	29 (I/O)	20	30 (I/O)
21	31 (I/O)	22	32 (I/O)
23	33 (I/O)	24	36 (I/O)
25	37 (I/O)	26	GND

ตารางที่ 3 ความสัมพันธ์ของ J7 กับตำแหน่ง I/O ของ ACEX1K FPGA

J7 Pin Number	ACEX1K Pin Number	J7 Pin Number	ACEX1K Pin Number
1	38 (I/O)	2	39 (I/O)
3	41 (I/O)	4	42 (I/O)
5	43 (I/O)	6	44 (I/O)
7	46 (I/O)	8	47 (I/O)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9	48 (I/O)	10	49 (I/O)
11	51 (I/O)	12	54 (Ded. Input)
13	56 (Ded. Input)	14	59 (I/O)
15	60 (I/O)	16	62 (I/O)
17	63 (I/O)	18	64 (I/O)
19	65 (I/O)	20	67 (I/O)
21	68 (I/O)	22	69 (I/O)
23	77 (I/O)	24	72 (I/O)
25	73 (I/O)	26	GND

ตารางที่ 4 ความสัมพันธ์ของ J8 กับตำแหน่ง I/O ของ ACEX1K FPGA

J8 Pin Number	ACEX1K Pin Number	J8 Pin Number	ACEX1K Pin Number
1	78 (I/O)	2	79 (I/O)
3	80 (I/O)	4	81 (I/O)
5	82 (I/O)	6	83 (I/O)
7	86 (I/O)	8	87 (I/O)
9	88 (I/O)	10	89 (I/O)
11	90 (I/O)	12	91 (I/O)
13	92 (I/O)	14	95 (I/O)
15	96 (I/O)	16	97 (I/O)
17	98 (I/O)	18	99 (I/O)
19	100 (I/O)	20	101 (I/O)
21	102 (I/O)	22	109 (I/O)
23	110 (I/O)	24	111 (I/O)
25	112 (I/O)	26	GND

ตารางที่ 5 ความสัมพันธ์ของ J9 กับตำแหน่ง I/O ของ ACEX1K FPGA

J9 Pin Number	ACEX1K Pin Number	J9 Pin Number	ACEX1K Pin Number
1	113 (I/O)	2	114 (I/O)
3	116 (I/O)	4	117 (I/O)
5	118 (I/O)	6	119 (I/O)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	120 (I/O)	8	121 (I/O)
9	122 (I/O)	10	124 (Ded. Input)
11	126 (Ded. Input)	12	128 (I/O)
13	130 (I/O)	14	131 (I/O)
15	132 (I/O)	16	133 (I/O)
17	135 (I/O)	18	136 (I/O)
19	137 (I/O)	20	138 (I/O)
21	140 (I/O)	22	141 (I/O)
23	142 (I/O)	24	143 (I/O)
25	144 (I/O)	26	GND

ก.1.5 Module Oscillator

บอร์ด POWER ACEX 1K SERIES มีโมดูลออสซิเลเตอร์ความถี่ 1 MHz (สามารถถอดเปลี่ยนเป็นค่าความถี่ที่ต้องการได้) เป็นแหล่งกำเนิดความถี่สำหรับใช้ในการออกแบบวงจรที่ต้องการอ้างอิงกับสัญญาณนาฬิกา โดยการป้อนเข้าที่ขา 55 (GCLK1) ของ FPGA



รูปที่ ก.6 พื้นที่สำหรับ Module Oscillator

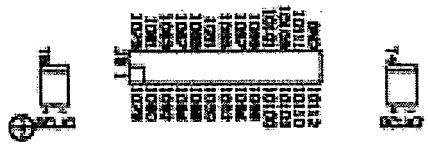
ตารางที่ 6 ความสัมพันธ์ระหว่างสัญญาณ Oscillator กับตำแหน่งขาของ ACEX1K FPGA

Module Oscillator Pin Number	ACEX1K Pin Number
CLK	55 (GLK1)

ก.2.6 Terminal แบบ 2 ขั้ว สำหรับ GCLK1 และ GCLK2

Terminal ตัวผู้แบบ 2 ขั้ว ใช้สำหรับต่อกับสัญญาณนาฬิกาจากภายนอกเพื่อป้อนให้แก่ FPGA ทางขา Global Clock 1 (GCLK1) หรือขา Global Clock 2 (GCLK2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



