ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB

Demonstrator of Real-Time Workshop via MATLAB

ชื่อนักศึกษา 1. นายอมรเทพ ผันสิน รหัสประจำตัว 46035293

2. นางสาวอุทัยวรรณ มานะมุ่งประเสริฐ รหัสประจำตัว 46035296

3. นายสมชาย เสนาปิน รหัสประจำตัว 46035697

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** วิศวกรรมโทรคมนาคม

อาจารย์ที่ปรึกษา อ.อำพล ทองระอา

อาจารย์ที่ปรึกษาร่วม ผศ.กิติพงค์ มะโน

คณะกรรมการสอบปริญญานิพนธ์	ลายมือชื่อ
1. อ.สุระชัย พิมพ์สาลี	
2. อ.อำพล ทองระอา	
3. อ.ปิยะ ศุภวราสุวัฒน์	
4. อ.โกศล ตราชู	
5. อ.ประเสริฐ เคนพันค้อ	

วัน/เดือน/ปีที่สอบ 25 เมษายน 2548 เวลา 14.00 น.

สถานที่สอบ ห้อง ค.427 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.สุรสิทธิ์ ราตรี)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่.....เดือน.....พ.ศ.....

<BT4720252>

ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB

DEMONSTRATOR OF REAL-TIME WORKSHOP VIA MATLAB

นายอมรเทพ ผันสิน
นางสาวอุทัยวรรณ มานะมุ่งประเสริฐ
นายสมชาย เสนาปิน

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

วพ. ว 281 ช 2547

เลขหมู่ 59489
เลขทะเบียน
วัน,เดือน,ปี - 6 ส.ย. 2549

b 15 ก.ย. 49
1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB

Demonstrator of Real-Time Workshop via MATLAB

วัตถุประสงค์

1. เพื่อศึกษาหลักการทำงานของการปฏิบัติงานที่เวลาจริงด้วย MATLAB
2. เพื่อออกแบบชุดทดลองการปฏิบัติงานที่เวลาจริง
3. เพื่อสร้างชุดทดลองการปฏิบัติงานที่เวลาจริง
4. เพื่อทดลองชุดทดลองการปฏิบัติงานที่เวลาจริง
5. เพื่อหาคุณภาพของการทำชุดทดลองและใบงานการทดลองการปฏิบัติงานที่เวลาจริง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยผู้ทรงคุณวุฒิด้านวิชาการอย่างน้อย 3 ท่าน และด้านสื่อการเรียนการสอนอย่างน้อย 3 ท่าน

ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการปฏิบัติงานที่เวลาจริงด้วย MATLAB
2. ได้โครงสร้างของชุดทดลองปฏิบัติงานที่เวลาจริงด้วย MATLAB
3. ได้ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB
4. ชุดทดลองและใบงานการทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB ที่มีคุณภาพทางด้านวิชาการและด้านสื่อการเรียนการสอนได้นำไปใช้เป็นสื่อในการเรียนการสอนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB	
นักศึกษา	นายอมรเทพ	ผันสิน
	นางสาวอุทัยวรรณ	มานะมุ่งประเสริฐ
	นายสมชาย	เสนาปิน
อาจารย์ที่ปรึกษา	อาจารย์อำพล	ทองระอา
อาจารย์ที่ปรึกษาร่วม	ผศ.กิติพงค์	มะโน
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต	
สาขาวิชา	วิศวกรรมโทรคมนาคม	
ปีการศึกษา	2547	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอ ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วยโปรแกรม MATLAB โดยใช้โปรแกรม MATLAB เชื่อมต่อกับฮาร์ดแวร์ภายนอก ซึ่งจะออกแบบเป็นใบงานการทดลองการประยุกต์ใช้งานในด้านระบบควบคุมและการประมวลผลสัญญาณดิจิตอล ประกอบด้วย การเชื่อมต่อกับคอมพิวเตอร์ในระบบเครือข่าย การเชื่อมต่อคอมพิวเตอร์ผ่านพอร์ตอนุกรม การเชื่อมต่อกับอุปกรณ์ GPIB (General Purpose Interface Bus) และการเชื่อมต่อกับ FPGA (Field Programmable Gate Array)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Demonstrator of Real-Time Workshop via MATLAB	
Students	Mr.Amornthep	Phansin
	Miss Autaiwan	Manamungprasert
	Mr.Somchai	Sanapin
Advisor	Mr.Amphon	Thongra-ar
Co-Advisor	Asst.Prof. Kitipong	Mano
Education Level	Bachelor of Science in Industrial Education	
Program in	Telecommunication Engineering	
Academic Year	2004	

ABSTRACT

This thesis presents the Demonstrator of Real-Time Workshop via MATLAB. This demonstrator consists of experiment set including Control System and Digital Processing Application.The experiment set comes with experiment sheet, network communication, serial communicator, GPIB (General Purpose Interface Bus) and FPGA (Field Programmable Gate-Array)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์จาก อาจารย์อำพล ทองระอา อาจารย์ผู้ควบคุมปริญญานิพนธ์ ผศ.กิติพงค์ มะโน อาจารย์ผู้ควบคุมปริญญานิพนธ์ร่วม ที่ได้กรุณาให้คำแนะนำ แนวความคิด ความรู้ต่างๆ แนวทางแก้ไขปัญหา และให้ความช่วยเหลือทางด้านเครื่องมือและอุปกรณ์ ในการจัดทำปริญญานิพนธ์ และช่วยตรวจสอบแก้ไขปริญญานิพนธ์นี้ให้สำเร็จได้อย่างสมบูรณ์ คณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาและขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ ดร.พรพิมล ฉายรัศมี อาจารย์พิชญ์สินี มะโน และอาจารย์วรวิทย์ สมหา เป็นอย่างสูงที่ได้กรุณาสละเวลาในการประเมินคุณภาพของชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB และได้ให้ความรู้ต่างๆ แนวความคิด และแนวทางแก้ไขปัญหา รวมถึงคำแนะนำในการแก้ไขข้อบกพร่องในการจัดทำปริญญานิพนธ์ เพื่อให้ปริญญานิพนธ์นี้สมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณ อาจารย์ปิยะ ศุภวราสุวัฒน์ เป็นอย่างสูงที่ให้กำลังใจ คำแนะนำ และแนวความคิด ตลอดระยะเวลาการจัดทำปริญญานิพนธ์ แก่คณะผู้จัดทำจนผลงานสำเร็จลุล่วงไปด้วยดี และได้ช่วยตรวจสอบความถูกต้องของปริญญานิพนธ์ รวมถึงได้ให้คำแนะนำในการแก้ไขข้อบกพร่องเพื่อให้ปริญญานิพนธ์นี้สมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณอาจารย์ทุกท่านที่ได้ประสิทธิประสาทความรู้ ตลอดจนให้ข้อคิดต่างๆ อันก่อให้เกิดประโยชน์ในการศึกษาค้นคว้า และเป็นแนวทางในการจัดทำปริญญานิพนธ์จนสำเร็จตามจุดมุ่งหมายที่ได้กำหนดไว้

ขอขอบพระคุณ คุณพ่อ และคุณแม่ ผู้เป็นที่เคารพยิ่ง ที่ได้ให้กำลังใจ และให้การสนับสนุนในการศึกษาตลอดมา และตลอดจนครอบครัว ญาติพี่น้อง ที่ให้การสนับสนุนเป็นอย่างดี

ขอขอบพระคุณเพื่อนๆ และบุคคลที่คณะผู้จัดทำไม่ได้กล่าวถึงไว้ในที่นี้ ที่ให้การสนับสนุนตลอดจนให้ความช่วยเหลือในด้านต่างๆ และเป็นกำลังใจแก่คณะผู้จัดทำจนผลงานสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอขอบพระคุณ คณะเจ้าหน้าที่ สำนักหอสมุดกลาง และห้องสมุดคณะครุศาสตร์อุตสาหกรรมทุกท่าน ที่ช่วยให้ความร่วมมือช่วยเหลือด้านการติดต่อสอบถาม และอำนวยความสะดวกรวมถึงเอื้อเฟื้อสถานที่ในการค้นคว้าหาข้อมูล ในการจัดทำปริญญานิพนธ์

คุณค่าและประโยชน์ใดๆ ที่เป็นผลมาจากปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	IX
สารบัญรูป	XI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 ขีดความสามารถโครงงาน	1
1.3 เนื้อหาโดยสังเขป	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กล่าวนำ	3
2.2 ความหมายของ Real-Time Workshop	3
2.2.1 ความต้องการทางด้านซอฟต์แวร์	4
2.2.2 การใช้งานตัวแปลภาษา	4
2.2.3 ขั้นตอนการทำงานของ Real-Time Workshop	5
2.2.4 ความหมายของไฟล์ภาษาซีที่ได้จากการสร้างโค๊ด	11
2.3 การใช้งาน xPCTarget	14
2.3.1 ความหมายของ xPCTarget	14
2.3.2 ความต้องการของระบบ	15
2.3.3 การใช้งานอุปกรณ์อินพุตและเอาต์พุต	17
2.3.4 ลักษณะการเชื่อมต่อแบบ HostPC กับ TargetPC	18
2.3.5 ขั้นตอนการทำงานของ xPCTarget	19
2.3.6 การใช้งาน xPCTargetExplorer	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4 การพัฒนาการประมวลผลสัญญาณดิจิตอลด้วยโปรแกรม MATLAB และโปรแกรม System Generator	25
2.4.1 ขั้นตอนการทำงานการประมวลผลสัญญาณดิจิตอลด้วย MATLAB และ System Generator	25
2.4.2 ความต้องการทางด้านซอฟต์แวร์	27
2.5 โครงสร้างของ GPIB	27
2.5.1 ขีดจำกัดของ IEEE-488	28
2.5.2 รายละเอียดเกี่ยวกับ IEEE-488	28
2.5.3 ความหมายของสัญญาณต่างๆ ในระบบ IEEE-488	29
2.5.4 การเชื่อมต่ออุปกรณ์ต่างๆ ในระบบ IEEE-488 BOS	31
2.5.5 คำสั่งใช้งานของ GPIB	31
2.5.6 Device clear และ Interface clear	34
2.5.7 การขอบริการและการตรวจสอบ	35
2.5.8 รูปแบบของข้อมูล	35
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	37
3.1 กล่าวนำ	37
3.2 เครื่องมือที่ใช้ในการออกแบบและสร้าง	37
3.2.1 เครื่องมือทางด้านซอฟต์แวร์	37
3.2.2 เครื่องมือทางด้านฮาร์ดแวร์	37
3.3 การออกแบบ	37
3.3.1 การกำหนดวัตถุประสงค์	38
3.3.2 การวิเคราะห์เนื้อหา	41
3.3.3 การสร้างใบงานการทดลอง	41
3.3.4 การวัดและประเมินผล	41
3.4 การสร้างและการทำงาน	44
3.4.1 ใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.4.2 ใบงานที่ 2 เรื่องการใช้งาน Simulink เบื้องต้น	45
3.4.3 ใบงานที่ 3 เรื่อง การใช้งาน Real-Time Workshop	46
3.4.4 ใบงานที่ 4 เรื่อง การใช้งาน xPC Ta rget	47
3.4.5 ใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม	48
3.4.6 ใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่อง	49
3.4.7 ใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่อง	50
3.4.8 ใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น	51
3.4.9 ใบงานที่ 9 เรื่อง การออกแบบวงจรกรองดิจิตอล (Digital Filter)	52
3.4.10 ใบงานที่ 10 เรื่อง การสังเคราะห์วงจร โดยใช้โปรแกรม	53
บทที่ 4 การทดลองและผลการทดลอง	54
4.1 กล่าวนำ	54
4.2 การทดลองในใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น	54
4.2.1 ขั้นตอนการทดลอง	54
4.2.2 ผลการทดลอง	54
4.3 การทดลองในใบงานที่ 2 เรื่อง การใช้งาน Simulink เบื้องต้น	55
4.3.1 ขั้นตอนการทดลอง	55
4.3.2 ผลการทดลอง	55
4.4 การทดลองในใบงานที่ 3 เรื่อง การใช้งาน Real-Time Workshop	56
4.4.1 ขั้นตอนการทดลอง	56
4.4.2 ผลการทดลอง	57
4.5 การทดลองในใบงานที่ 4 เรื่อง การใช้งาน xPC Target	59
4.5.1 ขั้นตอนการทดลอง	59
4.5.2 ผลการทดลอง	59
4.6 การทดลองในใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับ พอร์ตอนุกรม (RS-232)	61
4.6.1 ขั้นตอนการทดลอง	61
4.6.2 ผลการทดลอง	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.7 การทดลองในใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ติดต่อกับเครื่องกำเนิดสัญญาณ	63
4.7.1 ขั้นตอนการทดลอง	63
4.7.2 ผลการทดลอง	63
4.8 การทดลองในใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณ	65
4.8.1 ขั้นตอนการทดลอง	65
4.8.2 ผลการทดลอง	65
4.9 การทดลองในใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น	66
4.9.1 ขั้นตอนการทดลอง	66
4.9.2 ผลการทดลอง	67
4.10 การทดลองในใบงานที่ 9 เรื่อง ออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator	67
4.10.1 ขั้นตอนการทดลอง	67
4.10.2 ผลการทดลอง	68
4.11 การทดลองในใบงานที่ 10 เรื่อง การใช้งานโปรแกรม Xilinx ISE Foundation ในการสังเคราะห์วงจร	69
4.11.1 ขั้นตอนการทดลอง	69
4.11.2 ผลการทดลอง	70
4.12 สรุปผลการประเมินคุณภาพใบงานการทดลองชุดการทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB	71
บทที่ 5 บทสรุป	73
5.1 สรุป	73
5.2 ปัญหาและแนวทางการแก้ไข	73
5.3 แนวทางการพัฒนา	74
บรรณานุกรม	75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
ภาคผนวก ก ใบงานการทดลอง	76
ภาคผนวก ข รูปแบบคำสั่งการติดต่อกับเครื่องวัดสัญญาณ	238
ภาคผนวก ค แบบประเมินคุณภาพ	250
ภาคผนวก ง หนังสือแต่งตั้งผู้ทรงคุณวุฒิ	255
ประวัติผู้แต่ง	259

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 แปลภาษาซีที่โปรแกรม MATLAB รองรับ	5
2.2 ความต้องการทางด้านซอฟต์แวร์ขั้นต่ำ	15
2.3 ความต้องการทางด้านฮาร์ดแวร์ขั้นต่ำ	16
2.4 ความต้องการทางด้านซอฟต์แวร์ขั้นต่ำ	16
2.5 ความต้องการทางด้านฮาร์ดแวร์ขั้นต่ำ	17
2.6 สภาพการ interface ที่ได้กำหนดไว้ในระบบประกอบด้วยฟังก์ชั่นต่างๆ	34
4.1 ผลการทดลองโปรแกรมการหาผลลัพธ์ของการคูณเลข 2 จำนวน	54
4.1 (ต่อ) ผลการทดลองโปรแกรมการหาผลลัพธ์ของการคูณเลข 2 จำนวน	55
4.2 ผลการปรับค่าที่หน้าจอของเครื่องกำเนิดสัญญาณ	64
4.3 สรุปผลการประเมินคุณภาพใบงานชุดทดลองการปฏิบัติงาน ที่เวลาจริงด้วย MATLAB	71
4.3 (ต่อ) สรุปผลการประเมินคุณภาพใบงานชุดทดลองการปฏิบัติงาน ที่เวลาจริงด้วย MATLAB	72
ข.1 สัญลักษณ์ของ BNF และความหมาย	239
ข.2 ส่วนประกอบของประโยคคำสั่ง	240
ข.3 การเปรียบเทียบระหว่าง Header ที่มีผลตอบสนองกับไม่มี	242
ข.4 การแสดงตัวเลือกเขียนย่อ ตำแหน่ง Cursor ของ Mnemonic เมื่อ Oscilloscope และ Cursor	245
ข.5 การแสดงเครื่องมือวัดอัตโนมัติได้ 4 ตัว ด้วยการแสดง Waveform ของแต่ละตัว	246
ข.6 Channel Mnemonic เป็นคำสั่งระบุ channel ที่ใช้งานที่ Mnemonic ใน Header	246
ข.7 Math Waveform Mnemonic เป็นคำสั่งระบุ Waveform ในทางคณิตศาสตร์ที่ใช้งาน	246
ข.8 Reference Waveform Mnemonic เป็นคำสั่งระบุ Reference Waveform ที่ใช้งานได้	246

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
ข.9 Waveform Mnemonic ในบางคำสั่งสามารถระบุ Waveform คือ Channel Mnemonic, Math Waveform, หรือ Reference Waveform	247
ข.10 Numeric Argument	247
ข.11 คำสั่งของ Quoted String Argument	247

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ผังงานขั้นตอนการทำงานของ Real-Time Workshop	5
2.1 (ต่อ) ผังงานขั้นตอนการทำงานของ Real-Time Workshop	6
2.2 ตัวอย่างการออกแบบระบบจาก Simulink	7
2.3 หน้าต่างการกำหนดค่าพารามิเตอร์	7
2.4 หน้าต่างการตรวจสอบค่าพารามิเตอร์	8
2.5 ผังงานขั้นตอนการสร้างโค๊ด	8
2.6 หน้าต่างการแสดงชนิดของ TLC	9
2.7 หน้าต่างการตรวจสอบโค๊ดภาษาซี	10
2.8 หน้าต่างการบันทึกรูปแบบการกำหนดค่าพารามิเตอร์	11
2.9 การเชื่อมต่อแบบอนุกรม	18
2.10 การเชื่อมต่อแบบระบบเครือข่าย	19
2.11 ผังงานแสดงขั้นตอนการทำงานของ xPC Target	19
2.12 หน้าต่างการเรียกใช้งาน xPC Target Explorer	22
2.13 หน้าต่าง Compiler Configuration	22
2.14 หน้าต่าง xPC Target Hierachy	22
2.15 หน้าต่าง TargetPC1 Communication Component	23
2.16 การใช้งาน xPC Target Explorer	23
2.17 หน้าต่าง TargetPC1 Communication Component	24
2.18 หน้าต่าง TargetPC1 Target Configuration	25
2.19 ขั้นตอนการทำงานพัฒนาการประมวลผลสัญญาณดิจิตอล ด้วย MATLAB และ System Generator	26
2.20 การแบ่งเส้นสายนำสัญญาณ	29
2.21 ขั้วต่อของ GPIB และการจัดขาของสัญญาณต่างๆ	31
2.22 การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)	32
2.23 การเชื่อมต่อแบบกระจาย (Star Configuration)	32
2.24 รูปแบบของข้อมูล	35
2.25 รูปแบบของ HR	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.26 สัญญาณแบ่งข้อมูลแต่ละชุด	36
3.1 ผังงานของใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น	44
3.2 ผังงานของใบงานที่ 2 เรื่อง การใช้งาน Simulink เบื้องต้น	45
3.3 ผังงานของใบงานที่ 3 เรื่อง การใช้งาน Real-Time Workshop	46
3.4 ผังงานของใบงานที่ 4 เรื่อง การใช้งาน xPC Target	47
3.5 ผังงานของใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม	48
3.6 ผังงานของใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุม เครื่องกำเนิดสัญญาณ	49
3.7 ผังงานของใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุม เครื่องวัดสัญญาณ	50
3.8 ผังงานของใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น	51
3.9 ผังงานของใบงานที่ 9 เรื่อง การออกแบบวงจรกรองดิจิตอล โดยใช้ System Generator	52
3.10 ผังงานของใบงานที่ 10 เรื่อง การสังเคราะห์วงจร โดยใช้โปรแกรม Xilinx ISE Foundation	53
4.1 โปรแกรมการหาผลลัพธ์ของการคูณเลข 2 จำนวน	54
4.2 ลักษณะของโมเดลเพื่อการทดสอบการใช้งาน Simulink เบื้องต้น	55
4.3 ลักษณะสัญญาณที่สโคป	56
4.4 การออกแบบโมเดลเพื่อทดสอบการสร้างโค๊ดภาษาซี	56
4.5 ผลการจำลองการทำงานของโมเดลซึ่งเป็นผลของสัญญาณที่ผ่าน บล็อก Gain = 1	57
4.6 ผลการจำลองการทำงานของโมเดลซึ่งเป็นผลของสัญญาณที่ผ่าน บล็อก Gain = 2	57
4.7 ไฟล์ที่ได้หลังจากการสร้างโค๊ดภาษาซี	57
4.8 ผลการนำข้อมูลจากไฟล์ model.exe มาพล็อตกราฟซึ่งเป็นผลการพล็อต กราฟของข้อมูลที่ถูกคูณด้วยค่าคงที่เท่ากับ 1	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.9 ผลการนำข้อมูลจากไฟล์ model.exe มาพล็อตกราฟซึ่งเป็นผลการพล็อตกราฟของข้อมูลที่ถูกคูณด้วยค่าคงที่เท่ากับ 2	58
4.10 การออกแบบโมเดลเพื่อทดสอบการใช้งาน xPC Target	59
4.11 ผลการจำลองการทำงานของโมเดล	59
4.12 ผลการทดสอบการทำงานของ Target PC	60
4.13 ผลที่เกิดขึ้นที่ Target PC เมื่อทดลองเปลี่ยนแปลงค่าความถี่	60
4.14 การออกแบบโมเดลเพื่อทดสอบการติดต่อกับพอร์ตอนุกรม	61
4.15 ผลที่เกิดขึ้นที่ Target PC เมื่อเชื่อมต่อสายสื่อสารอนุกรมระหว่าง COM1 และ COM2	62
4.16 ผลที่เกิดขึ้นที่ Target PC เมื่อถอดสายสื่อสารอนุกรมระหว่าง COM1 และ COM2 ออก	62
4.17 ผลการอ่านค่าข้อมูลของเครื่องกำเนิดสัญญาณ	63
4.18 ผลการอ่านค่าลักษณะสัญญาณของเครื่องกำเนิดสัญญาณ	64
4.19 ผลการอ่านค่าความถี่ของเครื่องกำเนิดสัญญาณ	64
4.20 ผลการอ่านค่าระดับสัญญาณของเครื่องกำเนิดสัญญาณ	64
4.21 ผลการอ่านค่าข้อมูลของเครื่องวัดสัญญาณ	65
4.22 ผลการพล็อตกราฟของสัญญาณที่รับมาจากเครื่องมือวัดสัญญาณ	66
4.23 โมเดลเพื่อทดสอบการใช้งาน	66
4.24 ผลการจำลองการทำงานที่หน้าต่างสโคป เพื่อทดสอบการใช้งานบล็อก Xilinx	67
4.25 ลักษณะโมเดลวงจรกรองแถบความถี่ผ่าน	68
4.26 ผลการจำลองการทำงานของวงจรกรองแถบความถี่ผ่าน	68
4.27 ขนาดพื้นที่ของโมเดลวงจรกรองเมื่อโปรแกรมลงบนชิพ FPGA	69
4.28 ขั้นตอนการสังเคราะห์วงจรและเชื่อมต่อสายสัญญาณ	70
4.29 ลักษณะของวงจรกรองที่ผ่านการสังเคราะห์วงจรแล้ว	70
4.30 ผลรายงานขนาดพื้นที่ที่ใช้งาน	71
4.31 ผลรายงานความเร็วสูงสุดของวงจรที่สามารถใช้งานได้	71
ข.1 ส่วนประกอบสำคัญของประโยคคำสั่ง	241

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ข.2 ตัวอย่างของ block Argument	249

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันเทคโนโลยีด้านการประมวลผลที่เวลาจริง (Real-Time) ถูกนำมาประยุกต์การใช้งานในด้านต่างๆ มากมาย เช่น การประมวลผลสัญญาณเชิงดิจิตอลหรืองานด้านระบบควบคุม เป็นต้น ซึ่งการศึกษาการทำงานของระบบที่เวลาจริงนี้จะต้องทำการเขียนโปรแกรมด้วยภาษาโปรแกรมต่างๆ แล้วโปรแกรมบนฮาร์ดแวร์เพื่อดูผลการทำงานของฮาร์ดแวร์ที่เวลาจริง แต่การเรียนการสอนในสถานศึกษาต่างๆ นั้น ยังไม่มีสื่อการเรียนการสอนโดยตรงทำให้การเรียนการสอนเรื่อง การปฏิบัติงานที่เวลาจริง นั้นเป็นไปได้ยาก

โปรแกรม MATLAB มีความสามารถในการ ออกแบบ จำลองการทำงาน และเชื่อมต่อกับฮาร์ดแวร์ที่เวลาจริง โดยลักษณะที่โต้ตอบกับผู้ใช้งานได้ ดังนั้นทำให้การศึกษาการปฏิบัติงานที่เวลาจริง นั้นสามารถเข้าได้ง่ายและรวดเร็วขึ้น

1.2 ขีดความสามารถของโครงงาน

โครงงานนี้มีขีดความสามารถดังนี้

1. ใบงานการทดลองการปฏิบัติงานที่เวลาจริงอย่างน้อย 10 ใบงาน
2. มีฮาร์ดแวร์ประกอบภายนอก ประกอบการทดลองเพื่อให้เห็นการทำงานที่เวลาจริง
3. เชื่อมต่อฮาร์ดแวร์ประกอบภายนอกได้ไม่น้อยกว่านี้ คือ RS232, GPIB, Network

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปริญญานิพนธ์ ขีดความสามารถของโครงงาน และเนื้อหาในบทต่างๆ โดยสังเขป

บทที่ 2 ประกอบด้วย ทฤษฎีต่างๆ เกี่ยวกับการใช้งาน Real-Time Workshop, การใช้งาน xPC Target, ทฤษฎีของ GPIB และการพัฒนาการประมวลผลสัญญาณดิจิตอลด้วยโปรแกรม MATLAB และโปรแกรม System Generator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 กล่าวถึงเนื้อหาที่เกี่ยวกับการออกแบบและการสร้างใบงานการทดลอง ซึ่งเกี่ยวกับการออกแบบใบงานการทดลองทั้ง 10 ใบงาน ซึ่งจะมีหลักเกณฑ์ในการออกแบบคือ เครื่องมือที่ใช้ในการออกแบบและการสร้างจะเกี่ยวกับเครื่องมือทางด้านซอฟต์แวร์และฮาร์ดแวร์ ส่วนในการออกแบบนั้นจะมีขั้นตอนการออกแบบประกอบด้วย การกำหนดวัตถุประสงค์ การวิเคราะห์เนื้อหา ขั้นตอนการสร้างใบงานการทดลอง และการประเมินผลใบงานการทดลอง ส่วนการสร้างและการทำงานคือ ขั้นตอนการสร้างใบงานการทดลองจะเป็นการกำหนดเนื้อหาในใบงานอย่างละเอียด โดยเนื้อหาในแต่ละใบงานจะต้องสอดคล้องกับวัตถุประสงค์ที่กำหนดไว้และต้องสามารถประเมินผลได้จากการตอบคำถามท้ายใบงานการทดลอง

บทที่ 4 ประกอบด้วยการทดลองและผลการทดลองของใบงานการทดลองซึ่งประกอบด้วย 10 ใบงานดังนี้

ใบงานที่ 1 การใช้งานโปรแกรม MATLAB เบื้องต้น

ใบงานที่ 2 การใช้งาน Simulink เบื้องต้น

ใบงานที่ 3 การใช้งาน Real-Time Workshop

ใบงานที่ 4 การใช้งาน xPC Target

ใบงานที่ 5 การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม

ใบงานที่ 6 การใช้งานโปรแกรม MATLAB ควบคุมเครื่องกำเนิดสัญญาณ

ใบงานที่ 7 การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณ

ใบงานที่ 8 การใช้งานโปรแกรม System Generator เบื้องต้น

ใบงานที่ 9 การออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator

ใบงานที่ 10 การสังเคราะห์วงจรโดยใช้โปรแกรม Xilinx ISE Foundation

บทที่ 5 เป็นการสรุปผลการจัดทำโครงงาน ปัญหาที่เกิดขึ้นและแนวทางในการแก้ไข รวมทั้งแนวทางการพัฒนา

ภาคผนวก ก ใบงานการทดลอง

ภาคผนวก ข รูปแบบคำสั่งการติดต่อกับเครื่องวัดสัญญาณ

ภาคผนวก ค แบบประเมินคุณภาพ

ภาคผนวก ง หนังสือแต่งตั้งผู้ทรงคุณวุฒิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 กล่าวนำ

ในปัจจุบันการพัฒนาและสร้างโปรแกรมเพื่อใช้งานจริงในฮาร์ดแวร์ (Hardware) ส่วนใหญ่จะต้องเขียนโปรแกรมขึ้นมาเองโดยใช้ภาษาระดับสูงต่างๆ เช่น ภาษาซี (C language) ภาษาฟอร์แทรน (Fortran language) ภาษาบรรยายพฤติกรรม (HDL : Hardware Description Language) เป็นต้น จะเห็นว่าถ้าเป็นระบบที่มีความซับซ้อนสูงในเชิงวิศวกรรม จะต้องอาศัยความรู้และประสบการณ์ของนักเขียนโปรแกรมมาก ถ้าเป็นผู้พัฒนาโปรแกรมที่ไม่มีประสบการณ์ในการออกแบบมาก่อนยิ่งจะต้องเสียเวลาในการศึกษาขั้นตอนต่างๆ มากเลยทีเดียว

โปรแกรมหนึ่งที่นิยมใช้ในการแก้ปัญหาเชิงวิศวกรรม คือ โปรแกรม MATLAB ซึ่งมีประสิทธิภาพสูง เนื่องจากมีฟังก์ชั่นสำเร็จรูปในงานวิศวกรรมมากมายและภาษาที่ใช้จะมีความสอดคล้องกับภาษาโปรแกรมทั่วๆ ไป ส่วนใหญ่จะใช้ โปรแกรม MATLAB ในการจำลองการทำงานก่อนเพื่อดูผลการทำงานที่เกิดขึ้นก่อนนำไปใช้งานจริงหรือในงานเชิงวิจัย เป็นต้น

ขีดความสามารถของโปรแกรม MATLAB ถูกพัฒนาขึ้นมาก เพื่อพัฒนาการประยุกต์ใช้งานไม่ให้จำกัดเพียงในงานเชิงวิจัยเท่านั้น ความสามารถหนึ่งของโปรแกรม MATLAB ที่น่าสนใจก็คือ Real-Time Workshop จะช่วยให้นักออกแบบโปรแกรมสามารถทำงานได้รวดเร็วมากขึ้นและรองรับกับฮาร์ดแวร์ต่างๆ ได้หลายชนิด

2.2 ความหมายของ Real-Time Workshop

Real-Time Workshop เป็นกล่องเครื่องมือหนึ่งที่มีอยู่ในโปรแกรม MALAB มีความสามารถในการสร้างโค๊ดภาษาซีจากระบบที่ออกแบบจาก Simulink ซึ่งไฟล์ที่ได้หลังการสร้างโค๊ดจะเป็นไฟล์ภาษาซีหรือไฟล์กระทำการ (Execute Program) ที่มีคุณสมบัติสอดคล้องกับฮาร์ดแวร์ปลายทางที่กำหนด จะเหมือนกับการออกแบบโดยการเขียนโปรแกรมด้วยภาษาระดับสูงทั่วๆ ไป และสามารถทำงานบนฮาร์ดแวร์ได้หลายๆ แบบ ดังนี้

1) คอมพิวเตอร์ (Computer)

2) ไมโครคอนโทรลเลอร์ (Microcontroller)

3) ไมโครโปรเซสเซอร์ (Microprocessor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) DSP (Digital Signal Processor)

5) FPGA (Field Programmable Gate Array)

กระบวนการออกแบบโปรแกรมจากเดิมที่ต้องศึกษาคุณลักษณะของระบบให้เข้าใจก่อน แล้วจึงใช้ภาษาโปรแกรมบรรยายพฤติกรรมของระบบออกมาตามคุณลักษณะของภาษาที่ใช้ จะเห็นว่าผู้ออกแบบจะต้องมีความรู้ด้านการทำงานของระบบ การใช้งานภาษาโปรแกรมและการประยุกต์ใช้งานโปรแกรม ดังนั้นโปรแกรม MATLAB เป็นอีกทางเลือกหนึ่งของนักออกแบบโปรแกรม โดยอาศัยการออกแบบลักษณะที่มีการโต้ตอบกับผู้ใช้งานได้ซึ่งจะใช้ Simulink ในการออกแบบ ซึ่งมีลักษณะเป็นบล็อกของระบบย่อยๆ แต่ละชนิดมาเชื่อมต่อกันเป็นระบบใหญ่ แล้วใช้ Real-Time Workshop ทำการสร้างโค๊ดภาษาซีทั้งระบบหรือเป็นไฟล์กระทำการก็ได้ ซึ่งผู้ออกแบบโปรแกรมไม่ต้องเขียนโปรแกรมในลักษณะเป็นคำสั่งอีกต่อไป ลดเวลาในการศึกษาภาษาโปรแกรม ลดขั้นตอนในการพัฒนาโปรแกรมและลดเวลาในการออกแบบได้อีกด้วย ซึ่งจะเหมาะกับผู้ออกแบบระบบในเชิงวิศวกรรมที่มีความซับซ้อนสูงแต่ต้องการลดเวลาในการออกแบบ

2.2.1 ความต้องการด้านซอฟต์แวร์

การใช้งาน Real-Time Workshop จะต้องมีการติดตั้งซอฟต์แวร์ต่างๆ ก่อนจึงจะสามารถใช้งานได้ มีรายละเอียดดังนี้

1) โปรแกรม MATLAB

2) โปรแกรม Simulink

3) กล่องเครื่องมือ Real-Time Workshop

4) กล่องเครื่องมือ Real-Time Workshop Embedded Coder

5) โปรแกรมแปลภาษาซี (C/C++ Compiler)

2.2.2 การใช้งานตัวแปลภาษา

ก่อนการใช้งาน Real-Time Workshop จะต้องมีการติดตั้งตัวแปลภาษาซีก่อน ซึ่งผู้ใช้งานต้องติดตั้งเองหรืออาจจะใช้ตัวแปลภาษาที่โปรแกรม MATLAB มีให้แล้วก็ได้ โดยจะมีหน้าที่ในการสร้างโค๊ดภาษาซีจากระบบที่ออกแบบจาก Simulink ซึ่งจะนำค่าพารามิเตอร์ต่างๆ ของระบบมาเขียนเป็นตัวแปร ค่าคงที่ และจะบรรยายความสัมพันธ์ของข้อมูลทั้งระบบในแบบโครงสร้างภาษาซี โดยจะอาศัยความสามารถของตัวแปลภาษาซีในการสร้างโค๊ด ถือว่าเป็นส่วนที่สำคัญที่สุดในการใช้งาน ซึ่งก็จะมีโปรแกรมแปลภาษาที่โปรแกรม MATLAB รองรับอยู่มีดังนี้

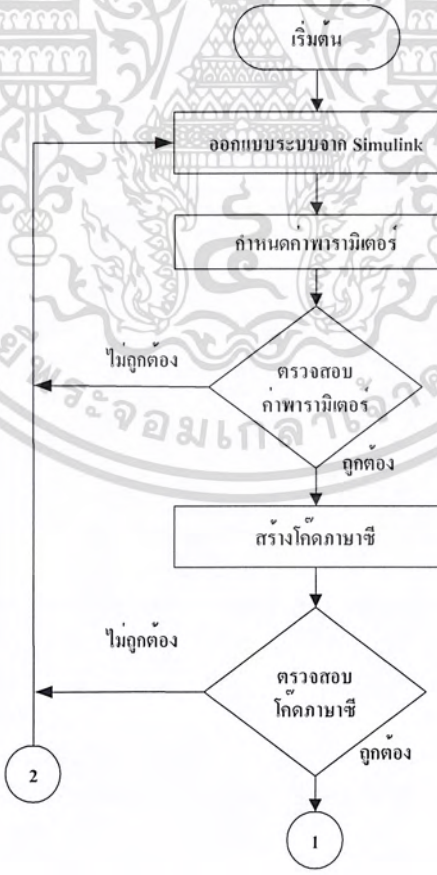
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 โปรแกรมแปลภาษาซีที่โปรแกรม MATLAB รองรับ

โปรแกรม	เวอร์ชั่น
Borland	5.2, 5.3, 5.4, 5.5 หรือ 5.6
Intel	7.1
LCC	เป็น Free Ware ติดตั้งพร้อมกับโปรแกรม MATLAB
Microsoft Visual C/C++	5.0, 6.0 หรือ 7.0
Watcom	11.0

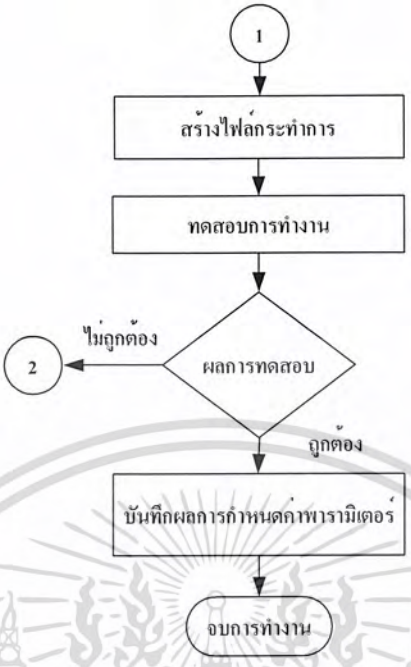
2.2.3 ขั้นตอนการทำงานของ Real-Time Workshop

การใช้งาน Real-Time Workshop นั้นจะมีขั้นตอนการทำงาน แสดงผังงานดังรูป



รูปที่ 2.1 ผังงานขั้นตอนการทำงานของ Real-Time Workshop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 (ต่อ) ผังงานขั้นตอนการทำงานของ Real-Time Workshop

การใช้งาน Real-Time Workshop นั้นแบ่งเป็นขั้นตอนได้ดังนี้

1) การออกแบบระบบจาก Simulink

ขั้นตอนนี้เป็นขั้นตอนแรกในการใช้งาน โดยจะใช้ Simulink ในการออกแบบระบบที่ต้องการและสามารถจำลองการทำงานของระบบก่อนนำไปใช้งานจริงได้ ไฟล์ที่ได้จะมีชื่อ model.mdl ตัวอย่างระบบที่ออกแบบจาก Simulink แสดงดังรูปที่ 2.2

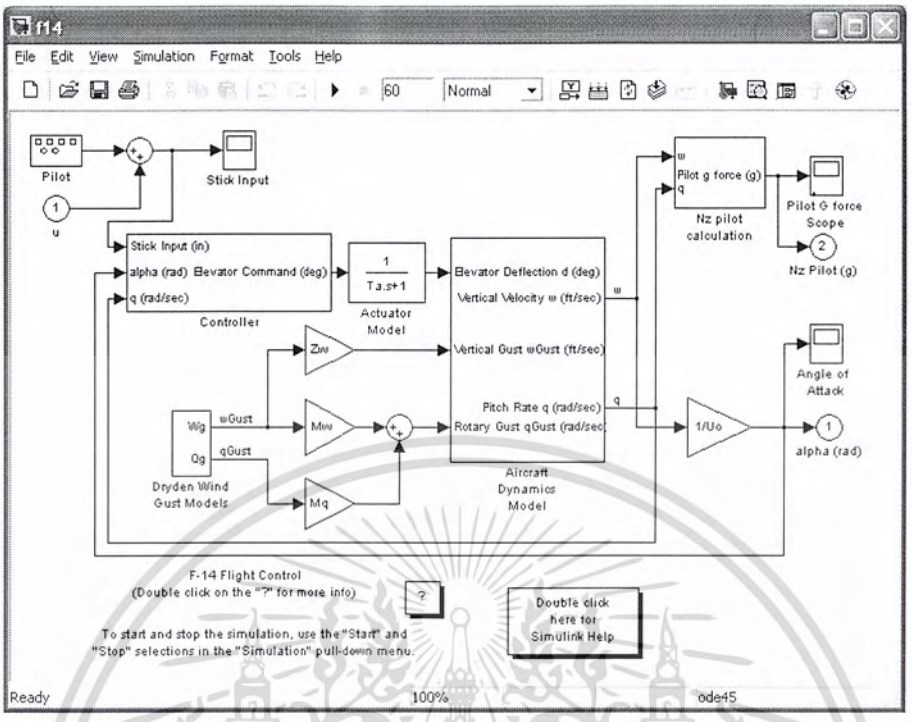
2) การกำหนดค่าพารามิเตอร์

หลังจากที่ได้ออกแบบระบบเป็นที่เรียบร้อยแล้วขั้นตอนนี้จะเป็นการกำหนดค่าพารามิเตอร์ต่างๆ ที่สำคัญ ซึ่งเป็นค่าที่กำหนดคุณสมบัติต่างๆ ของการสร้างโค๊ดภาษาซีทั้งหมด เช่น การกำหนดว่าจะให้ระบบที่ออกแบบทำงานบนไมโครโปรเซสเซอร์ หน้าต่างการกำหนดค่าพารามิเตอร์แสดงดังรูปที่ 2.3

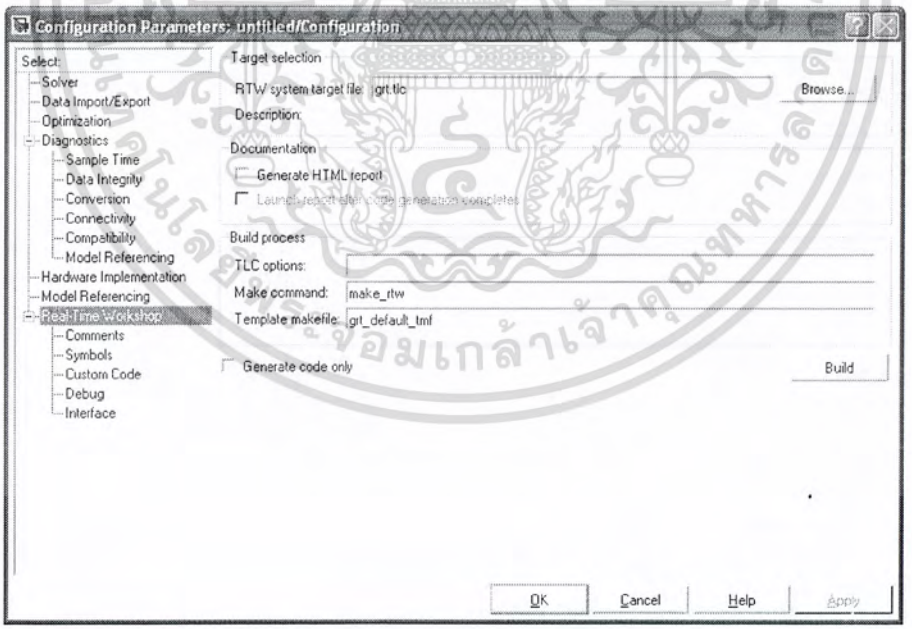
3) การตรวจสอบค่าพารามิเตอร์

หลังจากที่เราได้กำหนดค่าพารามิเตอร์ต่างๆ แล้ว สามารถที่จะตรวจสอบความถูกต้องของค่าพารามิเตอร์ต่างๆ อีกครั้งได้หรืออาจจะข้ามขั้นตอนนี้ไปได้เลย หน้าต่างการตรวจสอบค่าพารามิเตอร์แสดงดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

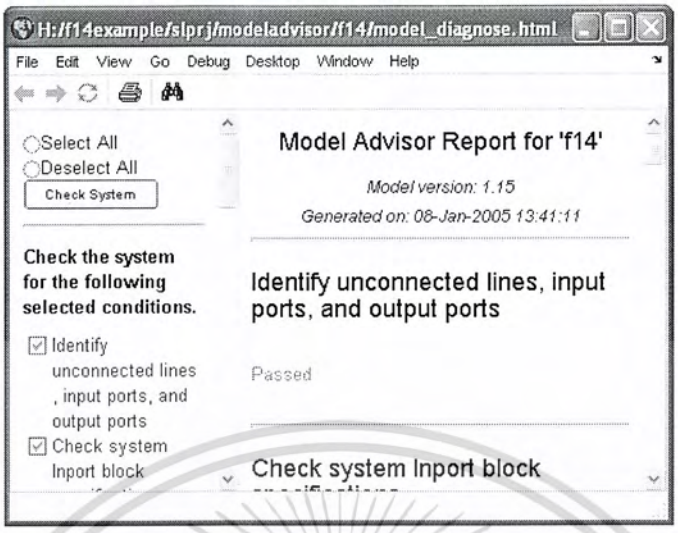


รูปที่ 2.2 ตัวอย่างการออกแบบระบบจาก Simulink



รูปที่ 2.3 หน้าต่างการกำหนดค่าพารามิเตอร์

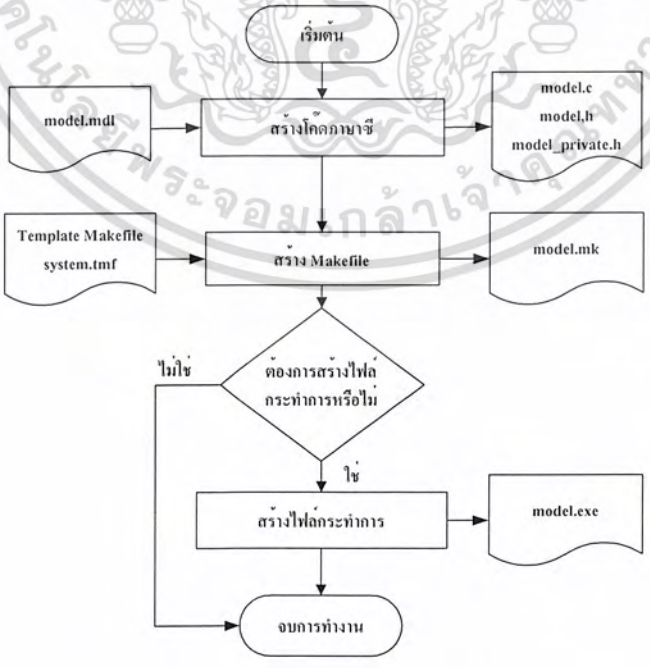
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 หน้าต่างการตรวจสอบค่าพารามิเตอร์

4) การสร้างโค๊ดภาษาซี

หลังจากที่ทำการตรวจสอบค่าพารามิเตอร์ว่าถูกต้องแล้ว ขั้นตอนนี้จะทำการสร้างโค๊ดภาษาซีจากโมเดลที่ออกแบบ ซึ่งจะได้ไฟล์ภาษาซีต่างๆ แสดงขั้นตอนการสร้างโค๊ดเป็นผังงานแสดงดังรูปที่ 2.5



รูปที่ 2.5 ผังงานขั้นตอนการสร้างโค๊ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

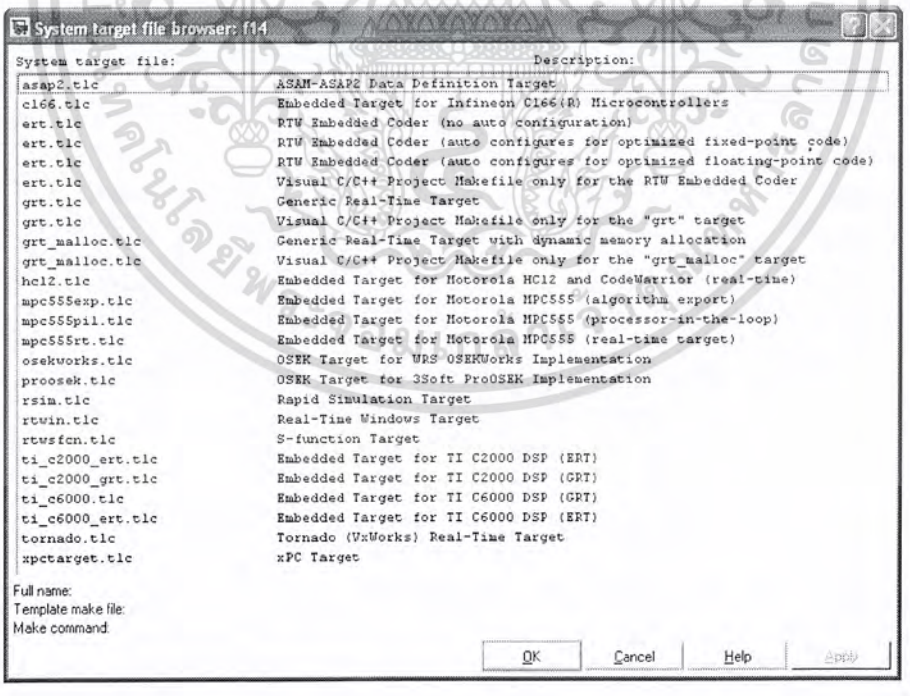
การสร้างโค๊ดภาษาซีจะมีขั้นตอนดังนี้

4.1) การสร้างโค๊ดภาษาซี

จะเป็นการสร้างโค๊ดภาษาซีจากโมเดลที่ออกแบบจาก Simulink ซึ่งจะเป็นไฟล์นามสกุล .mdl โดยตัวแปลภาษาซีจะนำค่าพารามิเตอร์ต่างๆ จากโมเดล มาทำการแปลงให้อยู่ในรูปของภาษาโปรแกรม หลังจากที่ได้ทำกระบวนการสร้างโค๊ดแล้วจะได้โค๊ดภาษาซี เช่น model.c, model.h หรือ model_private.h เป็นต้น

4.2) การสร้าง Makefile

เมื่อเราทำการสร้างโค๊ดภาษาซีจะได้ไฟล์ชนิดหนึ่งที่มีชื่อเป็น model.mk (Makefile) คือเป็นไฟล์ที่กำหนดคุณลักษณะของไฟล์กระทำการ model.exe (Execute) ว่าจะนำไปใช้ที่ปลายทางระบบใด เช่น นำไปใช้กับชิพ DSP (Digital Signal Processor) หรือไมโครโปรเซสเซอร์ เป็นต้น ไฟล์ model.mk นั้นจะถูกสร้างมาจาก Template Makefile ที่มีชื่อเป็น system.tmf โดย Template Makefile นี้จะถูกกำหนดในขั้นตอนการกำหนดค่าพารามิเตอร์ ถ้ากำหนด TLC (Target Language-Compiler) เป็น ti_c6000_grt.tlc จะหมายความว่า ต้องการให้ระบบที่เราออกแบบทำงานร่วมกับชิพ DSP ตระกูล 6000 ดังนั้นไฟล์ system.tmf และ model.mk ก็จะเก็บคุณสมบัติการเชื่อมต่อกับชิพ DSP ไว้นั้นเอง โดย TLC จะมีอยู่หลายชนิดแสดงดังรูปที่ 2.6



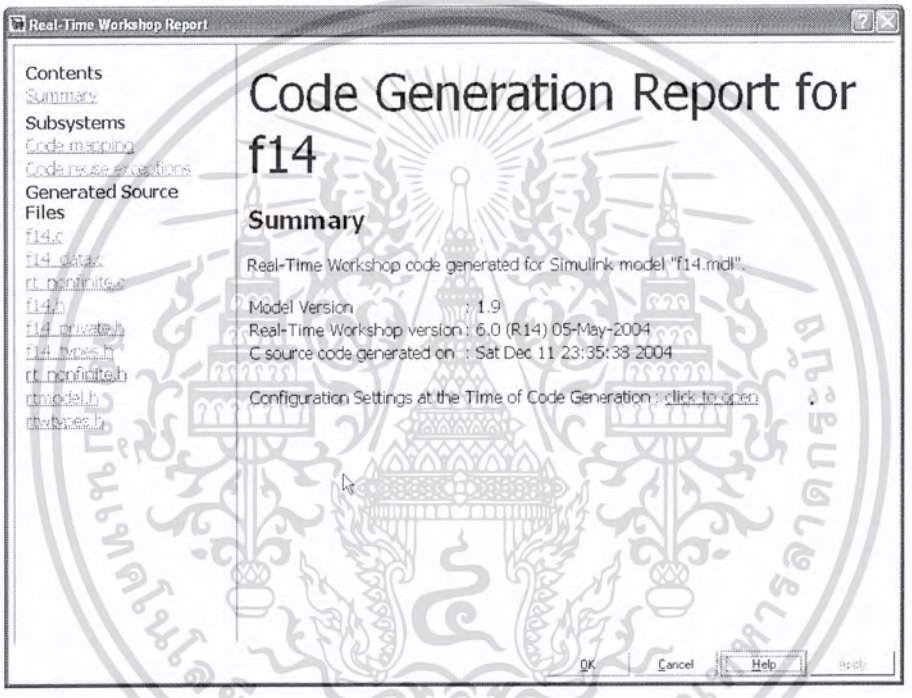
รูปที่ 2.6 หน้าต่างการแสดงชนิดของ TLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3) จะเป็นการตรวจสอบว่าต้องการสร้างโค๊ดภาษาซีอย่างเดียวหรือต้องการสร้างไฟล์กระทำการด้วย ซึ่งถ้ากำหนดให้สร้างโค๊ดภาษาซีอย่างเดียว ก็จะข้ามขั้นตอนการสร้างไฟล์กระทำการไป เหตุผลที่ต้องกำหนดให้สร้างโค๊ดภาษาซีอย่างเดียวเนื่องจากจะต้องดูความถูกต้องของโค๊ดทั้งหมดก่อนที่จะสร้างเป็นไฟล์กระทำการเพื่อใช้งานจริง

5) การตรวจสอบโค๊ดภาษาซี

ขั้นตอนนี้จะเป็นการตรวจสอบความถูกต้องของโค๊ดภาษาซีที่ได้สร้างขึ้นมาก่อน โดยสามารถดูได้จากรายงานผลการสร้างโค๊ด แสดงดังรูปที่ 2.7



รูปที่ 2.7 หน้าต่างการตรวจสอบโค๊ดภาษาซี

6) การสร้างไฟล์กระทำการ

ขั้นตอนนี้จะคล้ายกับขั้นตอนของการสร้างโค๊ดภาษาซี แตกต่างกันที่เมื่อสร้างโค๊ดภาษาซีเสร็จแล้วจะทำการอ่านค่าจากไฟล์ model.mk แล้วจึงทำการสร้างไฟล์กระทำการได้ โดยจะมีชื่อเป็น model.exe

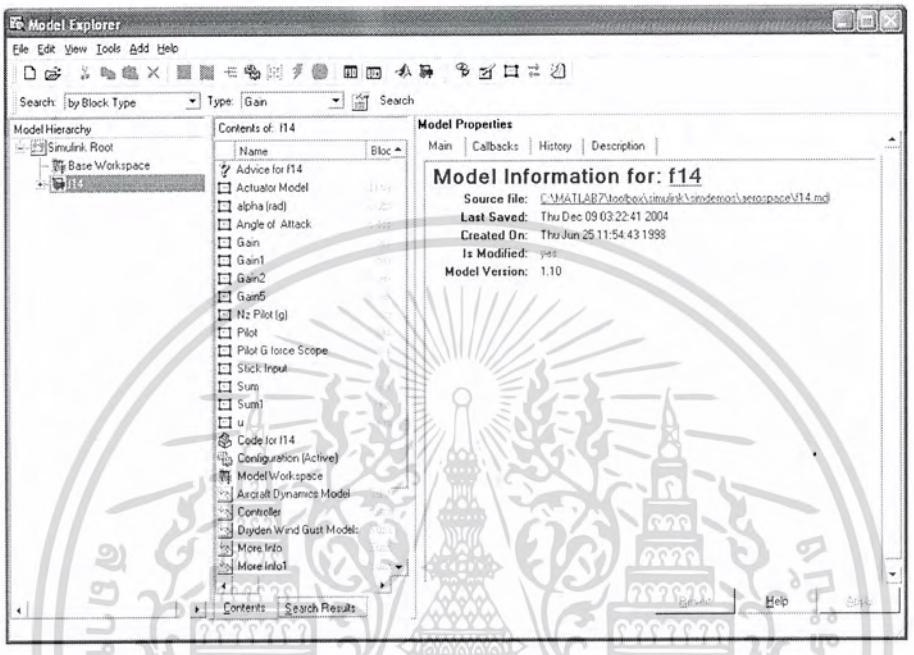
7) การทดสอบการทำงาน

เมื่อได้ไฟล์กระทำการแล้วขั้นตอนต่อไปจะทำการตรวจสอบความถูกต้องของไฟล์ โดยจะใช้โปรแกรม MATLAB หรือ Simulink ในการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) การบันทึกรูปแบบการกำหนดค่าพารามิเตอร์

เมื่อได้ไฟล์กระทำการเรียบร้อยแล้วสามารถที่จะเก็บรูปแบบการกำหนดค่าพารามิเตอร์ไว้ได้ แสดงดังรูปที่ 2.8



รูปที่ 2.8 หน้าต่างการบันทึกรูปแบบการกำหนดค่าพารามิเตอร์

2.2.4 ความหมายของไฟล์ภาษาซีที่ได้จากการสร้างโค๊ด

ความหมายของไฟล์ภาษาซีต่างๆ มีดังนี้

1) ไฟล์ model.c

เป็นโค๊ดภาษาซีที่ถูกสร้างขึ้นโดยตัวแปรภาษา ซึ่งจะเป็นไฟล์หลักของไฟล์ทั้งหมด การทำงานของ model.c จะเหมือนกับโมเดลที่ออกแบบโดย Simulink ทุกประการ มีรายละเอียดดังนี้

1.1) จะทำการเรียกฟังก์ชั่นย่อย model.h และ model _private.h มาใช้งาน

1.2) ค่าพารามิเตอร์ทั้งหมดจะทำการเรียกจาก model_data.c

1.3) จะมีโค๊ดของอัลกอลิทึมซึ่งสร้างจากโมเดลใน Simulink

1.4) ถ้าเรากำหนดคุณสมบัติพิเศษเป็น GRT wrapper จะปรากฏไฟล์เพิ่มขึ้นดังนี้

MdlStart, MdlOutputs, MdlUpdate, MdlInitializeSizes, MdlInitialize Sample Times

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ไฟล์ model.h

เป็นไฟล์ฟังก์ชั่นที่จะเก็บพารามิเตอร์ทั่วไปและพารามิเตอร์ที่กำหนดการเชื่อมต่อของไฟล์กับปลายทาง (Target) ที่เรากำหนดไว้ โดยข้อมูลโครงสร้าง (Data Structure) ของโปรแกรมจะถูกกำหนดโดยไฟล์ model_rtM ซึ่งจะถูกทำการเรียกจากไฟล์ model.h มีรายละเอียดดังนี้

2.1) เป็นค่าข้อมูลที่ได้จากโมเดลใน Simulink

2.2) ประกอบด้วยข้อมูลโครงสร้างทั้งหมดของโปรแกรมซึ่งเรียกจาก rtM

3) ไฟล์ model _private.h

จะประกอบด้วยข้อมูลและค่าพารามิเตอร์ที่สร้างระบบย่อย (Subsystem) ในกรณีที่เราออกแบบจาก Simulink ที่มีระบบย่อย

4) ไฟล์ model _types.h

เป็นไฟล์การประกาศค่าตัวแปรและชนิดของตัวแปรทั้งหมด

5) ไฟล์ model _data.c

เป็นไฟล์ที่ประกาศค่าพารามิเตอร์ทั้งหมด เช่น ข้อมูลโครงสร้างค่าคงที่ ค่าข้อมูลอินพุตและเอาต์พุต ชนิดของข้อมูล ในกรณีที่ไม่มีค่าข้อมูลที่ต้องใช้ในโมเดล ไฟล์ model _data.c ไม่ถูกสร้างขึ้น มีรายละเอียดดังนี้

5.1) ค่าพารามิเตอร์อินพุตและเอาต์พุตที่ใช้งานในโปรแกรมทั้งหมด

5.2) จะทำการอ้างอิงข้อมูลจาก model.h และ model_private.h

6) ไฟล์ model.exe

เป็นไฟล์กระทำการซึ่งสามารถนำไปใช้งานได้เลยการใช้งานขึ้นอยู่กับการกำหนด TLC (Target Language Compiler) เช่น ถ้ากำหนดเป็น GRT จะใช้งานทั่วไป โดยไฟล์นี้จะถูกสร้างขึ้นจากไฟล์ภาษาซีทั้งหมดที่ได้สร้าง ซึ่งจะประกอบด้วยคุณสมบัติการทำงานเหมือนกับโมเดลที่ออกแบบจาก Simulink

7) ไฟล์ model.mk

เป็นไฟล์ .mk (Make File) ถูกสร้างขึ้นเพื่อทำการสร้างไฟล์ model.exe

8) ไฟล์ rtmodel.h

เป็นไฟล์ที่ถูกสร้างขึ้นมากรณีที่เรากำหนด TLC เป็น grt_malloc.tlc จะทำการเก็บค่าพารามิเตอร์ต่างๆ โดยจะแตกต่างจาก TLC ชนิดอื่นๆ ซึ่งจะเป็น Memory Application ต่างๆ

9) ไฟล์ rtwtypes.h

เป็นไฟล์ที่ใช้อ้างอิงชนิดของ TLC เช่น ในกรณีที่เราทำการเลือก TLC เป็น grt.tlc ไฟล์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะกำหนดให้ไฟล์อ้างอิงเป็น simstruc_types.h และ rtwtypes.h โดยจะอ้างอิงข้อมูลจาก TLC เพื่อให้เหมาะสมกับการใช้งานมากที่สุด

10) ไฟล์ rt_nonfinite.c

เป็นไฟล์ที่กำหนดค่าพารามิเตอร์ที่สามารถปรับแต่งได้ เช่น inf, minus, inf และ nan

11) ไฟล์ rt_nonfinite.h

เป็นไฟล์หัวโปรแกรมที่จะเก็บค่าที่สามารถเปลี่ยนแปลงได้ให้กับไฟล์ rt_nonfinite.c

12) ไฟล์ rtw_proj.tmw

เป็นไฟล์เก็บค่าอ้างอิงการสร้างโค๊ดภาษาซี เช่น ที่เก็บตัวแปลภาษาซีหรือที่เก็บไฟล์ที่จำเป็นในการสร้างโค๊ดภาษาซี

13) ไฟล์ F14.bat

เป็นไฟล์ Windows Batch file ที่สำหรับติดตั้งค่าพารามิเตอร์ต่างๆ ที่ในการสร้างโค๊ดภาษาซีให้สอดคล้องกับระบบปฏิบัติการ Windows

14) ไฟล์ modelsources.txt

เป็นไฟล์ที่รายละเอียดเกี่ยวกับไฟล์ทั้งหมด

15) ไฟล์ model_targ_data_map.m

จะเป็นไฟล์ .m (โปรแกรม MATLAB file) ซึ่งจะเป็นค่าข้อมูลเริ่มต้นใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ซึ่งจะมีไฟล์นี้ในกรณีเลือกโหมดการติดต่อกับภายนอก (External Mode) เท่านั้น

16) ไฟล์ model_dt.h

เป็นไฟล์หัวโปรแกรมที่เก็บค่าพารามิเตอร์ ชนิดของข้อมูล และข้อมูลโครงสร้างต่างๆ ในโหมดการติดต่อกับภายนอก

17) ไฟล์ subsystem.c

จะเป็นไฟล์กำหนดค่าพารามิเตอร์ ในกรณีที่เรามีออกแบบโมเดลใน Simulink ที่มีระบบย่อย

18) ไฟล์ subsystem.h

เป็นไฟล์หัวโปรแกรมที่ใช้อ้างอิงในไฟล์ subsystem.c

19) ไฟล์ model_capi.h

เป็นไฟล์หัวโปรแกรมที่ใช้อ้างอิงระหว่างไฟล์โมเดลที่สร้างขึ้นกับไฟล์ C-API จะมีไฟล์นี้ในกรณีที่เรากำหนดเป็นโหมดการติดต่อภายนอก

20) ไฟล์ model_capi.c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไฟล์หลักในการกำหนดค่าข้อมูลหรือสัญญาณที่ใช้เชื่อมต่อระหว่างไฟล์ที่สร้างขึ้นกับไฟล์ C-API

21) ไฟล์ model_rt_sfcn_helper.h, rt_sfcn_helper.c

เป็นไฟล์หัวโปรแกรมและไฟล์หลักที่เก็บฟังก์ชั่นต่างๆ เช่น rt_CallSys, rt_enableSys, and rt_DisableSys จะมีไฟล์นี้ในกรณีที่เราออกแบบเป็น S-functions

2.3 การใช้งาน xPC Target

การประยุกต์ใช้งานโปรแกรม MATLAB ในการเชื่อมต่อกับฮาร์ดแวร์นั้น โปรแกรม MATLAB สามารถที่จะใช้เชื่อมต่อได้หลายชนิด เช่น ไมโครคอนโทรลเลอร์ ไมโครโปรเซสเซอร์ หรือชิพ DSP (Digital Signal Processor) โดยฮาร์ดแวร์อีกประเภทหนึ่งที่โปรแกรม MATLAB สามารถเชื่อมต่อได้ก็คือเครื่องคอมพิวเตอร์คือ ใช้กล่องเครื่องมือ xPC Target ในการเชื่อมต่อคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไป ซึ่งจะมีข้อดีกว่าการเชื่อมต่อกับฮาร์ดแวร์ประเภทอื่น เนื่องจากคอมพิวเตอร์สามารถที่จะประยุกต์ใช้งานได้กว้างกว่าแบบอื่นและสามารถที่จะเชื่อมต่อกับอุปกรณ์อินพุตและเอาต์พุตต่างๆ ได้ โดยผ่านระบบบัสต่างๆ ของคอมพิวเตอร์

2.3.1 ความหมายของ xPC Target

xPC Target เป็นกล่องเครื่องมือหนึ่งที่อยู่ในโปรแกรม MATLAB เป็นการประยุกต์ใช้งาน Real-Time Workshop รูปแบบหนึ่ง มีความสามารถในการเชื่อมต่อเครื่องคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไป โดยใช้เครื่องคอมพิวเตอร์ต้นทาง (Host PC) ควบคุม เครื่องคอมพิวเตอร์ปลายทาง (Target PC) ให้สามารถถ่ายโอนข้อมูลกันได้ โดยจะใช้กล่องเครื่องมือ Real-Time Workshop ทำหน้าที่สร้างไฟล์กระทำการ (Execute Program) ที่ใช้งานแบบ xPC Target การเชื่อมต่อ Host PC และ Target-PC นั้นสามารถเชื่อมต่อทางการสื่อสารแบบอนุกรม (Serial Communication) และแบบระบบเครือข่าย (Network Communication) คุณสมบัติพิเศษอีกประการหนึ่งของ xPC Target คือ ทำงานร่วมกับอุปกรณ์อินพุตหรือเอาต์พุตหลายชนิดได้ เช่น ที่ Target PC มีการ์ดแปลงสัญญาณดิจิตอลเป็นแอนะลอกต่ออยู่ (Digital-to-analog converter : DAC) ดังนั้นที่ Host PC จะสามารถส่งข้อมูลดิจิตอลไปยัง Target PC แล้วที่ Target PC จะแปลงข้อมูลดิจิตอลนั้นเป็นแอนะลอก เพื่อเชื่อมต่อกับอุปกรณ์อื่นๆ ได้ทำให้สามารถประยุกต์ใช้งานได้กว้างมาก

ลักษณะเด่นอีกประการหนึ่งของ xPC Target คือ มีความต้องการทางระบบ (System - Requirements) ทั้งซอฟต์แวร์ (Software Requirement) และฮาร์ดแวร์ (Hardware Requirement) ของ Target PC ต่ำมาก เช่น ที่ Target PC ไม่จำเป็นมีระบบปฏิบัติการ (Operating System) ใดๆ เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก ทาง xPC Target จะสร้างระบบระบบปฏิบัติการของตนเองขึ้นมาใช้งาน เรียกว่า xPC Target kernel

2.3.2 ความต้องการของระบบ (System Requirements)

จะแบ่งเป็น 2 หัวข้อ คือ ที่ Host PC และ Target PC

1) คอมพิวเตอร์ต้นทาง (Host PC)

ที่คอมพิวเตอร์ต้นทางเราจะต้องมีระบบปฏิบัติการอย่างใดอย่างหนึ่งที่สามารถใช้งานกับโปรแกรม MATLAB ได้ เช่น Microsoft Windows Platform และต้องมี Floppy Disk Drive ขนาด 3.5 นิ้ว และต้องมี Serial port หรือ Ethernet adapter card เพื่อใช้ในการเชื่อมต่อ

ตารางที่ 2.2 ความต้องการทางด้านซอฟต์แวร์ขั้นต่ำ (Minimum Software Requirement)

ซอฟต์แวร์	รายละเอียด
ระบบปฏิบัติการ (Operating System)	ต้องมีระบบปฏิบัติการที่ใช้งานโปรแกรม MATLAB ได้ดังนี้ -Windows Platform -Linux ix86 2.4.x, glibc 2.2.5 -Sun Solaris 2.8 หรือ 2.9 -HPUX 11.0 หรือ 11.i -Mac OS X 10.3.2
โปรแกรม MATLAB	เช่น โปรแกรม MATLAB เวอร์ชั่น 7.0
โปรแกรม Simulink	เช่น Simulink 6.0
กล่องเครื่องมือ Real-Time Workshop	เช่น Real-Time Workshop 6.0
โปรแกรมแปลภาษาซี	-Borland 5.2, 5.3, 5.4, 5.5, 5.6 -Intel 7.1 -LCC -Watcom C/C++ 11.0 -Microsoft Visual C/C++ Professional 5.0, 6.0, 7.0
กล่องเครื่องมือ xPC Target	เช่น xPC Target 2.6.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ความต้องการทางด้านฮาร์ดแวร์ขั้นต่ำ (Minimum Hardware Requirement)

ฮาร์ดแวร์	รายละเอียด
อุปกรณ์เชื่อมต่อ	-พอร์ตอนุกรม (Serial Port) ขนาด 9pin หรือ 25pin -การ์ด LAN (Ethernet Adapter Card)
หน่วยประมวลผลกลาง (CPU)	-Pentium -Athlon -หรือรุ่นอื่นๆ
อุปกรณ์ที่เกี่ยวข้อง	แผ่นฟล็อปปี้ดิสก์ 3.5 นิ้ว
หน่วยความจำ (RAM)	ขั้นต่ำ128 เมกะไบต์

2) คอมพิวเตอร์ปลายทาง (Target PC)

การใช้งาน xPC Target นั้นที่ Target PC จะมีได้มากกว่า 1 ตัวควบคุมด้วย Host PC เพียงเครื่องเดียว โดย Target PC ต้องมี Floppy disk drive ขนาด 3.5 นิ้ว และต้องมีพอร์ตอนุกรม (Serial-Port) หรือการ์ด LAN (Ethernet Adapter Card) เพื่อใช้ในการเชื่อมต่อ

ลักษณะคอมพิวเตอร์สามารถเลือกได้ดังนี้

2.1) เครื่องคอมพิวเตอร์ส่วนบุคคล (Desktop PC)

2.2) เครื่องคอมพิวเตอร์ในงานอุตสาหกรรม (Industrial PC) เช่น PC104, PC104+, CompactPCI หรือ Single-board computer (SBC)

ตารางที่ 2.4 ความต้องการทางด้านซอฟต์แวร์ขั้นต่ำ (Minimum Software Requirement)

ซอฟต์แวร์	รายละเอียด
ระบบปฏิบัติการ (Operating System)	ไม่ต้องมี เพราะ xPC Target จะสร้างระบบปฏิบัติการของตนเองชื่อ xPC Target kernel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่ 2.5 ความต้องการทางด้านฮาร์ดแวร์ขั้นต่ำ (Minimum Hardware Requirement)

ฮาร์ดแวร์	รายละเอียด
อุปกรณ์เชื่อมต่อ	-พอร์ตอนุกรม (Serial Port) ขนาด 9 pin หรือ 25 pin -การ์ด LAN (Ethernet Adapter Card)
หน่วยประมวลผลกลาง (CPU)	-Intel 386/486/Pentium -AMD K5/K6/Athlon
อุปกรณ์ที่เกี่ยวข้อง	แผ่นฟล็อปปี้ดิสก์ 3.5 นิ้ว
หน่วยความจำ (RAM)	ขั้นต่ำ 8 เมกะไบต์

2.3.3 การใช้งานอุปกรณ์อินพุตและเอาต์พุต (I/O Support)

ใน xPC Target มีรองรับอุปกรณ์อินพุตและเอาต์พุตมาตรฐานกว่า 150 ชนิด ซึ่งการใช้งานจะเชื่อมต่ออยู่ที่ target PC ซึ่งจะต้องเป็นอุปกรณ์ของบริษัทที่ร่วมมือกับ Mathworks เท่านั้น เช่น Advantech, Humusoft, National Instrument, Texas Instrument, Motorola, Infineon เป็นต้น

การแบ่งอุปกรณ์อินพุตและเอาต์พุตที่รองรับตามฟังก์ชั่นการทำงานสามารถแบ่งได้ดังนี้

1) อินพุตแบบแอนะลอกและเอาต์พุตแบบแอนะลอก

เป็นการเชื่อมต่อเพื่อรับส่งข้อมูลระหว่าง Target PC กับอุปกรณ์ภายนอกที่เป็นแอนะลอก ขึ้นอยู่กับการประยุกต์ใช้งาน

2) อินพุตแบบดิจิตอลและเอาต์พุตแบบดิจิตอล

เป็นการเชื่อมต่อเพื่อรับส่งข้อมูลระหว่าง Target PC กับอุปกรณ์ภายนอกที่เป็นดิจิตอล ขึ้นอยู่กับการประยุกต์ใช้งาน

3) การสื่อสารแบบอนุกรม RS-232/422/485

เป็นการใช้งานการสื่อสารแบบอนุกรมเชื่อมต่อทางพอร์ต COM1 หรือ COM2 เพื่อรับส่งข้อมูลระหว่าง Target PC กับอุปกรณ์ภายนอกที่รองรับการสื่อสารแบบอนุกรม ซึ่งจะรองรับมาตรฐานการเชื่อมต่อแบบอนุกรมหลายชนิด คือ RS-232, RS-422 และ RS-485 อาจจะใช้จากเมนบอร์ดหรือการ์ดจากบริษัทต่างๆ ก็ได้ เช่น บริษัท Quatech เป็นต้น

4) CAN : Controller Area Network

เป็นการใช้งานการควบคุมระบบเครือข่ายที่เรียกว่า CAN : Controller Area Network จะรองรับมาตรฐาน CAN-AC2, CAN-AC2-PCI และ CAN-AC2-104 ซึ่งเป็นบอร์ด CAN ของบริษัท

59489

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Softing โดย xPC Target จะรองรับมาตรฐานของระบบบัสแบบ CAN 2.0A และ 2.0B การทำงานจะเชื่อมต่อกับระบบเครือข่ายเพื่อใช้ในการติดต่อสื่อสารหรือจะประยุกต์การใช้งานในด้านอื่นๆ

5) GPIB : General Purpose Interface Bus

เป็นการใช้งานควบคุมอุปกรณ์เครื่องมือวัดในอุตสาหกรรมหรือเครื่องมือวัดทั่วๆ ไป ซึ่งจะเชื่อมต่อโดยใช้มาตรฐาน IEEE โดยใน xPC Target จะรองรับอุปกรณ์ของ GPIB ของบริษัท National Instruments เท่านั้น

6) UDP/IP : User Datagram Protocol Internet Protocol

เป็นการใช้งานการเชื่อมต่อกับระบบเครือข่ายตามมาตรฐาน UDP/IP

7) Watchdog

เป็นการใช้งานในลักษณะตรวจจับการเปลี่ยนแปลงที่เกิดขึ้นที่ปลายทาง เช่น เป็นมอนิเตอร์ดูการทำงานของเครื่องคอมพิวเตอร์ที่อยู่ปลายทางหรือสั่งเปิดหรือปิดเครื่องจากเครื่องต้นทาง

8) Shared memory

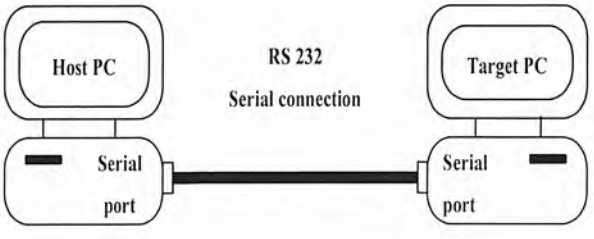
เป็นการใช้งานกรณีที่เรามีคอมพิวเตอร์ปลายทางหลายเครื่องเราสามารถใช้หน่วยความจำของคอมพิวเตอร์ร่วมกันได้โดยการเรียกบล็อก Shared Memory ใน xPC Target

2.3.4 ลักษณะการเชื่อมต่อระหว่าง Host PC กับ Target PC

ลักษณะการเชื่อมต่อ Host PC กับ Target PC จะมีอยู่ 2 ลักษณะคือ ผ่านทางการสื่อสารแบบอนุกรมและระบบเครือข่าย รายละเอียดมีดังนี้

1) การสื่อสารแบบอนุกรม (Serial Communication)

เป็นการติดต่อสื่อสารแบบอนุกรมใช้พอร์ตของคอมพิวเตอร์ COM1 หรือ COM2 หรืออาจเป็นการ์ดการสื่อสารแบบอนุกรมของบริษัทต่างๆ ก็ได้แต่ต้องรองรับกับ xPC Target ด้วยจะใช้สาย null modem ในการเชื่อมต่อ ความยาวของสายไม่ควรเกิน 5 เมตร โดยสามารถกำหนดความเร็วในการส่งถ่ายข้อมูลระหว่าง 1200 ถึง 115200 บอด (Baud) ลักษณะการเชื่อมต่อแสดงดังรูปที่ 2.9

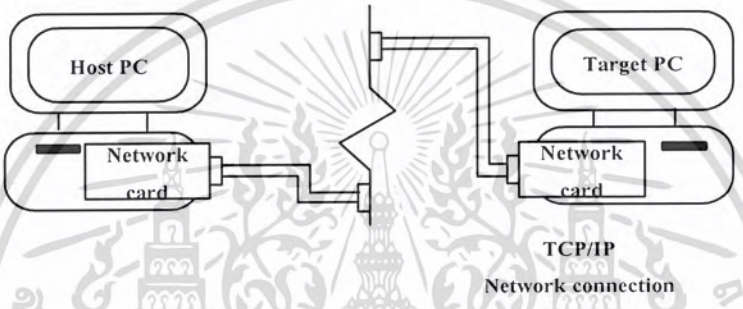


รูปที่ 2.9 การเชื่อมต่อแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การสื่อสารแบบระบบเครือข่าย

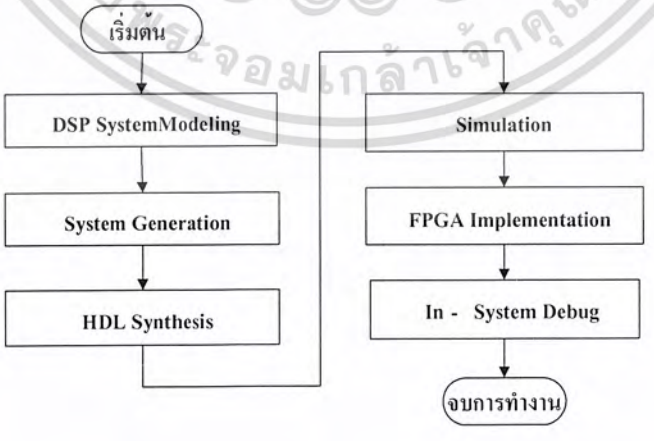
เป็นการเชื่อมต่อแบบระบบเครือข่าย เช่น ระบบเครือข่ายขนาดเล็ก อินเตอร์เน็ต หรือการเชื่อมต่อโดยตรงผ่านสาย ซึ่งการเชื่อมต่อแบบนี้จะต้องมีการ์ด LAN ซึ่งจะอาศัยโปรโตคอล TCP/IP ในการติดต่อสื่อสารระหว่าง Host PC และ Target PC ชนิดของการ์ด LAN นี้ทาง xPC Target จะต้องรองรับด้วย ถัดไปความเร็วในการส่งถ่ายข้อมูลจะขึ้นอยู่กับระบบเครือข่ายและขีดความสามารถของอุปกรณ์ในระบบแต่ที่ใช้กันในปัจจุบันจะอยู่ที่ 10 Mb/s หรือ 100 Mb/s และระยะทางในการติดต่อสื่อสารจะได้ไกลกว่าแบบอนุกรมมาก แสดงดังรูปที่ 2.10



รูปที่ 2.10 การเชื่อมต่อแบบระบบเครือข่าย

2.3.5 ขั้นตอนการทำงานของ xPC Target

ขั้นตอนการทำงานของ xPC Target สามารถแสดงเป็นผังงานดังนี้



รูปที่ 2.11 ผังงานแสดงขั้นตอนการทำงานของ Xpc Target

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนนี้จะเป็นขั้นตอนแรก โดยจะต้องทำการออกแบบระบบที่ต้องการจาก Simulink ซึ่งจะมีลักษณะเป็นโมเดลมีชื่อเป็น model.mdl

2) กำหนดการใช้งานเป็นแบบ xPC Target

ในขั้นตอนนี้จะทำงานร่วมกับ Real-Time Workshop ในการกำหนดค่าพารามิเตอร์ของการสร้างโค๊ดเป็นแบบ xPC Target เพื่อกำหนดให้ไฟล์กระทำการสามารถนำไปใช้ในการเชื่อมต่อแบบ xPC Target ได้

3) จำลองการทำงานจาก Simulink

ทำการจำลองการทำงานของระบบก่อนว่าถูกต้องหรือไม่ โดยใช้ Simulink ในการทดสอบ

ถ้าไม่ถูกต้องจะต้องกลับไปออกแบบระบบใหม่หรือถ้าถูกต้องแล้วก็สามารถสร้างไฟล์กระทำการได้เลย

4) สร้างไฟล์กระทำการ

ในขั้นตอนนี้จะใช้ Real-Time Workshop ในการสร้างโค๊ดภาษาซีและไฟล์กระทำการ ซึ่งหลังจากการสร้างไฟล์กระทำการ จะได้ไฟล์ model.exe

5) สร้างแผ่นบูทระบบ

ในขั้นตอนนี้จะสร้างแผ่นบูทระบบขึ้นมาเพื่อที่จะสั่งให้ Target PC เรียกระบบปฏิบัติการ xPC Target Real-Time kernel ขึ้นมาติดต่อกับ Host PC

6) ทดสอบการทำงานที่ Target PC

เมื่อที่ Target PC เรียกระบบปฏิบัติการ xPC Target kernel ขึ้นมาแล้วเราจะทราบได้ทันทีว่า ระหว่าง Host PC และ Target PC สามารถเชื่อมต่อกันได้หรือไม่ มีข้อผิดพลาดหรือไม่ ถ้ามีข้อผิดพลาดจะต้องกลับไปออกแบบระบบใหม่หรือแก้ไขค่าพารามิเตอร์บางค่าที่ผิด ถ้าถูกต้องก็สามารถที่จะส่งข้อมูลจาก Host PC ให้กับ Target PC ได้

7) ส่งและรับข้อมูลระหว่าง Host PC กับ Target PC

ในขั้นตอนนี้ที่ Host PC จะดาวน์โหลดไฟล์กระทำการให้กับ Target PC โดยไฟล์กระทำการนี้จะทำงานอยู่ที่ Target PC ตลอดเวลาเพื่อรอคำสั่งจาก Host PC ว่าต้องการทำอะไร เช่น การส่งและรับค่าข้อมูลระหว่างกันหรือใช้ Target PC เป็นมอนิเตอร์ดูสัญญาณที่รับมาจาก Host PC หรือสั่งให้ Target PC ทำการติดต่อกับอุปกรณ์อินพุตและเอาท์พุตภายนอก เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 การใช้งาน xPC Target Explorer

การใช้งาน xPC Target นั้นจะต้องทำการเรียกหน้าต่าง xPC Target Explorer ขึ้นมาเพื่อทำการกำหนดค่าพารามิเตอร์ต่างๆ ในการเชื่อมต่อเป็นหน้าต่างหลักที่มีทำหน้าที่ควบคุมการเชื่อมต่อระหว่างต้นทางและปลายทางทั้งหมด แต่เราอาจจะไม่ใช้หน้าต่าง xPC Target Explorer ก็ได้ เนื่องจากเราสามารถควบคุมผ่านหน้าต่าง Simulink หรือโปรแกรม MATLAB ได้เหมือนกัน

1) ความสามารถของ xPC Target Explorer

1.1) รับแต่งค่าพารามิเตอร์ที่ใช้ในการเชื่อมต่อของเครื่องคอมพิวเตอร์ต้นทาง

1.2) ปรับแต่งข้อมูลที่ใช้ในการเชื่อมต่อของเครื่องคอมพิวเตอร์ปลายทาง

1.3) สร้างแผ่นบูทระบบ (Target Boot Disk)

1.4) เชื่อมต่อเครื่องคอมพิวเตอร์ต้นทางและเครื่องคอมพิวเตอร์ปลายทางเข้าด้วยกัน

1.5) ดาว์นโหลดไฟล์กระทำการจากคอมพิวเตอร์ต้นทางสู่คอมพิวเตอร์ปลายทาง

1.6) กำหนดการเปิดหรือปิดการเชื่อมต่อของต้นทางและปลายทาง

1.7) สร้างสโคป (Scope) เพื่อทำการดูผลสัญญาณที่ปลายทางและต้นทาง

1.8) เพิ่มสัญญาณหรือลบสัญญาณในที่แสดงผลในสโคป

1.9) กำหนดการทำงานของสโคปให้ทำงานหรือหยุดการทำงาน

1.10) ทำการปรับแต่งค่าพารามิเตอร์หรือข้อมูลในขณะที่กำลังรันการทำงานอยู่ที่ Target PC

2) การเรียกใช้งาน xPC Target Explorer

ที่หน้าต่าง MATLAB Command Window พิมพ์คำสั่ง xpcexplr ที่ จะปรากฏหน้าต่าง xPC Target Explorer ดังรูปที่ 2.12

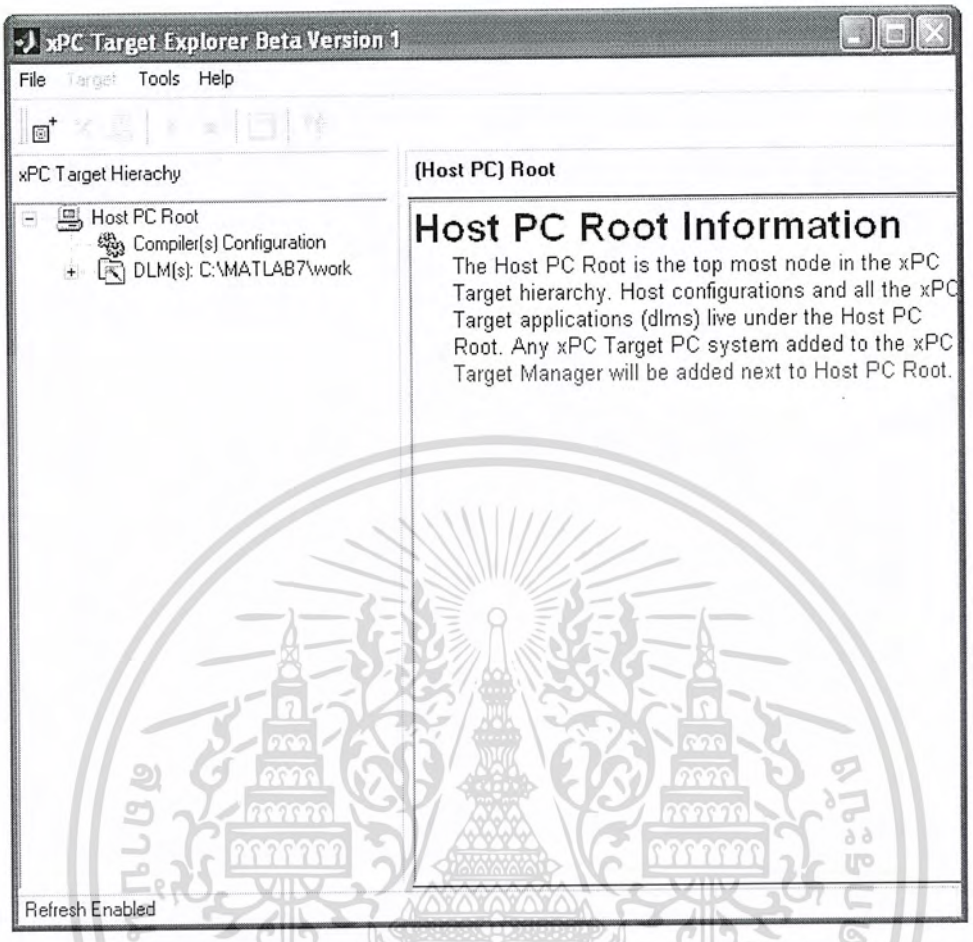
ก่อนการใช้งานต้องทำการเลือกตัวแปลภาษาซีก่อน โดยคลิกที่ Compiler (s) Configuration ทางด้านขวาให้ทำการเลือกตัวแปลภาษาที่ต้องการ ดังรูปที่ 2.13

คลิกขวาที่ Host PC Root เลือก Add Target เพื่อจะทำการเพิ่ม Target โดยการเพิ่ม Target นี้เพิ่มได้ตามความต้องการแสดงดังรูปที่ 2.14

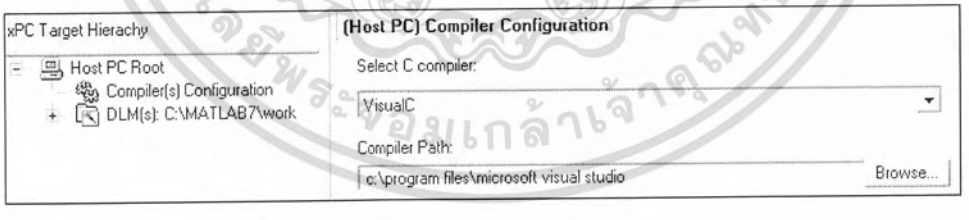
3) การใช้งาน xPC Target Explorer ในการเชื่อมต่อแบบอนุกรม (Serial)

คลิกที่หัวข้อ Communication ของ Target PC1 ทางด้านขวาจะสังเกตเห็นว่าสามารถทำการกำหนดลักษณะการเชื่อมต่อได้ ให้เลือกเป็น RS 232 กำหนดพอร์ตที่เชื่อมต่อ ค่าที่ COM1 หรือ COM2 และกำหนดอัตราการรับส่งข้อมูล ในโปรแกรมจะกำหนดไว้ที่ 115200 บอด เป็นค่าเริ่มต้น

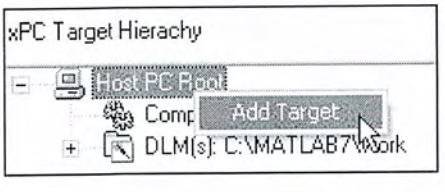
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 หน้าต่างการเรียกใช้งาน xPC Target Explorer



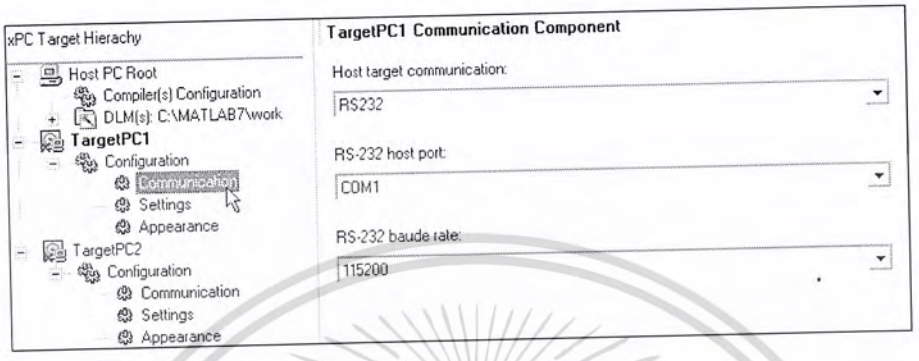
รูปที่ 2.13 หน้าต่าง Compiler Configuration



รูปที่ 2.14 หน้าต่าง xPC Target Hierachy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

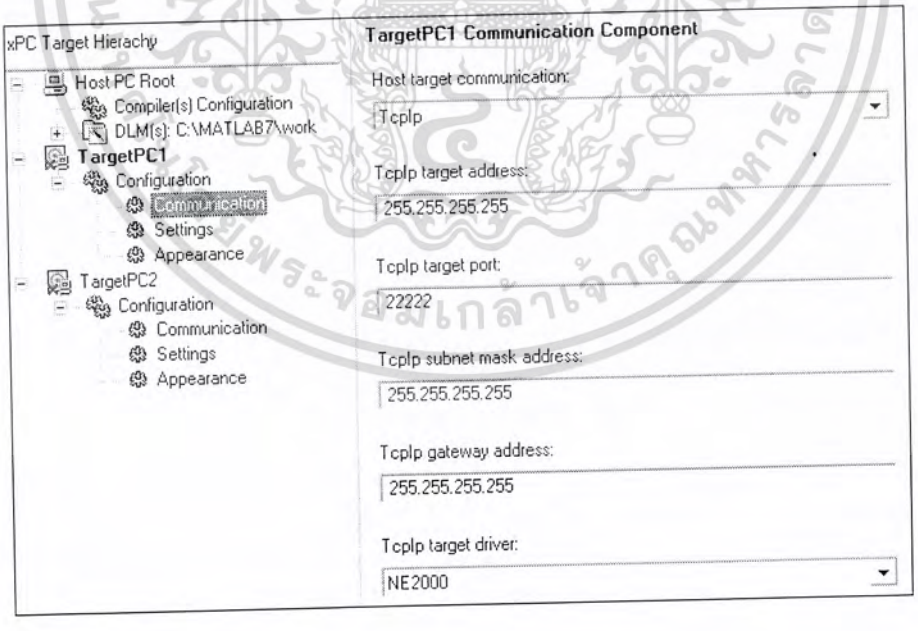
ในกรณีที่เรามี Target PC มากกว่า 1 ตัว สามารถปรับแต่งค่าต่างๆ แต่ละตัวได้เลยโดยเข้าไปที่หัวข้อ Communication ของ Target PC นั้นๆ เช่น ที่ Target PC2 จะทำการเชื่อมต่ออยู่กับ COM 2 และอาจกำหนดอัตราการรับส่งข้อมูลเป็น 9600 บอด เป็นต้น แสดงดังรูปที่ 2.15



รูปที่ 2.15 หน้าต่าง TargetPC1 Communication Component

4) การใช้งาน xPC Target Explorer ในการเชื่อมต่อแบบระบบเครือข่าย (Network)

คลิกที่หัวข้อ Communication ให้เลือกเป็น TcpIp จะปรากฏค่าพารามิเตอร์ต่างๆ ที่เราต้องทราบมีรายละเอียดดังนี้ ดังรูปที่ 2.16



รูปที่ 2.16 การใช้งาน xPC Target Explorer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Target PC IP address เป็น IP Address ของ Target PC เช่น กำหนดเป็น 161.246.63.95

LAN Subnet mask Address เป็น subnet mask address ของ target PC เช่นกำหนดเป็น 255.255.255.192

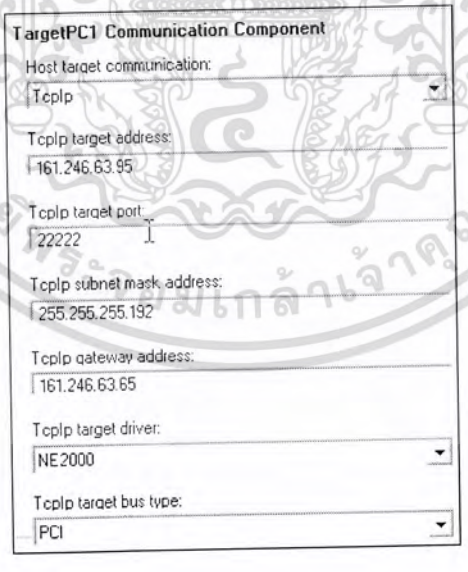
TcpIp Target Port เป็นพอร์ตที่ใช้ในการติดต่อกับ Target PC โดยโปรแกรมจะตั้งเป็น 22222 เป็นค่าเริ่มต้น แต่เราสามารถกำหนดใหม่ได้โดยให้อยู่ระหว่างพอร์ต 20000 ถึง 65536

TcpIp Gateway Address เป็น Gateway Address ของ Target PC ในกรณีที่เราเชื่อมต่อกับ Target PC ในวง LAN เดียวกันจะมีค่าเหมือนกัน ไม่ต้องกำหนดก็ได้ แต่ในกรณีที่อยู่คนละวง LAN กันจะต้องทำการกำหนดค่า Gateway Address ด้วย เช่น กำหนดค่าเป็น 161.246.63.65

TcpIp Target Driver เป็นการกำหนดตัวไดร์ฟเวอร์ของการ์ด LAN โดยจะกำหนดค่า NE2000 เป็นค่าเริ่มต้น

TcpIp Target Bus Type เป็นการกำหนดรูปแบบบัสในใช้กับการ์ด LAN ที่ Target PC โปรแกรมจะตั้งเป็น PCI

ในกรณีที่เรามี Target PC หลายตัวเราจะต้องทำการปรับแต่งค่าพารามิเตอร์ให้ตรงกับ Target PC แต่ละตัว ซึ่งขั้นตอนจะเหมือนกับที่ได้อธิบายมาแล้ว เช่น ถ้า Target PC2 อยู่คนละวง LAN กัน แต่เราต้องการใช้งาน Target PC ทั้ง 2 ตัว เราต้องทำการกำหนดค่า TcpIp Gateway Address ใหม่ เป็นต้น การกำหนดค่าต่างๆ แสดงดังรูปที่ 2.17



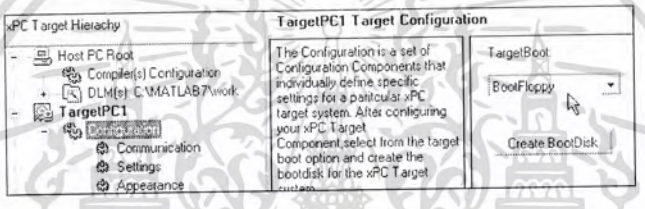
รูปที่ 2.17 หน้าต่าง TargetPC1 Communication Component

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) สร้างแผ่นบูทระบบ (Target Boot Disk)

ก่อนการสร้างแผ่นบูทระบบผู้ใช้งานต้องเตรียมแผ่น Floppy Disk ขนาด 3.5 นิ้ว ซึ่งจะต้องไม่มีข้อมูลใดๆ อยู่ก่อน เมื่อสร้างแผ่นบูทระบบเสร็จในแผ่นบูทระบบจะประกอบด้วยระบบปฏิบัติการที่ชื่อว่า xPC Target kernel ซึ่งเป็นระบบปฏิบัติการของ xPC Target โดยเฉพาะและจะเก็บค่าพารามิเตอร์ต่างๆ ที่สำคัญในการเชื่อมต่อกับ Target PC เช่น กำหนดข้อมูลในการเชื่อมต่อคือ การสื่อสารแบบอนุกรมหรือระบบเครือข่าย

คลิกที่หัวข้อ Configuration ใน Target PC ที่ต้องการ ในที่นี้เลือกการเชื่อมต่อแบบ RS 232 และในหน้าต่างทางด้านขวาจะแสดงการสร้างแผ่นบูทระบบ ให้ทำการเลือกเป็น BootFloppy ใส่แผ่น Floppy Disk ใน Floppy Drive คลิกที่ปุ่ม Create BootDisk เพื่อเริ่มขั้นตอนการสร้างแผ่นบูทระบบ แสดงดังรูปที่ 2.18



รูปที่ 2.18 หน้าต่าง TargetPC1 Target Configuration

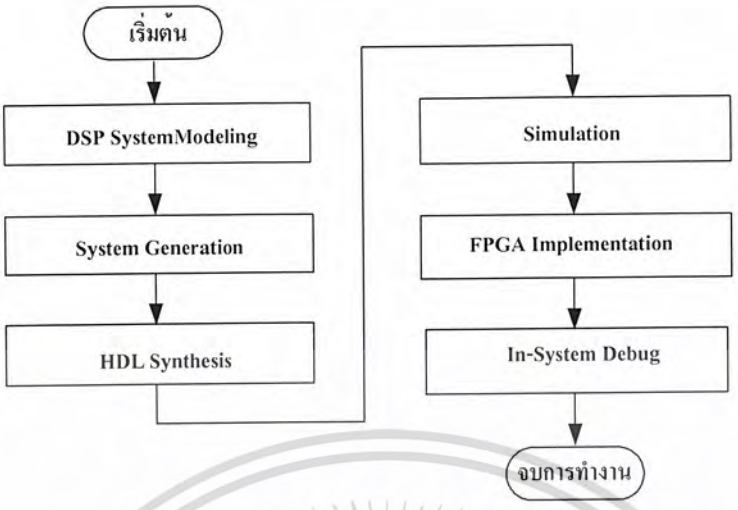
2.4 การพัฒนาการประมวลผลสัญญาณดิจิตอลด้วยโปรแกรม MATLAB และโปรแกรม System Generator

ในปัจจุบันเทคโนโลยีการออกแบบฮาร์ดแวร์ประมวลผลทางด้านดิจิตอลพัฒนาอย่างรวดเร็วและฮาร์ดแวร์ที่ได้รับความนิยมมากที่สุดในปัจจุบันคือ FPGA (Field Programmable Gate-Array) เพราะสามารถประยุกต์ใช้งานในด้านดิจิตอลได้หลากหลาย ซอฟต์แวร์ที่ใช้ในการพัฒนา FPGA ก็มีมากในโปรแกรม MATLAB เองก็สามารถพัฒนาการประยุกต์ใช้งาน FPGA ได้เช่นกัน

2.4.1 ขั้นตอนการทำงานพัฒนาการประมวลผลสัญญาณดิจิตอลด้วยโปรแกรม MATLAB และโปรแกรม System Generator

สามารถแสดงขั้นตอนการทำงานเป็นผังงานแสดงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 ขั้นตอนการทำงานพัฒนาการประมวลผลสัญญาณดิจิตอลด้วยโปรแกรม MATLAB และโปรแกรม System Generator

1) การออกแบบระบบ DSP (DSP System Modeling)

เมื่อทำการติดตั้งโปรแกรม System Generator แล้วจะมีกล่องเครื่องมือของ Xilinx เพิ่มเติมเข้าไปใน Simulink การออกแบบระบบ DSP เพียงแต่นำฟังก์ชั่นบล็อกของ Xilinx มาต่อกันเป็นระบบ DSP

2) การแปลงระบบให้เป็นไฟล์สำหรับ FPGA (System Generator)

ในขั้นตอนนี้ใช้เครื่องมือที่ชื่อว่า System Generator แปลงระบบที่ออกแบบจาก Simulink ให้เป็นไฟล์สำหรับโปรแกรม FPGA เช่น HDL Netlist, NGC Netlist หรือ Bitstream

3) การสังเคราะห์วงจร (HDL Synthesis)

เป็นขั้นตอนการสังเคราะห์วงจร โดยการนำไฟล์ HDL มาสังเคราะห์ให้เหมาะสมกับคุณสมบัติของ FPGA เพื่อจะนำไปโปรแกรมลงบน FPGA ต่อไป ซึ่งสามารถเลือกโปรแกรมที่ทำหน้าที่สังเคราะห์ได้หลายตัว เช่น Xilinx Synthesis Tool, Leonardo Spectrum หรือ Synplify Pro

4) การจำลองการทำงาน (Simulation)

เป็นการนำไฟล์ทดสอบการทำงาน (Testbench file) ที่ได้ในขั้นตอนการแปลงระบบเป็นไฟล์ FPGA มาใช้ในการจำลองการทำงานเพื่อตรวจสอบความถูกต้อง ซึ่งสามารถใช้โปรแกรมได้หลายตัว เช่น ModelSim XE หรือ ModelSim SE เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) การโปรแกรมลงบน FPGA (FPGA Implementation)

เป็นการนำไฟล์ HDL มาแปลงเป็นไฟล์ที่ใช้ในการโปรแกรมลงบน FPGA คือ ไฟล์ Bitstream นั้นเอง

6) การทดสอบการทำงานของ FPGA (In-System Debug)

ขั้นตอนนี้เป็นขั้นตอนพิเศษเพราะจะต้องมีซอฟต์แวร์ในการตรวจสอบการทำงานของ FPGA ว่าถูกต้องหรือไม่ โปรแกรมที่ใช้ได้เช่น ChipScope Pro

2.4.2 ความต้องการทางด้านซอฟต์แวร์

ก่อนการใช้งานต้องมีการติดตั้งซอฟต์แวร์ต่างๆ โดยจะมีความต้องการขั้นต่ำดังนี้

1) โปรแกรม Xilinx ISE Foundation

2) โปรแกรม System Generator

3) โปรแกรม ModelSim

4) โปรแกรม ChipScope Pro

2.5 โครงสร้างของ GPIB

ในระบบพื้นฐานของ GPIB จะประกอบด้วยอุปกรณ์ คือ ผู้ส่ง (Talker) ผู้รับ (Listener) และผู้ควบคุม (Controller)

1) TalKer ทำหน้าที่ส่งข้อมูล โดยในระบบสามารถมี Talker ได้หลายตัว แต่จะมีเพียงตัวเดียวเท่านั้นที่กำลังทำงานอยู่

2) Listener ทำหน้าที่เป็นตัวรับข้อมูล โดยในระบบเดียวกันสามารถมี Listener ได้หลายตัวเช่นเดียวกัน แต่ Listener สามารถทำงานได้ครั้งละหลายๆ ตัวได้

3) Controller ทำหน้าที่ควบคุมอุปกรณ์ต่างๆ ในระบบ โดยจะกำหนดให้ Talker ทำการส่งข้อมูลหรือกำหนดให้ Listener ทำการรับข้อมูล

อุปกรณ์ที่มี GPIB นั้นสามารถแบ่งตามหน้าที่ได้ดังนี้

1) ทำหน้าที่เป็น Talker เท่านั้น เช่น เครื่องมือวัด เป็นต้น

2) ทำหน้าที่เป็น Listener เท่านั้น เช่น เครื่องพิมพ์ (Printer), เครื่องบันทึก (Recorder) เป็นต้น

3) ทำหน้าที่เป็นทั้ง Talker และ Listener เช่น คอมพิวเตอร์, เครื่องมือวัดที่สามารถควบคุมได้จากภายนอก เป็นต้น

4) ทำหน้าที่เป็น Talker Listener และ Controller ในตัวเดียวกัน เช่น คอมพิวเตอร์ที่ทำหน้าที่ ควบคุมระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 ขีดจำกัดของ IEEE-488

1) จำนวนอุปกรณ์ในระบบ (Talker, Listener, Controller) ที่ต่อกับสายนำสัญญาณ 1 เส้น จะต้องไม่เกิน 15 เครื่อง

2) สายเคเบิ้ลที่จะต่อระหว่างอุปกรณ์ จะต้องยาวไม่เกิน 4 เมตร และความยาวรวมของสายเคเบิ้ลในระบบจะต้องไม่เกิน 20 เมตร

3) ความเร็วในการส่งข้อมูลจะต้องไม่เกิน 1Mb/Sec (1 ล้านไบต์ต่อวินาที)

4) ต้องมีการจ่ายไฟให้กับอุปกรณ์มากกว่าครึ่งหนึ่งของระบบ

2.5.2 รายละเอียดเกี่ยวกับ IEEE-488

ลักษณะทางกายภาพ IEEE-488 นั้นคือ เป็นสายสัญญาณแบบ 24 เส้นขนานกันและมีขั้วต่ออยู่ทางปลายทั้งสองของสาย เพื่อต่อกับอุปกรณ์หรือต่อกันเพื่อให้สายสัญญาณมีความยาวเพิ่มขึ้น ในจำนวนสายสัญญาณ 24 เส้น มีเพียง 16 เส้นเท่านั้น ที่ทำหน้าที่นำสัญญาณ ส่วนเหลืออีก 8 เส้น ทำหน้าที่กราวนด์ (ground) และ (shield)

โดยจำนวนสายที่ใช้นำสัญญาณ 16 เส้นนั้นยังแบ่งได้เป็น 3 ประเภท ตามรูปที่ 2.25 คือ

1) บัสข้อมูล (Data Bus) จำนวน 8 สาย คือ

DI01 – DI08

2) สายสัญญาณควบคุม (Control Line) จำนวน 5 สาย คือ

IFC (Interface Clear)

ATN (Attention)

SRQ (Servive Enable)

REN (Remote Enable)

EOI (End or Identify)

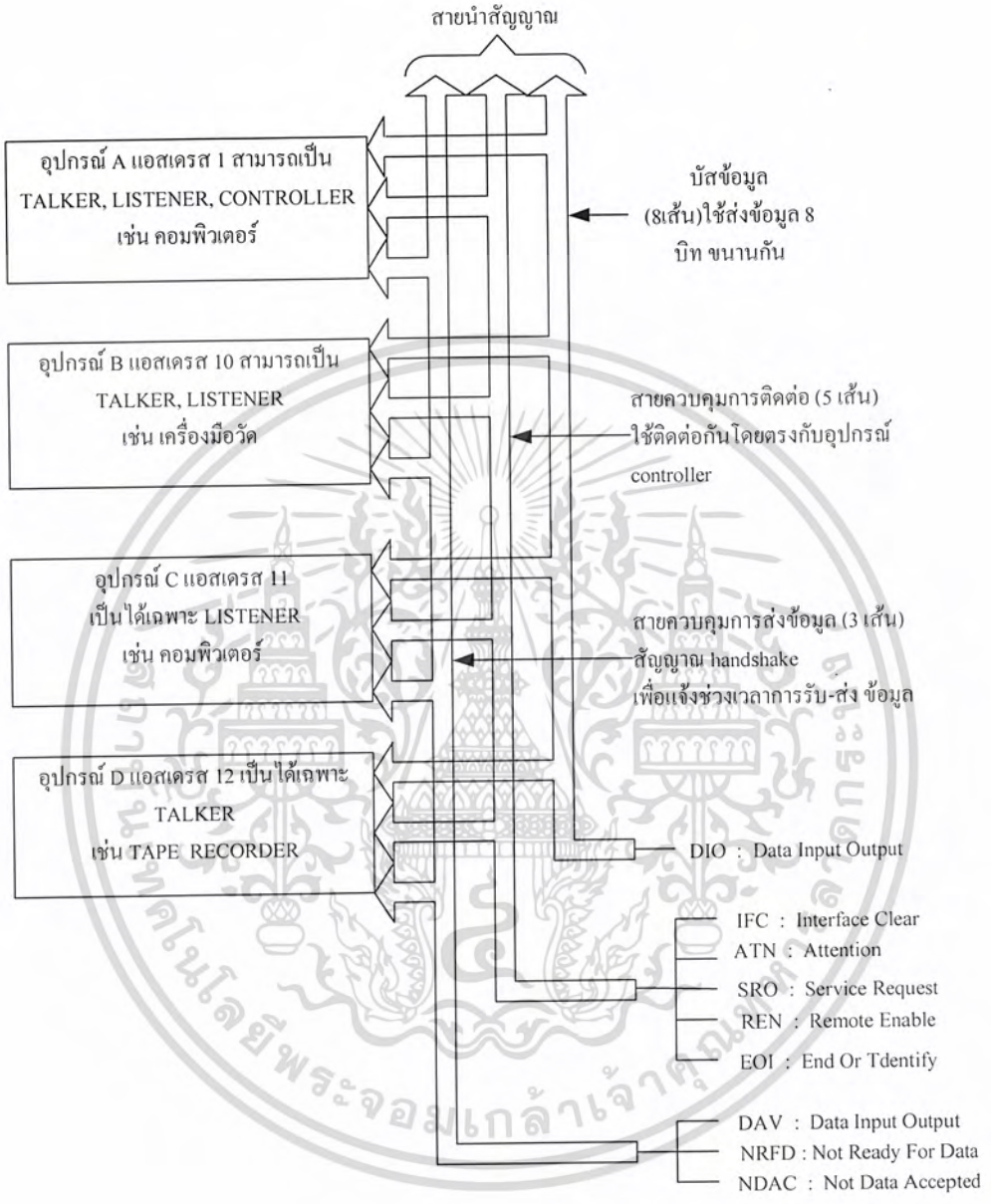
3) สายแฮนด์เชค (Hand Shake) 3 สาย คือ

DAV (Data Valid)