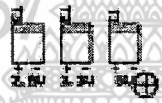
รูปที่ ก.7 การจัดวาง T4 และ T5

ตารางที่ 7 ความสัมพันธ์ของ T4, T5 กับตำแหน่งขาของ ACEX1K FPGA

T4 Pin Number	ACEX1K Pin Number	T5 Pin Number	ACEX1K Pin Number
1 (+)	125 (GCLK2)	1 (+)	55 (GCLK1)
2 (-)	GND	2 (-)	GND

ก.2.7 Terminal แบบ 2 ขั้ว สำหรับแรงดัน 2.5 โวลท์, 3.3 โวลท์ และ 5.0โวลท์

Terminal ตัวผู้แบบ 2 ขั้ว สำหรับต่อกับแหล่งจ่ายแรงดันภายในบอร์ด POWER ACEX1K SERIES เพื่อเป็นแหล่งจ่ายแรงดันไฟตรงให้กับวงจรภายนอก โดยมีแรงดันให้เลือกใช้งาน 3 ระดับ คือ 2.5 โวลท์, 3.3 โวลท์ และ 5.0 โวลท์



รูปที่ ก.8 ตำแหน่งของ T1 ,T2 และ T3

ตารางที่ 8 ความสัมพันธ์ของ T1, T2 และ T3 กับแหล่งจ่ายแรงดันภายในบอร์ด POWER ACEX1K SERIES

T1 Pin number	Volt	T2 Pin number	Volt	T3 Pin Number	Volt
1 (+)	5	1 (+)	3.3	1 (+)	2.5
2 (-)	GND	2 (-)	GND	2 (-)	GND

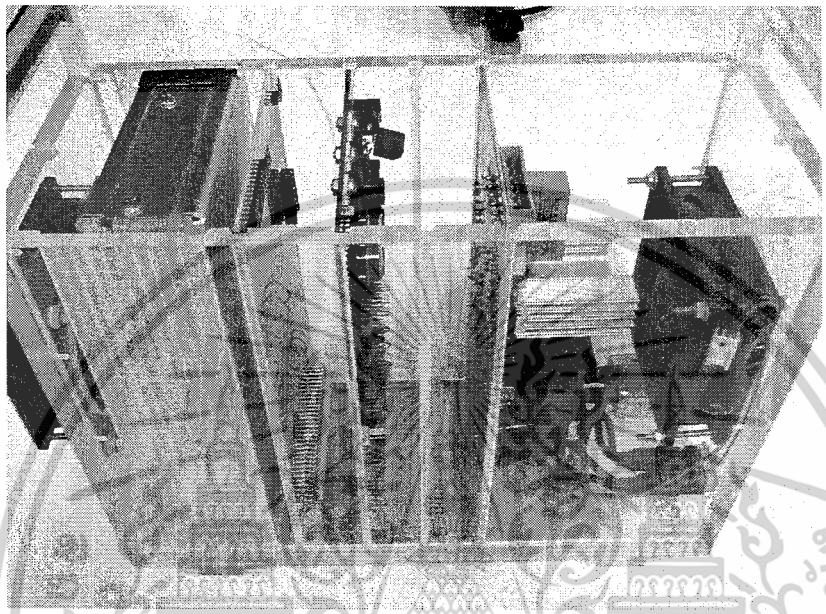
ก.2.8 คอนเน็คเตอร์แหล่งจ่ายไฟ

บอร์ด POWER ACEX1K SERIES ถูกออกแบบมาให้ใช้กับไฟกระแสตรงแรงดัน 9 โวลท์ ถึง 12 โวลท์ โดยแกนในของคอนเน็คเตอร์ J1 เป็นขั้วบวกและแกนนอกเป็นขั้วลบ นอกจากนี้ยังมีวงจรกลับขั้วแรงดันสำหรับในกรณีที่มีการป้อนแรงดันผิดขั้วให้แก่บอร์ดอีกด้วย