NRFD (Not Ready For Data)

NDAC (Not Data Accepted)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 การแบ่งเส้นสายนำสัญญาณ

2.5.3 ความหมายของสัญญาณต่างๆ ภายใน IEEE-488

ดังที่ได้กล่าวมาแล้วว่าสายสัญญาณต่างๆ ใน GPIB ได้แบ่งออกเป็น 3 กลุ่ม ในหัวข้อนี้ จะอธิบายความหมายของสัญญาณต่างๆ ดังนี้

1) กลุ่มสัญญาณข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DI01-DI08 (Data input/Output) สายสัญญาณทั้งหมด 8 เส้นนี้ จะทำหน้าที่เป็นทางผ่านของข้อมูลในระบบ

2) กลุ่มสัญญาณควบคุมการเชื่อมต่อ (Interface)

2.1) IFC (Interface Clear) เป็นสัญญาณรีเซตหรือเคลียร์ระบบจะกำเนิดได้ โดยตัวควบคุม (Controller) เท่านั้น เมื่ออุปกรณ์ในบัสได้รับสัญญาณเคลียร์นี้จะกลับคืนสู่สภาวะเริ่มต้นใหม่ ซึ่งจะเป็นสภาวะแรกเริ่มก่อนการกำหนดฟังก์ชันเหมือนแรกเปิดสวิตช์

2.2) ATN (Attention) เป็นสัญญาณที่ถูกส่งโดยอุปกรณ์ที่เป็นตัวควบคุมเช่นเดียวกัน ใช้ในการสั่งให้อุปกรณ์ทุกตัวในระบบเตรียมพร้อมเพื่อรอรับคำสั่งต่อไป

2.3) SQR (Service Request) เป็นสัญญาณที่ถูกส่งจากอุปกรณ์ต่างๆ เพื่อเป็นการบอกแก่ระบบว่าขณะนี้อุปกรณ์ดังกล่าวต้องการติดต่อจากตัวควบคุม

2.4) REN (Remote Enable) สัญญาณนี้เป็นสัญญาณที่ถูกส่งมาจากตัวควบคุมเพียงตัวเดียวเท่านั้น เพื่อใช้สั่งให้อุปกรณ์ต่างๆ เปลี่ยนจากโหมดที่ใช้งานปกติ มาเป็นการควบคุมโดยตัวควบ-คุมแทน

2.5) EOI (End or Identify) เป็นสัญญาณที่ถูกส่งได้โดยอุปกรณ์ที่เป็นตัวควบคุม (Controller) หรืออุปกรณ์ที่เป็นตัวส่ง (Talker) ก็ได้ ใช้สำหรับแสดงว่าข่าวสารที่ส่งเป็นชุดนั้นได้เสร็จสิ้นลงแล้ว

3) กลุ่มสัญญาณควบคุมการรับ-ส่งข้อมูล

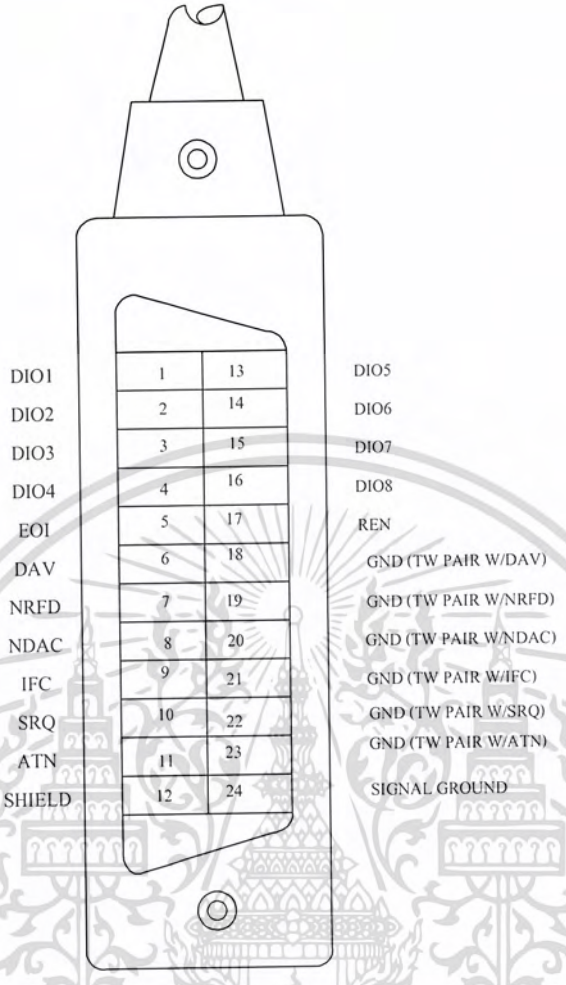
3.1) DAV (Data Valid) เมื่อสัญญาณนี้ถูกดึงเป็นลอจิก "Low" โดยอุปกรณ์ที่เป็นตัวส่ง (Talker) เป็นการแจ้งแก่ระบบบัสว่า ขณะนี้ตัวส่งได้ทำการส่งข้อมูลลงไปที่สายสัญญาณข้อมูลเรียบร้อยแล้ว

3.2) NRFD (Not Ready For Data) เมื่อสัญญาณนี้มีลอจิกเป็น "Low" จะเป็นการแสดงว่าในขณะนี้ระบบบัสยังไม่พร้อมที่จะรับข้อมูล เนื่องจากอุปกรณ์ทั้งหมดยังไม่พร้อมทุกตัว ซึ่งสัญญาณเส้นนี้จะไม่เป็น "Hi" จนกว่าอุปกรณ์ทุกตัวให้ลอจิกที่เป็น "Hi" ครบถ้วนแล้ว สัญญาณนี้มีประโยชน์ในกรณีที่อุปกรณ์ในระบบที่มีความเร็วแตกต่างกัน

3.3) NDAC (Not Data Accepted) สัญญาณเส้นนี้เป็นสัญญาณที่ถูกควบคุมโดยอุปกรณ์ที่เป็นตัวรับ (Listener) โดยสัญญาณนี้จะมีลอจิกเป็น "Low" ในขณะที่อุปกรณ์ที่เป็นตัวรับกำลังจะเก็บข้อมูล จากสายข้อมูล (Data Bus) และจะเป็น "Hi" เมื่ออุปกรณ์นั้นได้ทำการอ่านข้อมูลเสร็จเรียบร้อยแล้ว

โดยสัญญาณลอจิกที่ใช้ใน DATA BUS (D1-D8) ของ IEEE-488 มีลักษณะเป็นคอมพลี-เมนต์ทั้งหมดคือ "1" เท่ากับ "Low" และ "0" เท่ากับ "Hi" ซึ่งตรงข้ามกับในวงจรที่เราคุ้นเคย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 ขั้วต่อของ GPIB และการจัดขาของสัญญาณต่างๆ

2.5.4 การเชื่อมต่ออุปกรณ์ต่างๆ ในระบบ IEEE-488 BUS

สำหรับการเชื่อมต่อระหว่างอุปกรณ์ต่างๆ ในระบบ IEEE-488 Bus นั้น มีอยู่ 2 วิธี คือ

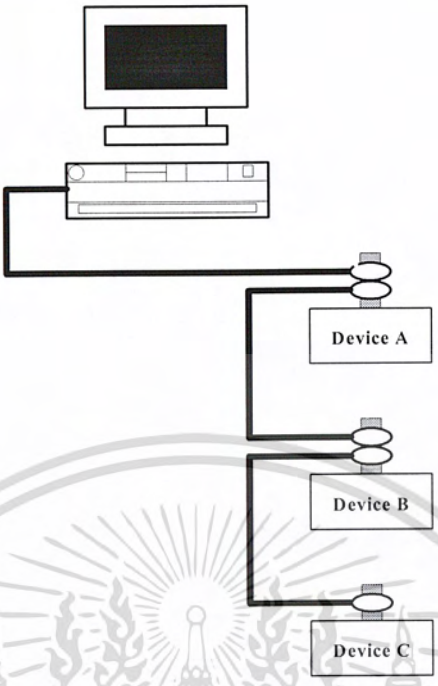
1) การเชื่อมต่อแบบเรียงต่อเนื่อง (Daisy Chain Configurating) ดังรูปที่ 2.22

2) การเชื่อมต่อแบบกระจาย (Star Configurating) ดังรูปที่ 2.23

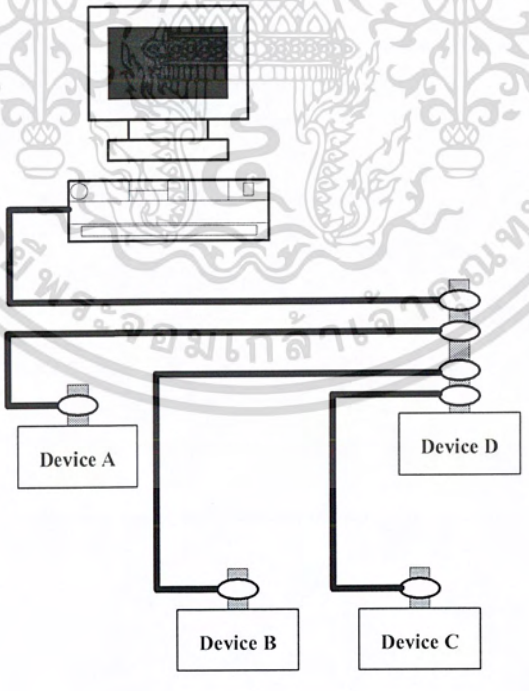
2.5.5 คำสั่งใช้งานของ GPIB

การสั่งการต่างๆ เพื่อกำหนดหน้าที่การทำงานและกำหนดฟังก์ชัน เช่น กำหนดช่วงการวัด โหมดการวัดหรืออื่นๆ แก่เครื่องวัดที่ตกอยู่เหล่านี้นั้น ตัวควบคุมจะเป็นตัวกำหนดการส่งรหัสคำสั่งไปที่อุปกรณ์โดยผ่าน DI1-DI6 รหัสคำสั่งนี้จะถูกส่งไปในช่วงที่สายสัญญาณ ATN เป็น LOW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)



รูปที่ 2.23 การเชื่อมต่อแบบกระจาย (Star Configuration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งสำหรับกำหนดหน้าที่การงานต่างๆ ตามมาตรฐานของ GPIB มีอยู่ด้วย 128 คำสั่ง ดังแสดงในตารางที่ 2.1 ต่อไปนี้ โดยแบ่งเป็น 5 กลุ่มคำสั่ง

รหัสที่ใช้ในระบบ GPIB บัสนั้นใช้ร่วมกัน ทั้งรหัสข้อมูลและรหัสคำสั่ง นั่นคือ รหัสเดียวกันมีความหมายได้ 2 อย่างคือ เมื่อ ATN เป็น LOW จะหมายถึงรหัสคำสั่ง แต่ ATN เป็น HIGH รหัสนี้ จะแทนข้อมูลที่เป็น ASCII แทน ซึ่งในตารางได้แบ่งความหมายออกเป็น 2 คอลัมน์

1) กลุ่มคำสั่งเจาะจงจุดหมาย (Addressed Command Group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ที่เป็นตัวส่งหรือตัวรับที่กำหนดไว้ล่วงหน้าแล้ว คำสั่งนี้ประกอบด้วย

1.1) GTL (Got To Local) สั่งให้อุปกรณ์กลับสู่สภาพการควบคุมปกติด้วยมือ

1.2) SDC (Selected Device Clear) สั่งให้อุปกรณ์เคลียร์ตัวเองสู่สภาพเริ่มต้นใหม่

1.3) PPC (Paralled Poll Configure) เป็นคำสั่งสำหรับทำการจัดสายสัญญาณของการทำการ จัดสรรสายสัญญาณของการทำกระบวนการตรวจสอบสภาพอุปกรณ์โดยวิธีขนาน โดยใช้กับกลุ่มคำสั่งรอง

1.4) GET (Group Excute Trigger) ใช้สั่งเริ่มต้นการทำงานของอุปกรณ์ทีละหลายตัว

1.5) TCT (Take Control) เป็นการกำหนดให้อุปกรณ์ตัวส่งทำหน้าที่เป็นตัวควบคุม

2) กลุ่มคำสั่งครอบคลุม (Universal Command Group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ทุกตัวที่ต่ออยู่ในบัสประกอบด้วย

2.1) LLO (Local Lockout) เป็นการสั่งให้อุปกรณ์ล็อคอยู่ที่สภาวะควบคุมโดยปุ่มปรับที่หน้าปัดตามปกติ

2.2) DCL (Device Clear) สั่งให้อุปกรณ์ทุกตัวกลับไปสู่สภาวะเริ่มต้น

2.3) PPU (Parallel Poll Unconfigure) ใช้ยกเลิกกระบวนการตรวจสอบสภาพแบบขนาดทั้งหมด

2.4) SPE (Serial Poll Enable) เปลี่ยนโหมดของการตรวจสอบสภาพเป็นแบบอนุกรม ในโหมดนี้ จะเป็นการส่งสถานะของเครื่องแทนการส่งข้อมูล

2.5) SPD (Serial Poll Disable) ยกเลิกโหมดการตรวจสอบแบบอนุกรม

3) กลุ่มคำสั่งกำหนดอุปกรณ์ตัวรับ (Listener Address Group) เป็นคำสั่งสำหรับกำหนดให้อุปกรณ์เป็นตัวรับ ตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNIT (Untalker) สำหรับยกเลิกเช่นกัน

4) กลุ่มคำสั่งกำหนดอุปกรณ์ตัวส่ง (Talker Address Group) สำหรับกำหนดให้อุปกรณ์เป็นตัวส่ง ตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNL (Unlisten) สำหรับยกเลิก

คำสั่งกลุ่มที่ 1 ถึง 4 นั้น จัดเป็นกลุ่มคำสั่งหลักที่มีความหมายตายตัวยังมีคำสั่งอีกกลุ่มที่ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดภายหลังนั่นคือ กลุ่มคำสั่งรอง

5) กลุ่มคำสั่งรอง (Secondary Command Group) เป็นคำสั่งที่กำหนดรายละเอียดย่อยของอุปกรณ์แต่ละตัวที่ต่ออยู่ในระบบ ให้มีการทำงานอย่างไร ตามจุดประสงค์ใช้งานของเครื่องมือนั้น เช่น เดียวกับการปรับปุ่มต่างๆ ด้วยมือนั่นเอง คำสั่งรองนี้จะตามหลังคำสั่งหลักคือ จะใช้หลังจากอุปกรณ์ต่างๆ ถูกกำหนดวางตัวในระบบเรียบร้อยแล้ว

คำสั่งต่างๆ ที่กล่าวไป ซึ่งใช้ในการกำหนดสภาวะการทำงานของอุปกรณ์ แต่ละสภาวะที่กำหนดไปนั้นเป็นอย่างไรและมีจุดประสงค์เพื่ออะไร ดังต่อไปนี้

2.5.6 Device clear/Interface clear

Device clear ใช้ในการทำให้อุปกรณ์ที่ต่ออยู่ในบัสกลับไปสู่สภาวะเริ่มต้น ยังไม่มีการกำหนดฟังก์ชันใดๆ สภาวะเริ่มต้นนี้จะแตกต่างกันไป แล้วแต่ว่าอุปกรณ์นั้นออกแบบไว้อย่างไร Device clear มีอยู่ 2 ลักษณะคือ เคลียร์หมดทุกตัวที่ต่ออยู่ (DCL) กับเคลียร์เจาะจงอุปกรณ์ตัวใดตัวหนึ่ง (SDC)

แต่ว่าในการเคลียร์อุปกรณ์ให้อยู่ในสภาวะเริ่มต้นนั้นไม่ได้หมายความว่า Interface Function ของ GPIB จะถูกเคลียร์อุปกรณ์ให้ไปอยู่ในสภาวะเริ่มต้นด้วยแต่อย่างใด Interface Function คือ สภาพการ Interface ที่ได้กำหนดไว้ในระบบประกอบด้วยฟังก์ชันต่างๆ ดังแสดงในตารางที่ 2.6

ตารางที่ 2.6 สภาพการ Interface ที่ได้กำหนดไว้ในระบบประกอบด้วยฟังก์ชั่นต่างๆ ดังนี้

ฟังก์ชัน	สัญลักษณ์	การกลับสู่จุดเริ่มต้นโดย IFC
Source handshake	SH	ได้
Acceptor handshake	AH	ได้
Talker or enlarge talker	T or TE	ได้
Listener or enlarge listener	L or LT	ได้
Service request	SR	ไม่ได้
Remote/local	RL	ไม่ได้
Parallel poll	PP	ไม่ได้
Device clear	DC	ได้
Device trigger	DT	ได้
Controller	C	ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7 การขอบริการและการตรวจสอบ (Service Request and Polling)

เมื่อตัวควบคุมได้รับ SRQ เป็น LOW จะให้อุปกรณ์ส่งข้อมูลแสดงสถานะการทำงาน ซึ่งมีอยู่ 2 วิธี คือ

1) การตรวจสอบแบบอนุกรม ซึ่งมีขั้นตอนดังนี้

1.1) ATN ถูกดึงเป็น LOW หลังจากได้รับ LOW จากสายสัญญาณ SRQ

1.2) คำสั่ง UNL ถูกส่งไปยังอุปกรณ์

1.3) ตัวควบคุมจะแจ้งรหัสตัวรับของตนและกำหนดรหัสตัวส่งอุปกรณ์ที่จะตรวจสอบไปที่บัส

1.4) ตามด้วยคำสั่ง SPE และสาย ATN กลายเป็น HI ซึ่งอุปกรณ์ที่ถูกเรียกจะส่งข้อมูลแสดงสถานะออกมา 1 ไบต์ โดยบิตที่ 7 จะเป็นตัวชี้ว่าอุปกรณ์เส้นเป็นตัวขอบริการ ถ้าใช่จะเป็น LOW ส่วนบิตอื่นๆ ก็ใช้บอกข้อมูลอื่นๆ ซึ่งมิได้กำหนดเฉพาะ

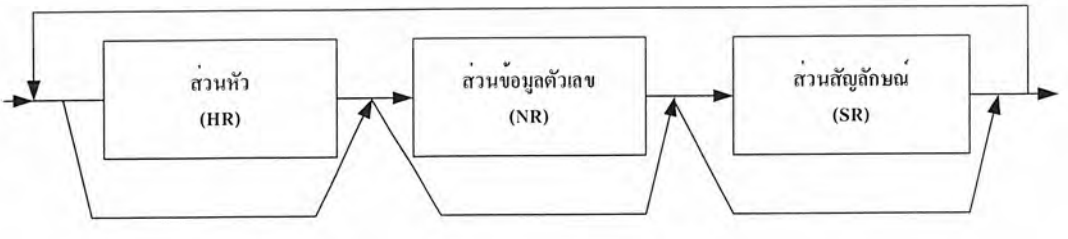
1.5) สาย ATN ถูกดึงเป็น LOW อีกที เพื่อส่งคำสั่งยกเลิกการตรวจสอบคือ SPD

1.6) จากนั้นคำสั่ง UNT ก็ถูกส่งไปยังอุปกรณ์เพื่อยกเลิกการเป็นตัวส่ง ซึ่งถ้าหาก SQR ยังคงเป็น LOW อยู่ ก็จะมีการตรวจสอบไปยังอุปกรณ์ตัวอื่นๆ ต่อไปตามขั้นตอนเดิม

2) การตรวจสอบแบบขนาน สามารถทำได้เร็วกว่าแบบอนุกรมทั้งนี้เพราะสามารถอ่านข้อมูล เพียงไบต์เดียวก็สามารถรู้ได้ทันทีว่าอุปกรณ์ตัวใดเป็นผู้ขอบริการ

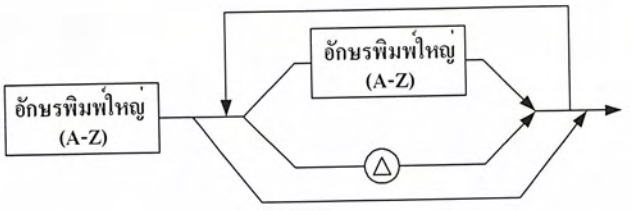
2.5.8 รูปแบบของข้อมูล

โดยทั่วไปข้อมูลจากอุปกรณ์ แบ่งได้ออกเป็น 3 ส่วนดังแสดงในรูปที่ 2.24 อันได้แก่ส่วนหัว (HR) ซึ่งจะอยู่ส่วนหน้าสุด เป็นตัวบอกชนิดข้อมูล ส่วนประกอบ HR แสดงในรูปที่ 2.25 จะเห็นว่าประกอบด้วยตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ช่องว่างที่เป็นเว้นวรรค () ปกติจะมีอักษรประมาณ 1-3 ตัว



รูปที่ 2.24 รูปแบบของข้อมูล

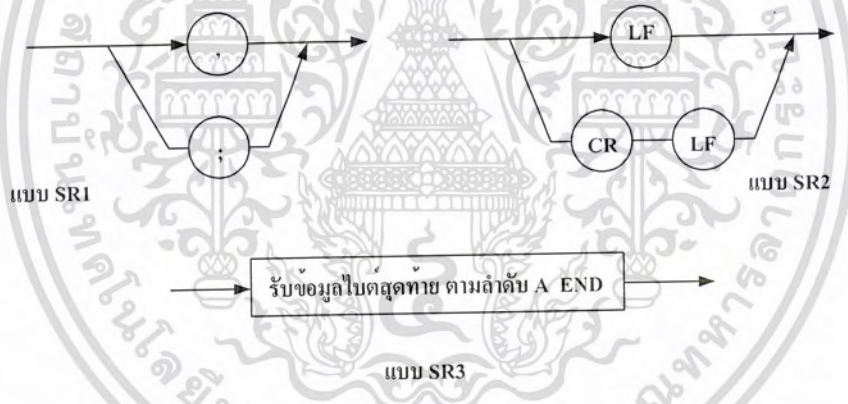
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 รูปแบบของ HR

ส่วนที่ 2 คือ เนื้อหาข้อมูล (NR) ซึ่งใช้แสดงค่าตัวเลข มีอยู่ 3 แบบ คือ NR1, NR2 และ NR3 ดังแสดงในรูปที่ 2.26 ส่วนท้ายของ NR ยังอาจมีตัวอักษรแสดงหน่วยตามมา

ส่วนที่ 3 คือ สัญญาณแบ่งข้อมูลแต่ละชุด (SR) ดังแสดงในรูปที่ 2.26 โดย SR1 ใช้แสดงการต่อเนื่องของข้อมูล (ข้อมูลยังมีต่อ) SR2 และ SR3 แสดงการสิ้นสุดของข้อมูลแต่ SR3 เป็นการบอกการเสร็จสิ้นข้อมูลทั้งหมดจากการวัด



รูปที่ 2.26 สัญญาณแบ่งข้อมูลแต่ละชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB ประกอบด้วยใบงานการทดลองทั้งหมด 10 ใบงาน ขั้นตอนการออกแบบนั้นจะต้องกำหนดวัตถุประสงค์ของแต่ละใบงาน ซึ่งก็คือจุดประสงค์เชิงพฤติกรรม ขั้นตอนต่อไปจะเป็นการกำหนดเนื้อหาและสร้างใบงานการทดลอง โดยการวัดและประเมินผลนั้นจะออกแบบเป็นคำถามท้ายการทดลอง ซึ่งขั้นตอนการออกแบบ การสร้าง และการทำงานมีดังนี้

3.2 เครื่องมือที่ใช้ในการออกแบบและสร้าง

3.2.1 เครื่องมือทางด้านซอฟต์แวร์

1) โปรแกรม MATLAB เวอร์ชั่น 7.0
2) โปรแกรม Microsoft Visual C/C++ เวอร์ชั่น 6.0
3) โปรแกรม Xilinx ISE Foundation 6.3i Evaluation
4) โปรแกรม System Generator 6.3 Evaluation

3.2.2 เครื่องมือทางด้านฮาร์ดแวร์

1) การ์ด GPIB ของบริษัท National Instruments รุ่น GPIB+
2) เครื่องกำเนิดสัญญาณ รุ่น Sony Tektronix AFG310 (Function Generator)
3) เครื่องวัดสัญญาณ รุ่น Tektronix TDS 340A (Oscilloscope)
4) สายสื่อสาร GPIB (GPIB Communication Cable) ของบริษัท National Instruments
5) สายสื่อสารอนุกรม RS 232 (Serial Communication Cable)
6) เครื่องคอมพิวเตอร์ปลายทาง (Target PC)

3.3 การออกแบบ

ขั้นตอนการออกแบบประกอบด้วย การกำหนดวัตถุประสงค์ การวิเคราะห์เนื้อหา ขั้นตอนการสร้างใบงานการทดลอง และการประเมินผลใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 การกำหนดวัตถุประสงค์

การกำหนดวัตถุประสงค์ในการสร้างใบงานการทดลองนั้นจะเป็นการกำหนดจุดประสงค์การเชิงพฤติกรรมของผู้เรียนว่าในแต่ละใบงานการทดลองจะต้องการให้ผู้เรียนได้บรรลุพฤติกรรมทางด้านใด ซึ่งแบ่งเป็นใบงานการทดลองทั้งหมด 10 ใบงานมีรายละเอียดดังนี้

1) ใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1.1) แสดงการใช้งานโปรแกรม MATLAB ในการคำนวณเบื้องต้นได้อย่างถูกต้อง

1.2) แสดงการเรียกใช้งาน Help และ Demo ได้อย่างถูกต้อง

1.3) แสดงการตรวจสอบค่าข้อมูลโดยใช้คำสั่ง who และ whos ได้อย่างถูกต้อง

1.4) เขียนโปรแกรมในลักษณะ M-file ได้อย่างถูกต้อง

2) ใบงานที่ 2 เรื่อง การใช้งาน Simulink เบื้องต้น

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

2.1) อธิบายข้อดีของการจำลองระบบด้วย Simulink ได้อย่างถูกต้อง

2.2) แสดงขั้นตอนการจำลองระบบโดยใช้ Simulink ได้อย่างถูกต้อง

2.3) ประยุกต์การใช้งาน Simulink ในการพัฒนาระบบหรือทฤษฎีในลักษณะแบบจำลองได้อย่างถูกต้อง

3) ใบงานที่ 3 เรื่อง การใช้งาน เรื่อง Real-Time Workshop

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

3.1) อธิบายขั้นตอนการทำงานของ Real-Time Workshop ได้อย่างถูกต้อง

3.2) อธิบายลักษณะการประยุกต์ใช้งานของ Real-Time Workshop ในด้านต่างๆ ได้อย่างถูกต้อง

3.3) บอกความแตกต่างระหว่างหน้าที่การทำงานของ Model Advisor และ Model Explorer ได้อย่างถูกต้อง

3.4) อธิบายการทำงานของไฟล์กระทำการได้อย่างถูกต้อง

4) ใบงานที่ 4 เรื่อง การใช้งาน xPC Target

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1) อธิบายขั้นตอนการทำงานของ xPC Target ได้อย่างถูกต้อง

4.2) บอกความแตกต่างระหว่างการเชื่อมต่อแบบอนุกรมและระบบเครือข่ายได้อย่างถูกต้อง

4.3) แสดงขั้นตอนการเชื่อมต่อระหว่าง Host PC กับ Target PC ด้วยการสื่อสารอนุกรมได้อย่างถูกต้อง

4.4) แสดงขั้นตอนการเชื่อมต่อระหว่าง Host PC กับ Target PC ด้วยการสื่อสารแบบระบบเครือข่ายได้อย่างถูกต้อง

4.5) แสดงขั้นตอนการดาวน์โหลดไฟล์กระทำการจาก Host PC ไปที่ Target PC ได้อย่างถูกต้อง

4.6) อธิบายการทำงานของไฟล์กระทำการที่ Target PC ได้อย่างถูกต้อง

5) ใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

5.1) อธิบายขั้นตอนการทำงานของ xPC Target ในการติดต่อกับพอร์ตอนุกรมได้อย่างถูกต้อง

5.2) แสดงการกำหนดค่าพารามิเตอร์ในการเชื่อมต่อกับพอร์ตอนุกรมได้อย่างถูกต้อง

5.3) อธิบายผลที่เกิดขึ้นเมื่อกำหนด TLC แบบ xPC Target ได้อย่างถูกต้อง

5.4) อธิบายผลที่เกิดขึ้นเมื่อมีการรับส่งข้อมูลกับพอร์ต RS-232 ได้อย่างถูกต้อง

6) ใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องกำเนิดสัญญาณ

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

6.1) อธิบายขั้นตอนการใช้งานโปรแกรม MATLAB เพื่อควบคุมเครื่องกำเนิดสัญญาณ AFG310 ได้อย่างถูกต้อง

6.2) แสดงขั้นตอนการใช้คำสั่งในการติดต่อกับเครื่องกำเนิดสัญญาณได้อย่างถูกต้อง

6.3) อธิบายผลที่เกิดขึ้นหลังจากการใช้คำสั่งต่างๆ ได้อย่างถูกต้อง

7) ใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณ

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

7.1) อธิบายขั้นตอนการใช้งานโปรแกรม MATLAB เพื่อควบคุมเครื่องวัดสัญญาณ Tektronix TDS-340A ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2) แสดงขั้นตอนการใช้คำสั่งในการติดต่อกับเครื่องวัดสัญญาณได้อย่างถูกต้อง

7.3) อธิบายผลที่เกิดขึ้นหลังจากการใช้คำสั่งควบคุมเครื่องมือวัดสัญญาณได้อย่างถูกต้อง

7.4) แสดงขั้นตอนการอ่านข้อมูลจากเครื่องมือวัดมาแสดงผลที่โปรแกรม MATLAB ได้อย่างถูกต้อง

8) ใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

8.1) อธิบายขั้นตอนการใช้งานกล่องเครื่องมือ System Generator ในการออกแบบร่วมกับโปรแกรม MATLAB ได้อย่างถูกต้อง

8.2) อธิบายขั้นตอนการเชื่อมต่อระหว่างบล็อกของ Xilinx และ Simulink ได้อย่างถูกต้อง

8.3) แสดงขั้นตอนการรับค่าข้อมูลจากโปรแกรม MATLAB ได้อย่างถูกต้อง

8.4) แสดงขั้นตอนการส่งค่าข้อมูลออกไปที่โปรแกรม MATLAB ได้อย่างถูกต้อง

9) ใบงานที่ 9 เรื่อง การออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

9.1) แสดงการออกแบบสัมประสิทธิ์ของวงจรกรองโดยใช้ FDATool ได้อย่างถูกต้อง

9.2) ออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator ได้อย่างถูกต้อง

9.3) แสดงวิธีการคำนวณขนาดพื้นที่ของวงจรโดยใช้โปรแกรม System Generator ได้อย่างถูกต้อง

10) ใบงานที่ 10 เรื่อง การสังเคราะห์วงจรโดยใช้โปรแกรม Xilinx ISE Foundation

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

10.1) แสดงขั้นตอนการสังเคราะห์วงจรได้อย่างถูกต้อง

10.2) แสดงขั้นตอนการตรวจสอบขนาดพื้นที่ที่ใช้งานได้อย่างถูกต้อง

10.3) แสดงขั้นตอนการตรวจสอบพอร์ตอินพุตและเอาต์พุตที่ใช้งานได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การวิเคราะห์เนื้อหา

ขั้นตอนการวิเคราะห์เนื้อหาในแต่ละใบงานจะเป็นการกำหนดเนื้อหาที่สามารถทำให้ผู้เรียนได้บรรลุจุดตามจุดประสงค์เชิงพฤติกรรมที่ได้กำหนดไว้ แบ่งเป็นใบงานมีรายละเอียดดังนี้

ใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น

ใบงานที่ 2 เรื่อง การใช้งาน Simulink เบื้องต้น

ใบงานที่ 3 เรื่อง การใช้งาน Real-Time Workshop

ใบงานที่ 4 เรื่อง การใช้งาน xPC Target

ใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม

ใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องกำเนิดสัญญาณ

ใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณ

ใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น

ใบงานที่ 9 เรื่อง การออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator

ใบงานที่ 10 เรื่อง การสังเคราะห์วงจรโดยใช้โปรแกรม Xilinx ISE Foundation

3.3.3 การสร้างใบงานการทดลอง

ขั้นตอนการสร้างใบงานการทดลองจะนำเอาวัตถุประสงค์ของแต่ละใบงาน มาเป็นการกำหนดรายละเอียดของเนื้อหาในใบงาน จากขั้นตอนการวิเคราะห์เนื้อหาเป็นการกำหนดขอบเขตของเนื้อหาเบื้องต้นว่าควรจะมีเนื้อหาอะไร ซึ่งจะนำมาแยกรายละเอียดและจัดทำเป็นใบงานการทดลองต่อไป โดยจะกล่าวถึงการสร้างในใบงานโดยละเอียดในหัวข้อ การสร้างและการทำงาน ซึ่งหลังจากการสร้างใบงานเสร็จสิ้นจะต้องออกแบบวิธีการประเมินผล ว่าผู้เรียนได้บรรลุตามวัตถุประสงค์ที่ได้ตั้งไว้หรือไม่ซึ่งรายละเอียดจะอยู่ในหัวข้อถัดไป

3.3.4 การวัดและประเมินผล

ในการวัดและประเมินว่าผู้เรียนสามารถบรรลุตามจุดประสงค์ที่ได้ตั้งไว้หรือไม่นั้น จะใช้วิธีการให้ผู้เรียนได้ตอบคำถามท้ายใบงานการทดลอง โดยคำตอบของคำถามสามารถที่จะนำมาประเมินได้ว่าผู้เรียนสามารถบรรลุตามจุดประสงค์ที่ได้ตั้งไว้หรือไม่ โดยคำถามจะแบ่งเป็นใบงานการทดลองดังนี้

1) ใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น คำถามท้ายการทดลองมีดังนี้

1.1) การเรียกใช้งาน Help และ Demos นั้นมีวิธีการอย่างไร?

1.2) คำสั่ง who และ whos แตกต่างกันอย่างไร?

1.3) ถ้าต้องการหาความหมายและลักษณะการใช้งานฟังก์ชั่นใดๆ เช่น Cos(x) มีวิธีการอย่างไร?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4) การเขียน M-file เป็นลักษณะฟังก์ชั่นนั้นมีประโยชน์อย่างไร?

2) ใบงานที่ 2 เรื่อง การใช้งาน Simulink เบื้องต้น คำถามท้ายการทดลองมีดังนี้

2.1) การจำลองการทำงาน (Simulation) มีประโยชน์อย่างไร?

2.2) การสร้างโมเดลด้วย Simulink เมื่อบันทึกไฟล์จะมีนามสกุลเรียกว่าอย่างไร?

3) ใบงานที่ 3 เรื่อง การใช้งาน Real-Time Workshop คำถามท้ายการทดลองมีดังนี้

3.1) จงอธิบายขั้นตอนการทำงานของ Real-Time Workshop มาพอสังเขป?

3.2) จงยกตัวอย่างไฟล์ TLC มา 3 ชนิดและอธิบายลักษณะการนำไปใช้งาน?

3.3) จงอธิบายหน้าที่ของหน้าต่าง Model Advisor และ Model Explorer?

3.4) จงเปรียบเทียบผลการทดลองผลของการจำลองการทำงานกับผลที่ได้จากการสร้างโค๊ดภาษาซีว่าเหมือนกันหรือแตกต่างกันเพราะเหตุใด จงอธิบาย?

4) ใบงานที่ 4 เรื่อง การใช้งาน xPC Target คำถามท้ายการทดลองมีดังนี้

4.1) จงอธิบายขั้นตอนการทำงานของ xPC Target มาพอสังเขป?

4.2) ในกรณีที่มีความต้องการติดต่อกับคอมพิวเตอร์ปลายทาง (Target PC) มากกว่า 1 ตัวจะมีวิธีการกำหนดอย่างไร?

4.3) จากการสังเกตสัญญาณที่คอมพิวเตอร์ปลายทาง (Target PC) จะเห็นว่าสัญญาณจะทำงานที่ช่วงเวลาหนึ่งเท่านั้นแล้วก็จะหยุดลง ช่วงเวลาที่สัญญาณทำงานคือเท่าใดและถ้าต้องการเปลี่ยนแปลงช่วงเวลานี้สามารถกำหนดได้อย่างไร จงอธิบาย?

4.4) จงอธิบายความแตกต่างระหว่างการกำหนดค่าเพื่อเชื่อมต่อแบบ RS232 และระบบเครือข่าย?

5) ใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม คำถามท้ายการทดลองมีดังนี้

5.1) จงอธิบายขั้นตอนการทำงานของ xPC Target ในการติดต่อกับพอร์ต RS-232 มาพอสังเขป?

5.2) ในการกำหนดพอร์ตและอัตราการรับส่งข้อมูลสามารถกำหนดได้อย่างไร?

5.3) ในขั้นตอนการสร้างโค๊ดภาษาซีนั้น ถ้ากำหนด TLC (Target Language Compiler) เป็นแบบ xpctarget.tlc จะได้ไฟล์ที่ใช้ดาวน์โหลดสู่ Target PC เป็นไฟล์ชื่อชนิดใดและแตกต่างกับการกำหนด TLC แบบ grt.tlc อย่างไร?

5.4) จะอธิบายผลที่เกิดขึ้นเมื่อทดลองถอดสายสื่อสารระหว่างพอร์ตอนุกรมออก?

6) ใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องกำเนิดสัญญาณคำถามท้ายการทดลองมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1) ถ้าต้องการทราบรายละเอียดของเครื่องมือวัดนั้นจะต้องใช้คำสั่งใดและมีลักษณะการใช้งานอย่างไร?

6.2) ในการเริ่มต้นการติดต่อกับการ์ด GPIB และเครื่องมือวัดนั้นต้องใช้คำสั่งใดและมีลักษณะการใช้งานอย่างไร?

6.3) เมื่อต้องการยกเลิกการติดต่อกับการ์ด GPIB และเครื่องมือวัดเสร็จสิ้นต้องใช้คำสั่งอะไรบ้างในการยกเลิกการติดต่อ?

7) ใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณคำถามท้ายการทดลองมีดังนี้

7.1) การกำหนดค่า Primary Address ของเครื่องวัดสัญญาณว่ามีขั้นตอนวิธีการกำหนดอย่างไร?

7.2) ค่า InputBufferSize และค่า OutputBufferSize มีความหมายว่าอย่างไร?

7.3) จงอธิบายความแตกต่างของข้อมูลที่พล็อตกราฟกับสัญญาณที่แสดงที่หน้าจอเครื่องวัดสัญญาณ?

8) ใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น คำถามท้ายการทดลองมีดังนี้

8.1) บล็อก Gateway In และบล็อก Gateway Out มีหน้าที่อะไร?

8.2) การกำหนดพารามิเตอร์ต่างๆ ของบล็อก Gateway In มีผลต่อสัญญาณที่ได้อย่างไร?

8.3) การนำเอาข้อมูลใน MATLAB Workspace มาเป็นสัญญาณอินพุตให้กับโมเดลมีวิธีกำหนดอย่างไร?

8.4) การส่งค่าข้อมูลหลังจากการจำลองการทำงานใน Simulink ที่จะให้กับโปรแกรม MATLAB นั้นมีวิธีกำหนดอย่างไร?

9) ใบงานที่ 9 เรื่อง การออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator คำถามท้ายการทดลองมีดังนี้

9.1) วงจรกรองที่ออกแบบมีจำนวน Order สูงสุดเท่ากับเท่าใด?

9.2) จำนวนบิตที่จะเก็บค่าข้อมูลของสัมประสิทธิ์ นั้นมีผลต่อคุณภาพของวงจรกรองหรือไม่ จงอธิบาย?

10) ใบงานที่ 10 เรื่อง การสังเคราะห์วงจร โดยใช้โปรแกรม Xilinx ISE Foundation คำถามท้ายการทดลองมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

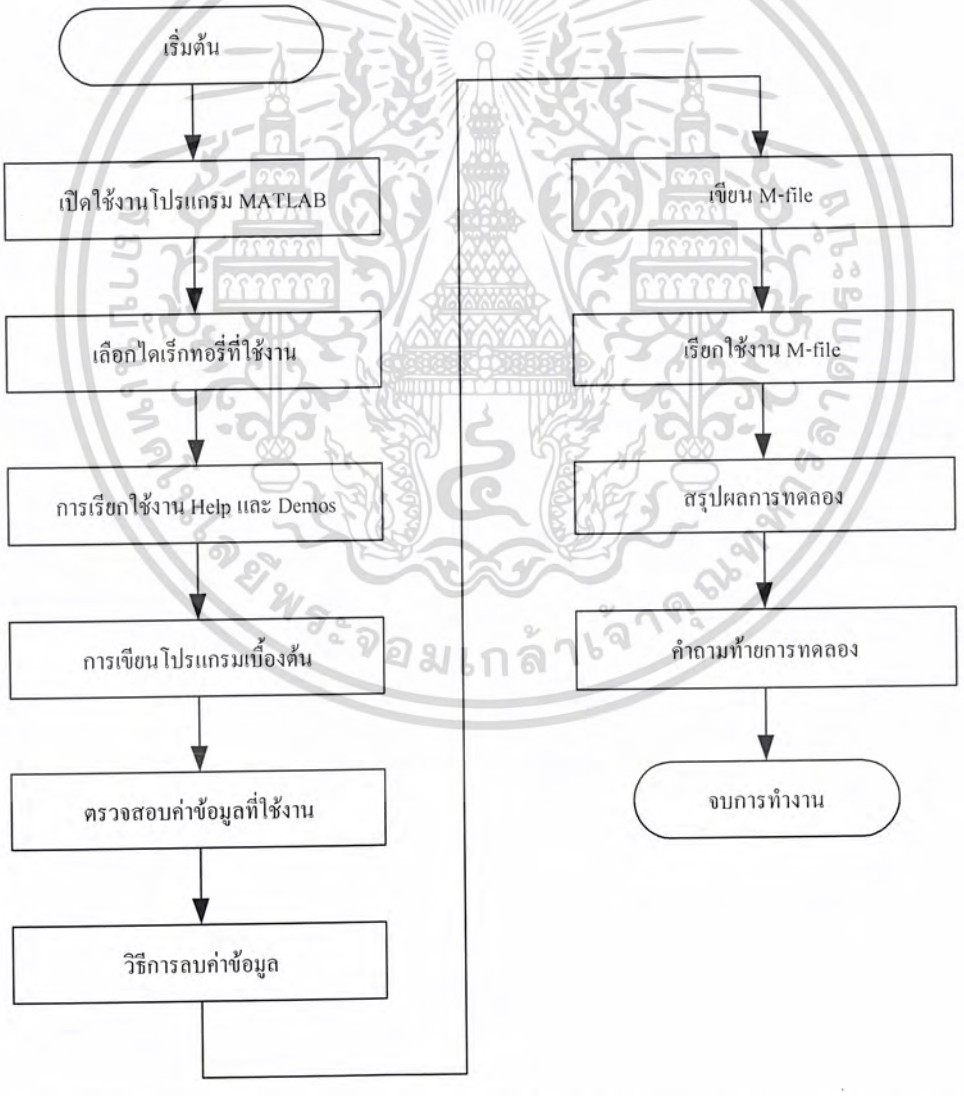
10.1) จงเปรียบเทียบผลการคำนวณขนาดพื้นที่จากโปรแกรม MATLAB กับโปรแกรม Xilinx ISE ว่ามีค่าเท่ากันหรือไม่เพราะเหตุใด?

3.4 การสร้างและการทำงาน

ขั้นตอนการสร้างใบงานการทดลองจะเป็นการกำหนดเนื้อหาในใบงานอย่างละเอียด โดยเนื้อหาในแต่ละใบงานจะต้องสอดคล้องกับวัตถุประสงค์ที่กำหนดไว้และต้องสามารถประเมินผลได้จากการตอบคำถามท้ายใบงานการทดลอง ซึ่งเนื้อหาแบ่งเป็นใบงานดังนี้

3.4.1 ใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น

จากผังงานในรูปที่ 3.1 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

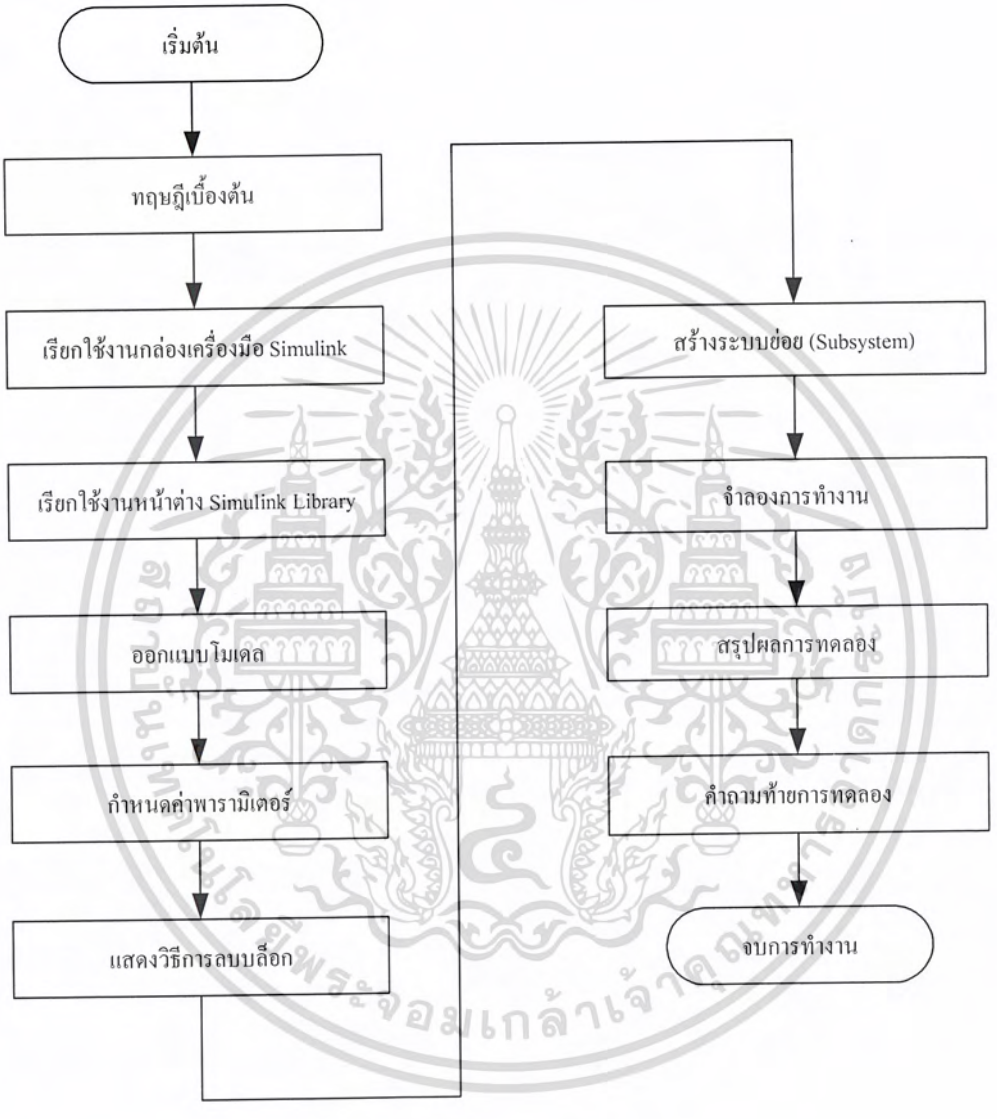


รูปที่ 3.1 ผังงานของใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 ใบงานที่ 2 เรื่องการใช้งาน Simulink เบื้องต้น

จากผังงานในรูปที่ 3.2 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

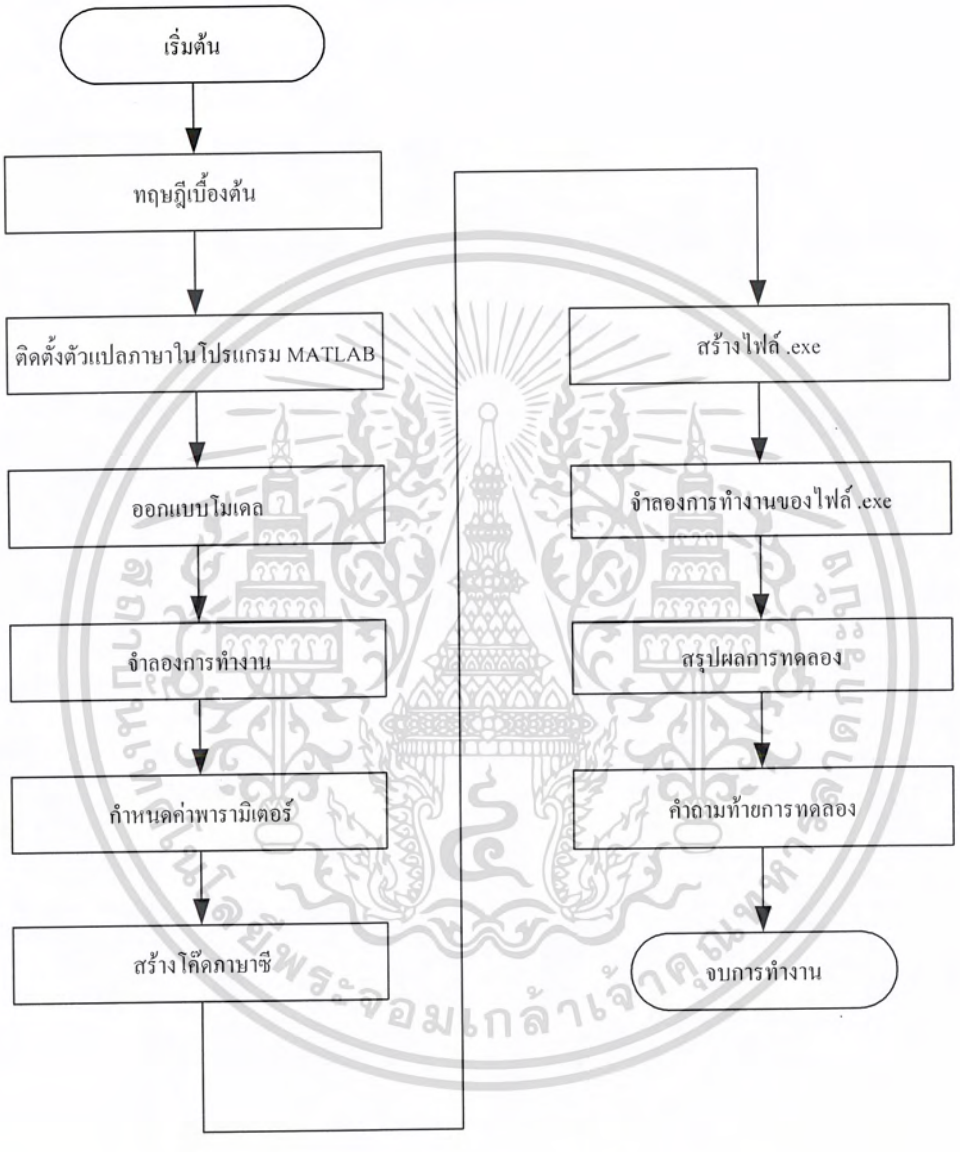


รูปที่ 3.2 ผังงานของใบงานที่ 2 เรื่อง การใช้งาน Simulink เบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 ใบงานที่ 3 เรื่องการใช้งาน Real-Time Workshop

จากผังงานในรูปที่ 3.3 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

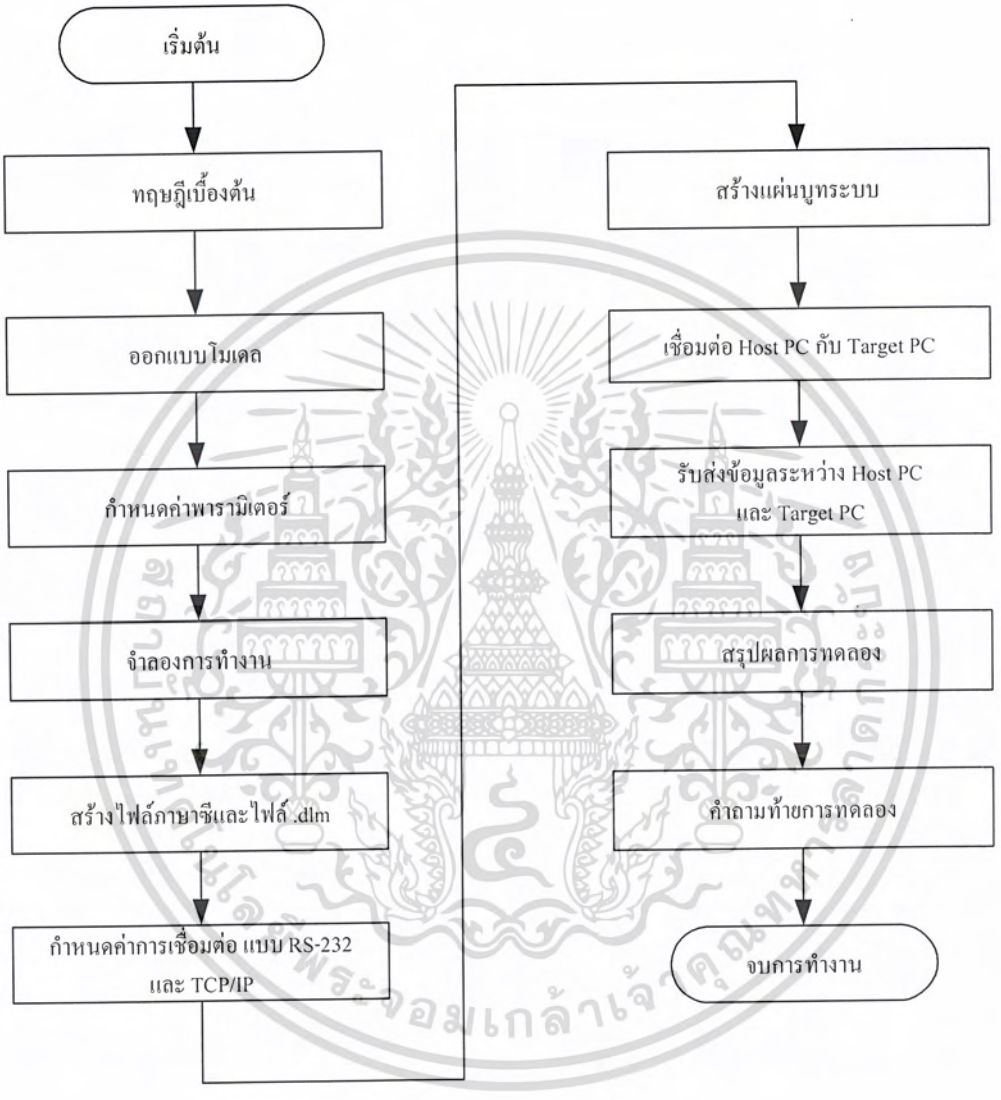


รูปที่ 3.3 ผังงานของใบงานที่ 3 เรื่อง การใช้งาน Real-Time Workshop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 ใบงานที่ 4 เรื่อง การใช้งาน xPC Ta rget

จากผังงานในรูปที่ 3.4 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

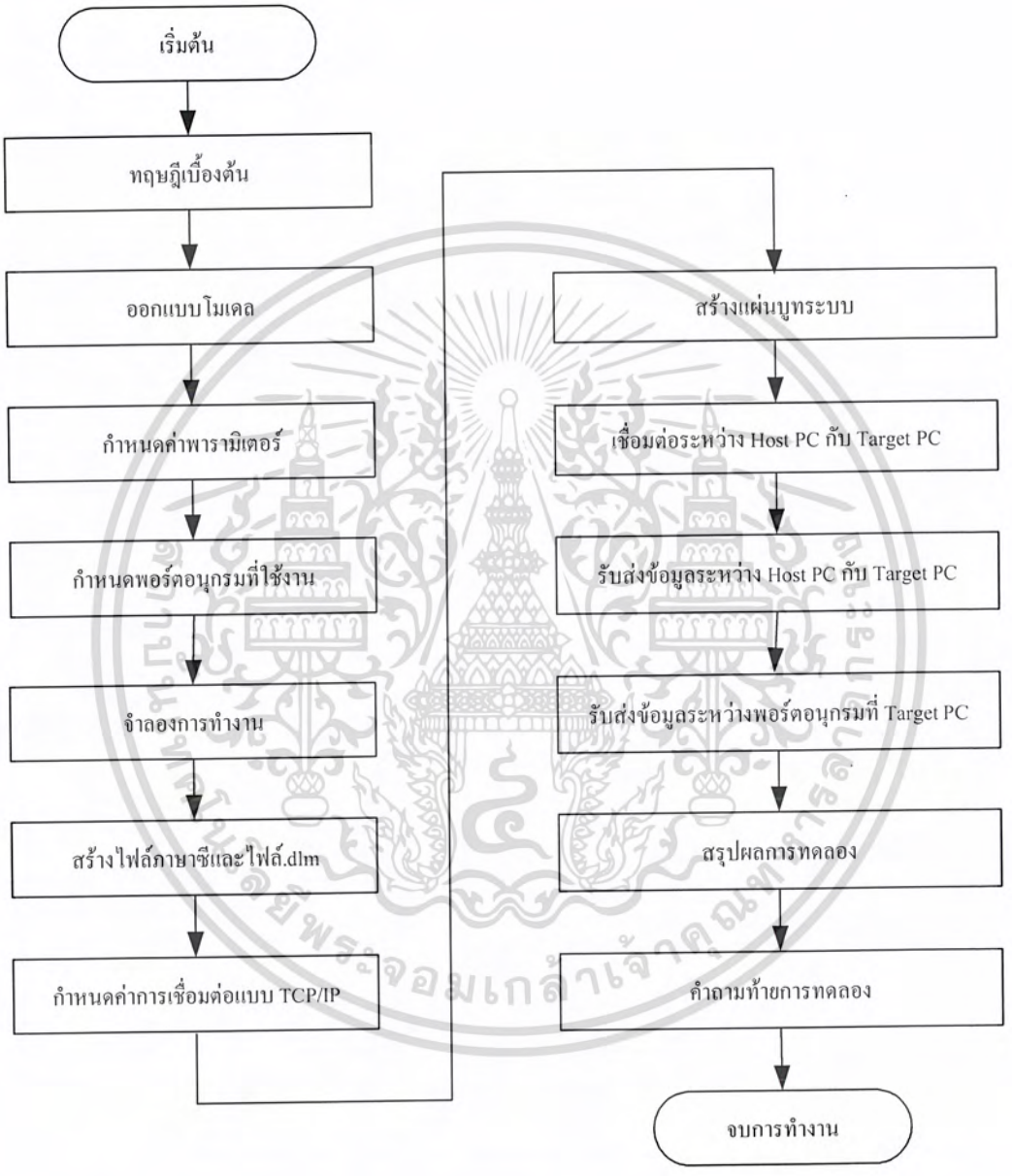


รูปที่ 3.4 ผังงานของใบงานที่ 4 เรื่อง การใช้งาน xPC Target

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 ใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม

จากผังงานในรูปที่ 3.5 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

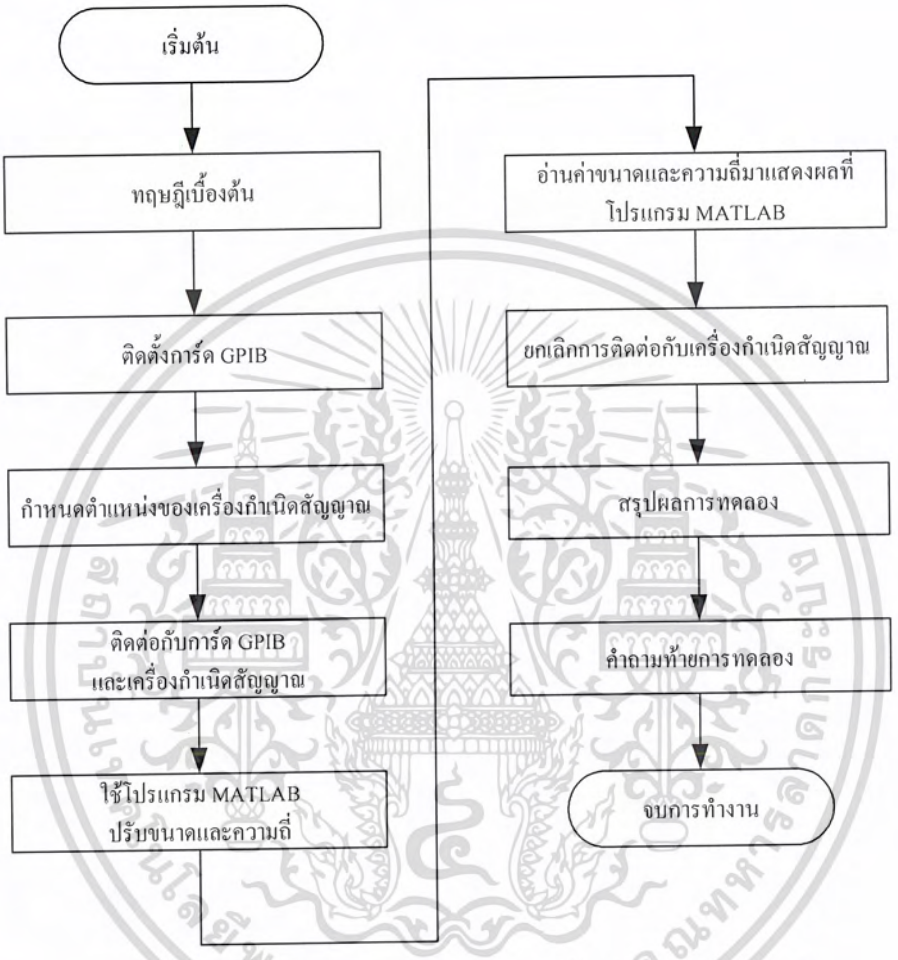


รูปที่ 3.5 ผังงานของใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.6 ใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องกำเนิดสัญญาณ

จากผังงานในรูปที่ 3.6 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

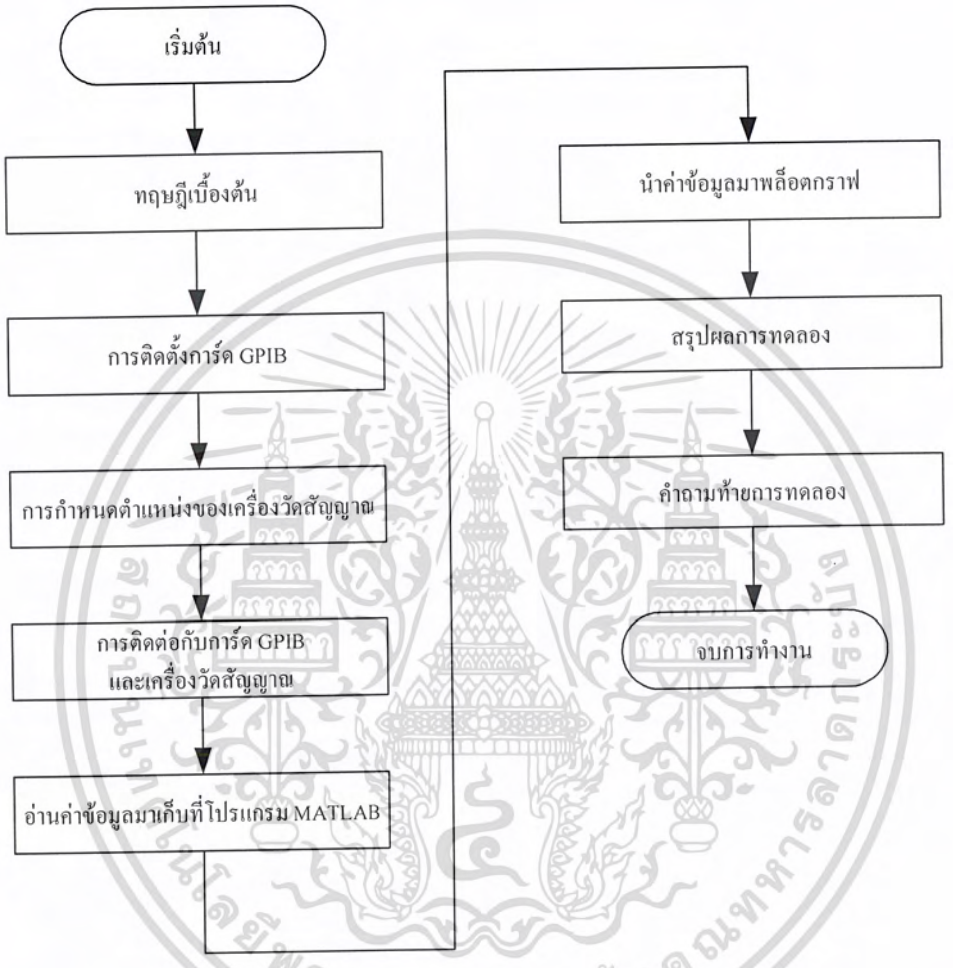


รูปที่ 3.6 ผังงานของใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องกำเนิดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.7 ใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณ

จากผังงานในรูปที่ 3.7 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

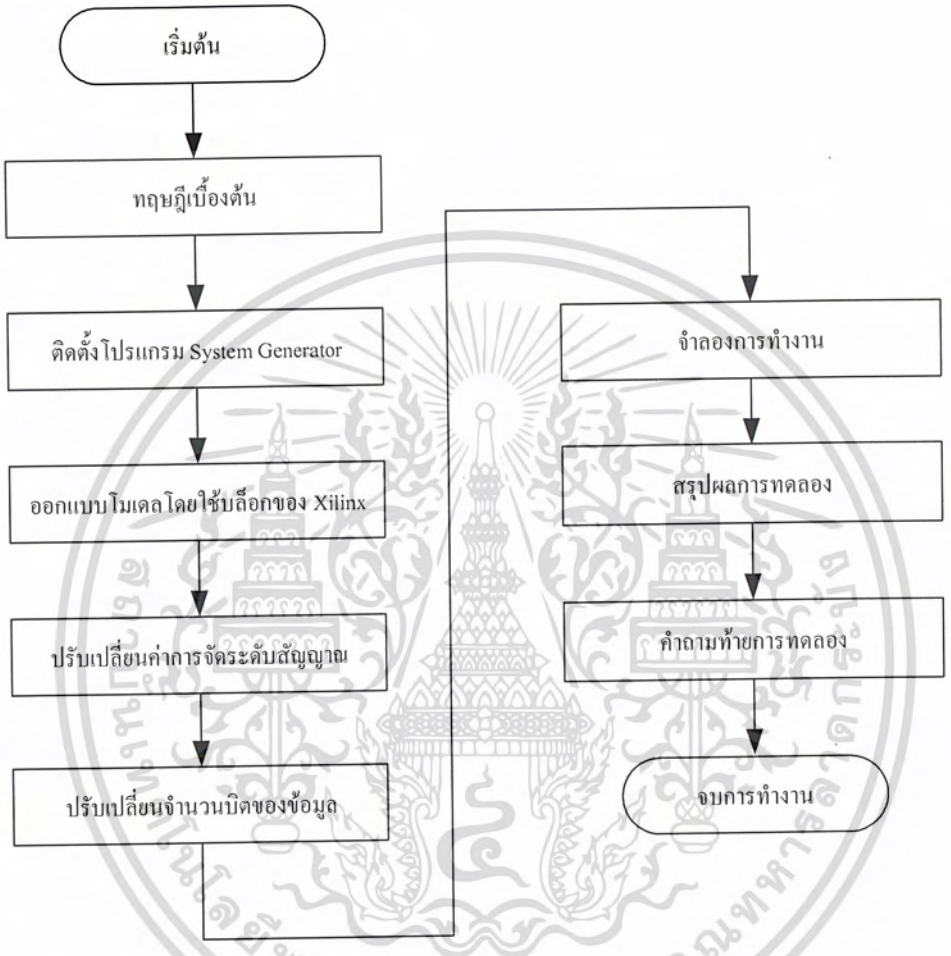


รูปที่ 3.7 ผังงานของใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.8 ใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น

จากผังงานในรูปที่ 3.8 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

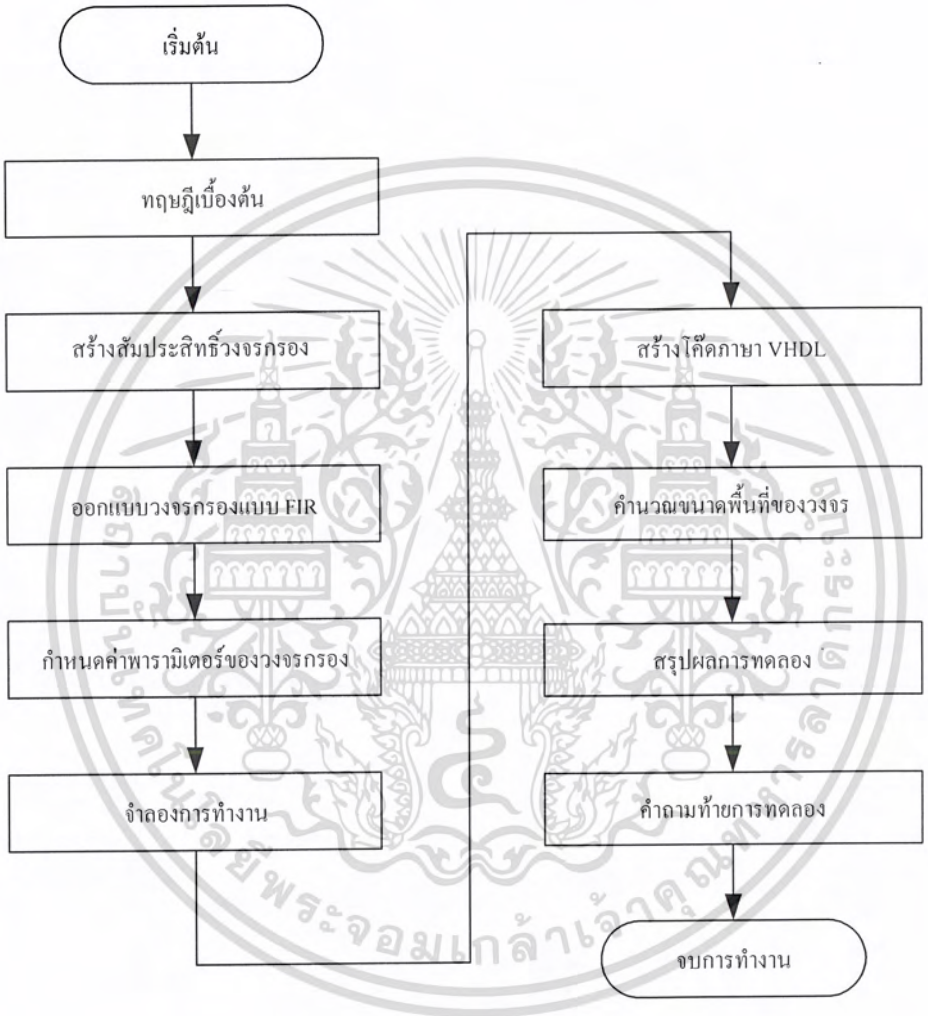


รูปที่ 3.8 ผังงานของใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.9 ใบงานที่ 9 เรื่อง การออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator

จากผังงานในรูปที่ 3.9 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก

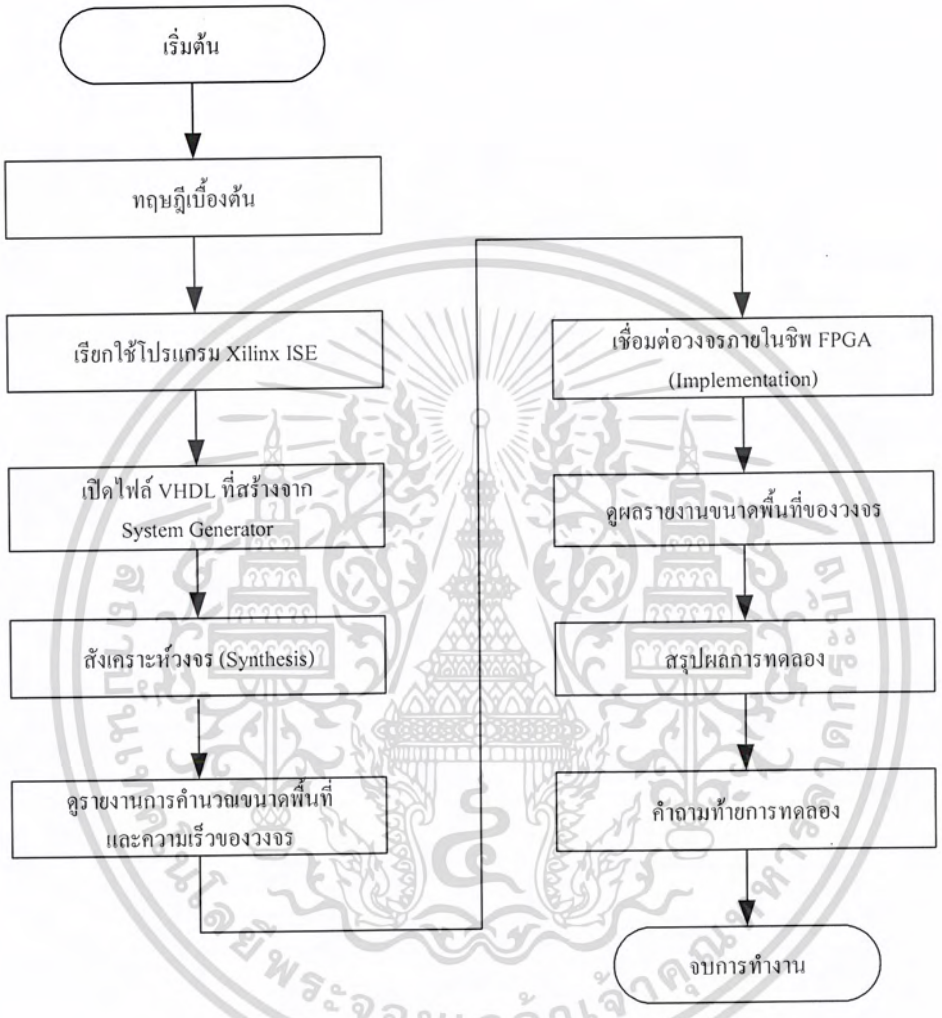


รูปที่ 3.9 ผังงานของใบงานที่ 9 เรื่อง การออกแบบวงจรกรองดิจิตอล โดยใช้ โปรแกรม System Generator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.10 ใบงานที่ 10 เรื่อง การสังเคราะห์วงจร โดยใช้โปรแกรม Xilinx ISE Foundation

จากผังงานในรูปที่ 3.10 ซึ่งจะมีรายละเอียดเนื้อหาดังภาคผนวก ก



รูปที่ 3.10 ผังงานของใบงานที่ 10 เรื่อง การสังเคราะห์วงจร
โดยใช้โปรแกรม Xilinx ISE Foundation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 กล่าวนำ

ในบทนี้จะกล่าวถึงการทดลองและผลการทดลองของใบงานการทดลองทั้งหมด 10 ใบงาน โดยจะแบ่งเป็นขั้นตอนการทดลองของเนื้อหาที่ได้ออกแบบไว้ในบทที่ 3 ซึ่งแบ่งเป็นขั้นตอนดังนี้

4.2 การทดลองในใบงานที่ 1 เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น

4.2.1 ขั้นตอนการทดลอง

เขียนโปรแกรมลักษณะเป็นฟังก์ชั่นชื่อ lab1 เพื่อหาผลลัพธ์ของการคูณเลข 2 จำนวนและบันทึกเป็น M-file แสดงดังรูปที่ 4.1

```
1    function ans = lab1(x,y)
2
3 -    ans = x*y;
4
5 -    end
```

รูปที่ 4.1 โปรแกรมการหาผลลัพธ์ของการคูณเลข 2 จำนวน

4.2.2 ผลการทดลอง

เมื่อทดสอบการทำงานของฟังก์ชั่น lab1 โดยการป้อนค่าจำนวน 2 จำนวนเพื่อหาค่าผลลัพธ์ ผลลัพธ์ที่ได้แสดงดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองโปรแกรมการหาผลลัพธ์ของการคูณเลข 2 จำนวน

โจทย์		ผลลัพธ์
x	y	ans
5	41	205

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 (ต่อ) ผลการทดลองโปรแกรมการหาผลลัพธ์ของการคูณเลข 2 จำนวน

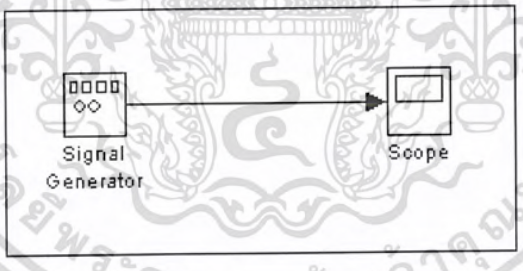
โจทย์		ผลลัพธ์
6	22	132
13	35	455
158	77	12166

จากผลการทดลองในตารางที่ 4.1 ซึ่งค่าผลลัพธ์ที่ได้เมื่อเปรียบเทียบกับการคำนวณด้วยเครื่องคำนวณมีค่าเท่ากัน แสดงว่าการทำงานของโปรแกรมการหาผลลัพธ์ของการคูณเลข 2 จำนวน ให้ผลลัพธ์ได้อย่างถูกต้อง

4.3 การทดลองในใบงานที่ 2 เรื่อง การใช้งาน Simulink เบื้องต้น

4.3.1 ขั้นตอนการทดลอง

ออกแบบโมเดลโดยสร้างสัญญาณจากบล็อก Signal Generator กำหนดค่าสัญญาณเป็น Sine Wave มีระดับสัญญาณเท่ากับ 5 V และมีความถี่เท่ากับ 1 Rad/s แสดงดังรูปที่ 4.2

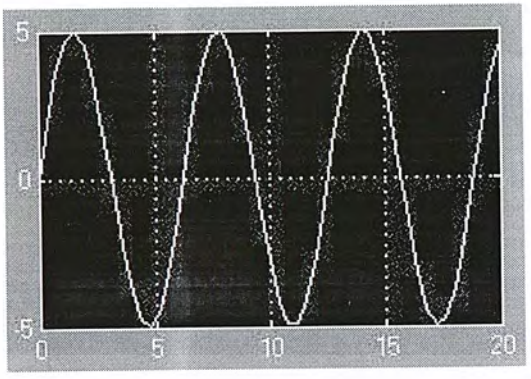


รูปที่ 4.2 ลักษณะของโมเดลเพื่อการทดสอบการใช้งาน Simulink เบื้องต้น

4.3.2 ผลการทดลอง

เมื่อทดสอบการทำงานของโมเดลที่ออกแบบไว้ แล้วสังเกตผลที่เกิดขึ้นจากหน้าต่างสโคป (Scope) ผลที่เกิดขึ้นแสดงดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



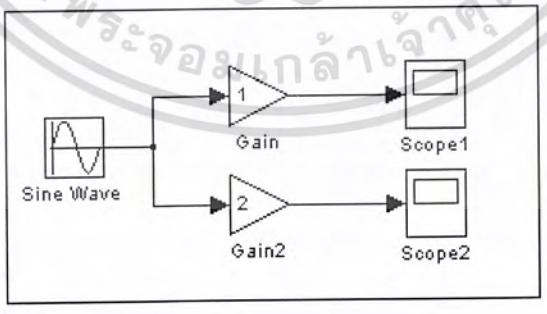
รูปที่ 4.3 ลักษณะสัญญาณที่สโคป

จากผลการทดลองที่สโคปจะแสดงสัญญาณ Sine Wave ที่ระดับสัญญาณเท่ากับ 5 V และมีความถี่เท่ากับ 1 Rad/s ซึ่งเท่ากับค่าที่กำหนดจากบล็อก Signal Generator แสดงว่าผลการทดลองถูกต้องตามที่ออกแบบไว้

4.4 การทดลองในใบงานที่ 3 เรื่อง การใช้งาน Real-Time Workshop

4.4.1 ขั้นตอนการทดลอง

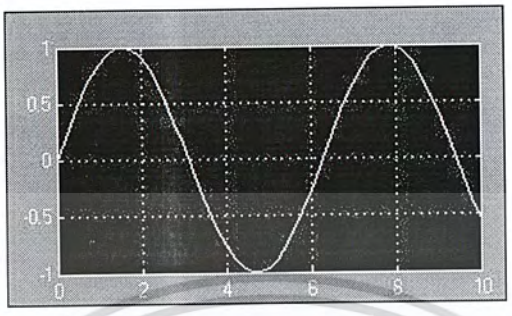
ออกแบบโมเดลซึ่งเป็นวงจรขยายสัญญาณ โดยสร้างบล็อก Sine Wave กำเนิดสัญญาณ Sine Wave มีค่าระดับสัญญาณเท่ากับ 1V และมีความถี่เท่ากับ 1 Rad/s โดยสัญญาณจะผ่านบล็อกขยายสัญญาณ 1 เท่าและ 2 เท่า แสดงผลที่หน้าจอสโคป แสดงดังรูปที่ 4.4 แล้วทำการสร้างโค๊ดภาษาซีจากกล่องเครื่องมือ Real–Time Workshop



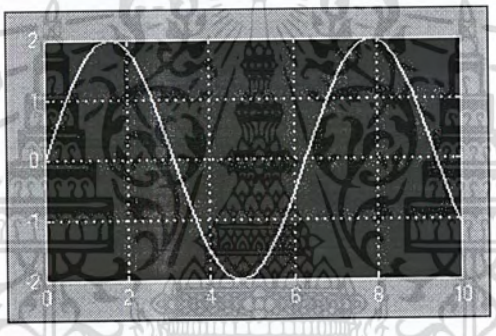
รูปที่ 4.4 การออกแบบโมเดลเพื่อทดสอบการสร้างโค๊ดภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองจำลองการทำงานของโมเดลที่ได้ออกแบบเพื่อดูผลที่หน้าต่างสโคป แสดงดังรูปที่ 4.5 และรูปที่ 4.6



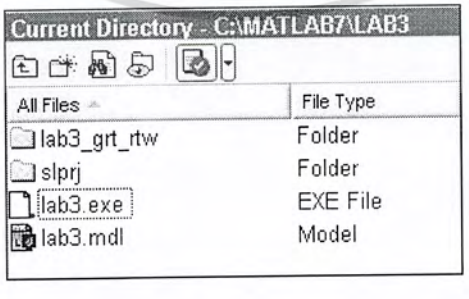
รูปที่ 4.5 ผลการจำลองการทำงานของโมเดลซึ่งเป็นผลของสัญญาณที่ผ่านบล็อก Gain = 1



รูปที่ 4.6 ผลการจำลองการทำงานของโมเดลซึ่งเป็นผลของสัญญาณที่ผ่านบล็อก Gain = 2

4.4.2 ผลการทดลอง

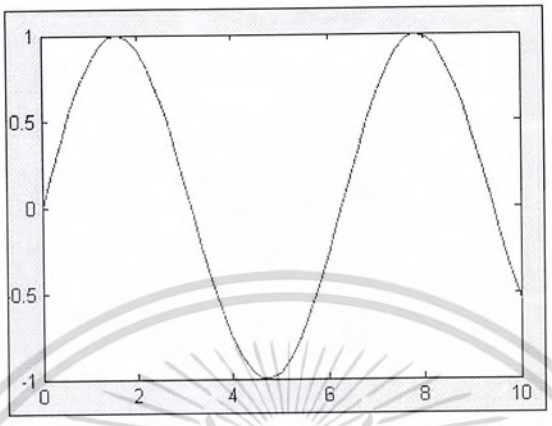
เมื่อขั้นตอนการสร้างโค๊ดภาษาซีเสร็จสิ้นจะได้ไฟล์ที่มีนามสกุล .exe แสดงดังรูปที่ 4.7



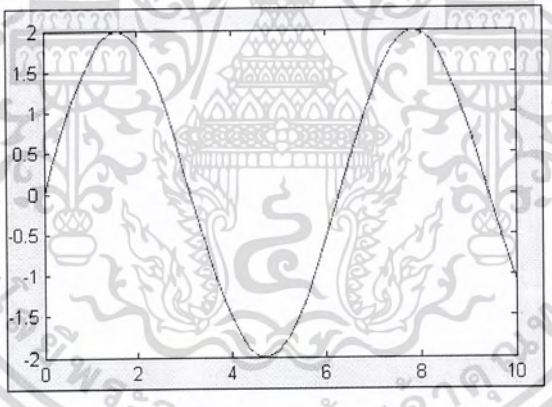
รูปที่ 4.7 ไฟล์ที่ได้หลังจากการสร้างโค๊ดภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองนำข้อมูลที่สร้างขึ้นจากไฟล์ model.exe มาพล็อตกราฟผลที่เกิดขึ้นแสดงดังรูปที่ 4.8 และรูปที่ 4.9



รูปที่ 4.8 ผลการนำข้อมูลจากไฟล์ model.exe มาพล็อตกราฟซึ่งเป็นผลการพล็อตกราฟของข้อมูลที่ถูกคูณด้วยค่าคงที่เท่ากับ 1



รูปที่ 4.9 ผลการนำข้อมูลจากไฟล์ model.exe มาพล็อตกราฟซึ่งเป็นผลการพล็อตกราฟของข้อมูลที่ถูกคูณด้วยค่าคงที่เท่ากับ 2

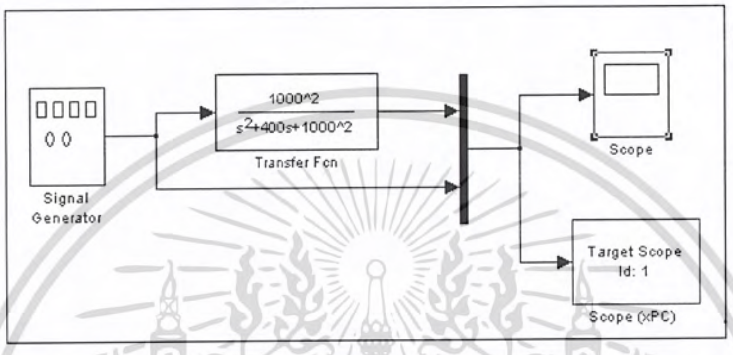
จากผลการทดลองจะเห็นว่าข้อมูลที่ได้จากไฟล์ model.exe เมื่อนำมาพล็อตกราฟจะมีลักษณะเท่ากับผลที่เกิดจากการจำลองการทำงานที่สโคป แสดงว่าสามารถโค๊ดภาษาซีจากโมเดลที่ออกแบบได้จริงและไฟล์ model.exe สามารถสร้างข้อมูลได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองในใบงานที่ 4 เรื่อง การใช้งาน xPC Target

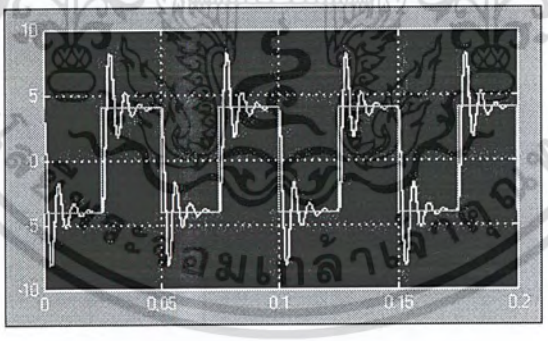
4.5.1 ขั้นตอนการทดลอง

ออกแบบโมเดลเพื่อทดสอบการใช้งาน xPC Target ติดต่อกับ Target PC ผ่านพอร์ตอนุกรมและระบบเครือข่าย โดยสร้างสัญญาณ Square Wave จากบล็อก Signal Generator ส่งผ่านบล็อกที่มี Transfer Function แล้วแสดงผลที่หน้าต่างสโคป แสดงดังรูปที่ 4.10



รูปที่ 4.10 การออกแบบโมเดลเพื่อทดสอบการใช้งาน xPC Target

ทดลองจำลองการทำงานของโมเดลผลที่เกิดขึ้นที่หน้าต่างสโคป แสดงดังรูปที่ 4.11

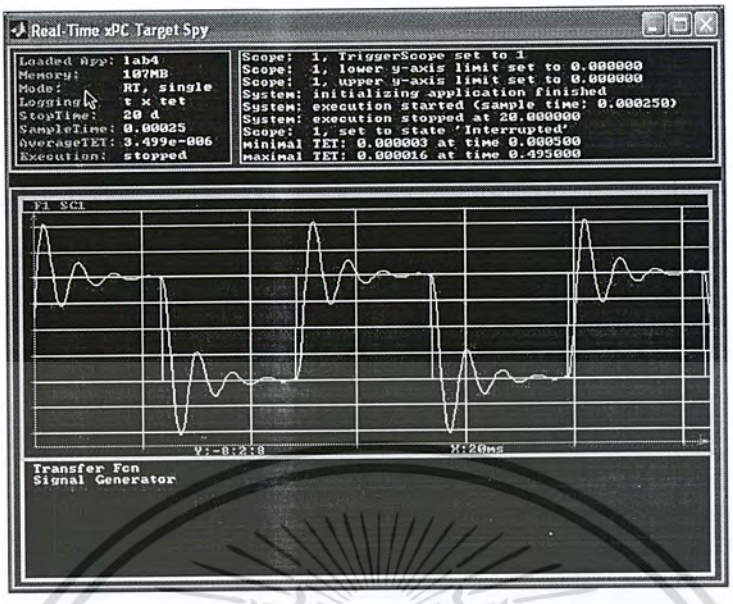


รูปที่ 4.11 ผลการจำลองการทำงานของโมเดล

4.5.2 ผลการทดลอง

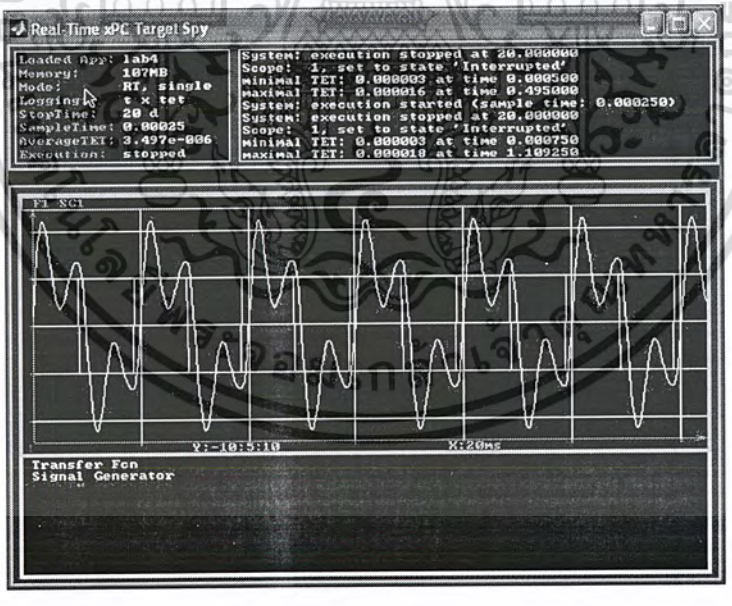
ทดลองสร้างโค๊ดภาษาซีจากกล่องเครื่องมือ Real–Time Workshop และติดต่อกับ Target PC ผลที่เกิดขึ้นที่ Target PC แสดงดังรูปที่ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ผลการทดสอบการทำงานของ Target PC

ทดลองปรับเปลี่ยนค่าความถี่ของสัญญาณที่ Host PC ผลที่เกิดขึ้น แสดงดังรูปที่ 4.13



รูปที่ 4.13 ผลที่เกิดขึ้นที่ Target PC เมื่อทดลองเปลี่ยนแปลงค่าความถี่

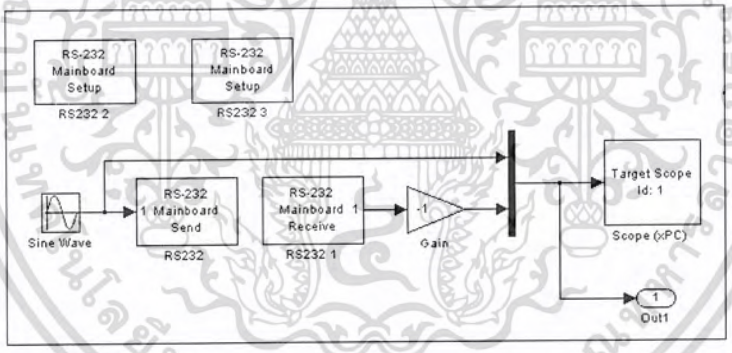
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองที่ได้จะเห็นว่าสัญญาณที่แสดงที่ Target PC นั้นจะมีลักษณะเหมือนกับสัญญาณที่ได้จากการจำลองการทำงานที่หน้าต่างสโคปและเมื่อทดลองปรับเปลี่ยนค่าความถี่ของสัญญาณที่ Host PC แล้วจะเกิดการเปลี่ยนแปลงสัญญาณที่ Target PC ที่เวลาจริงด้วย ดังนั้นแสดงว่าสามารถติดต่อกับ Target PC ได้จริงและสามารถรับและส่งข้อมูลระหว่าง Host PC และ Target PC ได้อย่างถูกต้อง

4.6 การทดลองในใบงานที่ 5 เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม (RS-232)

4.6.1 ขั้นตอนการทดลอง

ออกแบบโมเดลเพื่อทดสอบการติดต่อกับพอร์ตอนุกรม โดยโมเดลจะสร้างสัญญาณ Sine Wave ออกที่พอร์ตอนุกรม COM1 และรับค่าสัญญาณกลับมาที่พอร์ต COM2 แต่สัญญาณที่รับเข้ามาจะผ่านบล็อก Gain เท่ากับ -1 ทำให้สัญญาณที่รับเข้ามาจะมีลักษณะกลับเฟสกับสัญญาณที่ส่งออก 180 องศา โดยสัญญาณทั้งหมดจะแสดงที่ Target PC ลักษณะของโมเดลแสดงดังรูปที่ 4.14

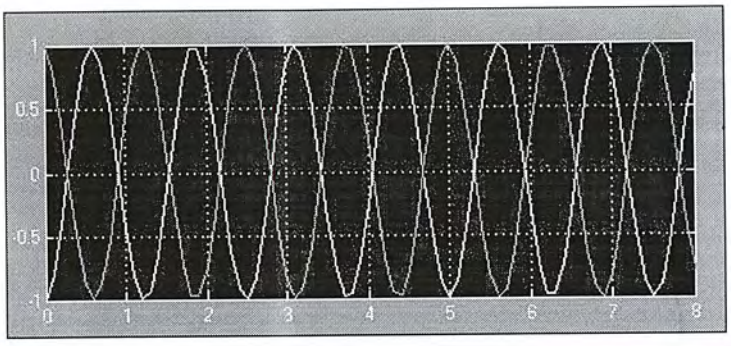


รูปที่ 4.14 การออกแบบโมเดลเพื่อทดสอบการติดต่อกับพอร์ตอนุกรม

4.6.2 ผลการทดลอง

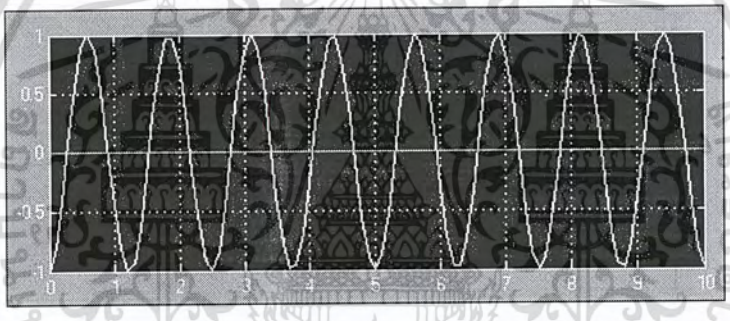
เมื่อทดลองติดต่อกับ Target PC และเชื่อมต่อสายสื่อสารอนุกรมระหว่าง COM1 และ COM2 ผลที่เกิดขึ้นที่ Target PC คือ จะแสดงสัญญาณที่ส่งออกที่พอร์ต COM1 และแสดงสัญญาณที่รับเข้ามาที่ COM2 โดยสัญญาณทั้ง 2 จะมีการกลับเฟสกัน 180 องศา แสดงดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 ผลที่เกิดขึ้นที่ Target PC เมื่อเชื่อมต่อสายสื่อสารอนุกรมระหว่าง COM1 และ COM2

เมื่อทดลองถอดสายสื่อสารอนุกรมระหว่าง COM1 และ COM2 ออกผลที่เกิดขึ้นที่ Target PC แสดงดังรูปที่ 4.16



รูปที่ 4.16 ผลที่เกิดขึ้นที่ Target PC เมื่อถอดสายสื่อสารอนุกรมระหว่าง COM1 และ COM2 ออก

จากผลการทดลองจะเห็นว่าลักษณะสัญญาณที่ Target PC จะแสดงสัญญาณที่ส่งออกที่ COM1 และแสดงสัญญาณที่รับเข้ามาจากพอร์ต COM2 และเมื่อทดลองถอดสายสื่อสารระหว่าง COM1 และ COM2 ออกจะเกิดการเปลี่ยนแปลงของสัญญาณที่ Target PC คือจะไม่แสดงสัญญาณที่รับเข้ามาจากพอร์ต COM2 แสดงว่าสามารถที่จะติดต่อกับพอร์ตอนุกรมที่ Target PC เพื่อรับและส่งข้อมูลได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การทดลองในใบงานที่ 6 เรื่อง การใช้งานโปรแกรม MATLAB ติดต่อกับเครื่องกำเนิดสัญญาณ

4.7.1 ขั้นตอนการทดลอง

1) ใช้คำสั่ง >> g=gpib('ni',0,2) เพื่อทดลองติดต่อกับการ์ด GPIB และเครื่องกำเนิดสัญญาณ

2) ใช้คำสั่ง >> fprintf(g,'*IDN?') และคำสั่ง >> data=fscanf(g) เพื่ออ่านค่าข้อมูลของเครื่องกำเนิดสัญญาณ

3) ทดลองกำหนดปรับลักษณะสัญญาณ ระดับสัญญาณและความถี่ของสัญญาณที่เครื่องกำเนิดสัญญาณ โดยใช้คำสั่งดังนี้

```
>>fprintf(g, 'Volt 2')
>>fprintf(g, 'Freq 10k')
>>fprintf(g, 'Func sin')
```

4) ทดลองอ่านค่าสัญญาณที่เครื่องกำเนิดสัญญาณ แล้วมาแสดงผลที่โปรแกรม MATLAB โดยใช้คำสั่งดังนี้

```
>>fprintf(g, 'Func?')
>>fprintf(g, 'Volt?')
>>fprintf(g, 'Freq?')
```

4.7.2 ผลการทดลอง

1) การทดลองติดต่อกับการ์ด GPIB และเครื่องกำเนิดสัญญาณผลที่ได้ คือ สามารถติดต่อได้จริง

2) ผลการอ่านค่าข้อมูลของเครื่องกำเนิดสัญญาณ แสดงดังรูปที่ 4.17

```
data =

SONY/TEK,AFG310,0,SCPI:94.0 FW:2.1
```

รูปที่ 4.17 ผลการอ่านค่าข้อมูลของเครื่องกำเนิดสัญญาณ

3) ผลการทดลองปรับลักษณะสัญญาณ ระดับสัญญาณ และความถี่ของสัญญาณ แสดงดังตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการปรับค่าที่หน้าจอของเครื่องกำเนิดสัญญาณ

Function	Frequency (Hz)	Amplitude (V)
Sine	10.00000 k	2.000

4) ผลการทดลองการอ่านค่าของเครื่องกำเนิดสัญญาณ แสดงดังรูปที่ 4.18, รูปที่ 4.19 และรูปที่ 4.20

```
>> fprintf(g, 'Func?')
>> data=fscanf(g)

data =

SIN
```

รูปที่ 4.18 ผลการอ่านค่าลักษณะสัญญาณของเครื่องกำเนิดสัญญาณ

```
>> fprintf(g,'Freq?')
>> data=fscanf(g)

data =

1.000000E+04
```

รูปที่ 4.19 ผลการอ่านค่าความถี่ของเครื่องกำเนิดสัญญาณ

```
>> fprintf(g,'Volt?')
>> data=fscanf(g)

data =

2.000000
```

รูปที่ 4.20 ผลการอ่านค่าระดับสัญญาณของเครื่องกำเนิดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองจะเห็นว่าสามารถติดต่อกับการ์ด GPIB และเครื่องกำเนิดสัญญาณได้จริง สามารถปรับเปลี่ยนสัญญาณที่เครื่องกำเนิดสัญญาณได้จริงและสามารถอ่านค่าลักษณะสัญญาณที่เครื่องกำเนิดสัญญาณได้อย่างถูกต้อง

4.8 การทดลองในใบงานที่ 7 เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณ

4.8.1 ขั้นตอนการทดลอง

1) ใช้คำสั่ง >> g=gpib('ni',0,1) เพื่อทดลองติดต่อกับการ์ด GPIB และเครื่องวัดสัญญาณ

2) ใช้คำสั่ง >> fprintf(g,'*IDN?') และคำสั่ง >> data=fscanf(g) เพื่อดูข้อมูลของเครื่องวัดสัญญาณ

3) ทดลองอ่านค่าสัญญาณที่เกิดขึ้นที่หน้าจอเครื่องวัดสัญญาณ แล้วนำข้อมูลที่ได้มาพล็อตกราฟและเปรียบเทียบกับลักษณะสัญญาณที่หน้าจอเครื่องวัดสัญญาณ

4.8.2 ผลการทดลอง

1) ผลการติดต่อกับการ์ด GPIB และเครื่องวัดสัญญาณผลที่ได้คือ สามารถติดต่อได้จริง

2) ผลการอ่านค่าข้อมูลของเครื่องวัดสัญญาณ แสดงดังรูปที่ 4.21

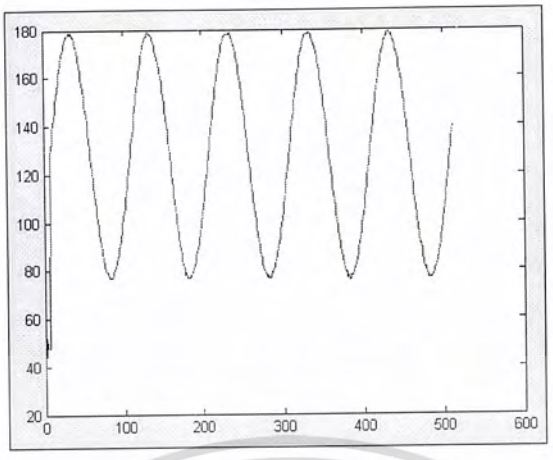
```
data =

TEKTRONIX,TDS 340A,0,CF:91.1CT FV:v1.09
```

รูปที่ 4.21 ผลการอ่านค่าข้อมูลของเครื่องวัดสัญญาณ

3) ผลการอ่านค่าสัญญาณที่เกิดขึ้นที่หน้าจอเครื่องวัดสัญญาณ แล้วมาทำการพล็อตกราฟ แสดงดังรูปที่ 4.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



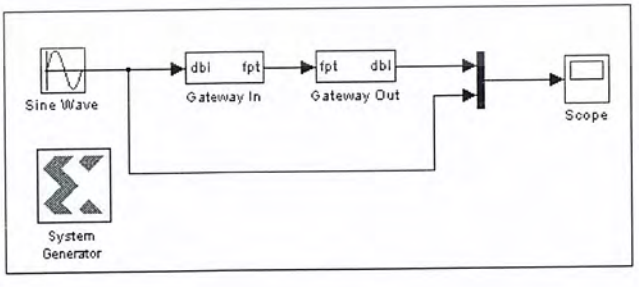
รูปที่ 4.22 ผลการพล็อตกราฟของสัญญาณที่รับมาจากเครื่องมือวัดสัญญาณ

จากผลการทดลองจะเห็นว่าสามารถติดต่อกับการ์ด GPIB และเครื่องวัดสัญญาณได้จริง และสามารถรับข้อมูลจากเครื่องวัดสัญญาณมาแสดงผลที่โปรแกรม MATLAB ได้อย่างถูกต้อง

4.9 การทดลองในใบงานที่ 8 เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น

4.9.1 ขั้นตอนการทดลอง

ออกแบบโมเดลเพื่อทดลองการใช้บล็อกของ Xilinx ร่วมกับบล็อกพื้นฐานของ Simulink โดยสร้างสัญญาณ Sine Wave ส่งผ่านบล็อก Gateway In เพื่อเปลี่ยนระบบเลขจำนวน Floating Point เป็น Fixed Point และผ่านบล็อก Gateway Out เพื่อเปลี่ยนระบบจำนวนเลขกลับเป็น Floating Point อีกครั้งและแสดงผลที่หน้าต่างสโคป โดยเปรียบเทียบสัญญาณระหว่าง สัญญาณที่ผ่านการเปลี่ยนระบบจำนวนกับสัญญาณที่สร้างจากบล็อกสโคป ลักษณะโมเดลแสดงดังรูปที่ 4.23

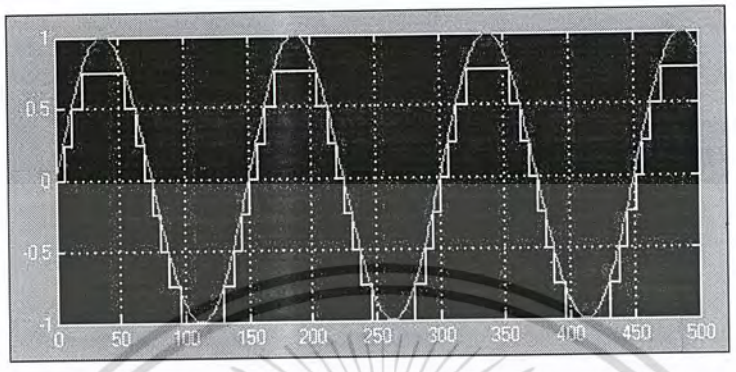


รูปที่ 4.23 โมเดลเพื่อทดสอบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9.2 ผลการทดลอง

เมื่อจำลองการทำงานของโมเดล ผลที่เกิดขึ้นที่หน้าต่างสโคป แสดงดังรูปที่ 4.24



รูปที่ 4.24 ผลการจำลองการทำงานที่หน้าต่างสโคป เพื่อทดสอบการใช้งานบล็อก Xilinx

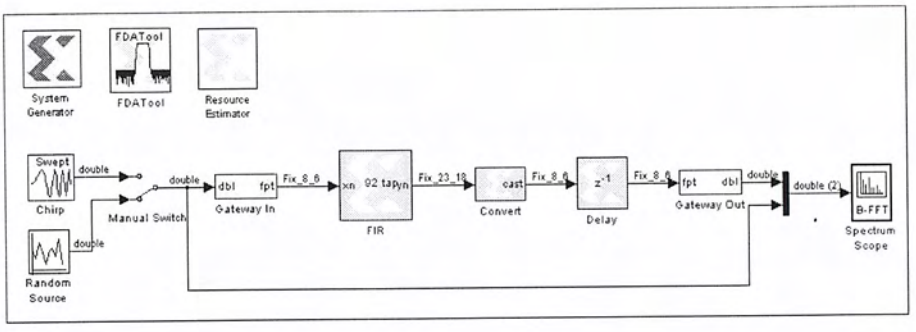
จากการทดลองจะเห็นว่าผลที่เกิดขึ้นที่หน้าต่างสโคป จะแสดงสัญญาณ Sine Wave ที่สร้างขึ้นและแสดงสัญญาณ Sine Wave ที่ผ่านการเปลี่ยนระบบจำนวนเลข ซึ่งสัญญาณที่ได้จะมีลักษณะที่หยาบกว่าสัญญาณเดิม เป็นผลมาจากความผิดพลาดในการจัดระดับสัญญาณของบล็อก Gateway In ดังนั้นแสดงว่าสามารถออกแบบโมเดลที่ใช้งานบล็อกของ Xilinx ร่วมกับบล็อกพื้นฐานของ Simulink ได้อย่างถูกต้อง

4.10 การทดลองในใบงานที่ 9 เรื่อง ออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator

4.10.1 ขั้นตอนการทดลอง

ออกแบบโมเดลวงจรกรองแถบความถี่ผ่าน โดยสร้างสัญญาณที่มีความถี่ทุกๆ ความถี่จากบล็อกพื้นฐานของ Simulink เพื่อเป็นอินพุตให้กับบล็อกต่างๆ ของ Xilinx ซึ่งออกแบบให้เป็นวงจรกรองแถบความถี่ผ่าน โดยแถบความถี่ที่ต้องการให้ผ่านอยู่ในช่วง 300KHz ถึง 450KHz สัญญาณที่ผ่านการกรองแล้วและสัญญาณอินพุตจะนำไปแสดงผลที่หน้าต่าง Spectrum Scope เพื่อเปรียบเทียบผลที่เกิดขึ้น หลังจากการจำลองการทำงานเสร็จสิ้นจะสร้างโค๊ดภาษา VHDL จากโมเดล และทดลองคำนวณขนาดพื้นที่ของโมเดลเมื่อโปรแกรมลงบนชิพ FPGA โดยลักษณะของโมเดลแสดงดังรูปที่ 4.25

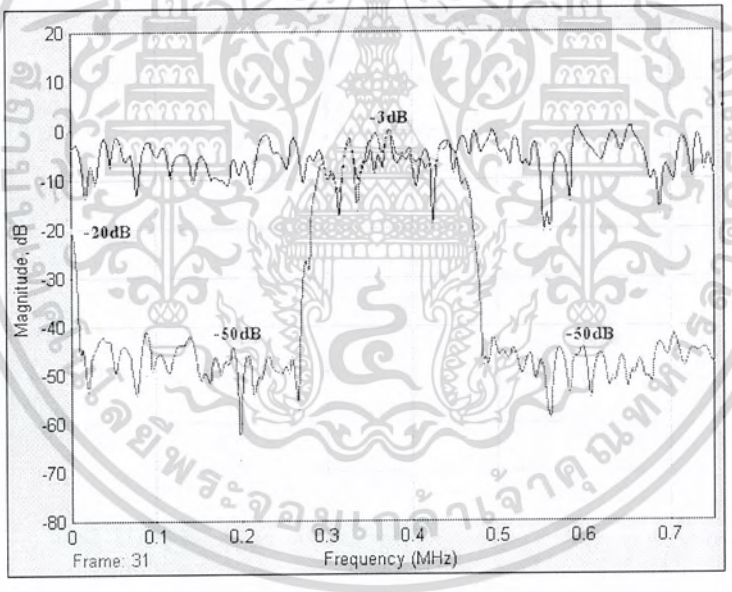
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 ลักษณะโมเดลวงจรกรองแถบความถี่ผ่าน