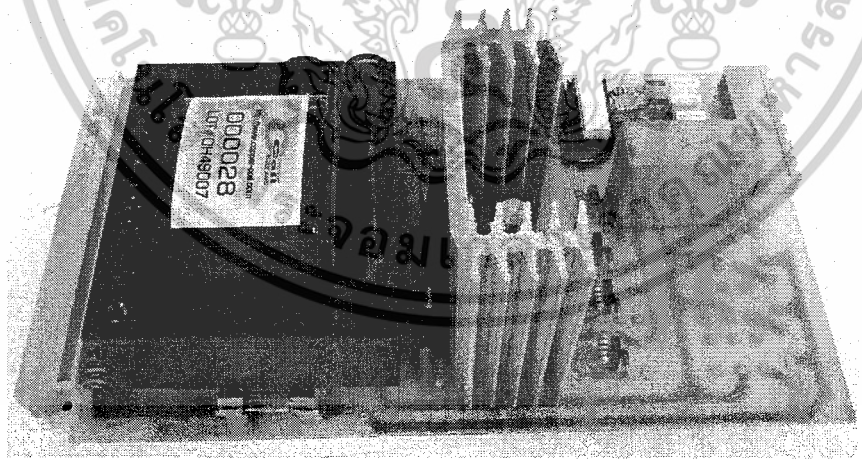
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. ส่วนของแหล่งจ่ายไฟ

ส่วนของแหล่งจ่ายไฟที่ใช้จ่ายสัญญาณไฟฟ้าให้กับบอร์ด FPGA และดีซีเซอร์โวมอเตอร์ นั้นได้ใช้แหล่งจ่ายไฟของ MEAN WELL รุ่น S-50-5 ให้ Input : 100-120 VAC (1.30A), 200-240 VAC (0.65A) 50/60 Hz Output : +5VDC 10A