4.10.2 ผลการทดลอง

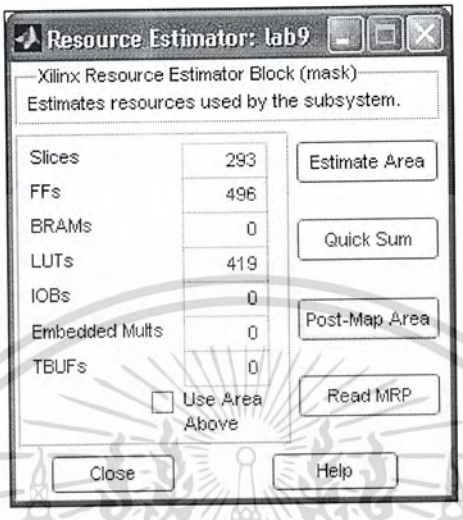
เมื่อทดลองจำลองการทำงานของโมเดลผลที่เกิดขึ้นที่หน้าต่าง Spectrum Scope แสดงดังรูปที่ 4.26



รูปที่ 4.26 ผลการจำลองการทำงานของวงจรกรองแถบความถี่ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองสร้างโค๊ดภาษา VHDL และคำนวณขนาดพื้นที่ของโมเดลเมื่อโปรแกรมลงบนชิพ FPGA แสดงดังรูปที่ 4.27



รูปที่ 4.27 ขนาดพื้นที่ของโมเดลวงจรกรองเมื่อโปรแกรมลงบนชิพ FPGA

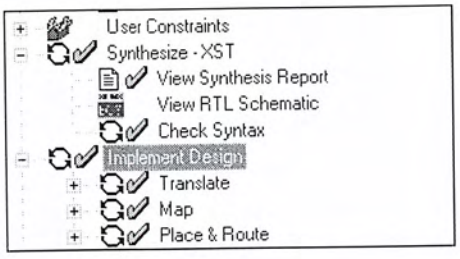
จากผลการจำลองการทำงานที่หน้าต่าง Spectrum Scope จะเห็นว่าสามารถออกแบบวงจรกรองแถบความถี่ผ่านจากบล็อกของ Xilinx ได้อย่างถูกต้องและสามารถที่จะสร้างโค๊ดภาษา VHDL และคำนวณขนาดพื้นที่ของโมเดลวงจรกรองเมื่อโปรแกรมลงบนชิพ FPGA ได้จริง

4.11 การทดลองในใบงานที่ 10 เรื่อง การใช้งานโปรแกรม Xilinx ISE Foundation ในการสังเคราะห์วงจร

4.11.1 ขั้นตอนการทดลอง

ใช้โปรแกรม Xilinx ISE Foundation นำโค๊ดภาษา VHDL ของวงจรกรองที่ออกแบบจากโปรแกรม MATLAB มาทำการสังเคราะห์วงจรและเชื่อมต่อสายสัญญาณบนชิพ FPGA เพื่อคำนวณขนาดพื้นที่ของวงจรที่ใช้โปรแกรมลงบนชิพ FPGA แสดงดังรูปที่ 4.28

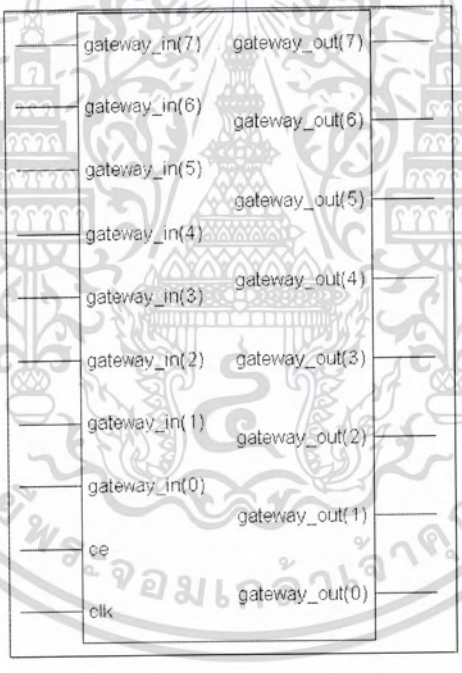
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 ขั้นตอนการสังเคราะห์วงจรและเชื่อมต่อสายสัญญาณ

4.11.2 ผลการทดลอง

เมื่อทดลองการสังเคราะห์วงจรและเชื่อมต่อสายสัญญาณ ผลที่เกิดขึ้นจะได้วงจรในระดับ RTL ซึ่งมีลักษณะเป็นบล็อกของวงจรกรอง แสดงดังรูปที่ 4.29



รูปที่ 4.29 ลักษณะของวงจรกรองที่ผ่านการสังเคราะห์วงจรแล้ว

เมื่อทดลองการเชื่อมต่อสายสัญญาณของวงจรกรองบนชิพ FPGA แล้วผลที่ได้จะเป็นรายงานขนาดพื้นที่ที่ใช้งานและความเร็วของวงจรที่สามารถใช้งานได้สูงสุด แสดงดังรูปที่ 4.30 และรูปที่ 4.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
22  Device utilization summary:
23
24     Number of External GCLKIOBs          1 out of 4      25%
25     Number of External IOBs             16 out of 158    10%
26        Number of LOCed External IOBs     0 out of 16      0%
27
28     Number of SLICEs                   293 out of 6912    4%
29
30     Number of GCLKs                      1 out of 4      25%
31
```

รูปที่ 4.30 ผลรายงานขนาดพื้นที่ที่ใช้งาน

```
712  Speed Grade: -6
713
714     Minimum period: 8.752ns (Maximum Frequency: 114.260MHz)
715     Minimum input arrival time before clock: 2.909ns
716     Maximum output required time after clock: 6.514ns
717     Maximum combinational path delay: No path found
```

รูปที่ 4.31 ผลรายงานความเร็วสูงสุดของวงจรที่สามารถใช้งานได้

4.12 สรุปผลการประเมินคุณภาพใบงานชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATAB โดยผู้ทรงคุณวุฒิ

ตารางที่ 4.3 สรุปผลการประเมินคุณภาพใบงานชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB

ด้านที่ประเมิน	ระดับความคิดเห็น			รวม	เฉลี่ย
	คนที่ 1	คนที่ 2	คนที่ 3		
1. ด้านวิชาการ					
1.1 เนื้อหามีความสอดคล้องกับวัตถุประสงค์	5	5	5	15	5.00
1.2 ความถูกต้องของภาษาที่ใช้	4	4	5	13	4.33
1.3 ความเหมาะสมในการจัดลำดับความสำคัญของเนื้อหา	5	5	5	15	5.00
1.4 ความถูกต้องของใบงานการทดลอง	5	5	5	15	5.00
1.5 ใบงานการทดลองสามารถนำไปใช้ในการเรียนการสอนได้	5	4	5	14	4.50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 (ต่อ) สรุปผลการประเมินคุณภาพใบงานชุดทดลองการปฏิบัติงานที่เวลาจริง ด้วย MATLAB

ด้านที่ประเมิน	ระดับความคิดเห็น			รวม	เฉลี่ย
	คนที่ 1	คนที่ 2	คนที่ 3		
1.6 มีความชัดเจนของขั้นตอนการทำงาน	4	5	5	14	4.50
2. ด้านสื่อการเรียนการสอน					
2.1 การวางรูปแบบของใบงานการทดลอง	5	4	5	14	4.50
2.2 ความเหมาะสมของภาพที่ประกอบใบงานการทดลอง	5	5	4	14	4.50
2.3 มีฮาร์ดแวร์ประกอบการทดลอง	4	5	5	14	4.50
รวม	**42**	**42**	**44**	**128**	**42.66**
เฉลี่ย	**4.66**	**4.66**	**4.88**	**14.22**	**4.74**

จากผลการประเมินใบงานการทดลองจากผู้ทรงคุณวุฒิด้านวิชาการ และด้านสื่อการเรียนการสอน ทั้งหมด 3 ท่าน จะเห็นได้ว่า ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB มีคะแนนเฉลี่ยเท่ากับ 4.74 ซึ่งแสดงว่าใบงานการทดลองมีคุณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุป

ปริญญานิพนธ์ฉบับนี้นำเสนอ ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB เนื้อหาประกอบด้วยใบงานการทดลองทั้งหมด 10 ใบงาน แบ่งเป็น 4 เรื่องหลักดังนี้คือ เรื่องที่ 1 จะเป็นการใช้งานโปรแกรม MATLAB และการใช้งาน Simulink เบื้องต้น เรื่องที่ 2 เป็นการใช้งานกล่องเครื่องมือ Real-Time Workshop และ xPC Target เรื่องที่ 3 เป็นการใช้งานโปรแกรม MATLAB ควบคุมเครื่องมือวัดผ่านพอร์ต GPIB และในเรื่องที่ 4 จะเป็นการใช้งานโปรแกรม MATAB ร่วมกับโปรแกรม System Generator

โดยเนื้อหาในแต่ละใบงานจะเรียงลำดับความสำคัญของเนื้อหาในใบงานและจะอธิบาย ทีละขั้นตอนการทดลองโดยละเอียดและมีฮาร์ดแวร์ประกอบการทดลองเพื่อเสริมความเข้าใจ จึงทำให้สามารถทำความเข้าใจได้ง่าย เหมาะสมกับนักศึกษาและผู้สนใจทั่วไป

5.2 ปัญหาและแนวทางแก้ไข

ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB ได้จัดทำขึ้นยังมีข้อบกพร่องบางประการ ซึ่งทางคณะผู้จัดทำได้รวบรวมปัญหาที่เกิดขึ้นและแนวทางในการพัฒนาโดยมีรายละเอียดดังต่อไปนี้

1. ปัญหา การรวบรวมข้อมูลและสร้างใบงานแต่ละใบนั้นมีความล่าช้า เนื่องจากข้อมูลการใช้งานยังไม่แพร่หลายมากนัก ประกอบกับขีดจำกัดทางด้านภาษาอังกฤษของคณะผู้จัดยังอยู่ในระดับต่ำ

แนวทางแก้ไข ศึกษาการใช้งานจากอินเตอร์เน็ตและคู่มือการใช้งานของโปรแกรมเป็นส่วนใหญ่และทดลองใช้งานจริงแล้วจึงสรุปเป็นวิธีการใช้งาน ตามความเข้าใจของคณะผู้จัดทำเอง

2. ปัญหา ขาดแคลนอุปกรณ์ในการแสดงการประยุกต์ใช้งานของ xPC Target เช่น การ์ดแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

แนวทางแก้ไข ใช้พอร์ตอนุกรม (RS-232) จำนวน 2 พอร์ตแสดงการรับและส่งข้อมูลแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนา

1. จัดหาฮาร์ดแวร์มาประกอบการทดลองในใบงาน xPC Target จะทำให้เห็นภาพการประยุกต์ใช้งานได้มากขึ้น เช่น การ์ดแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกหรือการ์ดแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล เป็นต้น

2. จัดหาเครื่องมือวัดสัญญาณมาทดลองเพิ่มเติมในใบงานการใช้งานโปรแกรม MATLAB ติดต่อกับเครื่องมือวัดสัญญาณ จะทำให้สามารถประยุกต์การใช้กว้างขึ้น เช่น เครื่องวิเคราะห์แถบความถี่ (Spectrum Analyzer) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

คมกฤษณ์ ชูเรือง, แวฮาซัน แวหะนะ และอนุชา หาญสมบัติเจริญ. "การควบคุมอุปกรณ์ GPIB ผ่านเครื่อข่ายอินเตอร์เน็ต." วิทยานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541

จุมพล เยรบุตร และอรนุช สัมฤทธ์ . "การควบคุมเครื่องมือวัดผ่านเครือข่ายอินเตอร์เน็ตโดยพอร์ต GPIB" ปริญญานิพนธ์ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2542

มนัส สังวรศิลป์ และวรรัตน์ ภัทรอมรกุล. คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์. กรุงเทพฯ: สำนักพิมพ์ อินโฟเพรส. 2543

Tektronix Inc. **AFG310 and AFG320 User Manual.** [Online]. Available: http://e-sites2.tek.com/pls/vignmn/tk_mn_download.main. n.d

Tektronix Inc. **TDS340A, TDS360, & TDS380 Programmer Manual.** [Online]. Available: http://e-sites2.tek.com/pls/vignmn/tk_mn_download.main. n.d

Xilinx Inc. **Desiging a FIR Filter.** [Online]. Available : http://www.xilinx.com/univ/xup/labsfnd/downld.htm 2004

Xilinx Inc. **Creating a 12 x 8 MAC Using the Xilinx System Generator.** [Online]. Available : http://www.xilinx.com/univ/xup/labsfnd/downld.htm 2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 1
เรื่อง การใช้งานโปรแกรม MATLAB เบื้องต้น

วัตถุประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. แสดงการใช้งานโปรแกรม MATLAB ในการคำนวณเบื้องต้นได้อย่างถูกต้อง
2. แสดงการเรียกใช้งาน Help และ Demo ได้อย่างถูกต้อง
3. แสดงการตรวจสอบค่าข้อมูลโดยใช้คำสั่ง who และ whos ได้อย่างถูกต้อง
4. เขียนโปรแกรมในลักษณะ M-file ได้อย่างถูกต้อง
5. แสดงการพล็อตกราฟของข้อมูลโดยใช้คำสั่ง plot ได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0

ความรู้และทักษะเดิม

1. การใช้งานคอมพิวเตอร์เบื้องต้น

ทฤษฎีเบื้องต้น

MATLAB เป็นภาษาคอมพิวเตอร์ชั้นสูง (High-level language) ใช้สำหรับการคำนวณด้านวิศวกรรมที่ประกอบด้วยการคำนวณเชิงตัวเลข กราฟิกที่ซับซ้อนและการจำลองแบบเพื่อให้มองเห็นภาพได้ง่ายและชัดเจน ชื่อของ MATLAB ย่อมาจาก MATrix LABoratory โดยพัฒนาโดยบริษัท Mathworks.Inc (http://www.mathwork.com/) ซึ่งในปัจจุบันได้พัฒนาถึงเวอร์ชั่น 7.2 (April-2005) โดยโปรแกรม MATLAB นิยมใช้สำหรับการคำนวณทางคณิตศาสตร์ เพราะมีฟังก์ชั่นทางคณิตศาสตร์ให้เลือกใช้อย่างมากมาย สามารถพัฒนาอัลกอริทึมต่างๆ และใช้ประมวลผลร่วมกับโปรแกรมอื่นได้เช่น FORTRAN, Borland C/C++, Microsoft Visual C++ เป็นต้น และยังนำไปใช้งานทางด้านกราฟิกได้เป็นอย่างดี

การเขียนโปรแกรม MATLAB เบื้องต้นในการคำนวณคณิตศาสตร์พื้นฐานจะต้องมีความรู้ทางเครื่องหมายทางคณิตศาสตร์ เช่น ตัวดำเนินการดังตารางที่ 1.1 ตัวดำเนินการเปรียบเทียบและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรรกะดังตารางที่ 1.2 และเครื่องหมายพิเศษดังตารางที่ 1.3 เพื่อนำมาใช้ในการคำนวณเพื่ออำนวยความสะดวกในการเขียนโปรแกรม

ตารางที่ 1.1 ตัวดำเนินการของโปรแกรม MATLAB

ลักษณะดำเนินงาน	ตัวดำเนินการ	รูปแบบของ MATLAB
การบวก	+	A+B
การลบ	-	A-B
การคูณ	*	A*B
การคูณเชิงสมาชิก	.*	A.*B
การหารทางขวา	/	A/B
การหารทางซ้าย	\	A\B
การหารเชิงสมาชิก	./	A./B
การยกกำลัง	^	A^b
การยกกำลังเชิงสมาชิก	.^	A.^b

ตารางที่ 1.2 ตัวดำเนินการเปรียบเทียบและตรรกะของโปรแกรม MATLAB

ลักษณะการดำเนินงาน	ตัวดำเนินการเปรียบเทียบทางตรรกะ	ตัวอย่างรูปแบบการใช้งาน
น้อยกว่า	<	x<10
น้อยกว่าหรือเท่ากับ	<=	x<=10
มากกว่า	>	x>10
มากกว่าหรือเท่ากับ	>=	x>=10
เท่ากับ	= =	x = = 1
ไม่เท่ากับ	~ =	x ~=5
และ	&	x>2 & y<1
หรือ	|	x>2 | y<1
ไม่	~	~x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

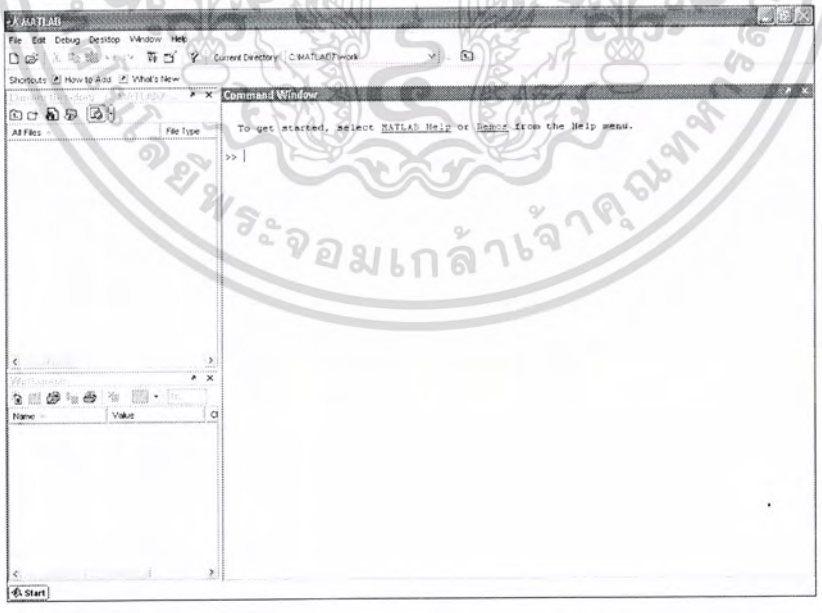
ตารางที่ 1.3 เครื่องหมายพิเศษของโปรแกรม MATLAB

เครื่องหมาย	รายละเอียด
.	จุดทศนิยม
()	กำหนด subscripts
=	กำหนดค่า
[]	สร้างเวกเตอร์และเมตริกซ์
:	สร้างเวกเตอร์
...	กระทำคำสั่งยังบรรทัดต่อไป
,	แยก element ภายในเมตริกซ์และ subscripts

ลำดับขั้นการทดลอง

1. เปิดโปรแกรม MATLAB เวอร์ชั่น 7.0 โดยดับเบิ้ลคลิกที่ไอคอน

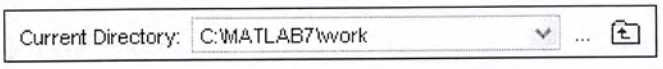
MATLAB เวอร์ชั่น 7.0 บนหน้าจอจะปรากฏหน้าต่าง MATLAB และที่หน้าต่าง Command Window จะพบเครื่องหมาย MATLAB prompt (>>) เครื่องหมายนี้แสดงว่าพร้อมที่จะรอรับคำสั่งต่างๆ เพื่อการคำนวณและประมวลผลโดยทุกคำสั่งจะป้อนเข้าไปหลังจากเครื่องหมายนี้ แสดงดังรูปที่ 1.1



รูปที่ 1.1 หน้าต่างโปรแกรม MATLAB เวอร์ชั่น 7.0

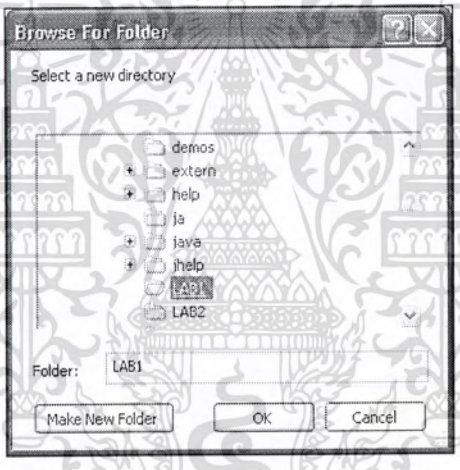
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เลือก Current Directory ที่ใช้งาน โดยที่ Current Directory ซึ่งเป็นโฟลเดอร์ที่เก็บข้อมูลขณะที่ใช้งาน โดยโฟลเดอร์เริ่มต้นของโปรแกรมจะกำหนดเป็น **C:\MATLAB7\work** แสดงดังรูปที่ 1.2



รูปที่ 1.2 โฟลเดอร์เริ่มต้นของโปรแกรม

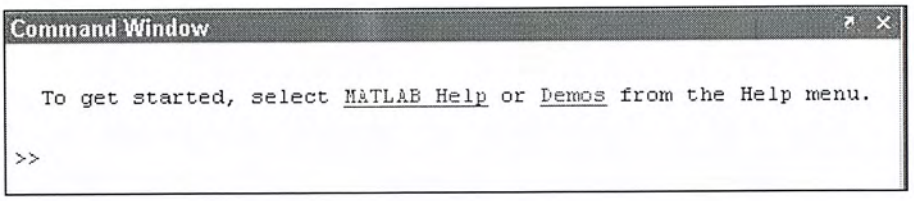
3. คลิกที่ปุ่ม ... เลือกไดเร็กทอรี่ที่ใช้งานเป็น **C:\MATLAB7\LAB1** ถ้าไม่มีโฟลเดอร์ LAB1 ให้สร้างขึ้นมาใหม่ แสดงดังรูปที่ 1.3



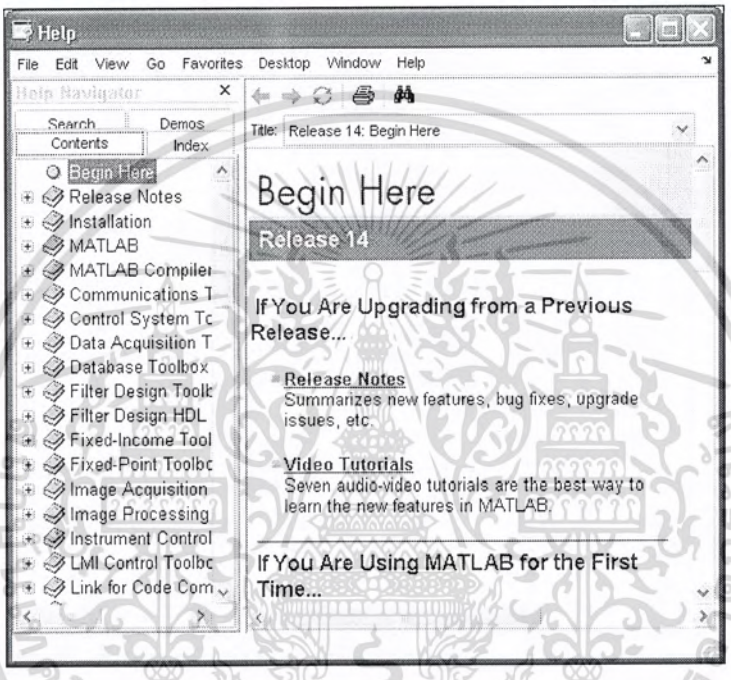
รูปที่ 1.3 การเลือกไดเร็กทอรี่ที่ใช้งานเป็น C:\MATLAB7\LAB1

4. เมื่อผู้ที่เริ่มต้นศึกษาใช้โปรแกรม MATLAB สามารถเรียกการใช้งาน Help และ Demos ได้โดยพิมพ์คำสั่งว่า **>>demo** และ **>>help** หรือที่หน้าต่าง Command Window แสดงดังรูปที่ 1.4 คลิกที่ MATLAB Help และ Demos จะปรากฏหน้าต่างการใช้งานโปรแกรมและตัวอย่างการประยุกต์ใช้งานต่างๆ มากมาย ดังรูปที่ 1.5 และรูปที่ 1.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

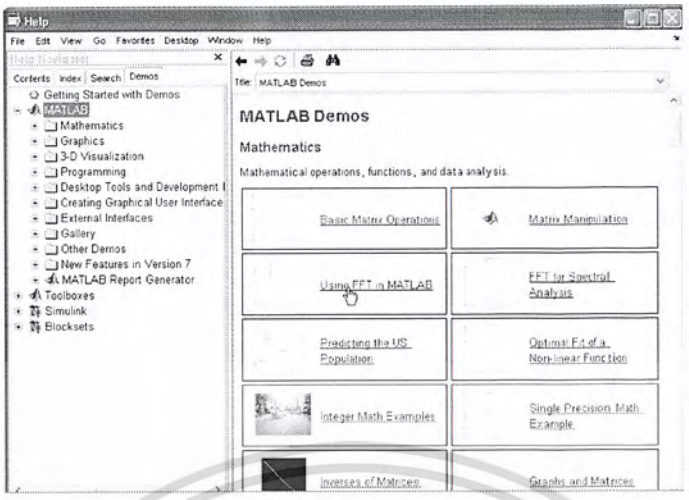


รูปที่ 1.4 การเรียกใช้งาน MATLAB Help และ Demos



รูปที่ 1.5 หน้าต่างการใช้งานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.6 หน้าต่างตัวอย่างการประยุกต์ใช้งาน

5. ขั้นตอนต่อไปจะเป็นการเขียนโปรแกรมโดยสามารถพิมพ์คำสั่งหลังเครื่องหมาย >> แล้วกดปุ่ม Enter ก็จะแสดงผลลัพธ์ออกมา แต่ถ้าไม่ต้องการให้แสดงผลลัพธ์ก็ให้พิมพ์เครื่องหมายเซมิ-โคลอน (;) ต่อท้ายบรรทัดคำสั่ง ให้พิมพ์คำสั่งดังนี้

>> (2+3)*(9-5)

>> (2+3)*(9-5);

บันทึกผลการทดลองและอธิบายความหมายของการใช้เครื่องหมายเซมิโคลอน (;)

.....

.....

.....

6. ในการทำงานของโปรแกรม MATLAB นั้นค่าข้อมูลต่างๆ ที่เกิดขึ้นซึ่งก็คือ ผลลัพธ์จากการคำนวณจะเก็บอยู่ที่หน้าต่าง MATLAB Workspace โดยสามารถที่จะตรวจสอบชื่อ ขนาด และกลุ่มของตัวแปร ได้โดยใช้คำสั่ง who และ whos ทดลองพิมพ์คำสั่งดังนี้

>> whos บันทึกผลการทดลอง

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

>> **who** บันทึกผลการทดลอง

.....

.....

.....

7. การลบข้อมูลที่อยู่ใน MATLAB Workspace นั้นสามารถทำได้โดยใช้คำสั่ง clear ทดลองพิมพ์คำสั่ง >> **clear** แล้วพิมพ์คำสั่ง >> **whos** อีกครั้ง สังเกตผลที่เกิดขึ้น

8. เขียนโปรแกรมคำนวณหาผลลัพธ์จากโจทย์ในตารางที่ 1.5 และบันทึกผลการทดลอง

ตารางที่ 1.5 การคำนวณจำนวนเลขโดยใช้โปรแกรม MATLAB

โจทย์	ผลลัพธ์
(2*5)+(3/9)	
((2/8)+9)+15	
(13*8)-(152/3)	
((3^6)+5^3)/8	
10*sin(100)	

9. การใช้งานฟังก์ชั่นในการคำนวณนั้นถ้าไม่ทราบลักษณะการใช้สามารถใช้คำสั่ง help แล้วตามด้วย ฟังก์ชั่นที่ต้องการทราบ ทดลองพิมพ์คำสั่ง >> **help sin** และ >> **help sind** จงอธิบายการใช้งานและความแตกต่างของฟังก์ชั่นทั้งสอง บันทึกผลการทดลอง

.....

.....

.....

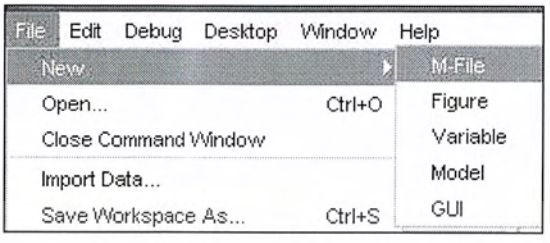
.....

.....

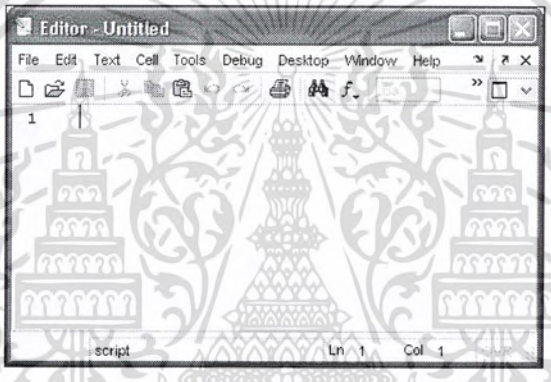
10. การสร้าง M-file เป็นการเขียนโปรแกรมที่จะเก็บไว้เป็นไฟล์ที่มีนามสกุลเป็น .m เป็นไฟล์เฉพาะของโปรแกรม MATLAB การใช้งานเพียงแต่เรียกชื่อไฟล์ขึ้นมาเท่านั้น M-file ก็จะทำงานตามคำสั่งที่เขียนไว้ เพื่อความสะดวกแก่ผู้ใช้งานไม่ต้องป้อนคำสั่งทีละคำ ขั้นตอนการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

M-file มีดังนี้ คลิกที่เมนู **File > New > M-file** จะปรากฏหน้าต่าง MATLAB Editor แสดงดังรูปที่ 1.7 และรูปที่ 1.8



รูปที่ 1.7 การสร้าง M-file



รูปที่ 1.8 หน้าต่าง Editor

11. เขียนโปรแกรมแสดงดังรูปที่ 1.9 และบันทึกชื่อเป็น **lab1.m**

```
%Test M-file LAB1
%ans = lab1(x,y)
%ans = x*y;

function ans = lab1(x,y)

ans = x*y;

end
```

รูปที่ 1.9 ตัวอย่างการเขียน M-file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ที่ Command Window พิมพ์คำสั่ง >> **help lab1** บันทึกผลการทดลอง

.....

.....

.....

13. พิมพ์คำสั่ง >> **z = lab1(5,6)** บันทึกผลการทดลอง

.....

.....

.....

14. ทดลองหาค่าผลลัพธ์ต่อไปนี้โดยใช้ฟังก์ชั่น lab1.m ที่สร้างขึ้นบันทึกผลการทดลองในตารางที่ 1.6 โดยลักษณะการใช้งานฟังก์ชั่น lab1.m นั้นจะเป็นการคำนวณหาค่าผลคูณของเลข 2 จำนวน มีรูปแบบการใช้คือ

ans = lab1(x,y)

โดยค่า ans = x*y

ตารางที่ 1.6 การทดสอบการคำนวณของฟังก์ชั่น lab1.m

โจทย์		ผลลัพธ์
x	y	ans
5	41	
6	22	
13	35	
158	77	

15. ทดลองสร้างกราฟจากข้อมูลโดยใช้คำสั่ง plot พิมพ์คำสั่งดังนี้

>> t=[0:0.1:10]; สร้างจำนวนที่มีค่าตั้งแต่ 0 ถึง 10 โดยเพิ่มที่ละ 0.1 เก็บไว้ในตัวแปรชื่อ t

>> y=[0;0.1:10];สร้างจำนวนที่มีค่าตั้งแต่ 0 ถึง 10 โดยเพิ่มที่ละ 0.1 เก็บไว้ในตัวแปรชื่อ y

>> z=cos(y); นำค่า y มาสร้างเป็นสัญญาณ Sin เก็บค่าไว้ในตัวแปรชื่อ z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

>> plot(t,y) นำค่า t และค่า z มาพล็อตกราฟ

บันทึกผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. การเรียกใช้งาน help และ Demos มีประโยชน์อย่างไร?

.....

.....

.....

.....

.....

2. คำสั่ง who และ whos แตกต่างกันอย่างไร?

.....

.....

.....

.....

.....

3. จงอธิบายความหมายและการใช้งานของฟังก์ชั่น Cosd ?

.....

.....

.....

.....

.....

4. จงเขียน M-file โดยสร้างฟังก์ชั่นขึ้นเองบันทึกชื่อเป็น lab1_ans.m และบันทึกผลการสร้าง M-file?

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 2
เรื่อง การใช้งาน Simulink เบื้องต้น

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. อธิบายข้อดีของการจำลองระบบด้วย Simulink ได้อย่างถูกต้อง
2. แสดงขั้นตอนการจำลองระบบโดยใช้ Simulink ได้อย่างถูกต้อง
3. ประยุกต์การใช้งาน Simulink ในการพัฒนาระบบในลักษณะแบบจำลองได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0
2. กล่องเครื่องมือ Simulink เวอร์ชั่น 6.0

ความรู้และทักษะเดิม

1. การใช้งานคอมพิวเตอร์เบื้องต้น

ทฤษฎีเบื้องต้น

การศึกษาและพัฒนางานทุกด้าน จำเป็นต้องมีเครื่องมือและอุปกรณ์ต่างๆ ใช้ทดสอบกระบวนการทำงาน เพื่อการตัดสินใจว่าควรจะต้องปรับปรุงส่วนใดของระบบหรือเพื่อใช้ศึกษาทำความเข้าใจให้เห็นภาพของกระบวนการทำงานที่ชัดเจนขึ้น ซึ่งการใช้เครื่องมือและอุปกรณ์ต่างๆ บางครั้งอาจมีอุปกรณ์ต่างๆ บางครั้งอาจมีอุปกรณ์ไม่พอเพียงต่อความต้องการหรือเครื่องมืออุปกรณ์มีมูลค่าสูงมาก จึงมีความจำเป็นต้องทำการทดลองทดสอบก่อนการทำงานจริงเพื่อความไม่สิ้นเปลืองและเพิ่มประสิทธิภาพสำหรับการทำงาน

การจำลองการทำงานของระบบ (Simulation) เป็นทางเลือกที่นิยม สำหรับการทดลองและทดสอบ เพื่อการพัฒนาทางด้านต่างๆ เนื่องจากเป็นการจำลองการทำงานบนเครื่องคอมพิวเตอร์จึงสามารถปรับปรุง พัฒนา ทดสอบการทำงาน และเก็บข้อมูลของการทำงานได้อย่างง่ายและรวดเร็ว ซึ่งในใบงานการทดลองนี้เป็นการทดลองจำลองการทำงานของโมเดลโดยใช้กล่องเครื่องมือ Simulink ของโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มต้นใช้งาน Simulink โปรแกรม MATLAB จะกำหนดชื่อของโมเดลในหน้าต่าง Simulink โดยอัตโนมัติเป็น Untitled หน้าต่าง Simulink นี้จะทำงานเชื่อมต่อกับหน้าต่างคำสั่งของโปรแกรม MATLAB โดย Simulink สามารถรับ/ส่งข้อมูลผ่าน Workspace ได้และการสร้างบล็อกในหมวดหมู่ต่างๆ ที่ไม่ได้กำหนดค่าตัวเลขแต่กำหนดเป็นค่าตัวแปรในหน้าต่าง Simulink ก็สามารถส่งข้อมูลให้กับ Workspace ได้เช่นกัน โดยเมื่อบันทึกระบบที่จำลองไว้ด้วย Simulink โปรแกรม MATLAB จะกำหนดไฟล์ที่สร้างจาก Simulink เป็นชื่อไฟล์ที่มีนามสกุล .mdl

ส่วนประกอบต่างๆ บนหน้าต่าง Library Simulink

เมื่อเริ่มใช้งาน Simulink จะพบหน้าต่าง Library Simulink ที่ประกอบด้วย Block diagram ที่เป็นหมวดหมู่ของอุปกรณ์ต่างๆ ที่จะนำไปใช้ในการจำลองระบบซึ่งอุปกรณ์เหล่านี้ไม่สามารถทำงานได้ด้วยตัวเอง แต่จะต้องเชื่อมต่อกับบล็อกอื่นๆ ถึงจะทำงานได้ การเลือกใช้งานหมวดหมู่ใดหมวดหมู่หนึ่งในหน้าต่าง Library Simulink ให้คลิกไปที่ไอคอนของหมวดหมู่นั้นแล้วจะปรากฏหน้าต่างแสดงรายละเอียดของ Block ต่างๆ บนหน้าต่าง Library Simulink ของหมวดหมู่ที่เลือกไว้ออกมา

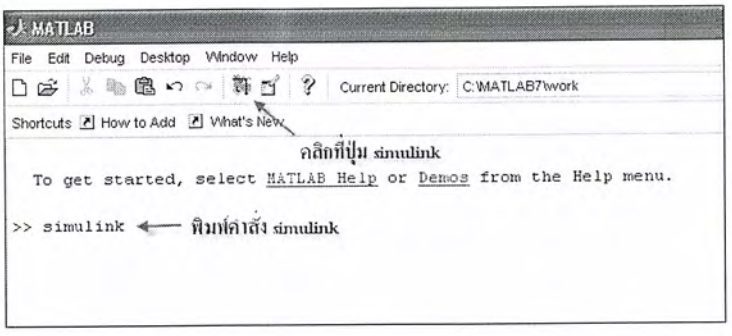
การใช้งานบล็อกของ Simulink

การเลือกใช้อุปกรณ์หรือ Block ที่อยู่ในหน้าต่าง Library Simulink ของหมวดหมู่นั้นสามารถกระทำได้โดยการเลื่อนเมาส์ไปยังไอคอน (Icon) ของอุปกรณ์ที่ต้องการเลือกใช้แล้วกดเมาส์ปุ่มซ้ายค้างไว้ลาก (Drag) รูปไอคอนนั้นมายังหน้าต่างทำงานของ Simulink โดยมีชื่อ Default เป็น Untitled (โปรแกรม MATLAB จะกำหนดชื่อนี้ไว้โดยอัตโนมัติก่อนที่จะ Save และเมื่อกำหนดชื่ออื่นแล้ว Save ก็จะเป็นชื่อไฟล์ที่ตั้งเอาไว้) เมื่อเลือก Block ที่ต้องการใช้งานแล้วสามารถทำการปิดหน้าต่าง Library Simulink ของ Block ที่เลือกโดยการเลื่อนเมาส์ไปยังเมนู File แล้วเลือกคำสั่ง Close

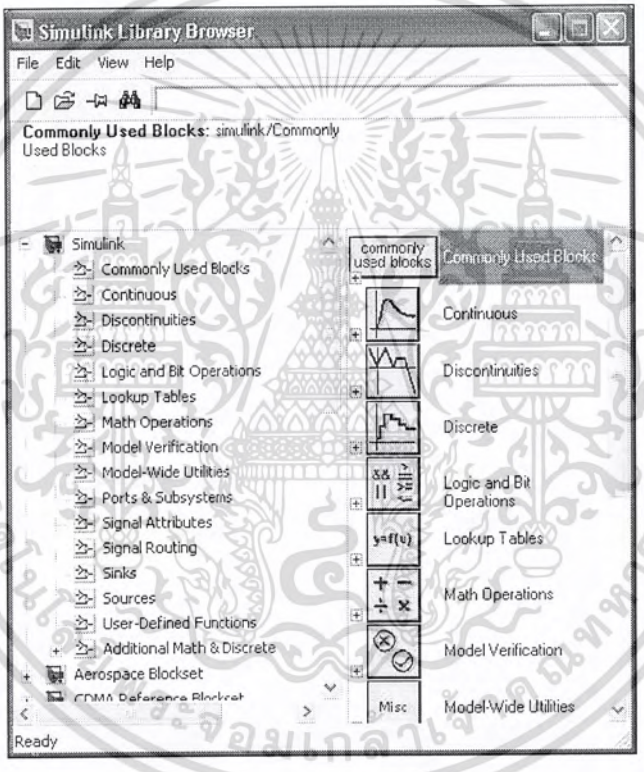
ลำดับขั้นการทดลอง

1. ทำการเปิดโปรแกรม MATLAB เวอร์ชั่น 7.0 เพื่อทำการเลือกไดเร็กทอรี่ที่ใช้งานเป็น **C:\MATLAB7\LAB2** จากนั้นพิมพ์คำสั่ง **>>Simulink** หรือคลิกที่ แสดงดังรูปที่ 2.1 และแสดงหน้าต่าง Simulink Library Browser ในรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



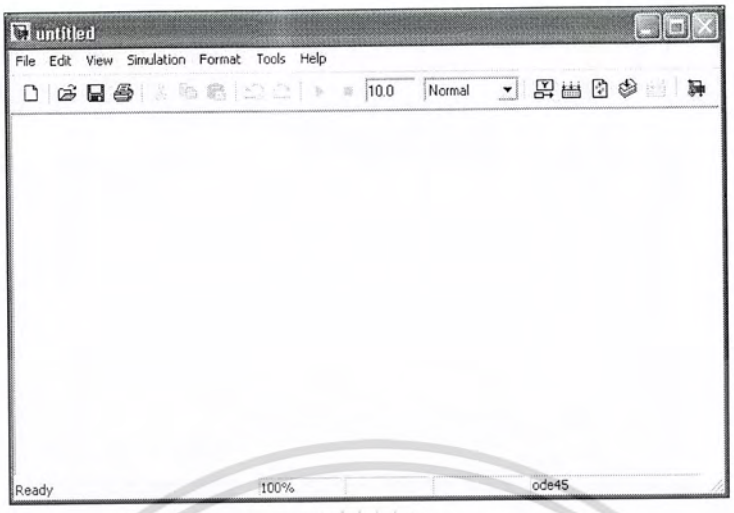
รูปที่ 2.1 การเรียกใช้งาน Simulink Library Browser



รูปที่ 2.2 หน้าต่าง Simulink Library Browser

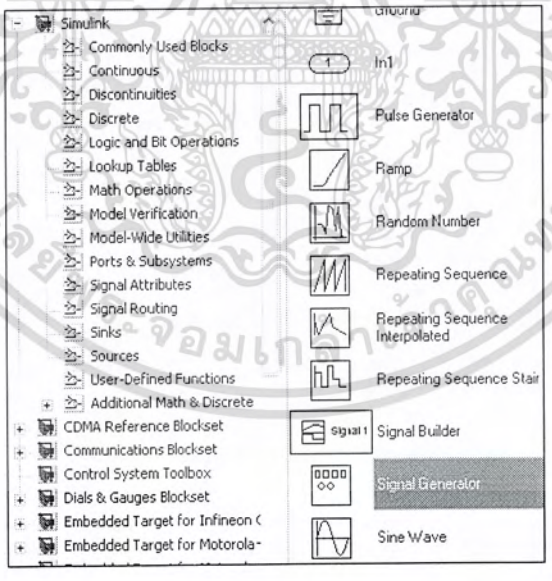
2. ทำการสร้างโมเดลใหม่โดยใช้ Simulink จะกำหนดชื่อของโมเดลโดยอัตโนมัติเป็น Untitled โดยสามารถจะคลิกที่ปุ่มไอคอน หรือคลิกที่บนเมนูบาร์ **File > New > Model** โดยจะปรากฏหน้าต่างดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 หน้าต่างโมเดลใหม่ชื่อ Untitled

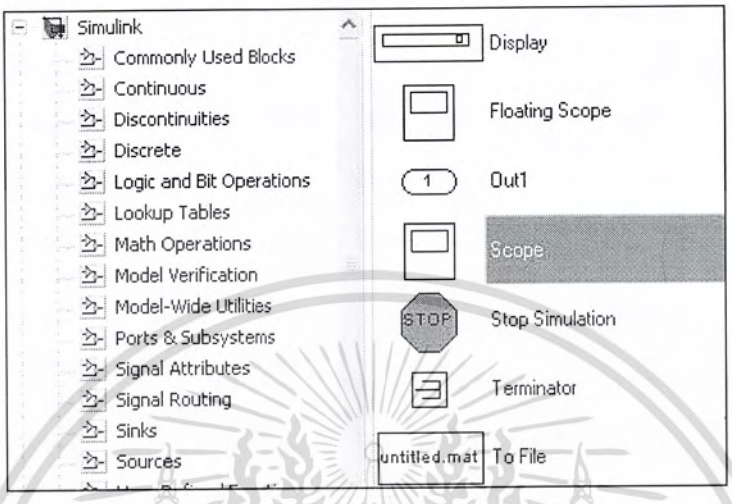
3. ทำการสร้างบล็อกในโมเดล คลิกที่หัวข้อ **Source** เลือกบล็อก **Signal Generator** โดยคลิกเมาส์ซ้ายค้างไว้บน Block แล้วลาก (Drag) บล็อกไปยังบริเวณหน้าต่าง Untitled แสดงดังรูปที่ 2.4



รูปที่ 2.4 การสร้างบล็อก Signal Generator

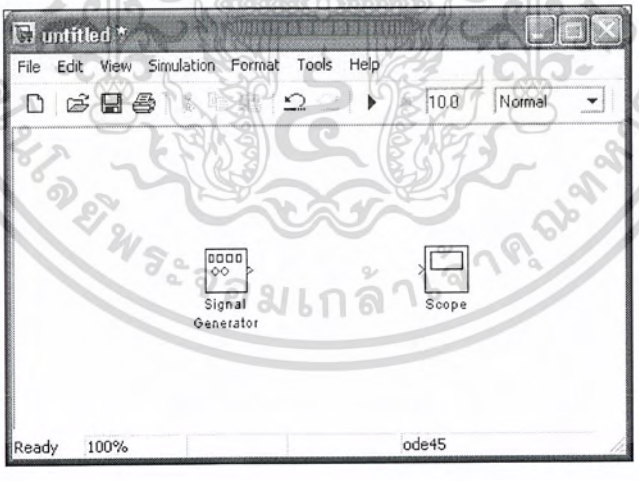
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คลิกที่หัวข้อ **Source** เลือกบล็อก **Scope** โดยคลิกเมาส์ซ้ายค้างไว้บน Block แล้วลาก (Drag) บล็อกไปยังหน้าต่าง Untitled แสดงดังรูปที่ 2.5



รูปที่ 2.5 การสร้างบล็อก Scope

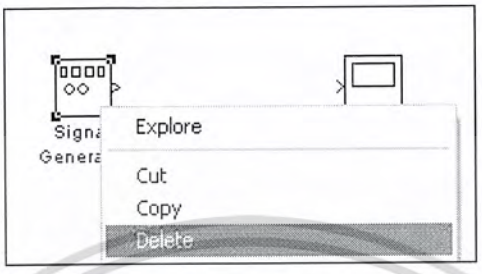
5. ผลที่ได้จากการสร้างบล็อก **Signal Generator** และบล็อก **Scope** แสดงดังรูปที่ 2.6



รูปที่ 2.6 การจัดวางตำแหน่งของบล็อก Signal Generator และ Scope

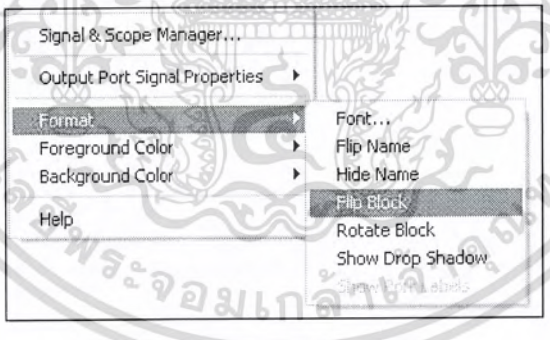
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การลบบล็อกนั้นทำได้โดยใช้เมาส์คลิกที่บล็อกนั้น แล้วกดปุ่ม **Delete** บนคีย์บอร์ดหรือใช้เมาส์คลิกที่บล็อกแล้วคลิกขวาแล้วเลือกเมนู Delete ดังรูปที่ 2.7 (เมื่อทดลองลบบล็อกแล้วให้-สร้างบล็อกใหม่ดังเดิม)



รูปที่ 2.7 วิธีการลบบล็อก

7. การกลับบล็อกให้อยู่ในลักษณะที่ต้องการทำได้โดยการใช้เมาส์คลิกขวาบน บล็อกที่ต้องการกลับเลือกเมนู **Format > Flib Block** หรือคลิกที่บล็อกแล้วใช้คีย์ลัด **Ctrl+I** แต่ถ้าต้องการกลับหมุนบล็อกทำได้ โดยเลือกเมนู **Format > Rotate Block** หรือคลิกที่บล็อกแล้วใช้คีย์ลัด **Ctrl+R** แสดงดังรูปที่ 2.8



รูปที่ 2.8 วิธีการเปลี่ยนตำแหน่งของบล็อก

คลิกที่บล็อก Signal Generator ทดลอง **Flib Block** และ **Rotate Block** อธิบายความแตกต่างของคำสั่งทั้งสอง

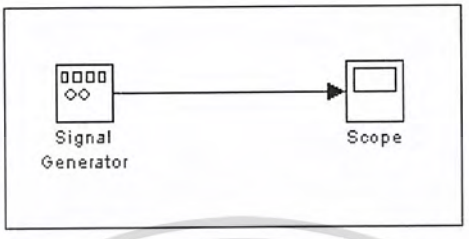
.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

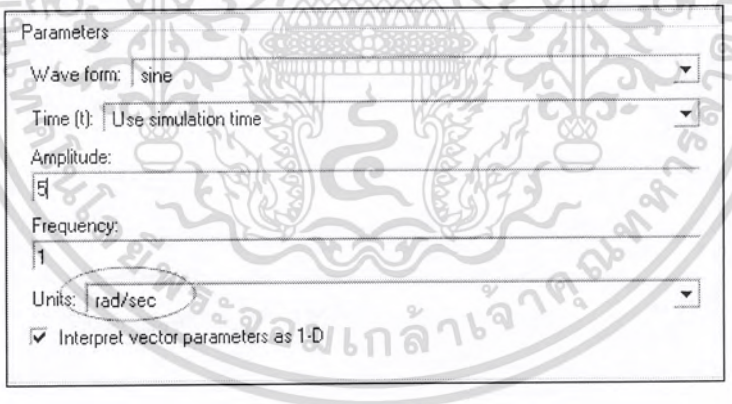
8. ทำการเชื่อมโยงบล็อกแต่ละบล็อกเข้าด้วยกัน โดยทำการคลิกเมาส์ซ้ายค้างแล้วลากเมาส์จาก Output Port ไปยัง Input Port ของบล็อกทั้งสอง เช่น การลากเมาส์จากบล็อก Signal Generator ไปยังบล็อก Scope แสดงดังรูปที่ 2.9



รูปที่ 2.9 การเชื่อมต่อบล็อก Signal Generator กับบล็อก Scope

9. กำหนดค่าพารามิเตอร์ของบล็อก Signal Generator โดยดับเบิ้ลคลิกที่บล็อก Signal Generator ซึ่งจะปรากฏหน้าต่างกำหนดค่าขึ้นมาแสดงดังรูปที่ 2.10 กำหนดค่าบล็อกต่างๆ ดังนี้

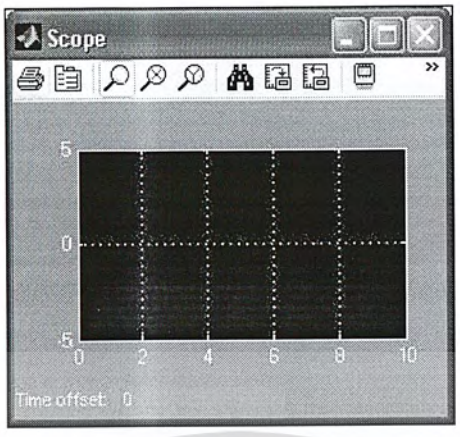
- กำหนดค่า **Amplitude** ของ Sine Wave มีค่าเท่ากับ **5**
- กำหนดค่า **Frequency** ของ Sine Wave มีค่าเท่ากับ **1 Rad/s**



รูปที่ 2.10 การกำหนดค่าพารามิเตอร์ของบล็อก Signal Generator

10. กำหนดค่าพารามิเตอร์ของบล็อก Scope โดยดับเบิ้ลคลิกที่บล็อก Scope ซึ่งจะปรากฏหน้าต่างกราฟของ Scope ขึ้นมาแสดงดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

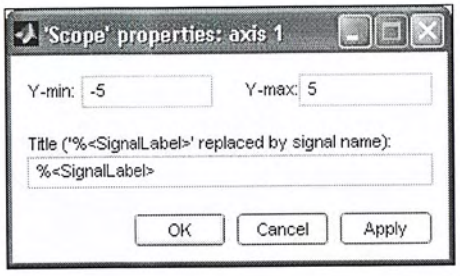


รูปที่ 2.11 หน้าต่างกราฟของ Scope

- **คลิกขวาที่บริเวณกราฟของ** Scope จะพบเมนูคำสั่งเลือก **Axes Properties** จะปรากฏหน้าต่าง Scope Properties กำหนดค่าดังนี้ **Ymin = -5** และ **Ymax = 5** จะแสดงดังรูปที่ 2.12 และรูปที่ 2.13

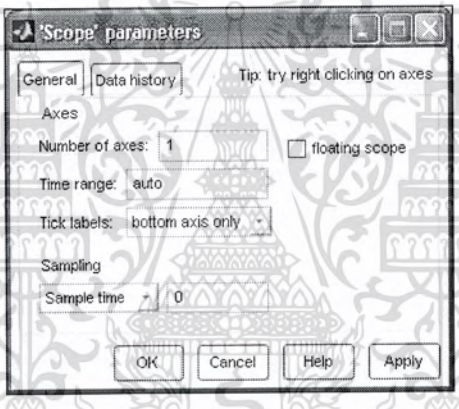
รูปที่ 2.12 กำหนดค่า Axes Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การกำหนดค่า Y-min และ Y-max

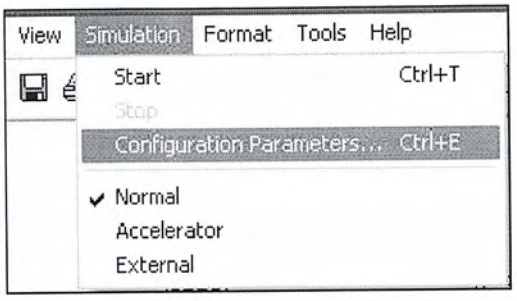
11. คลิกที่ไอคอน บนหน้าต่าง Scope จะปรากฏหน้าต่าง Scope Properties ขึ้นมาโดยกำหนด **Time range** เป็น **auto** และ **Sample Time = 0** ดังรูปที่ 2.14



รูปที่ 2.14 หน้าต่าง Scope Properties

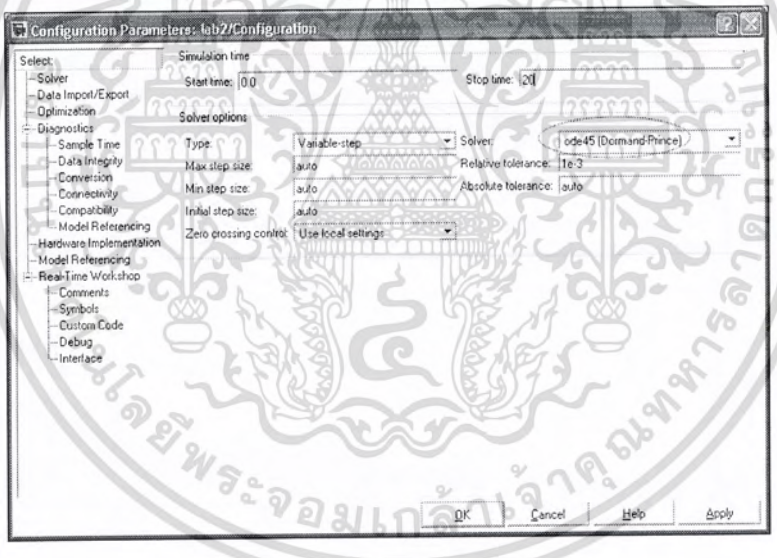
12. **บันทึกโมเดลตั้งชื่อเป็น lab2.mdl** ต่อไปจะเป็นการกำหนดวิธีการประมวลผลของโมเดล เวลาการทำงานเริ่มต้นและเวลาสิ้นสุด โดยที่เมนูบาร์ด้านบนคลิกที่ **Simulation > Configuration Parameters** แสดงดังรูปที่ 2.15 จะปรากฏหน้าต่าง Configuration Parameters ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 การเปิดหน้าต่าง Configuration Parameters

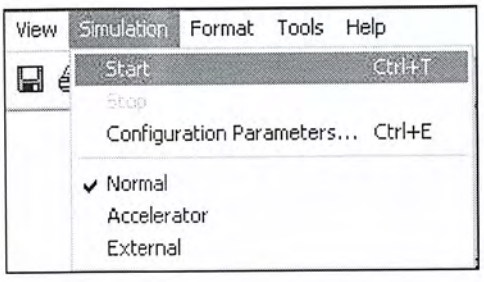
13. ที่หน้าต่าง Configuration Parameters กำหนดเวลาเริ่มต้น (Start Time) เท่ากับ **0** เวลาสิ้นสุด (Stop Time) เท่ากับ **20** และกำหนดวิธีในการแก้ปัญหาคือ **ODE45 (Dormand-Prince)** แล้วคลิกที่ปุ่ม OK ดังรูปที่ 2.16



รูปที่ 2.16 หน้าต่าง Simulation Parameters

14. กลับมาที่หน้าต่าง **lab2.mdl** คลิกที่ปุ่ม ▶ หรือคลิกที่เมนู **Simulation > Start** เพื่อจำลองการทำงาน แสดงดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 วิธีการจำลองการทำงาน

15. ดับเบิ้ลคลิกที่บล็อก Scope จะแสดงเอาต์พุตของระบบที่หน้าต่าง Scope จะเห็นว่าเป็นสัญญาณลักษณะ Sine Wave ที่มีค่าความถี่และแอมพลิจูด เท่ากับค่าที่กำหนดจากบล็อก Sine Wave แสดงดังรูปที่ 2.18 (ถ้าสัญญาณไม่แสดงดังรูปให้คลิกขวาที่กราฟ เลือก **Autoscale**) .

รูปที่ 2.18 ผลการกำเนิดสัญญาณ Sine Wave

16. ทดลองเปลี่ยนแปลงสัญญาณที่ Signal Generator ต่างๆ ดังนี้ บันทึกผลการทดลอง

- เปลี่ยนค่าเป็น **Square Wave** จำลองการทำงาน บันทึกผลการทดลอง

.....

.....

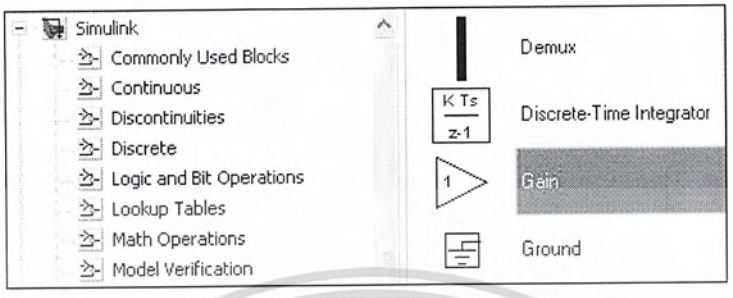
.....

.....

.....

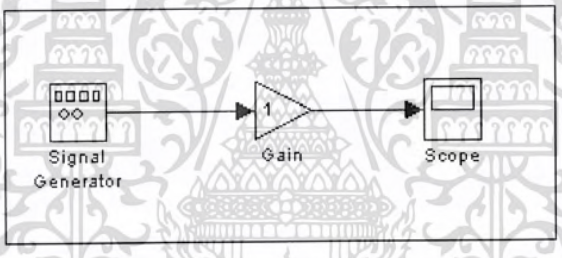
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. การเพิ่มบล็อก Gain ให้กับโมเดลโดยที่หน้าต่าง Simulink Library Browser โดยคลิกที่ **Commonly Used Blocks** เลือก บล็อก **Gain** แสดงดังรูปที่ 2.19



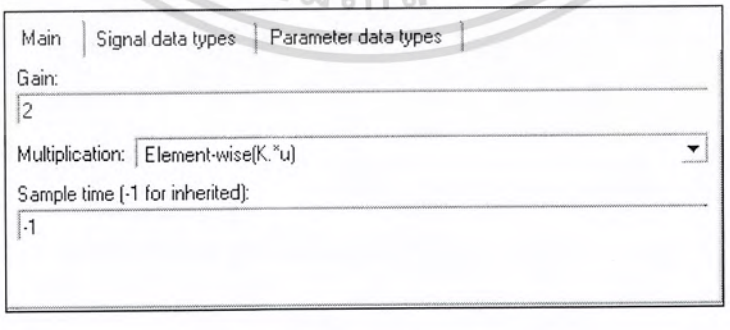
รูปที่ 2.19 หน้าต่างการเลือกบล็อก Gain

18. ลากเส้นเชื่อมต่อบล็อกทั้งหมดให้สมบูรณ์ แสดงดังรูปที่ 2.20



รูปที่ 2.20 การเชื่อมต่อบล็อก Gain กับบล็อกอื่นๆ

19. กำหนดค่าพารามิเตอร์ของบล็อก Gain โดยกำหนดเป็น 2 แสดงดังรูปที่ 2.21

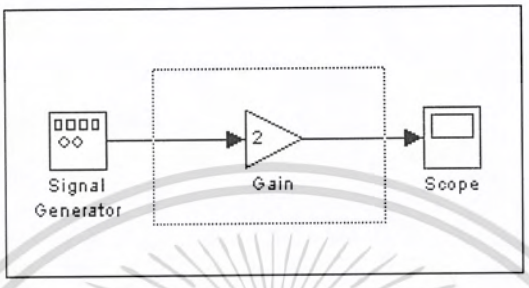


รูปที่ 2.21 การกำหนดค่าของบล็อก Gain

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

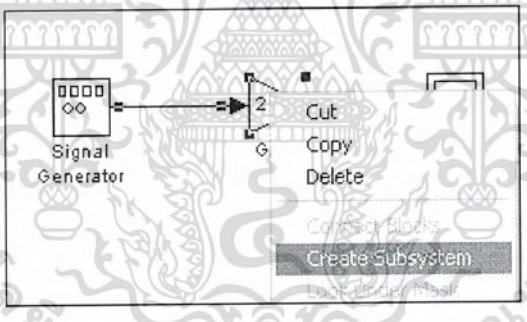
20. การสร้างระบบย่อย (Subsystems) เป็นการนำ Block ต่างๆ ที่เชื่อมต่อกันในระบบรวมไว้ด้วยกันเป็นหนึ่งระบบย่อยที่มี Port In 1 เป็นอินพุตและ Port Out 1 เป็นเอาต์พุต ซึ่งสามารถสร้างระบบย่อยของระบบที่กำหนดได้ดังนี้

- กดเมาส์ปุ่มซ้ายค้างไว้แล้วลากพื้นที่คลอบคลุมบล็อก Gain แสดงดังรูปที่ 2.22



รูปที่ 2.22 การเลือกพื้นที่เพื่อสร้างระบบย่อย

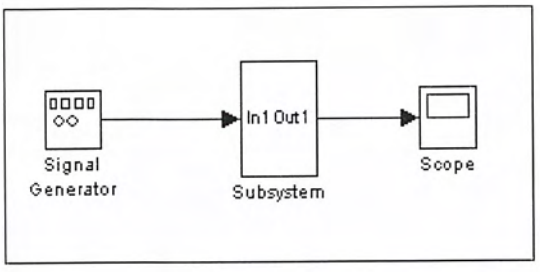
- คลิกขวาที่บล็อก Gain เลือก Create Subsystem แสดงดังรูปที่ 2.23



รูปที่ 2.23 การสร้างระบบย่อยของบล็อก Gain

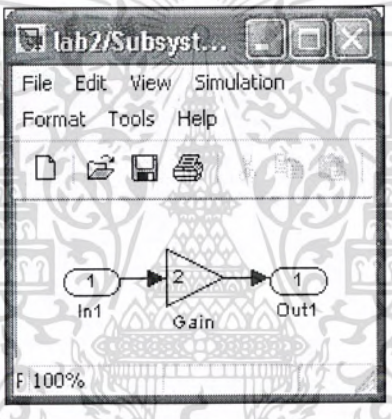
- ผลที่ได้เมื่อทำการสร้างระบบย่อยเสร็จสิ้น แสดงดังรูปที่ 2.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 การสร้างระบบย่อยเสร็จสิ้น

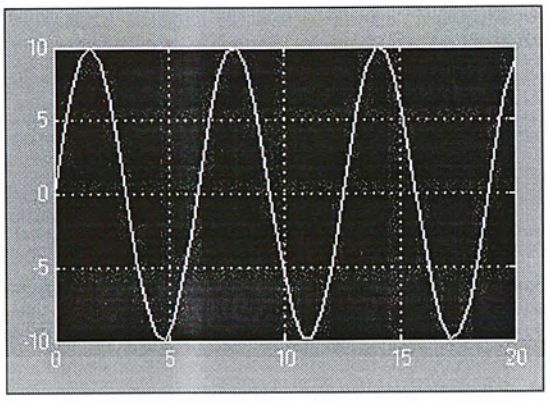
21. ดับเบิ้ลคลิกที่ระบบย่อย (Subsystem) จะแสดงระบบภายใน ซึ่งจะประกอบด้วย บล็อกรับสัญญาณอินพุต, บล็อก Scope และบล็อกส่งสัญญาณออก แสดงดังรูปที่ 2.25



รูปที่ 2.25 ระบบหลักภายในระบบย่อย

22. กำหนดค่าที่บล็อก Signal Generator เป็น Sine Wave และคลิกที่ปุ่ม ▶ เพื่อจำลองการทำงาน ดับเบิ้ลคลิกที่บล็อก Scope จะแสดงดังรูปที่ 2.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 ผลของสัญญาณที่ Scope เมื่อกำหนด Gain = 2

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. การจำลองการทำงาน (Simulation) มีประโยชน์อย่างไร?

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. จงออกแบบโมเดลที่มีสร้างสัญญาณ Saw tooth ที่มีแอมพลิจูดเท่ากับ 50 V และมีความถี่เท่ากับ 1 Rad/s ซึ่งจะต้องมีบล็อก Gain อยู่ในโมเดลด้วยแสดงผลสัญญาณที่สโคป บันทึกโมเดลชื่อ lab2_ans.mdl และบันทึกผลการทดลอง?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 3

เรื่อง การใช้งาน Real-Time Workshop

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. อธิบายขั้นตอนการทำงานของ Real-Time Workshop ได้อย่างถูกต้อง
2. อธิบายลักษณะการประยุกต์ใช้งานของ Real-Time Workshop ในด้านต่างๆ ได้อย่างถูกต้อง
3. บอกความแตกต่างระหว่างหน้าที่การทำงานของ Model Advisor และ Model Explorer ได้อย่างถูกต้อง
4. อธิบายการทำงานของไฟล์กระทำการได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0
2. กล่องเครื่องมือ Simulink เวอร์ชั่น 6.0
3. กล่องเครื่องมือ Real-Time Workshop เวอร์ชั่น 6.0
4. โปรแกรม Microsoft Visual C/C++ เวอร์ชั่น 6.0

ความรู้และทักษะเดิม

1. การใช้งานโปรแกรม MATLAB และ Simulink เบื้องต้น

ทฤษฎีเบื้องต้น

ในโปรแกรม MATLAB นั้นผู้ใช้งานสามารถออกแบบระบบ (Model) ในลักษณะโต้ตอบกับผู้ใช้งานได้ โดยออกแบบจาก Simulink ซึ่งมีลักษณะเป็นการนำบล็อกย่อยๆ มาต่อกันเป็นระบบใหญ่ แล้วทำการจำลองการทำงาน เพื่อดูผลการทำงานว่าถูกต้องหรือไม่ แต่ความสามารถของโปรแกรม MATLAB ที่น่าสนใจอีกอย่างคือ กล่องเครื่องมือ Real-Time Workshop จะสามารถสร้างโค๊ดภาษาซีจากระบบที่เราออกแบบได้ โดยจะทำงานร่วมกับโปรแกรมแปลภาษา ข้อดีก็คือผู้ใช้งานไม่ต้องทำการเขียนโปรแกรมด้วยตนเอง ลดขั้นตอนและเวลาในการพัฒนาโปรแกรมและเข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

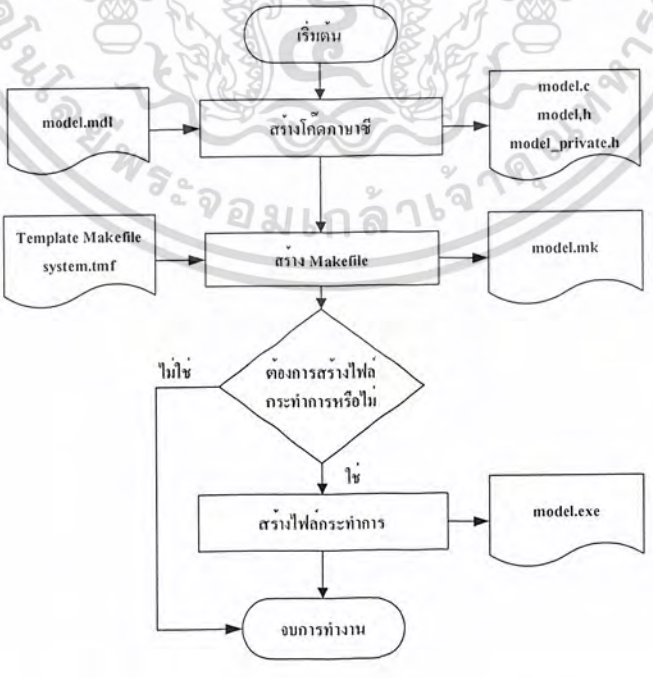
ภาพรวมของระบบได้ง่ายขึ้น ซึ่งโค๊ดภาษาซีที่สร้างขึ้นนั้นทางโปรแกรม MATLAB จะรองรับการประยุกต์ใช้งานในด้านต่างๆ มากมาย อาทิเช่น ใช้งานบนระบบปฏิบัติการของคอมพิวเตอร์ ใช้งานเป็นระบบฝังตัวบนฮาร์ดแวร์ (Embedded System) หรืออาจใช้งานเป็นโปรแกรมเพื่อการควบคุมคอมพิวเตอร์ระยะไกล เช่น ผ่านทางระบบเครือข่าย ซึ่งใช้โปรแกรม MATLAB ควบคุมการทำงานทั้งระบบ ซึ่งจะสามารถทำงานในเวลาจริงได้อีกด้วย

การสร้างโค๊ดภาษาซี

ขั้นตอนหลักในการใช้งาน Real-Time Workshop นั้นจะเป็นการสร้างโค๊ดภาษาซีจากระบบ ซึ่งโปรแกรม MATLAB นั้นไม่สามารถสร้างโค๊ดภาษาซีได้เอง จะต้องอาศัยการทำงานร่วมกับโปรแกรมแปลภาษาซีต่างๆ เช่น Visual C/C++ หรือ Borland C/C++ เป็นต้น การสร้างโค๊ดจากระบบนั้นจะเป็นการนำค่าพารามิเตอร์ต่างๆ ของระบบและโครงสร้างการแปลภาษาเพื่อใช้งานจริง (Template Makefile System) จากโปรแกรม MATLAB แล้วอาศัยโปรแกรมแปลภาษานำค่าพารามิเตอร์ต่างๆ มาเขียนเป็นโครงสร้างของภาษาซีหรือสร้างเป็นไฟล์กระทำการก็ได้

ขั้นตอนการทำงานของ Real-Time Workshop

การทำงานของ Real-Time Workshop นั้นจะแบ่งเป็นขั้นตอนหลักๆ คือ การออกแบบระบบ การกำหนดค่าพารามิเตอร์ การแปลงระบบเป็นโค๊ดภาษาซี และการสร้างไฟล์กระทำการ ซึ่งสามารถแสดงเป็นผังงานได้ดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการทำงานของ Real-Time Workshop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การออกแบบระบบจาก Simulink

เป็นการออกแบบระบบหรือโมเดลที่ผู้ใช้ต้องการ ซึ่งจะทำการออกแบบใน Simulink เป็นการนำบล็อกเซ็ต (Blocksets) ต่างๆ ที่มีใน Simulink มาต่อกันเป็นระบบ ซึ่งบล็อกเซ็ตบางชนิดทาง Real-Time Workshop ก็ไม่รองรับคือ ไม่สามารถสร้างโค๊ดภาษาซีจากฟังก์ชั่นบล็อกนั้นๆ ได้ซึ่งระบบที่ได้จะมีไฟล์เป็น model.mdl

2. การกำหนดค่าพารามิเตอร์ (Configuration Parameters)

ก่อนทำการสร้างโค๊ดภาษาซีจะต้องมีการกำหนดค่าพารามิเตอร์ต่างๆ ก่อน เป็นการกำหนดคุณสมบัติของระบบและโครงสร้างการแปลภาษาเพื่อใช้งานจริง ซึ่งค่าพารามิเตอร์ที่กำหนดจะมีผลในการสร้างโค๊ดภาษาซีทั้งหมด

3. การสร้างโค๊ดภาษาซี (C Code Generator)

ขั้นตอนนี้จะเป็นการสร้างโค๊ดภาษาซีจากระบบที่ออกแบบไว้และกำหนดค่าพารามิเตอร์แล้วเท่านั้น โดยค่าพารามิเตอร์ที่กำหนดไว้จะถูกแปลงเป็นไฟล์ model.rtw ซึ่งเป็นไฟล์เฉพาะของ Real-Time Workshop จะมีหน้าที่นำค่าพารามิเตอร์ต่างๆ ส่งให้กับตัวแปลภาษาสร้างโค๊ดภาษาซี เมื่อเสร็จขั้นตอนการสร้างโค๊ดจะได้ไฟล์ภาษาซี เช่น model.c และ model.h เป็นต้น

4. การสร้าง Makefile

เป็นขั้นตอนสร้าง Makefile เป็นไฟล์ที่มีหน้าที่กำหนดคุณสมบัติของไฟล์กระทำการ (Execute files : model.exe) โดย Makefile จะถูกสร้างขึ้นจากไฟล์ system.tmf อีกครั้งหนึ่ง ซึ่งได้มาจากขั้นตอนการกำหนดค่าพารามิเตอร์ ไฟล์ที่ได้ในขั้นตอนนี้จะเป็นไฟล์ model.mk

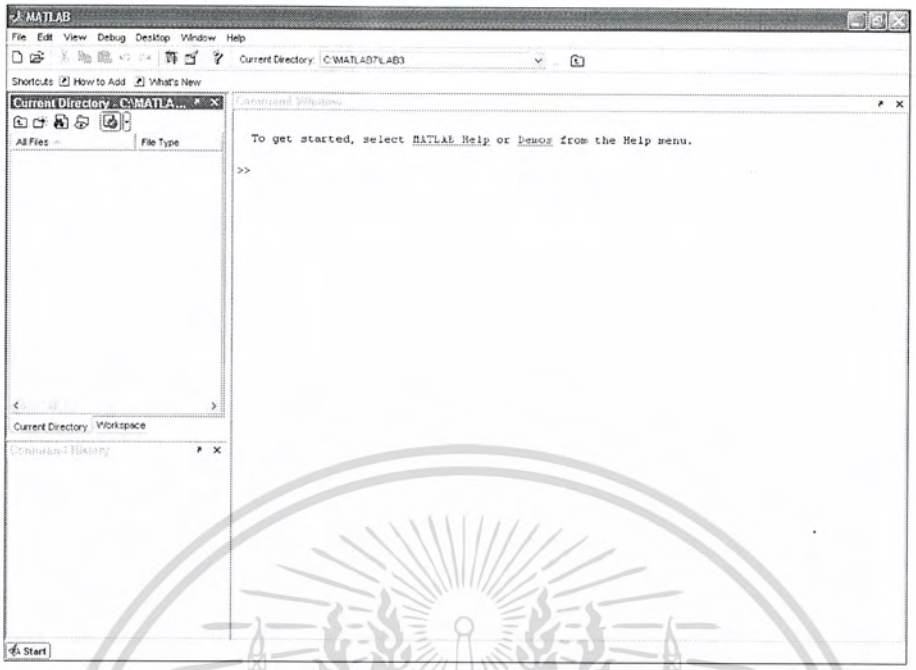
5. การสร้างไฟล์กระทำการ (Execute files Generator)

เป็นขั้นตอนการสร้างไฟล์กระทำการ model.exe ซึ่งถูกกำหนดคุณสมบัติต่างๆ จากไฟล์ model.mk จะเป็นไฟล์ที่นำไปใช้งานจริงตามคุณสมบัติที่ได้ออกแบบไว้ เช่น ใช้งานบนระบบปฏิบัติการของคอมพิวเตอร์หรือใช้งานเป็นระบบฝังตัวบนฮาร์ดแวร์ เป็นต้น

ลำดับขั้นการทดลอง

1. ก่อนการใช้งาน Real-Time Workshop ต้องติดตั้งโปรแกรมแปลภาษาซีก่อนในการทดลองนี้จะใช้โปรแกรม Microsoft Visual C/C ++ Professional Edition เวอร์ชั่น 6.0 เมื่อติดตั้งเรียบร้อยแล้วเปิดโปรแกรม MATLAB เวอร์ชั่น 7.0 เลือกไดเร็กทอรี่ที่ใช้งานในการทดลองนี้เลือกเป็น **C:\โปรแกรม MATLAB7\LAB3** แสดงดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 หน้าต่างโปรแกรม MATLAB เวอร์ชั่น 7.0

2. ติดตั้งตัวแปลภาษาซีในโปรแกรม MATLAB โดยพิมพ์คำสั่ง **>>mex – setup** เลือกตัวแปลภาษาเป็น Microsoft Visual C/C++ แสดงขั้นตอนดังรูปที่ 3.3