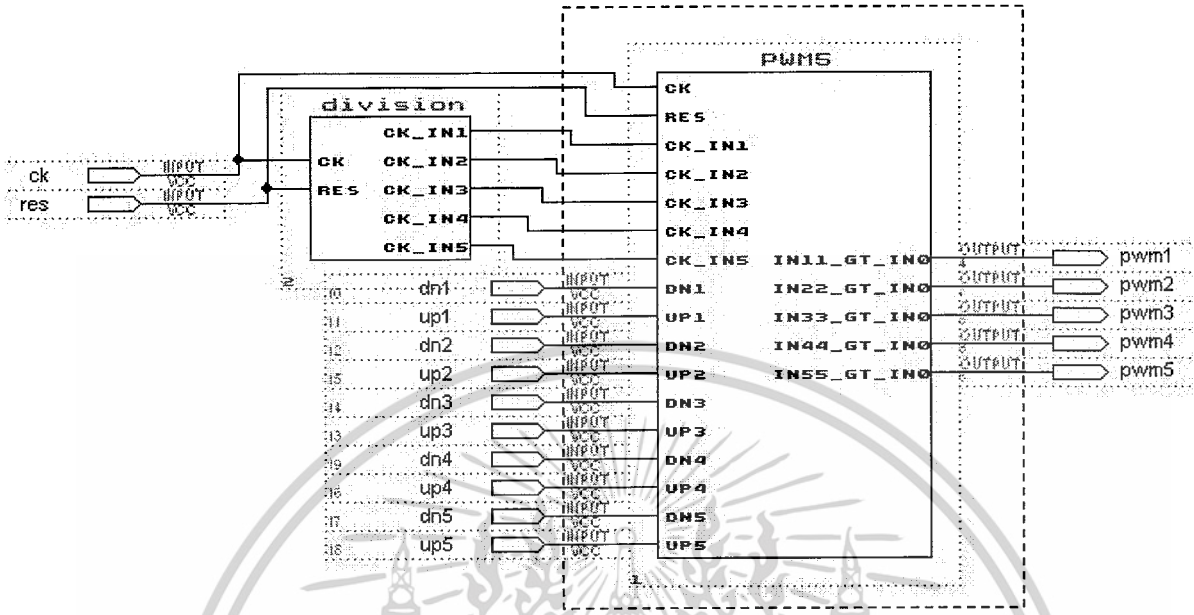
รูปที่ ข.1 กล่องควบคุม



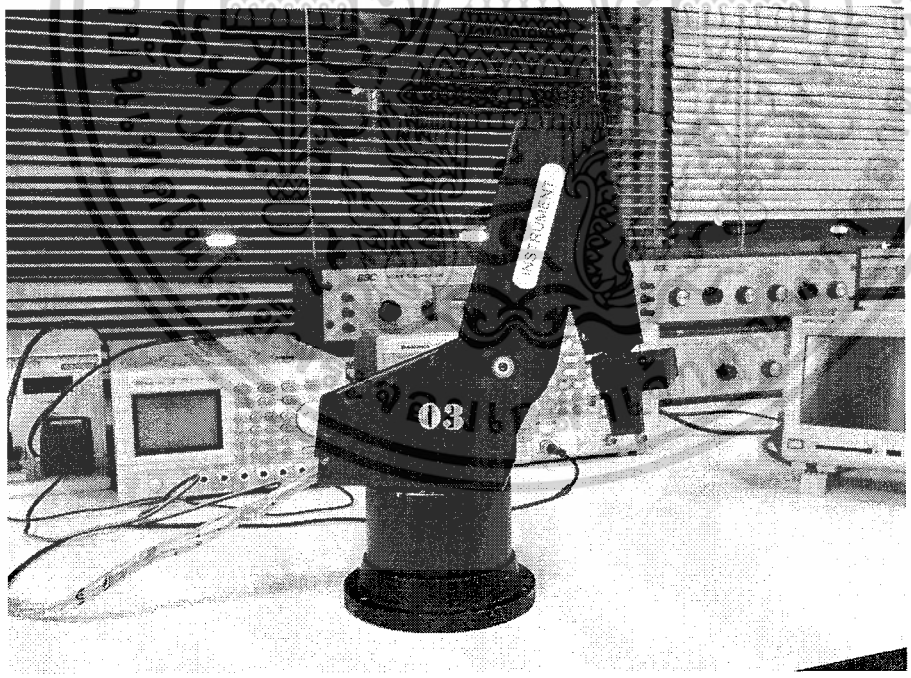
รูปที่ ข.2 แหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค. การออกแบบขับแขนกลด้วยโปรแกรม Veriolog HDL เชื่อมต่อกับปุ่มกดควบคุม

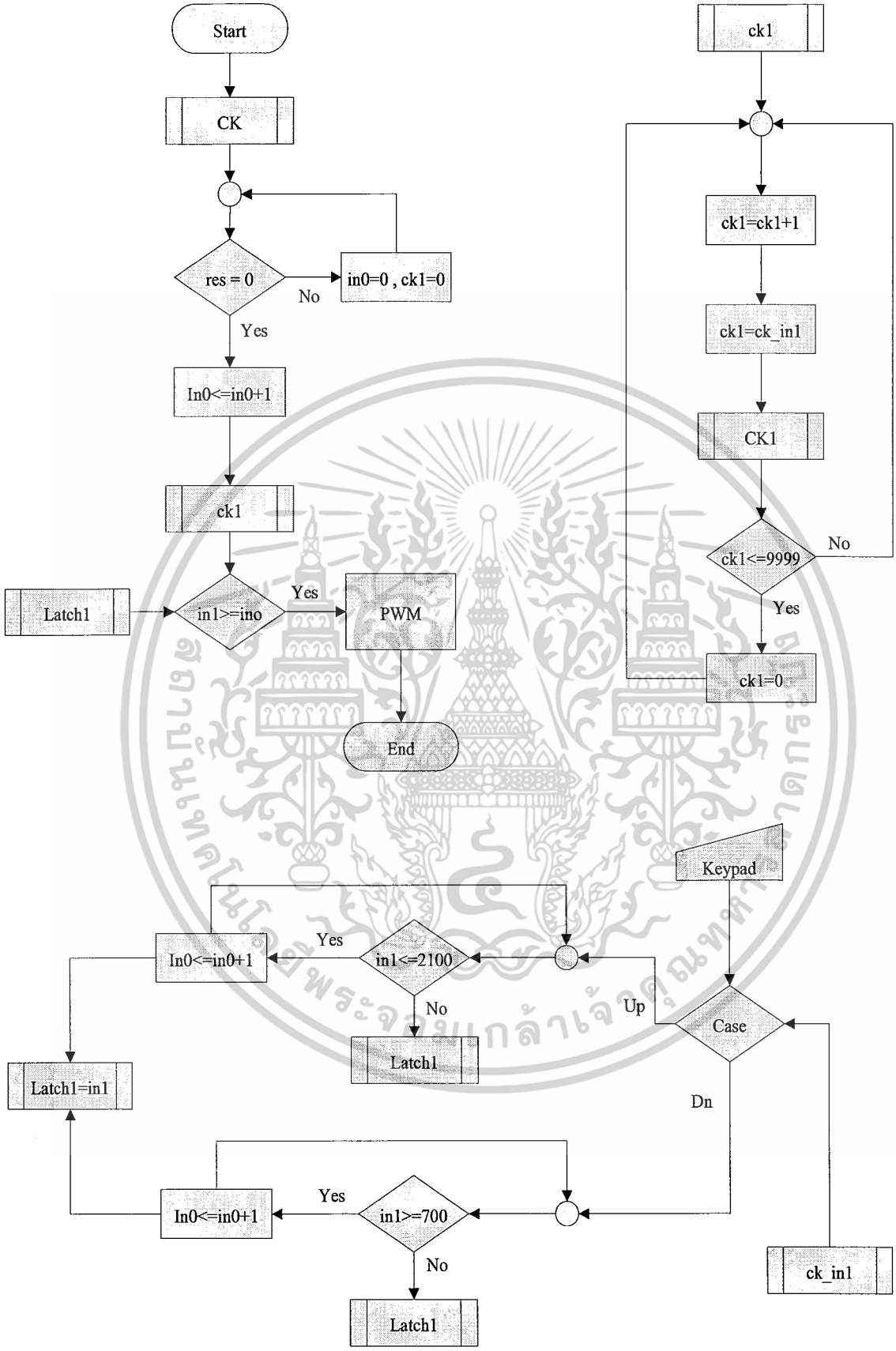


รูปที่ ค.1 การเชื่อมต่อโมดูลของโปรแกรมควบคุมแขนกลติดต่อกับปุ่มกดควบคุม



รูปที่ ค.2 แขนกลที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



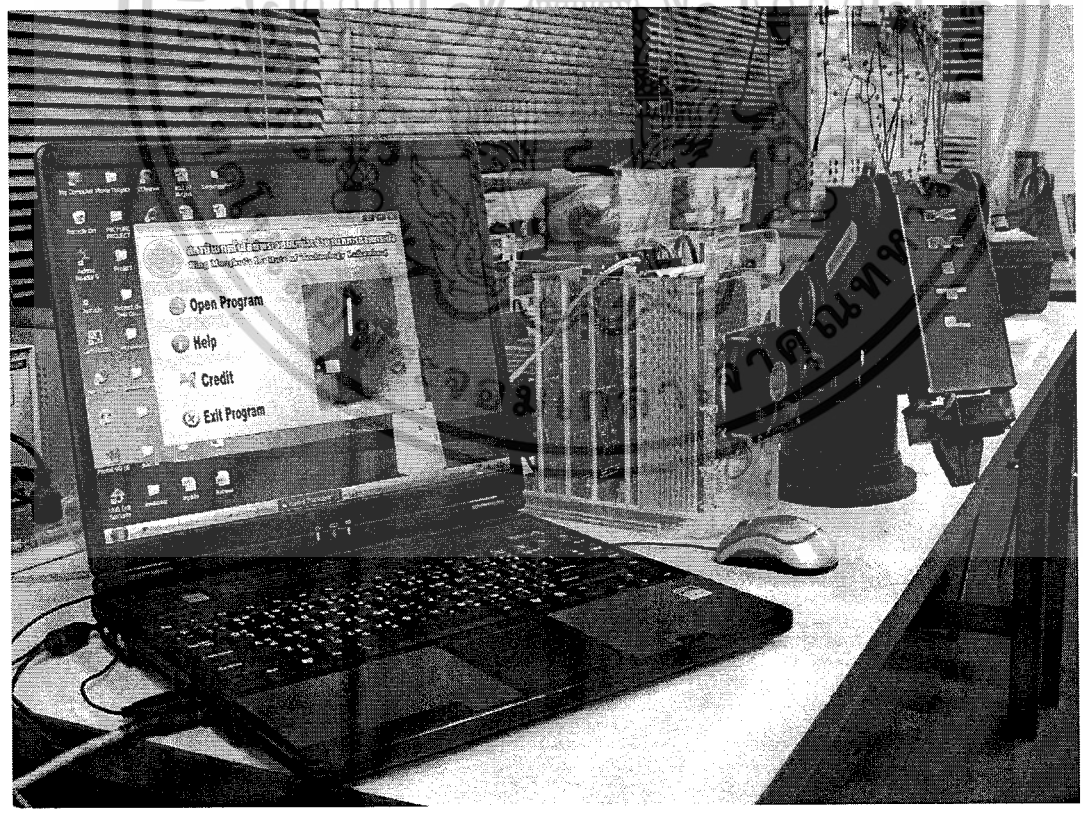
รูปที่ ค.3 Flowchart การสร้างสัญญาณ PWM ที่ FPGA เชื่อมต่อกับปุ่มกดควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง. เซอร์โวมอเตอร์และส่วนของชิ้นงานที่สำเร็จ