```
>> mex -setup
Please choose your compiler for building external interface (MEX) files:

Would you like mex to locate installed compilers [y]/n? y

Select a compiler:
[1] Digital Visual Fortran version 6.0 in C:\Program Files\Microsoft Visual Studio
[2] Lcc C version 2.4 in C:\MATLAB7\sys\lcc
[3] Microsoft Visual C/C++ version 6.0 in C:\Program Files\Microsoft Visual Studio

[0] None

Compiler: 3

Please verify your choices:

Compiler: Microsoft Visual C/C++ 6.0
Location: C:\Program Files\Microsoft Visual Studio

Are these correct?([y]/n): y

Try to update options file: C:\Documents and Settings\thep\Application Data\MathWorks
From template:              C:\MATLAB7\BIN\WIN32\mexopts\msvc60opts.bat

Done . . .
```

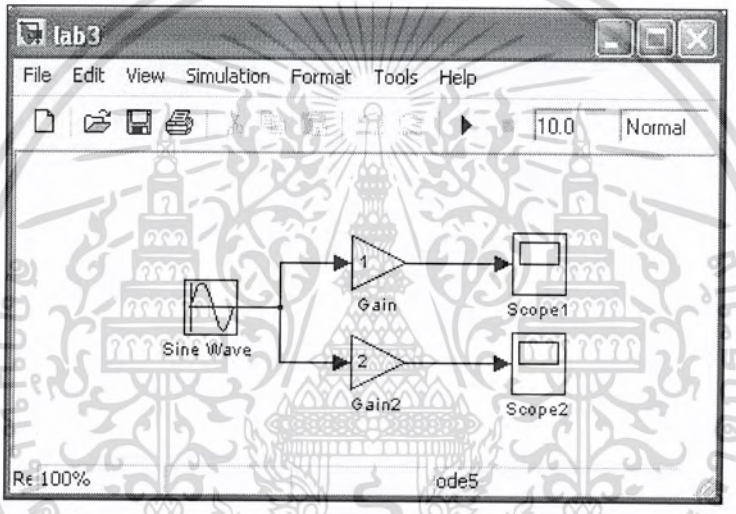
รูปที่ 3.3 การติดตั้งตัวแปลภาษาในโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เปิดกล่องที่เครื่องมือ Simulink โดยสามารถคลิกที่ หรือพิมพ์คำสั่ง >> **Simulink** ที่ Command Window จะปรากฏกล่องเครื่องมือ Simulink สร้างโมเดลใหม่ (New Model) โดยคลิกที่ จะปรากฏหน้าต่าง New Model ขั้นตอนต่อไปให้สร้างบล็อกต่างๆ มีรายละเอียดดังนี้

- คลิกที่หัวข้อ **Source** เลือก **Sine Wave**
- คลิกที่หัวข้อ **Sink** เลือก **Scope** จำนวน **2** บล็อก
- คลิกที่หัวข้อ **Commonly Used Blocks** เลือก **Gain** จำนวน **2** บล็อก

4. ลากเส้นเชื่อมต่อและตั้งชื่อบล็อกต่างๆ เสร็จแล้วบันทึกโมเดลตั้งชื่อเป็น **lab3** แสดงดังรูปที่ 3.4



รูปที่ 3.4 การเชื่อมต่อบล็อกของโมเดล lab3

5. กำหนดค่าพารามิเตอร์ในบล็อกต่างๆ ดังนี้

5.1 ดับเบิ้ลคลิกที่บล็อก **Sine wave** กำหนดค่าดังนี้

- **Sine Type** เลือกเป็น **Time Based**
- **Amplitude** กำหนดค่าเป็น **1V**
- **Frequency** กำหนดค่าเป็น **1 rad/sec**

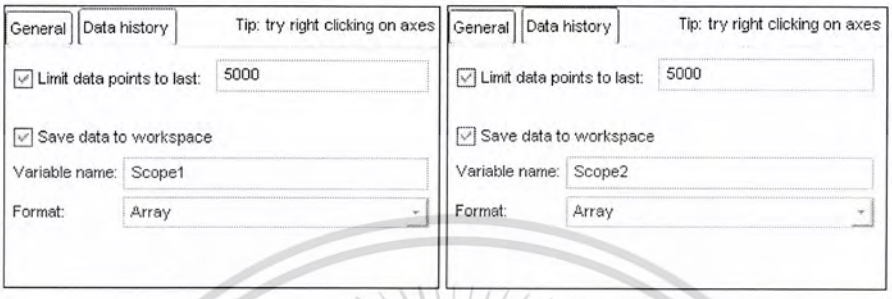
5.2 ดับเบิ้ลคลิกที่บล็อก **Gain** กำหนดค่าดังนี้

- บล็อก **Gain1** กำหนดค่า Gain เป็น **1**
- บล็อก **Gain2** กำหนดค่า Gain เป็น **2**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

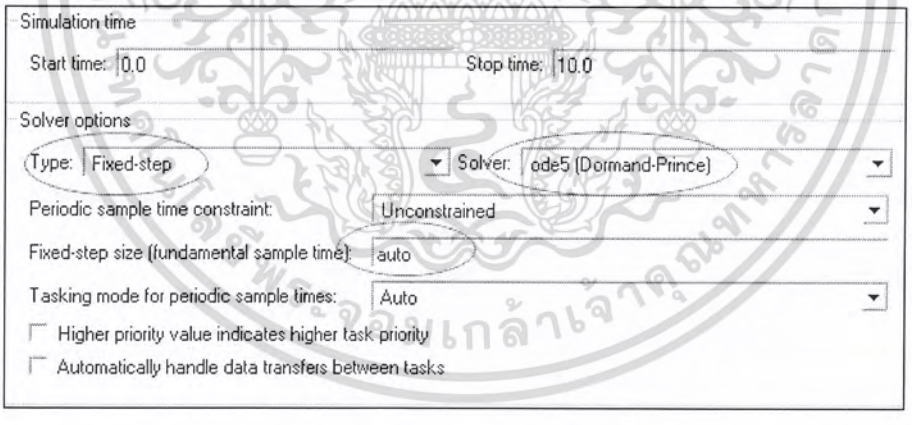
5.3 ดับเบิ้ลคลิกที่บล็อก **Scope1** และ **Scope2** คลิกที่ จะแสดงหน้าต่าง Scope Parameter ในแถบ Data history กำหนดดังรูปที่ 3.5

1) Scope 1 กำหนดค่าดังรูป 2) Scope 2 กำหนดค่าดังรูป



รูปที่ 3.5 การกำหนดค่าของ Scope ในหัวข้อ Data history

6. กำหนดค่าพารามิเตอร์ของโมเดล โดยที่แถบเมนูบาร์ด้านบนคลิกที่ **Simulation > Configuration Parameters** หรือใช้คีย์ลัด Ctrl+E จะแสดงหน้าต่าง Configuration Parameters คลิกที่หัวข้อ **Solver** กำหนดค่าพารามิเตอร์ต่างๆ ดังรูปที่ 3.6



รูปที่ 3.6 หน้าต่างการกำหนดค่าในหัวข้อ Solver Options

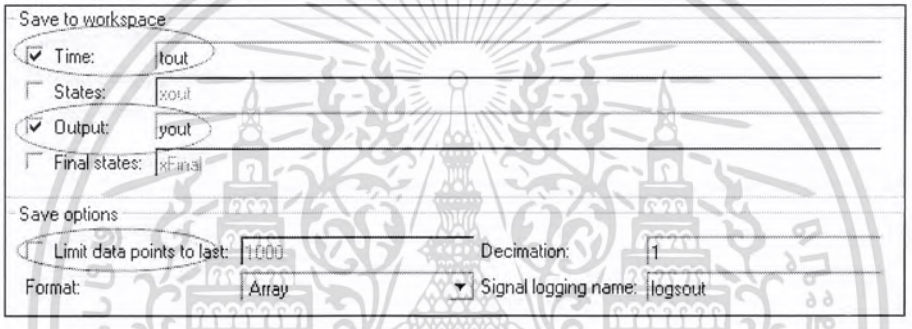
Simulation Time เป็นการกำหนดเวลาในการทำงานของโมเดลจะเริ่มต้นที่เวลาเท่าใดสิ้นสุดที่เวลาเท่าใดหรือหมายถึงเป็นช่วงเวลาที่สามารถสังเกตการทำงานที่ Target PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Solver Option เป็นการกำหนดวิธีการประมวลผลของโมเดล ที่ช่อง Type เป็นการกำหนดลักษณะของตัวเลขที่ในการคำนวณ จากเดิมที่โปรแกรมตั้งไว้จะเป็น Variable Step แต่เราจะกำหนดเป็น Fixed Step เนื่องจากเป็นการลดการคำนวณและลดขนาดของโมเดลเมื่อใช้งานที่ Target PC ส่วนในช่อง Solver เป็นการกำหนดอัลกอริธึมที่ใช้ในการคำนวณ โดยจะกำหนดเป็น ode5 โดยสามารถที่จะเปลี่ยนแปลงเป็นแบบอื่นได้

Fixed Step Size เป็นการกำหนดช่วงเวลาในการเปลี่ยนแปลงของสัญญาณเข้า

7. คลิกที่หัวข้อ **Data Import/Export** เป็นการกำหนดค่าพารามิเตอร์ที่ในการส่งค่าข้อมูลออกให้กับ MATLAB Workspace ในช่อง Save Option เป็นการกำหนดจำนวนข้อมูลที่ส่งให้กับ workspace กำหนดเป็นไม่จำกัดจำนวนของข้อมูลแสดงดังรูปที่ 3.7



รูปที่ 3.7 หน้าต่างการกำหนดค่าในหัวข้อ Data Import/Export

8. คลิกที่หัวข้อ **Real-Time Workshop** เป็นการกำหนดค่าพารามิเตอร์ในการสร้างโค๊ดภาษาซี ทั้งหมด โดยในหัวข้อ **Target Selection** จะเป็นการเลือกโครงสร้างการแปลภาษา (TLC : Target Language Compiler) ซึ่งจะต้องเลือกให้เหมาะสมกับการใช้งานโดยคลิกที่ปุ่ม Browse จะปรากฏหน้าต่างการเลือก TLC **บันทึกผลการทดลองในตารางที่ 3.1** หลังจากการบันทึกผลเลือก **grt.tlc** (Generic Real-Time Target) คลิก **Ok**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ไฟล์ TLC (Target Language Compiler) ที่สามารถใช้งานได้และลักษณะการใช้งาน

System Target File	Description
1. c166.tlc	
2. ert.tlc	
3. grt.tlc	
4. hc12.tlc	
5. mpc555rt.tlc	
6. rsim.tlc	
7. rtwin.tlc	
8. ti_c2000_grt.tlc	
9. ti_c6000.tlc	
10. xpctarget.tlc	

การเลือก Target Language Compiler เป็นส่วนที่สำคัญมาก เนื่องจากจะเป็นการกำหนดการประยุกต์ใช้งานของโมเดลหลังการแปลภาษา ซึ่งมีตัวอย่างการประยุกต์ใช้งานดังนี้

- **c166.tlc** เป็นการประยุกต์ใช้งานโปรแกรมลงบนไมโครคอนโทรลเลอร์ Infineon C166 ซึ่งจะต้องใช้กล่องเครื่องมือ Real-Time Workshop Embedded Coder
- **grt.tlc** เป็นการใช้งานบนเครื่องคอมพิวเตอร์ทั่วไปโดยไฟล์ที่ได้จะเป็น .exe
- **hc12.tlc** เป็นการประยุกต์ใช้งานโปรแกรมลงบนไมโครคอนโทรลเลอร์ Motorola HC12 ซึ่งจะต้องมีโปรแกรม CodeWarrior for Motorola HC12 และกล่องเครื่องมือ Real-Time Workshop Embedded Coder
- **rtwin.tlc** เป็นการใช้งานบนเครื่องคอมพิวเตอร์ โดยสามารถที่จะควบคุมคอมพิวเตอร์เครื่องอื่นๆ ได้ ซึ่งเป็นลักษณะ Host และ Target คล้ายกับ xPC Target แตกต่างกันที่ Target PC เป็น Windows เท่านั้น ซึ่งจะต้องใช้กล่องเครื่องมือ Real-Time Window Target
- **ti_c6000.tlc** เป็นการประยุกต์ใช้งานโปรแกรมลงบนชิฟ DSP รุ่น TI C6000 DSP ซึ่งจะต้องมีโปรแกรม Code Composer Studio IDE สำหรับการโปรแกรมลง DSP Chip รุ่น C6000 และกล่องเครื่องมือ Real-Time Workshop Embedded Coder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **xpctarget.tlc** เป็นการประยุกต์การใช้งานนำโมเดลที่ได้ไปทำงานที่เครื่อง Target PC และควบคุมการทำงานที่ Target PC ผ่านทาง Host PC โดยทำงานแบบ Real-Time ได้ การใช้งานนั้นจะอธิบายในใบงานที่ 4 และ 5

9. ในหัวข้อ **Document** เป็นการกำหนดให้มีการรายงานผลการสร้างโค๊ดหลังจากการสร้างโค๊ดเสร็จสิ้นแล้วหรือไม่โดยจะแสดงในรูปแบบเอกสาร HTML กำหนดค่าดังรูปที่ 3.8

รูปที่ 3.8 หน้าต่างกำหนดการรายงานผลการสร้างโค๊ดในรูปแบบเอกสาร HTML

10. คลิกที่หัวข้อ **Interface** กำหนดค่าลักษณะการเชื่อมต่อระหว่างโมเดลหลังการสร้างโค๊ดกับโปรแกรม MATLAB ในหัวข้อ **Verification** เป็นการกำหนดชื่อของข้อมูลที่จะส่งให้กับ MALAB Workspace โดยกำหนดค่าเป็นให้มีชื่อ _rt ต่อท้ายชื่อตัวแปรเดิมคลิกที่ **_rt** คลิก **Apply** คลิก **Ok** แล้วปิดหน้าต่าง **Configuration Parameters**

11. ขั้นตอนต่อไปเป็นการตรวจสอบความถูกต้องของโมเดลทั้งหมดก่อนการสร้างโค๊ดโดยกลับมาที่หน้าต่าง lab3.mdl ทำการบันทึกโมเดลโดยคลิกที่ปุ่ม เสร็จแล้วคลิก ที่ **Tools > Model Advisor** จะแสดงดังรูปที่ 3.9

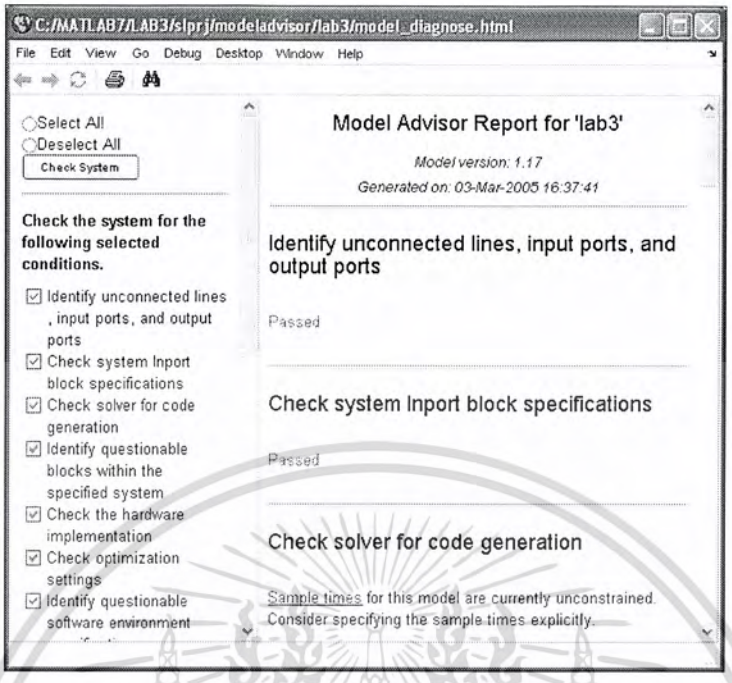
12. ตรวจสอบการกำหนดค่าพารามิเตอร์ทั้งหมด โดยคลิกที่ **Select All** แล้วคลิกที่ปุ่ม **Check System** เมื่อขั้นตอนการตรวจสอบเสร็จสิ้น คลิกที่หัวข้อ **Check the Hardware implementation** อยู่ทางเมนูด้านซ้าย สังเกตผลทางเมนูด้านขวา **บันทึกผลการทดลองในตารางที่ 3.2**

12.1 Your current system target file is _____.

12.2 You are using the following specification

ผลที่ได้ในขั้นตอนนี้จะเป็นการรายงานว่าโมเดลที่ออกแบบนั้นได้มีการกำหนดให้ใช้กับ Target ประเภทใด และบอกว่าได้กำหนดให้การแปลภาษาซี นั้นมีคุณสมบัติของตัวแปลอะไรบ้าง (สามารถย้อนกลับไปดูการกำหนดนี้ในหน้าต่าง Configuration Parameters ในหัวข้อ Hardware-Implementation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 หน้าต่าง Model Advisor

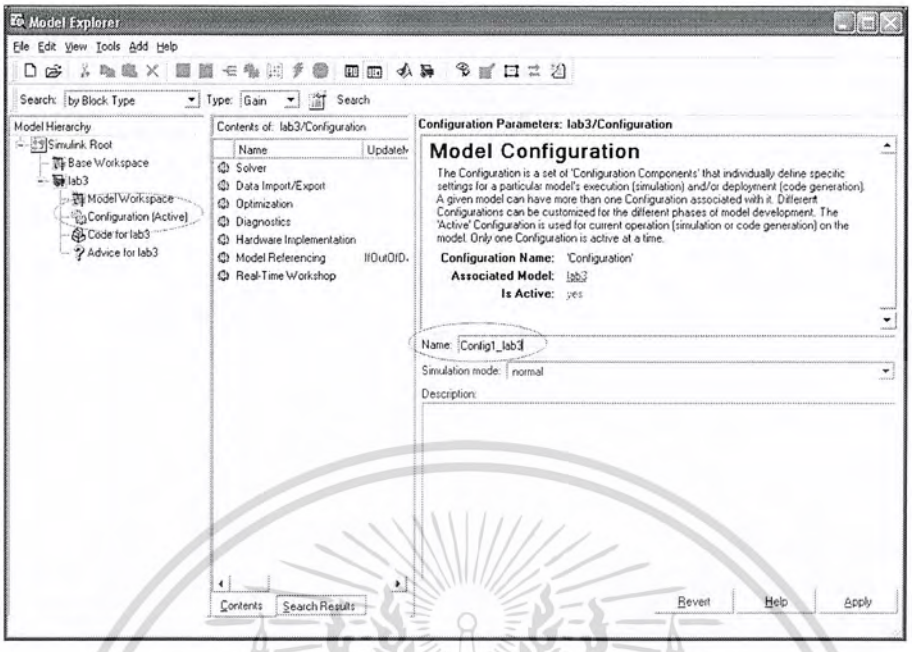
ตารางที่ 3.2 ค่าพารามิเตอร์ของ Hardware Implementation

	Embedded Hardware	**Emulation Hardware**
C Type char (bits)		
C Type short (bits)		
C Type int (bits)		
C Type long (bits)		

13. สังเกตว่าการใช้งาน Real-Time Workshop ทุกครั้งจะต้องมีการกำหนดค่าพารามิเตอร์ในหน้าต่าง Configuration Parameters ดังนั้นถ้าต้องการบันทึกรูปแบบการกำหนดค่าไว้ใช้งานในภายหลังก็สามารถทำได้โดยคลิกที่ **View > Model Explorer**

14. คลิกที่หัวข้อ **Configuration** จะแสดงหน้าต่างย่อย Model Configuration จะปรากฏในช่อง **Name** ตั้งชื่อเป็น **Config1_lab3** คลิก **Apply** แสดงดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 หน้าต่าง Model Explorer

ในขั้นตอนนี้รูปแบบการกำหนดค่าพารามิเตอร์ จะถูกบันทึกไว้แล้วในชื่อ Config1_lab3 ซึ่งกำลังถูกใช้งานอยู่สังเกตจะมีคำว่า Active ต่อท้าย

15. ทำการเปลี่ยนแปลงการกำหนดค่าพารามิเตอร์ โดยกลับมาที่หน้าต่าง lab3.mdl ใช้คีย์ลัด **Ctrl+E** จะแสดงหน้าต่าง **Configuration Parameters** ในหัวข้อ **Solver** กำหนดค่าใหม่เป็น **ode4 (Runge-Kutta)** คลิก **OK**

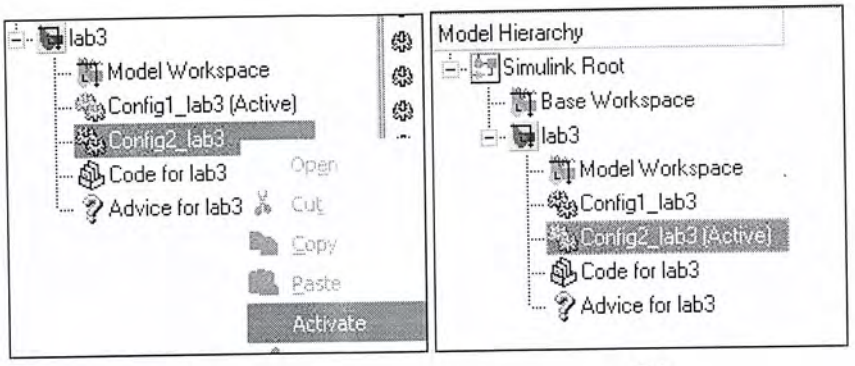
16. เปิดหน้าต่าง **Model Explorer** อีกครั้ง คลิกที่ **Add > Configuration Set** เพื่อที่จะทำรูปแบบการกำหนดค่าพารามิเตอร์

17. คลิกที่ **Configuration** ทางด้านขวา ตั้งชื่อใหม่เป็น **Config2_lab3** คลิก **Apply**

18. คลิกขวาที่ **Config2_lab3** แล้วเลือก **Active** เพื่อเป็นการเปิดใช้งาน ซึ่งจะเห็นว่ามีคำว่า Active ต่อท้ายแสดงว่ากำลังถูกเรียกใช้งานอยู่ ดังรูปที่ 3.11

จะเห็นว่าสามารถเลือกใช้งานรูปแบบการกำหนดค่าพารามิเตอร์ต่างๆ ได้ โดยที่รูปแบบเดิมจะไม่ถูกลบทิ้ง จะเป็นการปิดการใช้งานชั่วคราวเท่านั้น เพื่อสะดวกในการแก้ไขภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) (ข)

รูปที่ 3.11 หน้าต่างการเลือก Configuration Parameter Config2_lab3

19. ขั้นตอนต่อไปจะสร้างโค๊ดภาษาซีและไฟล์ EXE โดยทำได้ 2 วิธีคือ

19.1 ในหน้าต่าง **Configuration Parameter** คลิกที่หัวข้อ **Real-Time Workshop** คลิกที่ปุ่ม Build

19.2 ในหน้าต่าง lab3.mdl คลิกที่เมนู **Tool > Real-Time Workshop > Build Model** หรืออาจใช้คีย์ลัด **Ctrl+B**

20. เมื่อเริ่มขั้นตอนสร้างโค๊ดภาษาซีและไฟล์กระทำการ ที่หน้าต่างโปรแกรม MATLAB จะแสดงขั้นตอนต่างๆ บันทึกผลการทดลอง

Starting Real-Time Workshop build procedure for model: _____.

Generating code into build directory: C:\ MATLAB7\LAB3 _____.

Invoking Target Language Compiler on _____.

Creating HTML report file _____.

Creating project marker file: _____.

Creating lab3.mk from C:\โปรแกรม MATLAB7\rtw\c\grt_____.

Building lab3: ._____.

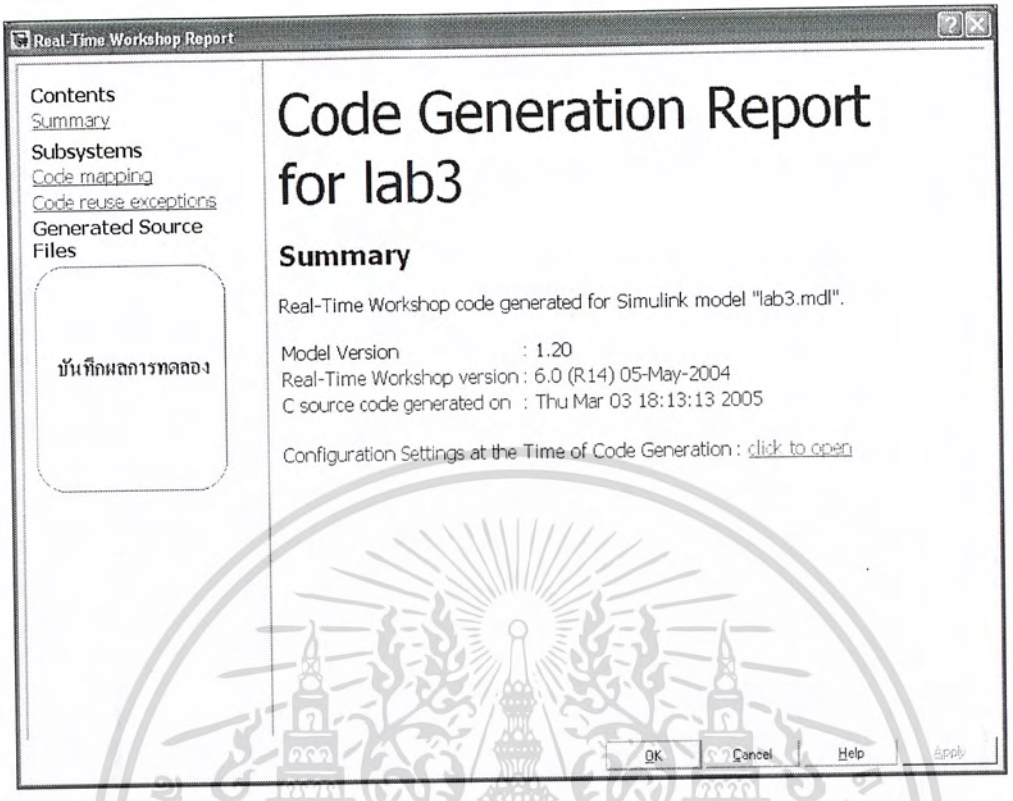
*** Created executable _____ .

เมื่อเสร็จสิ้นจะปรากฏข้อความ

_____.

21. เมื่อขั้นตอนการสร้างโค๊ดภาษาซีและไฟล์กระทำการเสร็จสิ้นจะปรากฏหน้าต่าง Real-Time Workshop Report บันทึกผลไฟล์ภาษาซี ที่ถูกสร้างขึ้น ในหัวข้อ **Generated Source Files** ดังรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

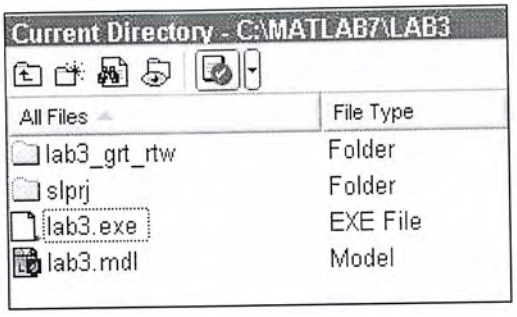


รูปที่ 3.12 หน้าต่าง Real-Time Workshop Report

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____

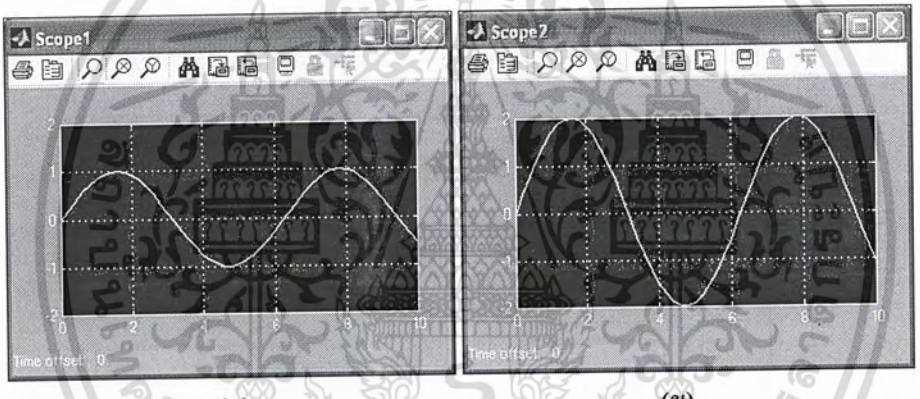
22. กลับมาที่หน้าต่างโปรแกรม MATLAB ที่หน้าต่างย่อย Current Directory จะแสดงไฟล์ต่างๆ ที่เพิ่มขึ้นแสดงดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงไฟล์ในหน้าต่าง Current Directory

23. กลับมาที่ lab3.mdl จำลองการทำงานของโมเดล โดยคลิกที่ ▶ เพื่อเริ่มต้นจำลองการทำงาน **ดับเบิ้ลคลิกที่ Scope ทั้งสอง** เพื่อดูผลการทำงาน แสดงดังรูปที่ 3.14



(ก) **(ข)**

รูปที่ 3.14 ผลการจำลองการทำงาน **(ก)** Scope1 **(ข)** Scope2

บันทึกผลการทดลอง (หมายเหตุ การคำนวณความถี่ของสัญญาณที่เป็น Rad/s (เรเดียนต่อวินาที) นั้นมีสูตรในการคำนวณคือ 2π Rad = 1 Hz)

1. สัญญาณที่ Scope1 มี Amplitude = ___V และ Frequency = _____ Hz = _____Rad/s
2. สัญญาณที่ Scope2 มี Amplitude = ___V และ Frequency = _____ Hz = _____Rad/s

24. ที่โปรแกรม MATLAB Command พิมพ์คำสั่ง **>>whos** เพื่อดูข้อมูลทั้งหมด **บันทึกผลการทดลองในตารางที่ 3.3**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

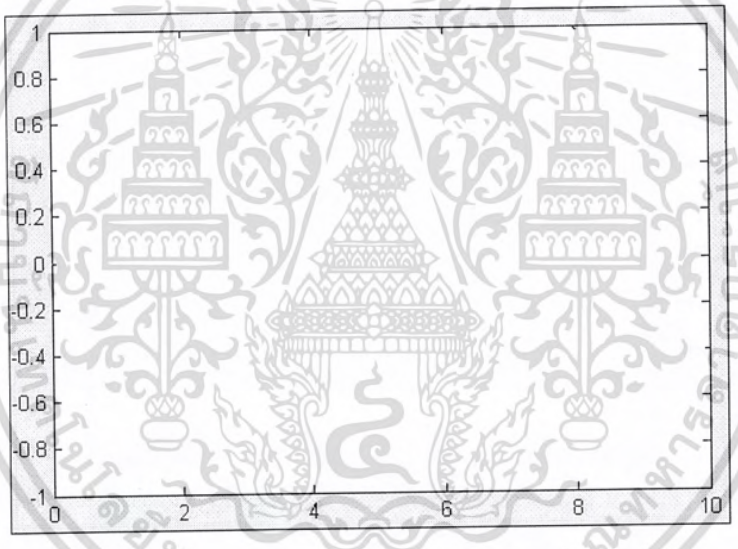
ตารางที่ 3.3 การแสดงผลของคำสั่ง whos

Name	Size	Bytes	Class

Grand total is _____ elements using _____ bytes

25. นำข้อมูลที่รับมาจาก lab32.mdl มาทำการพล็อตกราฟ มีขั้นตอนดังนี้

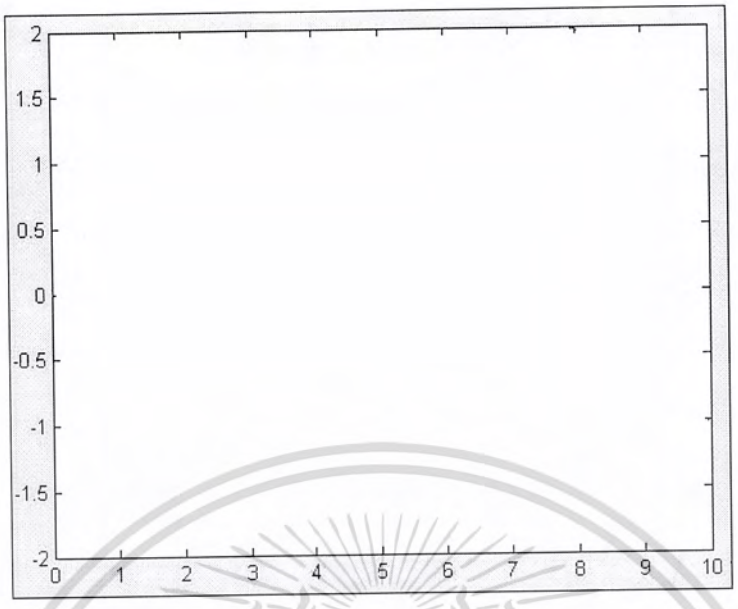
25.1 พิมพ์คำสั่ง >>**plot(tout,Scope1(:,2))** บันทึกผลการทดลอง



รูปที่ 3.15 กราฟที่เกิดขึ้นของข้อมูล tout กับ Scope1

25.2 พิมพ์คำสั่ง >>**plot(tout,Scope2(:,2))** บันทึกผลการทดลองในรูปที่ 3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 กราฟที่เกิดขึ้นของข้อมูล tout กับ Scope2

26. พิมพ์คำสั่ง >> **!lab3** (คำสั่ง ! เป็นการเรียกไฟล์ .exe ให้ทำงาน) ซึ่งเป็นการเรียกไฟล์ lab3.exe ให้ทำงาน ซึ่งจะทำการสร้างไฟล์ lab3.mat เป็นไฟล์ที่ใช้เชื่อมต่อข้อมูลระหว่าง lab3.exe กับโปรแกรม MATLAB

27. พิมพ์คำสั่ง >>**load lab3.mat** เพื่อเป็นการเรียกไฟล์ lab3.mat ให้ทำงาน

28. พิมพ์คำสั่ง >>**whos** เพื่อดูข้อมูลทั้งหมด **บันทึกผลการทดลองในตารางที่ 3.4**

ตารางที่ 3.4 การแสดงข้อมูลทั้งหมดของการใช้คำสั่ง whos

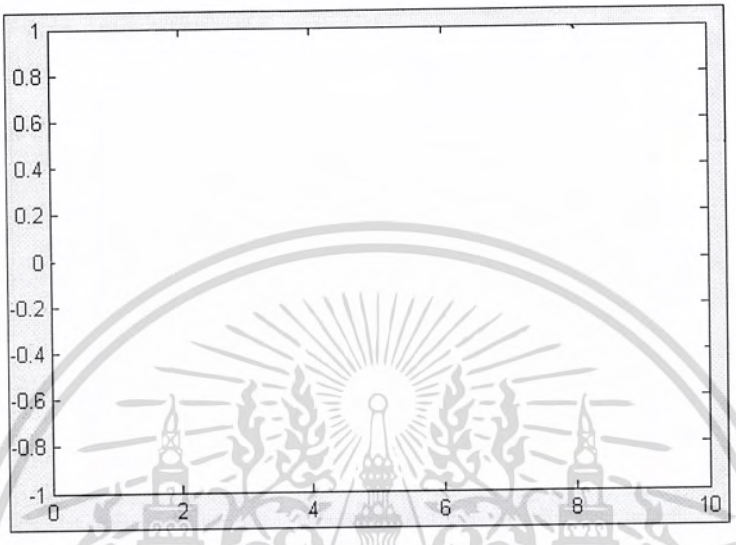
Name	Size	Bytes	Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Grand total is _____ elements using _____ bytes

29. นำข้อมูลที่รับมาจากไฟล์ lab3.exe มาทำการพล็อตกราฟ มีขั้นตอนดังนี้

29.1 พิมพ์คำสั่ง >>**plot(tout,Scope1_rt(:,2))** บันทึกผลการทดลองในรูปที่ 3.17



รูปที่ 3.17 กราฟที่เกิดขึ้นของข้อมูล tout กับ Scope1_rt

29.2 พิมพ์คำสั่ง >>**plot(tout,Scope2_rt(:,2))** บันทึกผลการทดลองในรูปที่ 3.18

รูปที่ 3.18 กราฟที่เกิดขึ้นของข้อมูล tout กับ Scope2_rt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

30. ทำการทดสอบความผิดพลาดของข้อมูลที่จะสร้างขึ้นระหว่าง lab3.mdl และ lab3.exe ที่ Command Windows มีขั้นตอนดังนี้

30.1 พิมพ์คำสั่ง >>**max(abs(Scope1-Scope1_rt))** บันทึกผลการทดลอง

.....

.....

.....

30.2 พิมพ์คำสั่ง >>**max(abs(Scope2-Scope2_rt))** บันทึกผลการทดลอง

.....

.....

.....

จากผลการทดลองจะเห็นว่าข้อมูลที่ได้จาก model.mdl เมื่อเปรียบเทียบกับ ข้อมูลที่ได้จาก model.exe แล้วนั้นจะมีค่าผิดพลาดอยู่ค่าหนึ่งแต่เป็นค่าที่น้อยมากๆ ซึ่งเป็นค่าที่ยอมรับได้

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. จงอธิบายขั้นตอนการทำงานของ Real-Time Workshop มาพอสังเขป?

.....

.....

.....

.....

.....

2. จากผลการทดลองในขั้นตอนที่ 8 จงยกตัวอย่างไฟล์ TLC มา 3 ชนิดและอธิบายลักษณะการนำไปใช้งาน?

.....

.....

.....

.....

.....

3. จงอธิบายหน้าที่ของหน้าต่าง Model Advisor และ Model Explorer?

.....

.....

.....

.....

.....

4. จงเปรียบเทียบผลการทดลองในขั้นตอนที่ 23, 25 และ29 ว่าเหมือนกันหรือแตกต่างกันเพราะเหตุใด จงอธิบาย?

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 4
เรื่อง การใช้งาน xPC Target

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. อธิบายขั้นตอนการทำงานของ xPC Target ได้อย่างถูกต้อง
2. บอกความแตกต่างระหว่างการเชื่อมต่อแบบอนุกรมและระบบเครือข่ายได้อย่างถูกต้อง
3. แสดงขั้นตอนการเชื่อมต่อระหว่าง Host PC กับ Target PC ด้วยการสื่อสารอนุกรมได้อย่างถูกต้อง
4. แสดงขั้นตอนการเชื่อมต่อระหว่าง Host PC กับ Target PC ด้วยการสื่อสารแบบระบบเครือข่ายได้อย่างถูกต้อง
5. แสดงขั้นตอนการดาวน์โหลดไฟล์กระทำการจาก Host PC ไปที่ Target PC ได้อย่างถูกต้อง
6. อธิบายการทำงานของไฟล์กระทำการที่ Target PCได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0
2. กล่องเครื่องมือ Simulink เวอร์ชั่น 6.0
3. กล่องเครื่องมือ Real-Time Workshop เวอร์ชั่น 6.0
4. โปรแกรม Microsoft Visual C/C++ เวอร์ชั่น 6.0
5. สายสื่อสารอนุกรม RS 232 (Serial Communication Cable)
6. เครื่องคอมพิวเตอร์ Target PC ที่มีพอร์ตอนุกรม
7. เครื่องคอมพิวเตอร์ Host PC และ Target PC ที่เชื่อมต่อกับระบบเครือข่าย
8. แผ่นฟล็อปปี้ดิสก์เปล่าขนาด 3.5 นิ้ว จำนวน 2 แผ่น

ความรู้และทักษะเดิม

1. การใช้งานโปรแกรม MATLAB และ Simulink เบื้องต้น
2. การใช้งาน Real-Time Workshop เวอร์ชั่น 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

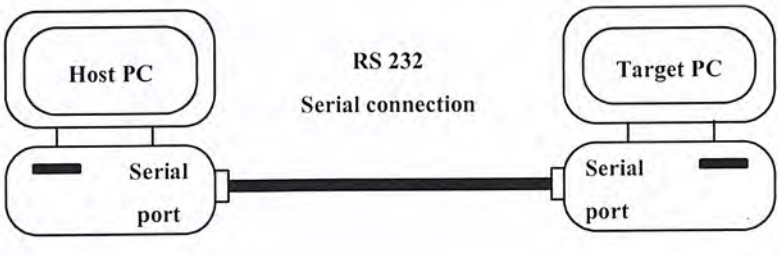
ทฤษฎีเบื้องต้น

xPC Target เป็นกล่องเครื่องมือหนึ่งที่มีอยู่ใน Simulink จะเป็นการประยุกต์ใช้งานโมเดลที่ได้สร้างโค๊ดภาษาซีจาก Real-Time Workshop รูปแบบหนึ่ง โดยไฟล์ที่ได้หลังจากการสร้างโค๊ด จะมีนามสกุล .dlm ซึ่งเป็นไฟล์เฉพาะของ xPC Target มีความสามารถเชื่อมต่อกับเครื่องคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไป ซึ่งจะมีเครื่องคอมพิวเตอร์ต้นทาง (Host PC) ควบคุมเครื่องคอมพิวเตอร์ปลายทาง (Target PC) ให้สามารถแลกเปลี่ยนข้อมูลกันได้ โดยจะรองรับการเชื่อมต่อ 2 รูปแบบคือ แบบที่หนึ่งเชื่อมต่อผ่านทางระบบการสื่อสารอนุกรม (Serial Communication) แบบที่สองเชื่อมต่อผ่านทางระบบเครือข่ายคอมพิวเตอร์ (Network Communication) คุณสมบัติพิเศษอีกประการหนึ่งก็คือ จะรองรับอุปกรณ์อินพุตและเอาต์พุต เพื่อเชื่อมต่อกับอุปกรณ์ภายนอกได้หลายชนิด โดยจะติดตั้งอยู่ที่ Target PC ดังนั้นนอกจากผู้ใช้จะสามารถควบคุมการทำงานของ Target PC จาก Host PC แล้ว ยังมีความสามารถที่จะควบคุมการทำงานของอุปกรณ์ภายนอกที่ Target PC ได้อีกด้วย ทำให้สามารถประยุกต์การใช้งานได้หลากหลาย

ลักษณะเด่นอีกประการหนึ่งของ xPC Target คือมีความต้องการของระบบ (System-Requirements) ทั้งซอฟต์แวร์ (Software Requirement) และฮาร์ดแวร์ (Hardware Requirement) ของ Target PC ต่ำมาก เช่น ที่ Target PC ไม่จำเป็นต้องมีระบบปฏิบัติการ (Operating System) ใดๆ เลยเนื่องจาก xPC Target จะสร้างระบบระบบปฏิบัติการของตนเองขึ้นมาใช้งาน เรียกว่า xPC-Target kernel

ลักษณะการเชื่อมต่อระหว่าง Host PC และ Target PC

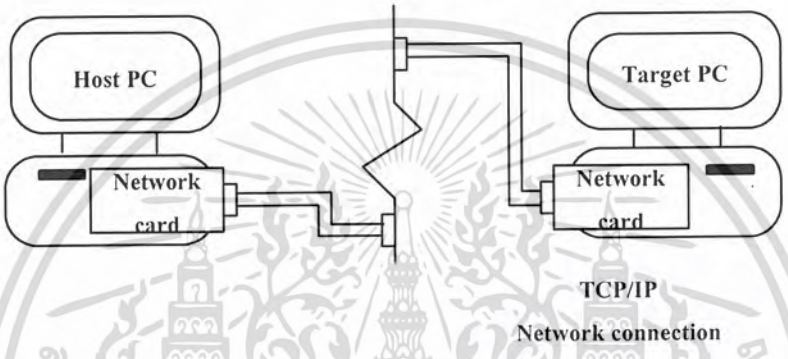
การสื่อสารแบบอนุกรม (Serial Communication) เป็นการติดต่อสื่อสารแบบอนุกรมโดยใช้พอร์ตอนุกรมของคอมพิวเตอร์ COM1 หรือ COM2 ซึ่งจะต้องมีสายสื่อสารอนุกรม โดยสามารถกำหนดความเร็วในการส่งถ่ายข้อมูลระหว่าง 1200 ถึง 115200 บอด (Baud) ลักษณะการเชื่อมต่อแสดงดังรูปที่ 4.1



รูปที่ 4.1 ลักษณะการเชื่อมต่อแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารแบบระบบเครือข่าย (Network Communication) เป็นการเชื่อมต่อกับเครือข่ายคอมพิวเตอร์ ซึ่งจะอาศัยโปรโตคอล TCP/IP ในการติดต่อสื่อสารระหว่าง Host PC และ Target PC เช่นในระบบเครือข่ายขนาดเล็ก (LAN : Local Area Network) หรือระบบเครือข่ายอินเตอร์เน็ต (Internet) เป็นต้น ซึ่งการเชื่อมต่อแบบนี้จะต้องมีการ์ด LAN (Ethernet Adapter Card) ที่ Host PC และ Target PC โดยชนิดของการ์ด LAN นั้นจะต้องเป็นการ์ด LAN ที่ทางโปรแกรม MATLAB รองรับด้วยลักษณะการเชื่อมต่อแสดงดังรูปที่ 4.2



รูปที่ 4.2 ลักษณะการเชื่อมต่อแบบระบบเครือข่าย

ขั้นตอนการทำงานของ xPC Target

การทำงานของ xPC Target นั้นจะแบ่งเป็นขั้นตอนหลักๆ คือ ออกแบบระบบ สร้างไฟล์ DLM สร้างแผ่นบูทระบบและเชื่อมต่อ Host PC กับ Target PC ซึ่งสามารถแสดงเป็นผังงานได้ดังรูปที่ 4.3

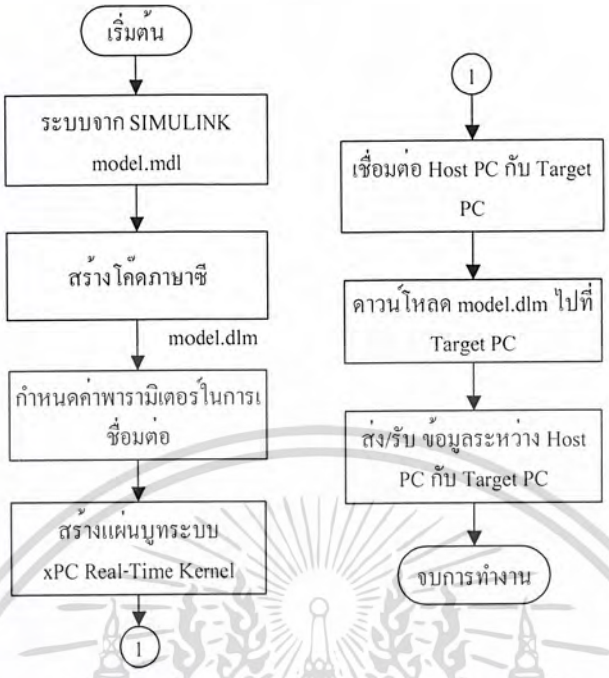
1. ออกแบบระบบจาก Simulink (System Modeling)

เป็นการออกแบบระบบหรือโมเดลที่ผู้ใช้ต้องการ ซึ่งจะทำการออกแบบใน Simulink โดยจะต้องคำนึงถึงการใช้งานที่ปลายทางด้วย เนื่องจากระบบที่ออกแบบจะนำไปใช้งานที่ปลายทาง ซึ่งการออกแบบจะมีคุณสมบัติพิเศษคือ บล็อกบางชนิดต้องเป็นบล็อกของ xPC Target เท่านั้น

2. สร้างโค๊ดภาษาซี (Generate C Code)

จะเป็นการสร้างโค๊ดภาษาซี ซึ่งขั้นตอนทั้งหมดจะอยู่ในการทำงานของ Real-Time Workshop ซึ่งผู้ใช้งานต้องกำหนดค่าพารามิเตอร์เป็นการติดต่อภายนอก คือขั้นตอนการเลือก TLC ต้องกำหนดเป็น xpctarget.tlc เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ขั้นตอนการทำงานของ xPC Target

3. กำหนดค่าพารามิเตอร์ในการเชื่อมต่อ (Configuration Parameters)

ในขั้นตอนนี้จะเป็นการกำหนดค่าพารามิเตอร์ในการเชื่อมต่อระหว่าง Host PC กับ Target โดยจะแบ่งเป็น 2 แบบ คือแบบที่หนึ่งกำหนดเป็นการสื่อสารอนุกรม จะต้องกำหนดค่าพารามิเตอร์ต่างๆ เช่น กำหนดพอร์ตในการสื่อสารและอัตราการรับส่งข้อมูล แบบที่สองจะเป็นการสื่อสารแบบระบบเครือข่าย จะต้องกำหนดค่าพารามิเตอร์ต่างๆ เช่น IP Address, Subnet Mask Address, Gateway Address และไดร์ฟเวอร์ (Driver) ของการ์ด LAN ที่ใช้งานที่ Target PC เป็นต้น

4. สร้างแผ่นบูทระบบ (Target Boot Disk)

เมื่อเสร็จสิ้นขั้นตอนการกำหนดค่าพารามิเตอร์แล้ว ขั้นตอนนี้จะเป็นการนำค่าพารามิเตอร์ที่กำหนดไว้ มาสร้างเป็นแผ่นบูทระบบ ซึ่งในแผ่นบูทระบบจะเก็บค่าข้อมูลที่สำคัญในการเรียกระบบปฏิบัติการ xPC Target Kernel ขึ้นมาใช้งาน

5. เชื่อมต่อ Host PCกับ Target PC (Host-Target Connection)

เมื่อที่ Target PC บูทระบบ xPC Target Kernel ขึ้นมาแล้วจะรอคำสั่งการทำงานจาก Host PC ซึ่งที่ Host PC ต้องกำหนดเริ่มการเชื่อมต่อกับ Target PC ก่อน ซึ่งที่ Host PC และ Target PC จะรายงานผลว่าการเชื่อมต่อนั้นสำเร็จหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ดาวน์โหลดไฟล์ DLM ไปที่ Target PC

เมื่อการเชื่อมต่อกับ Target PC สำเร็จที่ Host PC จะสามารถดาวน์โหลดไฟล์เฉพาะของ xPC Target ชื่อ model.dlm ให้ทำงานที่ Target PC ตามที่ได้ออกแบบไว้และสามารถปรับแต่งค่าพารามิเตอร์ต่างๆ ที่เวลาจริง (Real-Time) ได้อีกด้วย

7. รับ/ส่งข้อมูลระหว่าง Host PC กับ Target PC

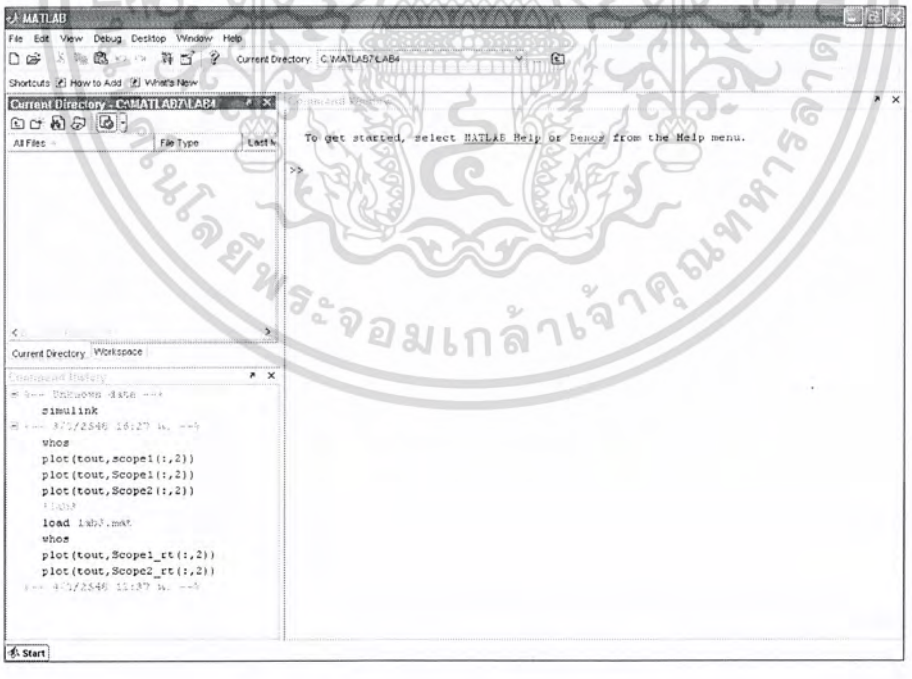
ในขณะที่โมเดลทำงานอยู่ที่ Target PC ระหว่าง Host PC และ Target PC จะสามารถที่จะแลกเปลี่ยนข้อมูลกันได้แบบเวลาจริง ขึ้นอยู่กับการใช้งาน

ลำดับขั้นการทดลอง

ลำดับขั้นตอนการทดลองจะแบ่งเป็น 2 ตอน คือ ตอนที่ 1 จะเป็นการเชื่อมต่อผ่านทางพอร์ตอนุกรมและตอนที่ 2 จะเป็นการเชื่อมต่อผ่านทางระบบเครือข่าย

ตอนที่ 1 การเชื่อมต่อกับ Target PC ผ่านทางพอร์ตอนุกรม RS232

1. เปิดโปรแกรม MATLAB เลือกไดเร็กทอรี่ที่ใช้งาน ในการทดลองนี้เลือกเป็น **C:\MATLAB7\LAB4** แสดงดังรูปที่ 4.4

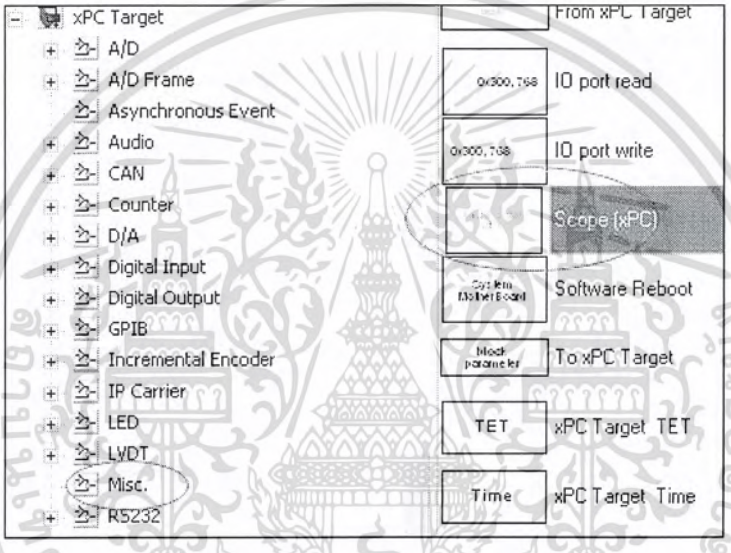


รูปที่ 4.4 หน้าต่างโปรแกรม MATLAB เวอร์ชั่น 7.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการเปิดกล่องที่เครื่องมือ Simulink โดยสามารถคลิกที่ หรือใช้คำสั่ง **Simulink** ที่ Command Window สร้างโมเดลใหม่ (New Model) รายละเอียดบล็อกต่างๆ มีดังนี้

- คลิกที่หัวข้อ **Source** เลือก **Signal Genrator**
- คลิกที่หัวข้อ **Continuous** เลือก **Transfer Fcn**
- คลิกที่หัวข้อ **Signal Routing** เลือก **Mux**
- คลิกที่หัวข้อ **Sinks** เลือก **Scope**
- คลิกที่หัวข้อ **xPC Target** หัวข้อย่อย **Misc** เลือก **Scope (xPC)** แสดงดังรูปที่ 4.5



รูปที่ 4.5 หน้าต่างการเลือก Scope (xPC)

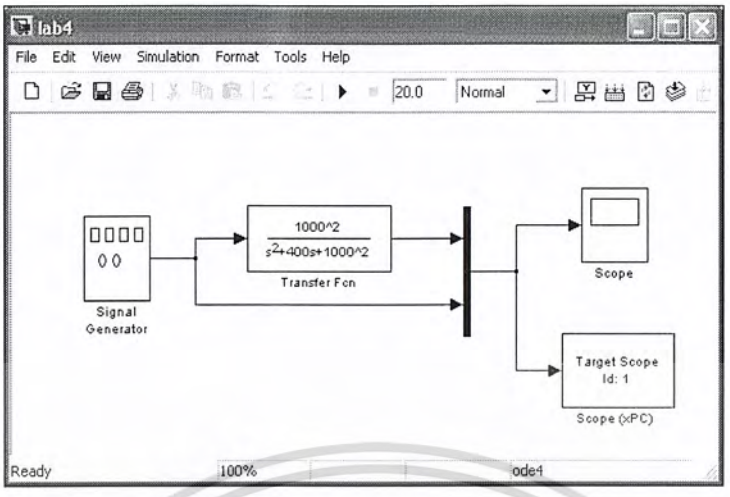
3. ลากเส้นเชื่อมต่อและตั้งชื่อบล็อกต่างๆ เสร็จแล้วบันทึกโมเดลตั้งชื่อเป็น **lab4** แสดงดังรูปที่ 4.6

4. กำหนดค่าพารามิเตอร์ของบล็อกต่างๆ ดังนี้

4.1 ดับเบิ้ลคลิกที่บล็อก **Signal Generator** กำหนดค่าดังนี้

- **Wave form** เลือกเป็น **Square**
- **Amplitude** กำหนดค่าเป็น **4V**
- **Frequency** กำหนดค่าเป็น **20Hz**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การลากเส้นเชื่อมต่อของโมเดล lab4

4.2 ดับเบิ้ลคลิกที่บล็อก Transfer Fcn กำหนดค่าดังนี้

- **Numerator** กำหนดค่าเป็น **[1000^2]**
- **Denominator** กำหนดค่าเป็น **[1 400 1000^2]**

4.3 ดับเบิ้ลคลิกที่บล็อก Scope กำหนดค่าดังนี้

- คลิกขวาที่พื้นที่ Scope เลือก **Axes Properties** จะแสดงหน้าต่าง Scope Properties กำหนดค่า **Y-min = -10** และ **Y-max = 10**
- คลิกที่ปุ่ม กำหนดค่าดังนี้

  ในหัวข้อ General กำหนดค่าดังนี้

  - **Time range** กำหนดค่าเป็น **0.2**
  - **Tick labels** เลือกเป็น **all**
  - **Sampling** เลือกเป็น **Sample time = 0**

  ในหัวข้อ Data history กำหนดค่าดังนี้

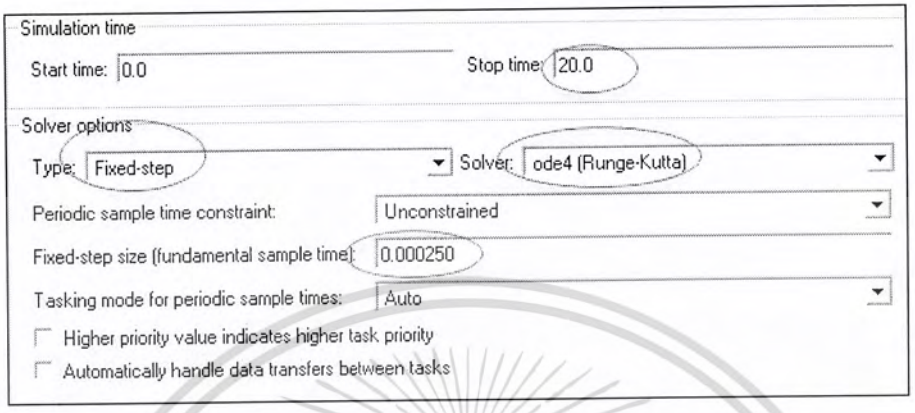
  - **ยกเลิก** การเลือกที่หัวข้อ **Limit data point to last**

4.4 ดับเบิ้ลคลิกที่บล็อก Scope (xPC) กำหนดค่าดังนี้

- **Scope type** กำหนดเป็น **Target**
- **Number of samples** กำหนดค่าเป็น **500**
- **Trigger mode** เลือกเป็น **Signal triggering**
- **Trigger slop** เลือกเป็น **Rising**

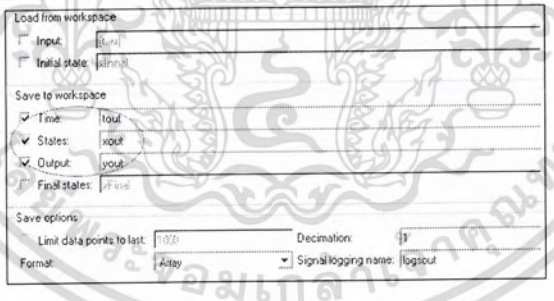
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เปิดหน้าต่าง Configuration Parameters โดยใช้คีย์ลัด Ctrl+E กำหนดค่าต่างๆ ดังนี้ในหัวข้อ **Solver** กำหนดค่าต่างๆ ดังรูปที่ 4.7



รูปที่ 4.7 หน้าต่างกำหนดค่าในหัวข้อ Solver Options

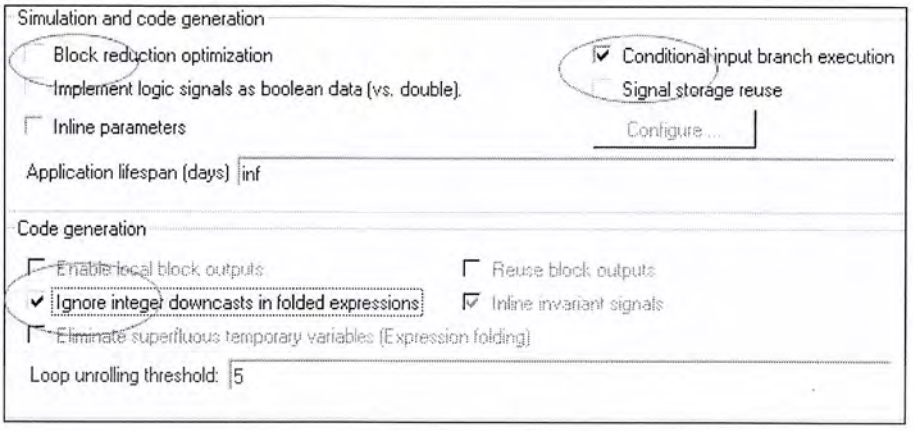
6. คลิกที่หัวข้อ **Data Import/Export** เป็นการกำหนดค่าพารามิเตอร์ในการส่งค่าข้อมูลเข้าและออกให้กับโปรแกรม MATLAB โดยข้อมูลที่รับเข้ามาจะอยู่ที่ MATLAB Workspace กำหนดค่าดังรูปที่ 4.8



รูปที่ 4.8 หน้าต่างกำหนดค่าในหัวข้อ Data Import/Export

7. คลิกที่หัวข้อ **Optimization** กำหนดค่าต่างๆ ดังรูปที่ 4.9

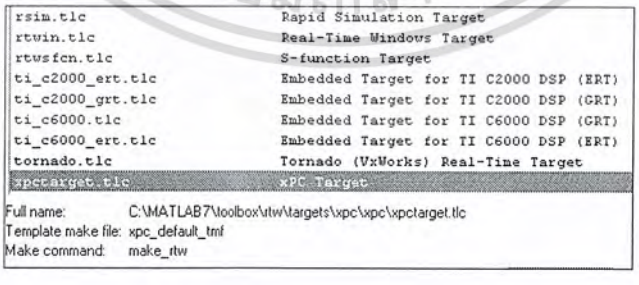
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 หน้าต่างกำหนดค่าในหัวข้อ Optimization

ในหัวข้อ Optimization นี้เป็นการกำหนดคุณสมบัติเพิ่มเติมให้กับการสร้างโค๊ดและการจำลองการทำงานทำให้มีประสิทธิภาพมากขึ้น เช่นในหัวข้อ Block reduction optimization เป็นการกำหนดว่าการสร้างโค๊ดจากบล็อกใดที่มีความซ้ำซ้อนกันก็ให้ใช้โค๊ดที่ได้สร้างไว้ก่อนหน้านี้ได้เลย ทำให้ช่วยลดเวลาในการจำลองการทำงานและการทำงานของโมเดลได้ แต่ในโมเดลนี้ไม่ได้กำหนด เนื่องจากโมเดลนี้ไม่ได้สร้างบล็อกที่มีความซ้ำซ้อนกัน ส่วนในหัวข้อ Condition input branch - execution เป็นการกำหนดจุดเชื่อมต่อที่มีความซ้ำซ้อนกันให้ใช้โค๊ดที่สร้างไว้แล้วได้เลย ซึ่งในการกำหนดค่า Optimization นี้จะมีผลต่อระบบใหญ่ๆ ที่มีความซ้ำซ้อนของบล็อกหรือจุดเชื่อมต่อมากๆ ซึ่งถ้าออกแบบระบบที่ไม่ซ้ำซ้อนมาก ก็ไม่ต้องกำหนดหรือกำหนดเฉพาะบางหัวข้อก็ได้

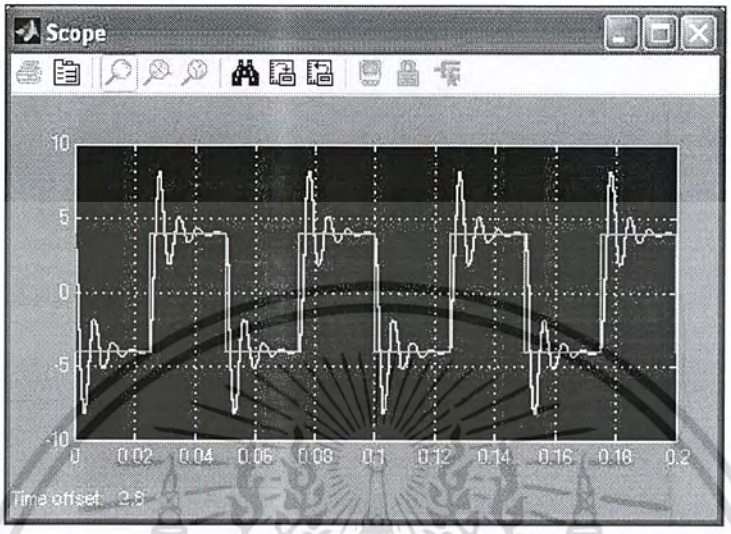
8. คลิกที่หัวข้อ Real-Time Workshop เป็นการกำหนดค่าพารามิเตอร์ในการสร้างโค๊ดภาษาซีโดยในหัวข้อ **Target Selection** คลิกปุ่ม **Browse** ให้เลือกเป็น **xpctarget.tlc** แสดงดังรูปที่ 4.10



รูปที่ 4.10 หน้าต่างการเลือกไฟล์ TLC

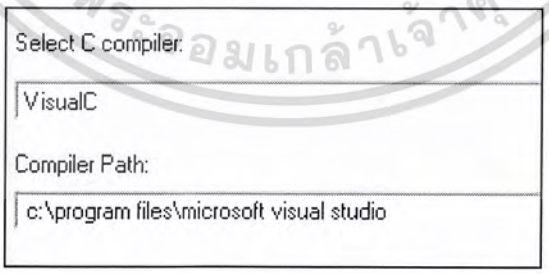
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. กลับมาที่หน้าต่าง lab4.mdl จำลองการทำงานของระบบโดยคลิกที่ปุ่ม ▶ และดับเบิ้ลคลิกที่บล็อก **Scope** ขึ้นมาจะแสดงผลดังรูปที่ 4.11



รูปที่ 4.11 ผลการจำลองการทำงานที่บล็อก Scope ของ lab4.mdl

10. ขั้นตอนต่อไปจะสร้างแผ่นบูทระบบ ที่หน้าต่าง MATLAB Command Window พิมพ์คำสั่ง >>**xpcexplr** จะแสดงหน้าต่าง **xPC Explorer** โดยจะเป็นหน้าต่างที่ควบคุมการเชื่อมระหว่าง Host กับ Target คลิกที่หัวข้อ **Compiler (s) Configuration** ในช่อง **Select C Compiler** เลือกเป็น **Visual C** ส่วนในช่อง **Compiler Parth** กำหนดเป็น **C:\program files\microsoft visual studio** แสดงดังรูปที่ 4.12



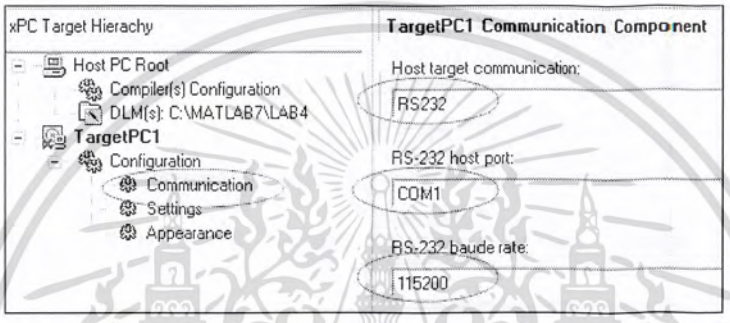
รูปที่ 4.12 การกำหนดโปรแกรมแปลภาษาของหน้าต่าง xPC Explorer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ทำการเพิ่ม Target PC คลิกที่ **File > Add Target** โดยการเพิ่ม Target PC นั้นสามารถที่จะเพิ่มจำนวน Target PC ที่ต้องการติดต่อได้มากกว่า 1 ตัว ซึ่งต้องกำหนดการเชื่อมต่อ Target PC แต่ละตัวให้แตกต่างกัน เช่น ตัวที่ 1 เชื่อมต่อแบบ RS232 และตัวที่ 2 เชื่อมต่อแบบระบบเครือข่าย โดยการกำหนดจะอยู่ในขั้นตอนต่อไป

12. จะปรากฏ **Target PC1** เพิ่มเข้ามาในหน้าต่างย่อย xPC Target Hierarchy กำหนดค่าดังนี้

- คลิกที่หัวข้อ **Communication** กำหนดค่า ดังรูปที่ 4.13



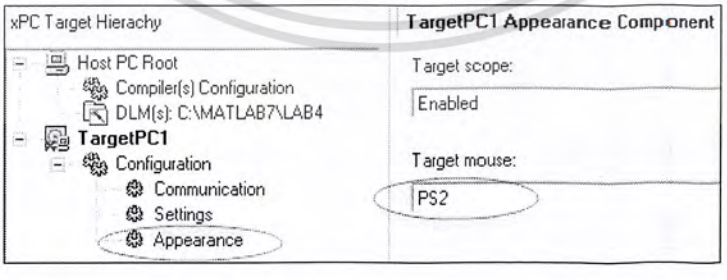
รูปที่ 4.13 หน้าต่างกำหนดค่าการเชื่อมต่อในหัวข้อ Communication

Host target communication เป็นการกำหนดลักษณะการเชื่อมต่อสามารถกำหนดได้ 2 แบบ คือ RS232 และ Tcp/Ip

RS-232 host port เป็นการกำหนดพอร์ตอนุกรมที่ใช้เชื่อมต่อกับ Target PC

RS-232 bauad rate เป็นการกำหนดอัตราการรับส่งข้อมูลระหว่าง Host กับ Target

- คลิกที่หัวข้อ **Appearance** เลือกที่ **Target Mouse** เป็น **PS2** แสดงดังรูปที่ 4.14



รูปที่ 4.14 หน้าต่างกำหนดค่าการเชื่อมต่อในหัวข้อ Appearance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Target scope เป็นการเปิดใช้งาน Scope ที่ Target PC ให้สามารถใช้งานได้

Target mouse เป็นการเลือกลักษณะของ mouse ที่ต่ออยู่ที่ Target PC โดยในหัวข้อนี้กำหนดเป็น **PS2**

13. นำแผ่นฟล็อปปี้ดิสก์เปล่าใส่ในไดร์ฟ A:/ เพื่อทำการสร้างแผ่นบูทระบบ คลิกที่หัวข้อ **Configuration** โดยเลือกเป็น **BootFloppy** คลิกที่ปุ่ม **Create BootDisk** จะแสดงขั้นตอนการสร้าง Target BootDisk ดังรูปที่ 4.15

รูปที่ 4.15 หน้าต่างการสร้าง Target BootDisk

14. เมื่อการสร้างแผ่นบูทเสร็จสิ้นนำแผ่นบูทที่สร้างขึ้นออก แล้วนำไปใส่ไดร์ฟ A:/ ที่ Target PC โดยจะต้องกำหนดให้ Target PC บูทจากแผ่นไดร์ฟ A:/ (ซึ่งต้องกำหนดค่าใน BIOS-ก่อนบูทเครื่อง Target PC) และ**ตรวจสอบการเชื่อมต่อสายสื่อสารอนุกรม RS-232** ระหว่าง Host PC และ Target PC **เปิดเครื่อง Target PC** สังเกตว่าจะบูทระบบ xPC Target Kernel ขึ้นมาจากแผ่นดิสก์ แสดงดังรูปที่ 4.16

15. ขั้นตอนต่อไปทำการติดต่อ (Connect) ระหว่าง Host PC กับ Target PC สามารถทำได้ 2 วิธีคือ

1. คลิกขวาที่ **Target PC1** เลือก **Connect**
2. คลิกที่ปุ่ม **Connect** บนเมนูบาร์ด้านบน

16. สังเกตการเปลี่ยนแปลงที่ **Host PC** ในหัวข้อ **TargetPC1 Target Status Display** บันทึกผลการทดลองในตารางที่ 4.1