ตารางที่ 9 ดีซีเซอร์โวมอเตอร์รุ่นต่าง ๆ

Model	STD	BBM	Size (L x W x H) Mm/in	Weight		4.8V			6V			Modify 360 °
				g	oz	SPEED (sec / 60°)	Torque Kg-cm	Torque Oz-in	SPEED (sec / 60°)	Torque Kg-cm	Torque Oz-in	
Micro	✓		28 x 14 x 29.8 1.1 x 0.55 x 1.17	18	0.63	0.16	1.8	25	0.13	2.30	32	✓
S03N	✓		39.5 x 20.0 x 35.6 1.56 x 0.79 x 1.40	41	1.44	0.23	2.40	47	0.18	4	56	✓
S03NXF	✓		39.5 x 20.0 x 35.6 1.56 x 0.79 x 1.40	41	1.45	0.15	2.20	31	0.12	2.45	34	✓
S03T	✓		39.5 x 20.0 x 39.6 1.56 x 0.79 x 1.56	46	1.62	0.33	7.20	100	0.27	8	111	✓
S03TXF	✓		39.5 x 20.0 x 39.6 1.56 x 0.79 x 1.56	46	1.62	0.21	5	69	0.17	6.20	86	✓
S04		✓	54.4 x 26.5 x 51.5 2.14 x 1.04 x 2.03	114	4	0.25	10	138.88	0.20	13	180.5	
S666	✓		63.0 x 32.0 x 61.6 2.48 x 1.26 x 2.43	142.4	5.02	0.28	13	181	0.22	15	208	✓
PICO	✓		22.8 x 9.5 x 15.5 0.90 x 0.37 x 0.61	5.40	0.19	0.12	0.70	10	0.09	0.84	12	✓
MICRO/ 2BBM		✓	28 x 14 x 29.8 1.1 x 0.55 x 1.17	18	0.63	0.16	1.80	25	0.13	2.30	32	✓
HS-322	✓		40 x 20 x 36.5 1.57 x 0.78 x 1.43	43	1.51	0.19	3	41.66	0.15	3.5	48.6	✓
S3003	✓		41 x 20 x 36 1.6 x 0.8 x 1.4	37.2	1.3	0.23	3.2	44	0.19	4.1	56.8	✓
S03T/ 2BB/J		✓	39.5 x 20.0 x 39.6 1.56 x 0.79 x 1.56	46.0	1.62	0.33	7.2	100	0.27	8	111	✓
S03T/2B BMG/J		✓	40.6 x20.0 x 42.8 1.60 x 0.79 x 1.70	73	2.57	0.33	7.4	103	0.27	8.6	119	✓
S666/ NMG/J		✓	63.0 x 32.0 x 61.6 2.48 x 1.26 x 2.43	180	6.35	0.24	24	333	0.2	28.8	400	✓



รูปที่ ง.1 โครงงานที่สำเร็จเรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้