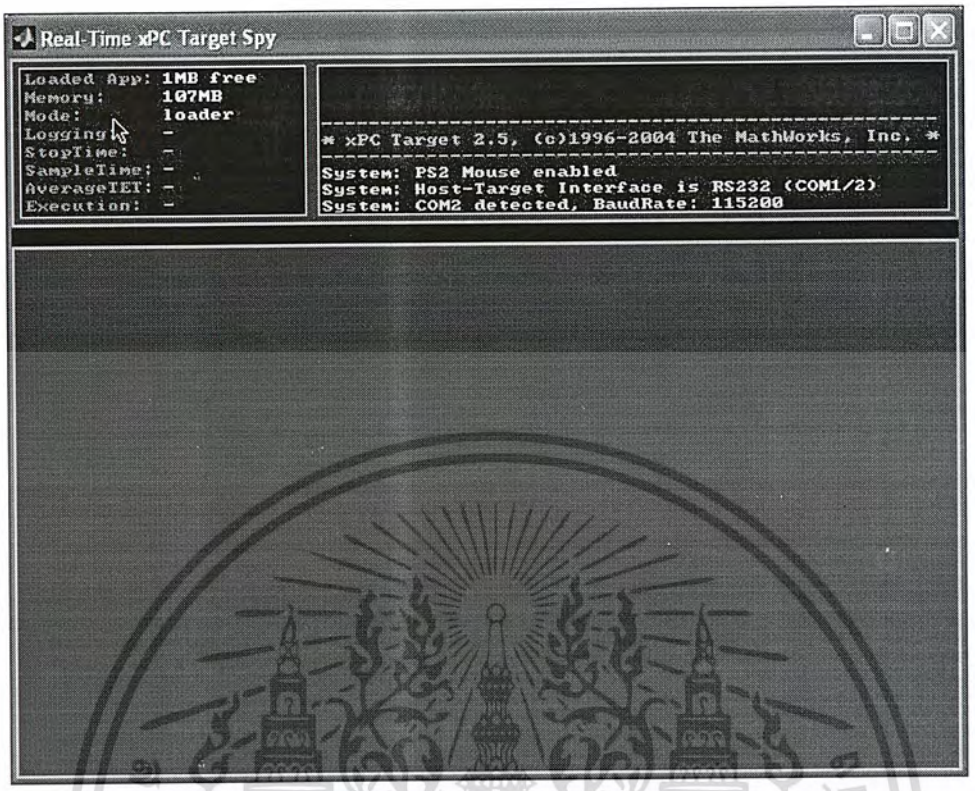
17. สังเกตการเปลี่ยนแปลงที่ Target PC ก่อนและหลังการติดต่อ (Connect) ว่ามีลักษณะเปลี่ยนแปลงอย่างไร บันทึกผลการทดลอง

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ระบบปฏิบัติการ xPC Target Kernel ที่ Target PC

ตารางที่ 4.1 สถานะของเครื่อง Target PC ที่เครื่อง Host PC

TargetPC1 Target Status Display	
Loaded App	
Mode	
Logging	
Stop Time	
Sample Time	
Average TET	
Execution Time	
Target Name	
Connected	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18. กลับมาที่หน้าต่าง lab4.mdl ทำการสร้างโค๊ดภาษาซี โดยเปิดหน้าต่าง **Configuration Parameters** ในหัวข้อ **Real-Time Workshop** ตรวจสอบดูว่า **Target Selection** ต้องเลือกเป็น **xpctarget.tlc** เมื่อถูกต้องคลิกที่ปุ่ม **Build** เพื่อเริ่มขั้นตอนการสร้างไฟล์โค๊ดภาษาซี

19. เมื่อเริ่มขั้นตอนการสร้างโค๊ดภาษาซี ที่หน้าต่างโปรแกรม MATLAB จะแสดงขั้นตอนต่างๆ และทดสอบการเชื่อมต่อกับ Target PC ด้วย บันทึกผลการทดลอง

Starting xPC Target build procedure for model: _____.

Generating code into build directory: C:\MATLAB7\LAB4_____.

Invoking Target Language Compiler on _____.

Looking for target

create xPC Object tg

Download Model onto Target

xPC Object

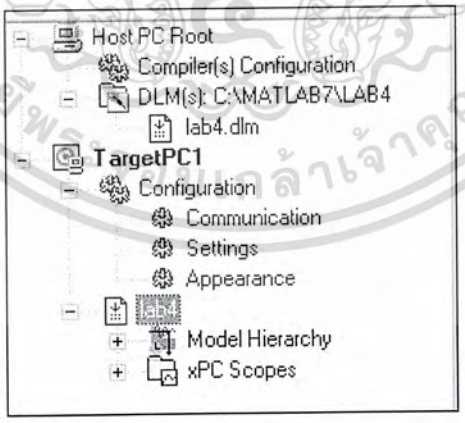
Connected = _____.

เมื่อเสร็จสิ้นจะปรากฏข้อความ

Successful completion of xPC Target build procedure for model:_____.

Successful completion of Real-Time Workshop build procedure for model:_____.

20. กลับมาที่หน้าต่าง **xPC Target Explorer** จะแสดงการเชื่อมต่อกับ Target โดยจะเห็นว่ามีไฟล์ lab4.dlm รอที่จะดาวน์โหลดอยู่ แสดงดังรูปที่ 4.17



รูปที่ 4.17 ไฟล์ lab4.dlm รอดาวน์โหลดอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกตว่าไฟล์ที่สามารถดาวน์โหลดไปที่ Target PC ได้นั้นจะเป็นไฟล์ที่มีนามสกุล .dlm ซึ่งเป็นไฟล์เฉพาะของ xPC Target เท่านั้น

21. ดาวน์ไฟล์ lab4.dlm ให้กับ Target PC โดยคลิกที่ปุ่ม ▶ หรือ**คลิกขวาที่ lab4** เลือก **Start** สังเกตการเปลี่ยนแปลงที่ Host PC และ Target PC บันทึกผลการทดลอง

21.1 บันทึกผลการทดลองในตารางที่ 4.2 ของเครื่อง Host PC

ตารางที่ 4.2 สถานะของเครื่อง Target PC ที่เครื่อง Host PC

TargetPC1 Target Status Display	
Loaded App	
Mode	
Logging	
Stop Time	
Sample Time	
Average TET	
Execution Time	
Target Name	
Connected	

21.2 ที่ Target PC จะแสดงผลดังรูปที่ 4.18

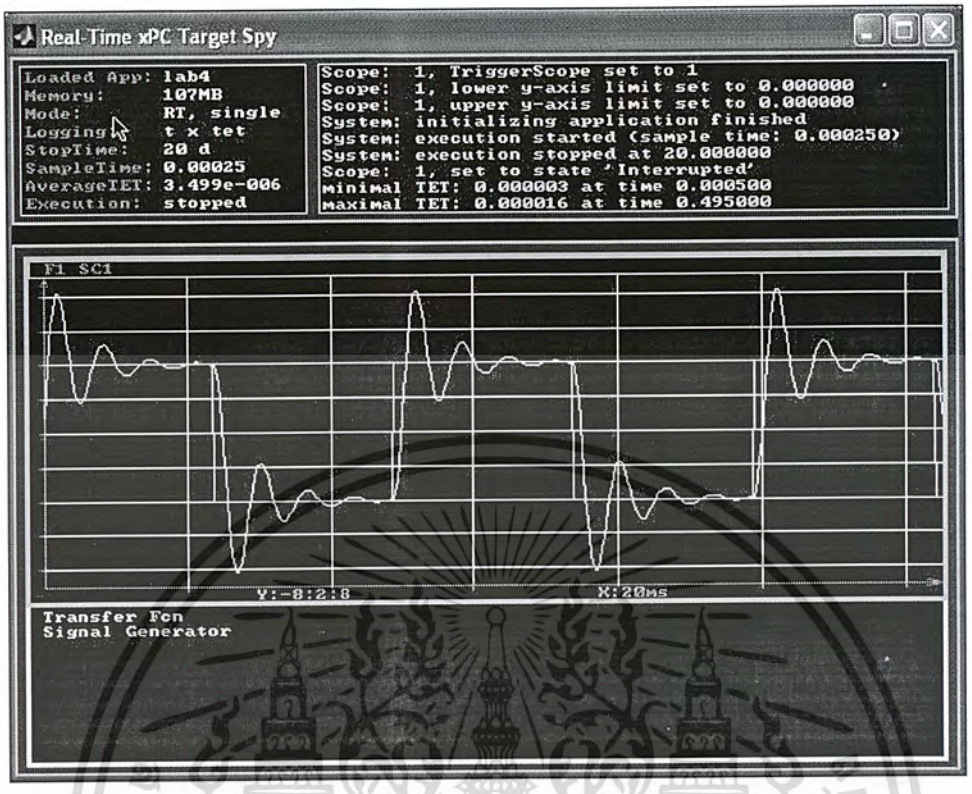
22. ที่ Target PC หลังการทำงานของสัญญาณสิ้นสุดลง ที่ด้านบนซ้ายของ Target PC ในหัวข้อ **Execution** จะเห็นว่ามีคำว่า **Stopped** แสดงอยู่ ใช้เมาส์**คลิกที่ คำว่า Stopped** และเมื่อเวลาผ่านไปสักครู่ใช้เมาส์**คลิกขวาที่**ตำแหน่งเดิมสังเกตการเปลี่ยนแปลงที่ Scope บันทึกผลการทดลอง

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 ผลของ Target PC หลังดาวน์โหลดไฟล์ lab4.dlm

23. บริเวณพื้นที่ Scope ใช้เมาส์คลิกขวาและคลิกซ้าย สังเกตผลที่เกิดขึ้น บันทึกผลการทดลอง

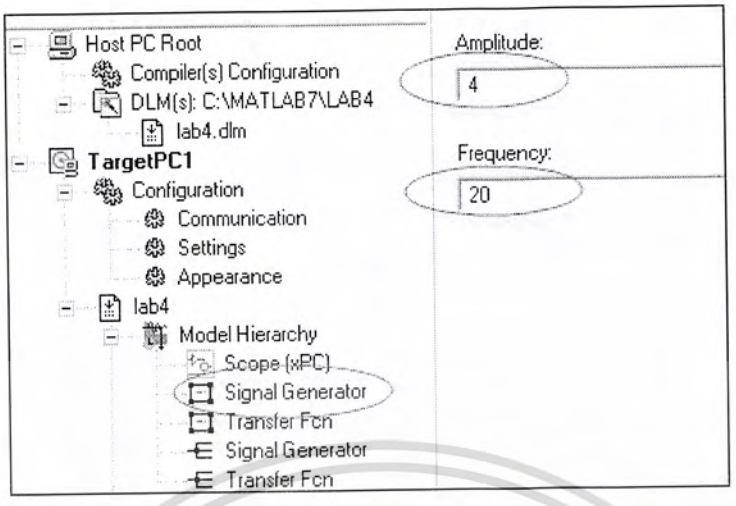
.....

.....

.....

24. ขั้นตอนต่อไปจะทดลองเปลี่ยนแปลงค่าพารามิเตอร์ที่ Host PC โดยคลิกที่ **Signal Generator** จะเห็นว่าสามารถเปลี่ยนแปลงค่าพารามิเตอร์ได้ 2 ค่า คือ **Amplitude** และ **Frequency** ดังรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 หน้าต่างกำหนดค่าพารามิเตอร์ของ Signal Generator

25. ทดลองเปลี่ยนแปลงค่าพารามิเตอร์ดังนี้

- ในช่อง **Amplitude** กำหนดค่าเป็น **5 V**
- ในช่อง **Frequency** กำหนดค่าเป็น **50 Hz**

คลิกปุ่ม ▶ สังเกตผลการเปลี่ยนแปลงที่ Target PC จะแสดงผลการเปลี่ยนแปลงดังรูปที่ 4.20

จงคำนวณหาค่า Amplitude และ Frequency ของสัญญาณ Signal Generator เส้นสีฟ้า (หมายเหตุ **Y:-10:5:10** หมายความว่าค่าของสเกลอยู่ระหว่าง -10 ถึง 10 V แต่ละช่องมีค่าเท่ากับ 5V)

.....

.....

.....

.....

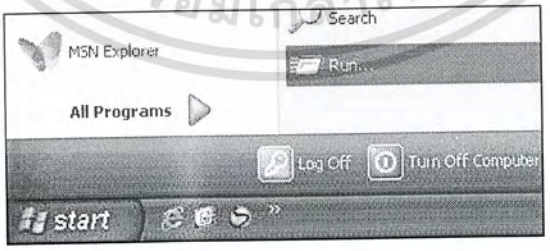
.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.20 แสดงผลที่ Target PC หลังการเปลี่ยนแปลงค่าพารามิเตอร์

ตอนที่ 2 การเชื่อมต่อผ่านทางระบบเครือข่าย

1. ขั้นตอนแรกที่ต้องเข้าระบบ Windows ที่ Target PC ก่อนแล้วจะต้องตรวจสอบค่า IP Address ที่ Target PC คลิกที่ปุ่ม **Start Menu** เลือกที่ **Run** แสดงดังรูปที่ 4.21 พิมพ์คำสั่ง **cmd** คลิก **Ok** แล้วจะปรากฏหน้าต่าง Command DOS พิมพ์คำสั่ง **ipconfig/all** เพื่อดูค่าพารามิเตอร์ต่างๆ ที่เกี่ยวกับการเชื่อมต่ออินเตอร์เน็ต บันทึกผลการทดลองในตารางที่ 4.3



รูปที่ 4.21 การเปิดหน้าต่าง Command DOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

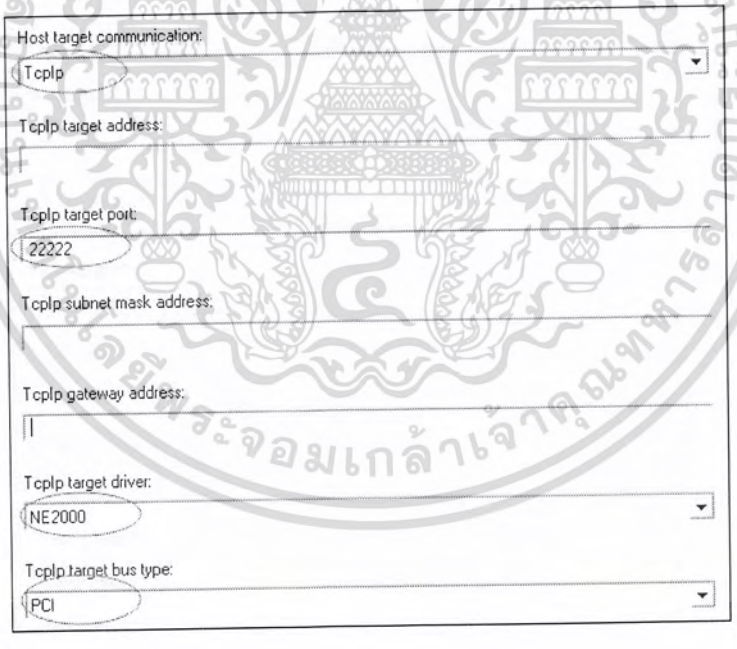
ตารางที่ 4.3 ค่าพารามิเตอร์ในการเชื่อมต่อกับระบบเครือข่าย

Description	
IP Address	
Subnet Mask	
Default Gateway	

2. การเชื่อมต่อแบบระบบเครือข่ายนี้ขั้นตอนจะแตกต่างกับการเชื่อมต่อแบบอนุกรมในขั้นตอนที่ 12-15 เท่านั้น ส่วนขั้นตอนอื่นๆ เหมือนเดิม

3. ตรวจสอบการเชื่อมต่อกับระบบเครือข่ายของ Host PC และ Target PC ว่าได้เชื่อมต่อเรียบร้อยแล้ว

4. เปลี่ยนแปลงการกำหนดค่าพารามิเตอร์ในขั้นตอนที่ 12 และกำหนดค่าพารามิเตอร์ของการเชื่อมต่อกับระบบเครือข่ายตามที่ได้บันทึกผลไว้และกำหนดค่าอื่นๆ แสดงดังรูปที่ 4.22



รูปที่ 4.22 หน้าต่างกำหนดค่าการเชื่อมต่อในหัวข้อ Communication

Host target communication เป็นการกำหนดลักษณะการเชื่อมต่อโดยกำหนดเป็น **Tcp/Ip**

Tcp/Ip target address เป็นหมายเลข IP Address ของ Target PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tcp/Ip subnet maskt address เป็นหมายเลข IP Address ของ Target PC

Tcp/Ip gateway address เป็นหมายเลข Subnet Mask ของ Target PC

Tcp/Ip target driver เป็นกำหนด Driver ของการ์ด LAN ที่ใช้ในการเชื่อมต่อ โดยที่ Target PC ใช้การ์ด LAN ของ SMC EZ 1208 ซึ่งต้องกำหนด Driver เป็น **NE200**

Tcp/Ip target bus type เป็นการกำหนดระบบบัสของการ์ด LAN ที่ใช้โดยกำหนดเป็น PCI

5. ทำซ้ำในขั้นตอนอื่นๆ สังเกตผลการทดลองเปรียบเทียบกับผลที่ได้ในตอนแรก บันทึกผลการเปลี่ยนแปลงที่เกิดขึ้น

.....

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. จงอธิบายขั้นตอนการทำงานของ xPC Target มาพอสังเขป?

.....

.....

.....

.....

.....

2. ในกรณีที่มีความต้องการติดต่อกับคอมพิวเตอร์ปลายทาง (Target PC) มากกว่า 1 ตัว จะมีวิธีการกำหนดอย่างไร?

.....

.....

.....

.....

.....

3. จากการสังเกตสัญญาณที่คอมพิวเตอร์ปลายทาง (Target PC) จะเห็นว่าสัญญาณจะทำงานที่ช่วงเวลาหนึ่งเท่านั้นแล้วก็จะหยุดลง ช่วงเวลาที่สัญญาณทำงานคือเท่าใดและถ้าต้องการเปลี่ยนแปลงช่วงเวลานี้สามารถกำหนดได้อย่างไร จงอธิบาย?

.....

.....

.....

.....

.....

4. จงอธิบายความแตกต่างระหว่างการกำหนดค่าเพื่อเชื่อมต่อแบบ RS232 และระบบเครือข่าย?

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 5

เรื่อง การใช้งาน xPC Target ติดต่อกับพอร์ตอนุกรม (RS-232)

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. อธิบายขั้นตอนการทำงานของ xPC Target ในการติดต่อกับพอร์ตอนุกรมได้อย่างถูกต้อง
2. แสดงการกำหนดค่าพารามิเตอร์ในการเชื่อมต่อกับพอร์ตอนุกรมได้อย่างถูกต้อง
3. อธิบายผลที่เกิดขึ้นเมื่อกำหนด TLC แบบ xPC Target ได้อย่างถูกต้อง
4. อธิบายผลที่เกิดขึ้นเมื่อมีการรับส่งข้อมูลกับพอร์ต RS-232 ได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0
2. กล่องเครื่องมือ Simulink เวอร์ชั่น 6.0
3. กล่องเครื่องมือ Real-Tinme Workshop เวอร์ชั่น 6.0
4. โปรแกรม Microsoft Visual C/C++ เวอร์ชั่น 6.0
5. เครื่องคอมพิวเตอร์ Host PC และ Target PC ที่เชื่อมต่อกับระบบเครือข่าย
6. สายสื่อสารอนุกรม RS 232 (Serial Communication Cable) จำนวน 1 เส้น
7. แผ่นฟล็อปปี้ดิสก์เปล่าขนาด 3.5 นิ้ว จำนวน 1 แผ่น

ความรู้และทักษะเดิม

1. การใช้งานโปรแกรม MATLAB และ Simulink เบื้องต้น
2. การใช้งาน Real-Tinme Workshop เวอร์ชั่น 6.0
3. การใช้งาน xPC Target

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การใช้งาน xPC Target นั้นมีคุณสมบัติพิเศษคือที่ Target PC นั้นสามารถที่จะเชื่อมต่อกับอุปกรณ์ภายนอกได้โดยผ่านทางระบบบัสต่างๆ ของคอมพิวเตอร์ ตัวอย่างเช่น การ์ด D/A หรือ A/D โดยการ์ดนั้นอาจต่ออยู่ที่ Slot PCI เพื่อที่จะเชื่อมต่อกับสัญญาณภายนอกที่เป็น Analog หรือใช้ติดต่อกับอุปกรณ์ภายนอกโดยผ่านพอร์ต RS-232/422/485 เป็นต้น โดยในการใช้งานนั้นจะมีข้อจำกัดที่สำคัญคือ การ์ดต่างๆ ที่นำมาใช้นั้นจะต้องเป็นของบริษัทผู้ผลิตที่ xPC Target รองรับด้วย ตัวอย่างเช่น การ์ด A/D ของบริษัท National Instruments เป็นต้น แต่การติดต่อกับอุปกรณ์ภายนอกผ่านทางพอร์ต RS-232 นั้นไม่จำเป็นต้องใช้การ์ด RS-232 เพราะคอมพิวเตอร์ทั่วไปจะมีพอร์ตนี้อยู่แล้ว โดยในใบงานนี้จะแสดงวิธีการใช้งาน xPC Target ติดต่อผ่านพอร์ต RS-232

ลักษณะการเชื่อมต่อระหว่าง Target PC และอุปกรณ์ภายนอก

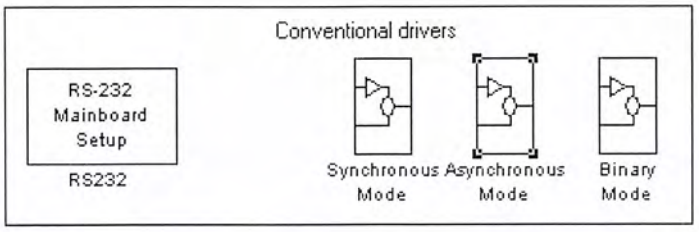
ลักษณะการเชื่อมต่ออุปกรณ์ภายนอกผ่านทางพอร์ต RS-232 แสดงดังรูปที่ 5.1 โดยจะต้องกำหนดค่าพารามิเตอร์ที่สำคัญได้แก่ อัตราในการรับส่งข้อมูล, พอร์ตที่ใช้งาน, จำนวนบิตข้อมูล และขนาดของ Buffer ที่ใช้ในการรับและส่งข้อมูล ทั้งนี้การกำหนดค่าพารามิเตอร์ต่างๆ นั้นจะต้องคำนึงถึงลักษณะของอุปกรณ์ที่จะเชื่อมต่อด้วย ดังรูปที่ 5.1

รูปที่ 5.1 ลักษณะการเชื่อมต่อกับอุปกรณ์ภายนอกผ่านพอร์ต RS-232

การกำหนดค่าพารามิเตอร์ในการติดต่อกับพอร์ต RS-232

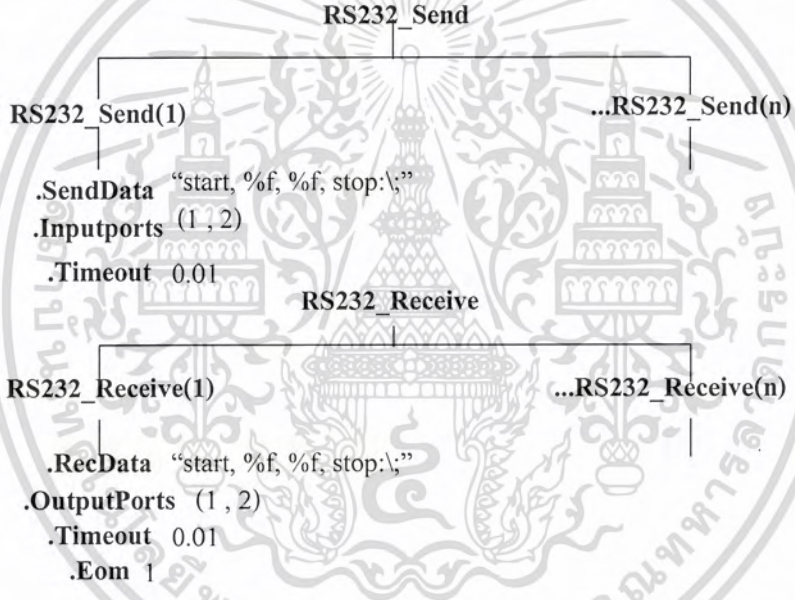
การติดต่อกับอุปกรณ์ภายนอกนั้นจะมีหลายแบบ ซึ่งการใช้งานต้องกำหนดด้วยว่าจะใช้งานในรูปแบบใด เช่น Synchronous Mode, Asynchronous Mode หรือ Binary Mode โดยสามารถกำหนดได้จาก Block ต่างๆ แสดงดังรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 บล็อกที่กำหนดลักษณะการเชื่อมต่อแบบ RS-232

ตัวอย่างการเขียน M-file เพื่อใช้กำหนดค่าในการรับ/ส่งข้อมูลผ่านพอร์ต RS-232 ดังรูปที่ 5.3



รูปที่ 5.3 การเขียน M-file เพื่อใช้กำหนดค่าในการรับและส่งข้อมูลผ่านพอร์ต RS-232

SendData , RecData เป็นการกำหนดลักษณะของข้อมูลที่จะส่งและรับ โดยข้อมูลที่จะส่งจะอยู่ในเครื่องหมาย '__' ซึ่งเป็นคำสั่งเฉพาะของโปรแกรม MATLAB เช่น %f, %f มีความหมายว่าต้องการส่งหรือรับข้อมูลที่เป็น Floating Point จำนวน 2 ข้อมูล

InputPorts, OutputPorts เป็นการกำหนดจำนวนพอร์ตที่ใช้ในการรับส่งข้อมูล

Timeout เป็นการกำหนดฐานเวลาในการรับส่งข้อมูล

EOM (End Of Message) เป็นการกำหนดจำนวนของข้อมูลเมื่อส่งหรือรับข้อมูลเสร็จสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

การทดลองนี้จะเป็นการติดต่อกับพอร์ต RS-232 แบบ Asynchronous Mode ซึ่งจะไม่มีการเชื่อมต่อกับอุปกรณ์ภายนอก แต่จะทดสอบการส่งข้อมูลออกที่พอร์ต COM1 และรับข้อมูลกลับมาที่พอร์ต COM2 แล้วแสดงผลสัญญาณทั้งสองที่ Target xPC Scope เพื่อเปรียบเทียบสัญญาณทั้งสอง โดยข้อมูลที่ส่งออกจะเป็นสัญญาณ Sine Wave ที่ถูกการแปลงเป็นข้อมูลแบบ Floating Point ก่อนที่จะส่งออกพอร์ต COM1 และข้อมูลที่รับเข้ามาที่พอร์ต COM2 จะผ่านบล็อก Gain ที่มีค่า = -1 หมายถึงการคูณข้อมูลที่รับเข้ามาด้วยค่าคงที่ -1 ดังนั้นผลที่เกิดขึ้นที่ Target xPC Scope คือจะได้สัญญาณ Sine Wave กลับเฟส 180 องศา เมื่อเทียบกับข้อมูลที่ส่งออกจาก COM1 โดยลำดับขั้นตอนการทดลองมีดังนี้

1. เปิดโปรแกรม MATLAB เวอร์ชั่น 7.0 เลือกไดเร็กทอรี่ที่ใช้งานเป็น **C:\MATLAB7\LAB5**

2. เปิดกล่องเครื่องมือ Simulink โดยคลิกที่ หรือใช้คำสั่ง **Simulink** ที่ Command Window สร้างโมเดลใหม่ (New Model) รายละเอียดบล็อกต่างๆ มีดังนี้ (*หมายเหตุการณ์กำหนดค่าพารามิเตอร์นั้นสามารถทำได้โดย ดับเบิ้ลคลิกที่บล็อกที่ต้องการกำหนดค่า)

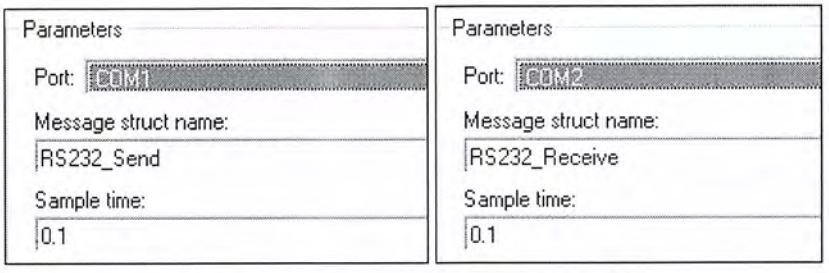
- คลิกที่หัวข้อ **Commonly Used Blocks** เลือก **Gain** กำหนดค่า **Gain = -1**
- คลิกที่หัวข้อ **Signal Routing** เลือก **Mux** กำหนดค่าเป็น **2 input** แบบ **bar**
- คลิกที่หัวข้อ **Sink** เลือก **Scope** กำหนดดังนี้

1. **คลิกขวาที่พื้นที่** Scope ขึ้นมาคลิกขวา เลือก **Axes Properties** กำหนดค่า **Ymin = -1 , Ymax = 1**

2. คลิกที่ กำหนดค่า **Time Range = 10** และ **Sampling** กำหนดเป็น **Sample time**

- คลิกที่หัวข้อ **Source** เลือก **Sine Wave** กำหนดค่า **Frequency** เป็น **5 Rad/s**
- คลิกที่หัวข้อ **xPC Target** หัวข้อย่อย **Misc** เลือก **Scope (xPC)** จำนวนกำหนดค่า **Scope Type** เป็น **Target**
- คลิกที่หัวข้อ **xPC Target** หัวข้อย่อย **RS-232** คลิกที่ **Asynchronous Mode** เลือก **RS-232** และเลือกบล็อก **RS-232 1** กำหนดค่าที่บล็อก Send และ Receive แสดงดังรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) (ข)

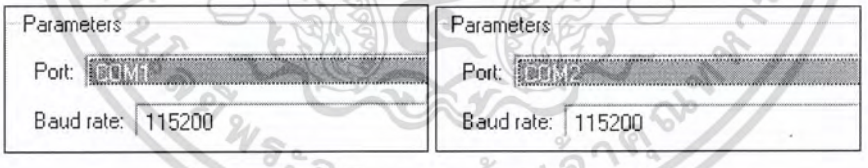
รูปที่ 5.4 การกำหนดค่าบล็อก RS232 Send (ก) และ RS232 Receive (ข)

Port เป็นการกำหนดพอร์ต RS232 ที่ใช้ในการเชื่อมต่อ

Massage struct name เป็นการกำหนดชื่อตัวแปรที่ใช้ในการเก็บค่าพารามิเตอร์ในการเชื่อมต่อ โดยจะต้องทำการเขียน M-file ให้สอดคล้องกับการตั้งชื่อตัวแปรนี้ รายละเอียดจะกล่าวในหัวข้อต่อไป

Sample Time เป็นฐานเวลาในการติดต่อสื่อสาร

- คลิกที่หัวข้อ **xPC Target** หัวข้อย่อย **RS-232** เลือก **RS-232** จำนวน **2** บล็อก โดยบล็อกที่ 1 จะกำหนดค่าพารามิเตอร์ของบล็อก **Send** ที่ต่ออยู่ที่ **Com1** และบล็อกที่ 2 จะกำหนดค่าพารามิเตอร์ของบล็อก **Receive** ที่ต่ออยู่ที่ **COM2** รายละเอียดการกำหนดค่าแสดงดังรูปที่ 5.5 ส่วนค่าอื่นๆ ให้ใช้ค่าเดิมที่ตั้งไว้



(ก) (ข)

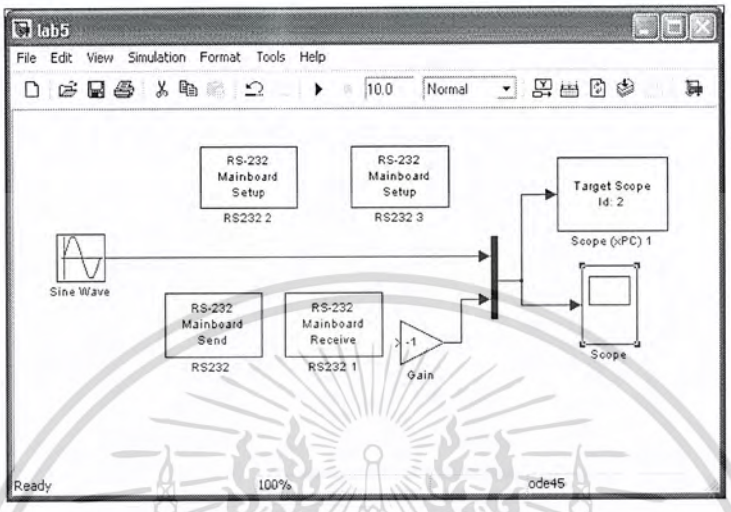
รูปที่ 5.5 การกำหนดค่าบล็อก RS232 Send Setup (ก) และ RS232 Receive Setup (ข)

Port เป็นการกำหนดพอร์ตที่ใช้งานซึ่งต้องตรงกับบล็อก Send และ Receive

Baud rate กำหนดอัตราการรับส่งข้อมูลที่พอร์ต RS232 มีหน่วยเป็นบอด (Baud) ในที่นี้กำหนดไว้สูงสุดคือ 115200 บอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ลากเส้นเชื่อมต่อและตั้งชื่อบล็อกต่างๆ เสร็จแล้วบันทึกโมเดลตั้งชื่อเป็น **lab5** แสดงดังรูปที่ 5.6



รูปที่ 5.6 การลากเส้นเชื่อมต่อบล็อกต่างๆ ของโมเดล lab5

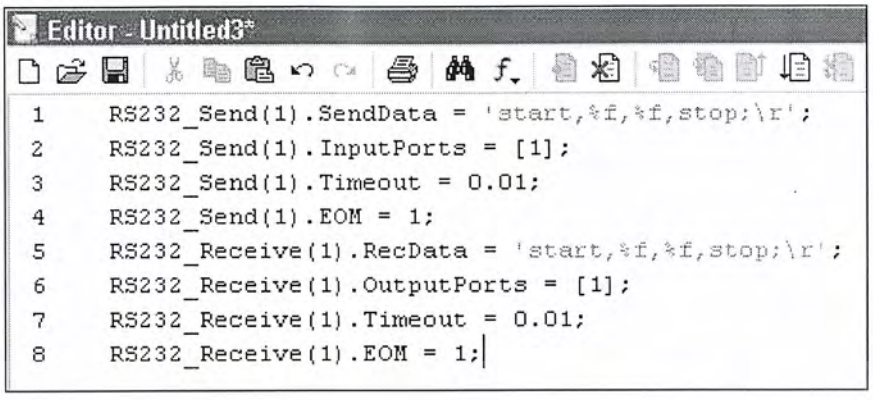
จากการลากเส้นเชื่อมต่อจะสังเกตว่าที่บล็อก RS232 Send จะไม่มีจุดเชื่อมต่ออินพุตและที่บล็อก RS232 Receive จะไม่มีจุดเชื่อมต่อเอาต์พุต เนื่องจากต้องมีการกำหนดค่าพารามิเตอร์สำหรับการเชื่อมต่อก่อน โดยจะต้องเขียนโปรแกรมเป็น M-file กำหนดให้บล็อก Send และ Receive อ่านค่าพารามิเตอร์จาก M-file นี้เมื่อสร้าง M-file แล้วที่บล็อก Send และ Receive จะมีจุดเชื่อมต่อปรากฏขึ้น ขั้นตอนการเขียน M-file แสดงในขั้นตอนถัดไป

4. กลับมาที่หน้าต่างโปรแกรม MATLAB บนเมนูบาร์ด้านบน คลิกที่ **File > New > M-file** หรืออาจจะใช้คีย์ลัด **Ctrl+N** ก็ได้ จะปรากฏหน้าต่าง Editor ขึ้นมาเป็นหน้าต่างที่เอาไว้เขียนโปรแกรมที่เป็น M-File เขียนโปรแกรมดังนี้

เมื่อเขียนโปรแกรมเสร็จสิ้น บนเมนูบาร์ด้านบน ให้ทำการคลิกที่ **File > Save As** ตั้งชื่อเป็น **RS232Async_Messages.m** บันทึกที่ **C:\MATLAB7\LAB5** เมื่อเสร็จสิ้นจะแสดงดังรูปที่ 5.8 และทำการปิดหน้าต่าง Editor

5. พิมพ์คำสั่ง **>>clear all** เพื่อลบข้อมูลทั้งหมดที่มีอยู่ใน Workspace พิมพ์คำสั่ง **>> whos** เพื่อดูข้อมูลใน Workspace จะสังเกตเห็นว่าจะไม่มีข้อมูลใดๆ อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 การเขียน M-file ที่ใช้กำหนดค่าให้กับบล็อก Send และ Receive

รูปที่ 5.8 ผลจากการบันทึกไฟล์ RS232 Async_Messages.m

6. กลับมาที่หน้าต่างโปรแกรม MATLAB พิมพ์คำสั่ง >>**RS232Async_Messages** เพื่อเป็นการเรียกไฟล์ RS232Async_Messages.m โดยเมื่อไฟล์นี้ทำงานจะทำการสร้างค่าพารามิเตอร์ต่างๆ ในการเชื่อมต่อกับพอร์ต RS232 ซึ่งไฟล์นี้จะทำการอ่านค่าจากบล็อก RS232 Send และ RS232 Receive เพื่อเป็นการสร้างจุดเชื่อมต่อให้กับบล็อกทั้งสอง พิมพ์คำสั่ง >>**whos** บันทึกผลที่เกิดขึ้นที่ Command Window ในตารางที่ 5.1

ตารางที่ 5.1 ผลการแสดงการสร้างค่าพารามิเตอร์ในการเชื่อมต่อกับพอร์ต RS232

Name	Size	Bytes	Class

Grand total is _____ elements using _____ bytes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

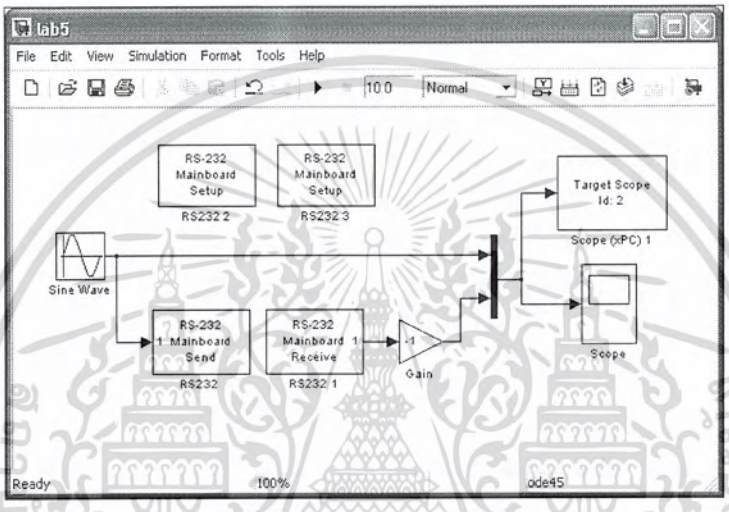
7. กลับมาที่หน้าต่าง Simulink คลิกที่ปุ่ม ▶ สังเกตการเปลี่ยนแปลงที่บล็อก RS232 Send และ RS232 Receive บันทึกผลการทดลองที่เกิดขึ้น

.....

.....

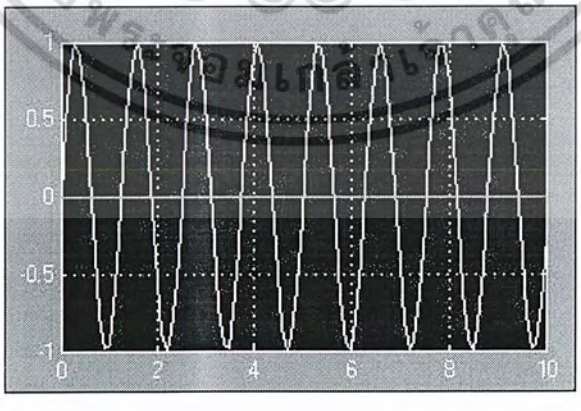
.....

8. ลากเส้นเชื่อมต่อของโมเดลให้สมบูรณ์แสดงดังรูปที่ 5.9 คลิกปุ่ม **Save**



รูปที่ 5.9 การลากเส้นเชื่อมต่อบล็อกของโมเดล lab4 ที่สมบูรณ์

9. คลิกที่ปุ่ม ▶ อีกครั้ง **ดับเบิ้ลคลิกที่บล็อก Scope** จะแสดงผลดังรูปที่ 5.10



รูปที่ 5.10 ผลการจำลองการทำงานของโมเดล lab5.mdl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ให้สังเกตว่าในการใช้งานบล็อก **RS232 Send** และ **RS232 Receive** ต้องเขียน **M-file** โดยกำหนดค่าพารามิเตอร์ทุกครั้งและเมื่อมีการเรียกใช้งานโมเดลนี้ตัวโมเดลจะทำการเรียก M-file ที่ได้เขียนขึ้นทุกครั้ง ดังนั้นจะต้องมีการกำหนดว่า เมื่อมีการเรียกไฟล์ lab5.mdl ขึ้นเมื่อใดให้ทำการเรียกค่าพารามิเตอร์จาก RS232Async_Messages.m นี้ ซึ่งขั้นตอนสามารถทำได้โดย ที่ MATLAB Command Window พิมพ์คำสั่ง

>>set_param(gcs, 'PreLoadFcn','RS232async_messages')

คลิก **Save** ในหน้าต่าง lab5.mdl ทดลองปิดโมเดลและเปิดขึ้นมาใหม่จะต้องแสดงโมเดลที่มีจุดเชื่อมต่อที่สมบูรณ์เหมือนกับตอนที่ Save ไว้แสดงว่าถูกต้อง

11. กำหนดค่าพารามิเตอร์ คลิกที่ **Simulation > Configuration parameters** กำหนดค่าดังนี้ คลิกที่หัวข้อ **Solver** กำหนดค่า **Start Time = 0, Stop Time = 300** และในช่อง **Type** กำหนดเป็น **Fixed Step** และ **Solve** เป็น **ode5 (Dormand-Prince)** การกำหนด Stop Time เป็น 300 นั้นเพื่อที่จะสังเกตการทำงานของสัญญาณที่ Target PC ให้นานขึ้น

12. คลิกที่หัวข้อ **Data Import/Export** คลิกเพิ่มการเลือกที่ ช่อง Save to Workspace ให้เลือกทั้ง 3 อย่าง คือ **Time, State และ Output** ส่วนในหัวข้อ **Save Option ยกเลิกการเลือกที่** Limit data points to lest

13. คลิกที่หัวข้อ **Real-Time Workshop** ในช่อง **Target Selection** เลือกเป็น **xpctarget.tlc**

14. กลับมาที่หน้าต่าง **lab5.mdl** คลิก **Save** แล้วคลิกที่ ▶ แล้วดับเบิ้ลคลิกที่ Scope **สังเกตการเปลี่ยนแปลงที่ Scope** เปรียบเทียบกับผลการทดลองในขั้นตอนที่ 9 **บันทึกผลในรูปที่ 5.11** การเปลี่ยนแปลงที่ Scope และอธิบายว่าสาเหตุที่เกิดการเปลี่ยนแปลงนั้นเนื่องจากอะไร?

.....

.....

.....

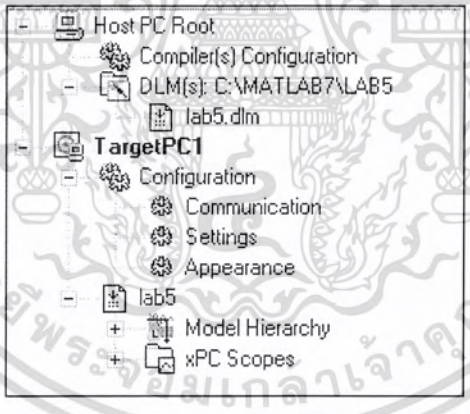
15. ขั้นตอนต่อไปจะเป็นการเชื่อมต่อกับ Target PC โดยจะใช้**ระบบเครือข่าย (Network)** โดยวิธีการเชื่อมต่อนั้นจะใช้หน้าต่าง **Model Explorer** ซึ่งวิธีการเชื่อมต่อนั้นสามารถดูได้จากใบงานที่ 4 แต่จะแตกต่างกันที่จะต้องใช้สายสื่อสาร RS232 เชื่อมต่อระหว่าง COM1 และ COM2 เมื่อเชื่อมต่อกับ Target PC ได้จะแสดงระบบ xPC Target Kernel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.11 กราฟการเปลี่ยนแปลงที่ Scope

16. กลับมาที่หน้าต่าง **lab5.mdl** คลิกสร้างโค๊ดและสร้างไฟล์ DLMโดยใช้คีย์ลัด Ctrl+B เมื่อเริ่มขั้นตอนสร้างโค๊ดภาษา C และไฟล์ DLM ที่หน้าต่างโปรแกรม MATLAB จะแสดงขั้นตอนต่างๆ และทดสอบการเชื่อมต่อกับ Target PC ด้วย

17. กลับมาที่หน้าต่าง **xPC Target Explorer** จะแสดงการเชื่อมต่อกับ Target แล้ว จะเห็นว่ามี ไฟล์ lab5 รอที่จะดาวน์โหลดอยู่ แสดงดังรูปที่ 5.12



รูปที่ 5.12 การเชื่อมต่อกับ Target PC เสร็จสิ้น

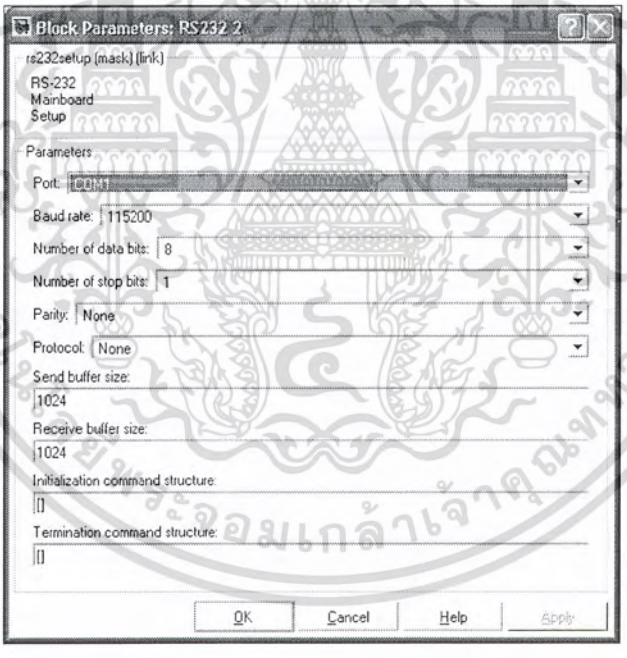
18. ทำการดาวน์ไฟล์ lab5 โดยคลิกที่ปุ่ม ▶ หรือ **คลิกขวาที่ lab5** เลือก **Start** จะเริ่มการทำงานที่ Target โดยจะมีการเปลี่ยนแปลงที่ Host แสดงผลดังรูปที่ 5.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TargetPC1 Target Status Display	
Loaded App	LAB2_xpc2
Mode	Real-Time Multi-Tasking
Logging	t tet
StopTime	Inf
Sample Time	0.00050
Average TET	0.0000034
Execution Time	53.00800
Target Name:	TargetPC1
Connected:	Yes

รูปที่ 5.13 การทำงานของ ไฟล์ lab5.dlm แสดงที่ Host

19. ที่ Target PC จะแสดงการทำงานของไฟล์ lab5 บันทึกผลลักษณะของสัญญาณที่ Scope ดังรูปที่ 5.14



รูปที่ 5.14 การทำงานของไฟล์ lab5 ที่ Target PC

20. จากสัญญาณที่ Target PC สังเกตว่าสัญญาณที่ส่งออกที่ **COM 1** (สีเหลือง) จะเป็นสัญญาณ Sine Wave และสัญญาณที่รับเข้ามาที่ **COM 2** (สีฟ้า) จะเป็นสัญญาณ Sine Wave เหมือนกันแต่จะกลับเฟสกันเพราะเนื่องจากสัญญาณที่รับเข้ามาที่ COM2 นั้นจะผ่านบล็อก Gain =

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-1 ดังนั้น Amplitude ของสัญญาณ Sine จะถูกคูณอยู่กับค่าคงที่ -1 จึงทำให้สัญญาณที่ได้เมื่อนำมาพล็อตกราฟจะมีลักษณะกลับเฟสกันนั่นเอง แสดงวิธีการคำนวณหาค่า **Amplitude** และ **Frequency** (Hz และ Rad/s) ของสัญญาณทั้งสอง บันทึกผลการทดลอง (หมายเหตุ 2π Rad = 1 Hz)

.....

.....

.....

.....

.....

21. ทดลอง**ถอดสาย RS232** ที่เชื่อมต่อระหว่าง **COM1** และ **COM2** ออก สังเกตการณ์เปลี่ยนแปลงที่ Target PC บันทึกผลการทดลอง

.....

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. จงอธิบายขั้นตอนการทำงานของ xPC Target ในการติดต่อกับพอร์ต RS-232 มาพอสังเขป?

.....

.....

.....

.....

.....

2. ในการกำหนดพอร์ตและอัตราการรับส่งข้อมูลสามารถกำหนดได้อย่างไร?

.....

.....

.....

.....

3. ในขั้นตอนการสร้างโค๊ดภาษา C นั้น ถ้ากำหนด TLC (Target Language Compiler) เป็นแบบ xpctarget.tlc จะได้ไฟล์ที่ใช้ดาวน์โหลดสู่ Target PC เป็นไฟล์ชื่อชนิดใดและแตกต่างกับการกำหนด TLC แบบ grt.tlc อย่างไร?

.....

.....

.....

.....

.....

4. ในผลการทดลองขั้นตอนการทดลองที่ 21 จะอธิบายผลการทดลองที่เกิดขึ้น?

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 6
เรื่อง การใช้งานโปรแกรม MATLAB ควบคุม เครื่องกำเนิดสัญญาณ

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. อธิบายขั้นตอนการใช้งานโปรแกรม MATLAB เพื่อควบคุมเครื่องกำเนิดสัญญาณ AFG310 ได้อย่างถูกต้อง
2. แสดงขั้นตอนการใช้คำสั่งในการติดต่อกับเครื่องกำเนิดสัญญาณได้อย่างถูกต้อง
3. อธิบายผลที่เกิดขึ้นหลังจากการใช้คำสั่งต่างๆ ได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0
2. เครื่องกำเนิดสัญญาณ Sony Tektronix AFG310 (Function Generator)
3. การ์ด GPIB ของบริษัท National Instruments รุ่น GPIB+
4. สาย GPIB ของบริษัท National Instruments จำนวน 1 เส้น

ความรู้และทักษะเดิม

1. การใช้งานโปรแกรม MATLAB เบื้องต้น

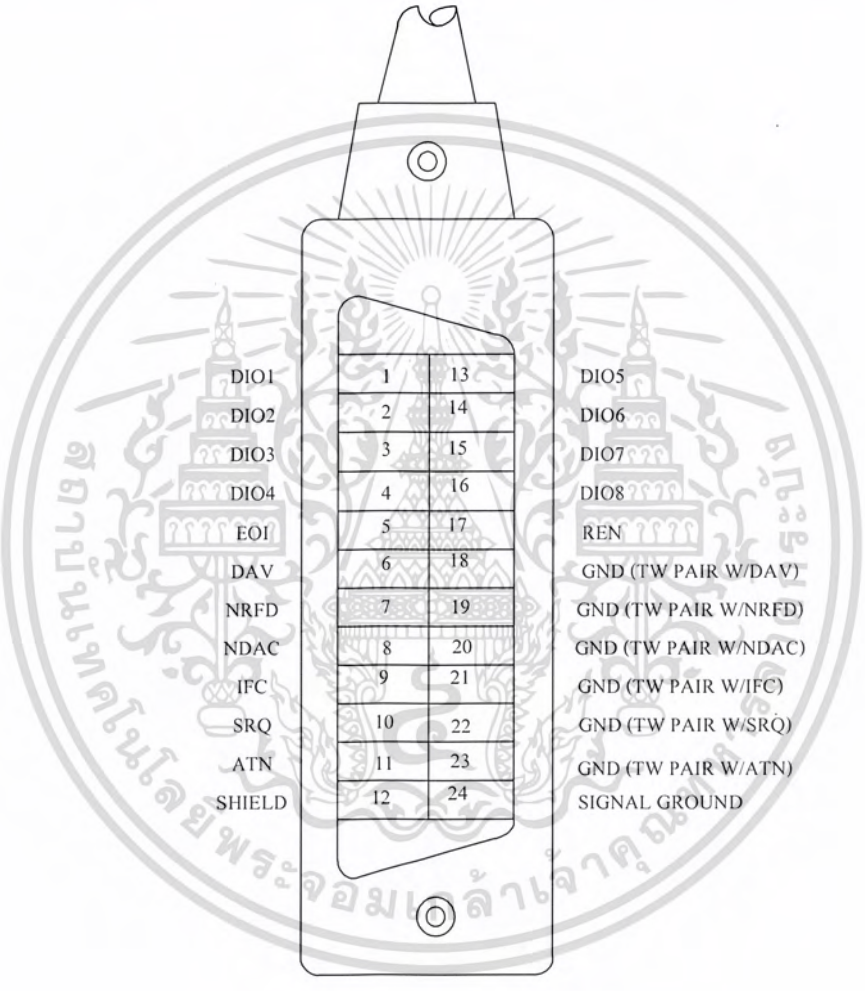
ทฤษฎีเบื้องต้น

GPIB : General Purpose Interface Bus เป็นระบบบัสรูปแบบหนึ่งที่ใช้ในการใช้งานควบคุมอุปกรณ์เครื่องมือวัดในอุตสาหกรรมหรือเครื่องมือวัดทั่วๆ ไป ซึ่งได้รองรับตามมาตรฐาน IEEE 488 และได้พัฒนาเป็นมาตรฐาน IEEE 488.2 ที่ใช้งานในปัจจุบัน ซึ่งลักษณะการใช้งานต้องมีการ์ด GPIB ซึ่งหลายบริษัทที่จัดทำการ์ด GPIB ขึ้น แต่บริษัทที่ได้รับความนิยมมากที่สุด คือ บริษัท National Instruments และเครื่องมือวัดที่นำมาใช้งานนั้นจะต้องมีพอร์ต GPIB ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ GPIB ตามมาตรฐาน IEEE 488

ลักษณะทางกายภาพ IEEE-488 นั้น คือ เป็นสายสัญญาณแบบ 24 เส้นขนานกันและมีขั้วต่ออยู่ทางปลายทั้งสองของสาย เพื่อต่อกับอุปกรณ์หรือต่อกันเพื่อให้สายสัญญาณมีความยาวเพิ่มขึ้น ในจำนวนสายสัญญาณ 24 เส้น มีเพียง 16 เส้นเท่านั้น ที่ทำหน้าที่นำสัญญาณ ส่วนเหลืออีก 8 เส้น ทำหน้าที่กราวด์ (ground) และชีลด์ (shield) ลักษณะของขั้วต่อ GPIB แสดงได้ดังรูปที่ 6.1



รูปที่ 6.1 ขั้วต่อของ GPIB และการจัดขาของสัญญาณต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขีดจำกัดของ GPIB ตามมาตรฐาน IEEE-488

1) จำนวนอุปกรณ์ในระบบ (Talker, Listener, Controller) ที่ต่อกับสายนำสัญญาณ 1 เส้น จะต้องไม่เกิน 15 เครื่อง

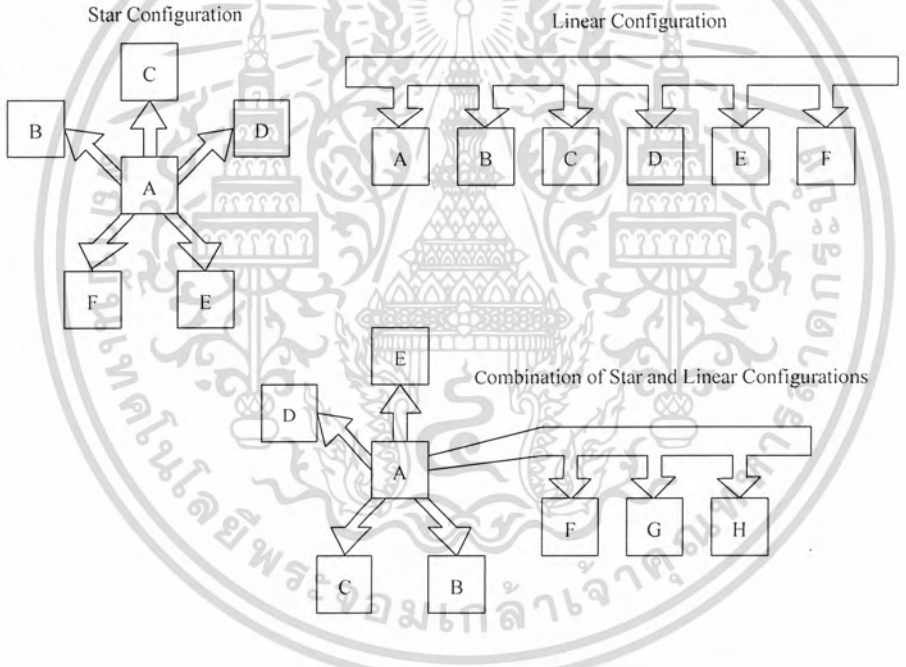
2) สายเคเบิ้ลที่ต่อระหว่างอุปกรณ์ จะต้องยาวไม่เกิน 4 เมตร และความยาวรวมของสายเคเบิ้ลในระบบจะต้องไม่เกิน 20 เมตร

3) ความเร็วในการส่งข้อมูลจะต้องไม่เกิน 1Mb/Sec (1 ล้านไบต์ต่อวินาที)

4) ต้องมีการจ่ายไฟให้กับอุปกรณ์มากกว่าครึ่งหนึ่งของระบบ

ลักษณะการเชื่อมต่อกับเครื่องมือวัด

ลักษณะการเชื่อมต่อการ์ด GPIB กับเครื่องมือวัดในระบบนั้นจะรองรับการเชื่อมต่อหลายรูปแบบ แสดงดังรูปที่ 6.2



รูปที่ 6.2 ลักษณะการเชื่อมต่อเครื่องมือวัดในระบบ

1. Star Configuration

การเชื่อมต่อแบบนี้จะมีลักษณะคือ เชื่อมต่อเครื่องมือวัดทั้งหมดกับเครื่องหลักเพียงตัวเดียว หรืออาจเรียกได้เป็นการเชื่อมต่อแบบขนาน ซึ่งที่เครื่องมือวัดแต่ละตัวจะต้องมีการกำหนดค่า Primary Address ที่แตกต่างกัน

2. Linear Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อแบบนี้จะมีลักษณะเชื่อมต่อเครื่องมือวัดแบบตัวต่อตัวไปเรื่อยๆ หรืออาจเรียกได้เป็นการเชื่อมต่อแบบอนุกรม ซึ่งที่เครื่องมือวัดแต่ละตัวจะต้องมีการกำหนดค่า Primary Address ที่แตกต่างกันเหมือนกับการเชื่อมต่อแบบขนาน

3. Combination of Star and Linear Configurations

การเชื่อมต่อแบบนี้จะเป็นการผสมกันระหว่างแบบอนุกรมและแบบขนาน ซึ่งทำให้ในระบบมีความยืดหยุ่นในการเชื่อมต่อสูง แต่ทั้งนี้ต้องขึ้นอยู่กับขีดความสามารถของ GPIB ตามมาตรฐาน IEEE 488 ด้วย

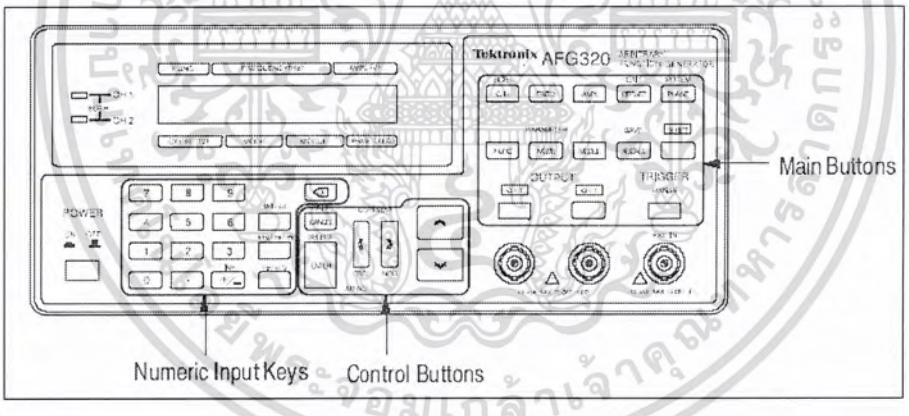
ลักษณะของเครื่องกำเนิดสัญญาณ Sony Tektronix AFG310

ลักษณะของเครื่องกำเนิดสัญญาณ Sony Tektronix AFG310 นั้นแสดงได้ดังรูปที่ 6.3 โดยจะมีเมนูการกำหนดค่าต่างๆ แบ่งเป็น 3 ส่วน

1. Main Buttons เป็นส่วนที่กำหนดหน้าที่การทำงานหลักของเครื่อง

2. Numeric Input Keys เป็นส่วนที่ป้อนตัวเลขในการที่จะกำหนดค่า Amplitude หรือ Frequency

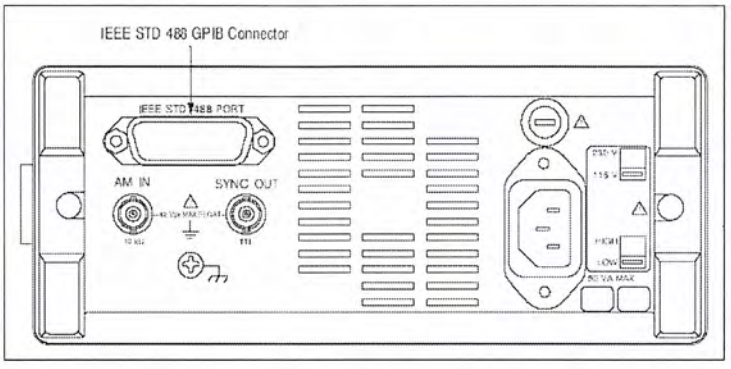
3. Control Buttons เป็นส่วนควบคุมการทำงานในการเลือกหรือกำหนดค่า



รูปที่ 6.3 ลักษณะทางด้านหน้าของเครื่องกำเนิดสัญญาณ Sony Tektronix AFG310

ลักษณะของเครื่องกำเนิดสัญญาณ Sony Tektronix AFG310 ทางด้านหลังแสดงดังรูปที่ 6.4 จะเห็นว่ามีพอร์ตที่รองรับการใช้งานกับการ์ด GPIB การใช้งานเพียงเชื่อมต่อสาย GPIB กับพอร์ต GPIB นี้เท่านั้น

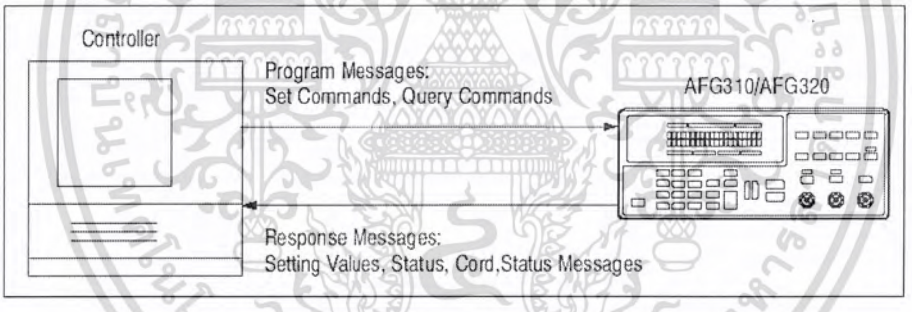
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 ลักษณะทางด้านหลังของเครื่องกำเนิดสัญญาณ Sony Tektronix AFG310

ลักษณะการเชื่อมต่อเครื่องกำเนิดสัญญาณกับเครื่องคอมพิวเตอร์

การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องมือวัดนั้นจะต้องมีสาย GPIB ในการเชื่อมต่อ โดยที่เครื่องคอมพิวเตอร์จะต้องมีการ์ด GPIB ต่ออยู่และที่เครื่องมือวัดจะต้องมีพอร์ต GPIB ลักษณะการเชื่อมต่อกับเครื่องคอมพิวเตอร์นั้นแสดงดังรูปที่ 6.5



รูปที่ 6.5 การเชื่อมต่อเครื่องกำเนิดสัญญาณกับเครื่องคอมพิวเตอร์

คำสั่งในการติดต่อกับเครื่องมือวัด (MATLAB Command)

- **fprintf** เป็นคำสั่งในการเขียนข้อมูลที่เป็นตัวอักษร (ASCII) ซึ่งเป็นลักษณะคำสั่งให้กับเครื่องมือวัด
- **fscanf** เป็นคำสั่งในการอ่านค่าข้อมูลในลักษณะตัวอักษรจากเครื่องมือวัด
- **valuesreceive** เป็นคำสั่งในการอ่านค่าข้อมูลที่รับมาจากเครื่องมือวัดทั้งหมด ในขณะนั้น
- **valuessent** เป็นคำสั่งในการอ่านค่าข้อมูลที่ส่งออกไปให้กับเครื่องมือวัดในขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **InputBufferSize** เป็นการตรวจสอบจำนวนของข้อมูลทั้งหมดที่สามารถรับเข้ามาได้ในการติดต่อ 1 ครั้ง
- **OutputBufferSize** เป็นการตรวจสอบจำนวนของข้อมูลทั้งหมดที่สามารถส่งออกไปให้กับเครื่องมือวัดในการติดต่อ 1 ครั้ง

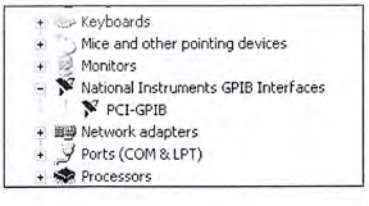
คำสั่งในการควบคุมเครื่องกำเนิดสัญญาณ Sony Tektronix AFG310 (Instruments Command)

คำสั่งในการควบคุมเครื่องกำเนิดสัญญาณ Sony Tektronix AFG310 จะเป็นคำสั่งเฉพาะไม่สามารถนำไปใช้กับเครื่องมือวัดรุ่นอื่นๆได้ แต่ก็มีบางคำสั่งที่เป็นมาตรฐานของ IEEE 488 ซึ่งสามารถนำไปใช้ควบคุมการทำงานของเครื่องมือวัดทั่วๆ ไปได้ สามารถศึกษาเพิ่มเติมจากคู่มือการใช้งานเครื่องกำเนิดสัญญาณรุ่น Sony Tektronix AFG310 ตัวอย่างคำสั่งในควบคุมเครื่องกำเนิดสัญญาณ มีรายละเอียดดังนี้

- **Freq** parameter เป็นการกำหนดค่าความถี่ของสัญญาณที่สร้างขึ้น ตามค่า parameter
- **Volt** parameter เป็นการกำหนดค่าระดับแรงดันของสัญญาณที่สร้างขึ้น ตามค่า parameter
- **Func** parameter เป็นการกำหนดลักษณะของสัญญาณที่สร้างขึ้น ตามค่า parameter
- ***IDN** เป็นคำสั่งในการบอกให้เครื่องมือวัดนั้นแจ้งรายละเอียดของเครื่องมือทั้งหมด
- ***RST** เป็นคำสั่งในการกำหนดให้เครื่องมือวัดรีเซตการกำหนดค่าต่างๆ ให้เป็นค่าเริ่มต้น

ลำดับขั้นการทดลอง

1. การใช้งาน GPIB นั้นนอกจากต้องมีการ์ด GPIB ซึ่งในการทดลองนี้จะใช้การ์ด GPIB ของบริษัท National Instruments รุ่น GPIB+ และต้องมีไดร์ฟเวอร์ของการ์ดด้วยซึ่งเป็นไดร์ฟเวอร์ตามมาตรฐาน IEE 488.2 เมื่อติดตั้งเสร็จสิ้น ทดลองเปิดที่ Device Manager ของ Widows จะปรากฏดังรูปที่ 6.6

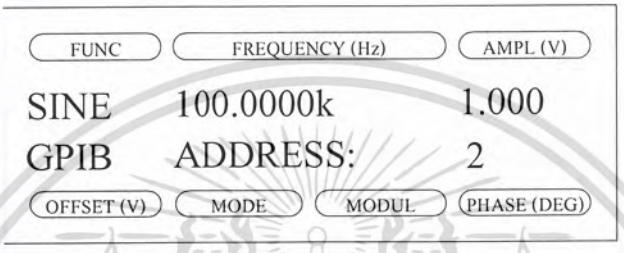


รูปที่ 6.6 หน้าต่าง Device Manager เมื่อติดตั้งการ์ด GPIB เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการเชื่อมต่อระหว่างการ์ด GPIB กับเครื่องกำเนิดสัญญาณ Tektronix AFG310 โดยใช้สาย GPIB เปิดเครื่องกำเนิดสัญญาณ

3. ขั้นตอนต่อไปจะเป็นการกำหนดลำดับของเครื่องกำเนิดสัญญาณ (Primary Address) ซึ่งจะต้องกำหนดจากหน้าปัดของเครื่องกำเนิดสัญญาณ กดปุ่ม **SHIFT** ตามด้วยปุ่ม **PHASE** จะแสดงหน้าต่างดังรูปที่ 6.7 บริเวณปุ่มเมนูกดปุ่ม **ENTER** กดตัวเลขที่ปุ่มป้อนตัวเลข ให้กดที่หมายเลข **2** กดปุ่ม **ENTER** กดปุ่ม **CANCEL** ออกจากเมนูกด **CANCEL** อีกครั้ง



รูปที่ 6.7 หน้าต่างการกำหนดตำแหน่ง Primary Address

4. เปิดโปรแกรม MATLAB เวอร์ชั่น 7.0 ในขั้นตอนนี้จะทำการติดต่อกับการ์ด GPIB และเครื่องกำเนิดสัญญาณ โดยจะมีรูปแบบคำสั่งคือ

```
>> Obj = gpib('Vender',Boardindex,Primary address)
```

- Obj คือ ชื่อตัวแปรที่ตั้งขึ้นเพื่อเก็บค่าพารามิเตอร์ต่างๆ ในการติดต่อเช่น g
- Vender คือ ชนิดของการ์ด GPIB ที่ใช้ซึ่งจะเป็นชื่อย่อของบริษัทเช่น NI
- Boardindex คือ ลำดับของการ์ด GPIB ที่ใช้ถ้ามีการ์ดเดียวจะกำหนดเป็น 0
- Primary address คือ ลำดับของเครื่องมือวัดที่ติดต่อ เช่น ตัวแรกกำหนดเป็น 1 ตัวถัดไปกำหนดเป็น 2

ที่หน้าต่าง Command Window พิมพ์คำสั่ง **>>clear all** เพื่อลบค่าข้อมูลที่อยู่ใน Workspace ทดลองใช้คำสั่ง **>>whos** จะไม่มีค่าข้อมูลใดๆ อยู่และพิมพ์คำสั่ง **>>g = gpib('ni',0,2)** เพื่อเริ่มต้นการติดต่อกับการ์ด GPIB และเครื่องมือวัด สังเกตผลที่เกิดขึ้นที่ Command Window บันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

5. จากผลการทดลองในขั้นตอนที่ 4 ในช่อง Status จะปรากฏคำว่า Close หมายความว่าถูกปิดการใช้งานอยู่ เนื่องจากเมื่อติดต่อกับการ์ด GPIB แล้วต้องมีการเปิดการใช้งานโดยการพิมพ์คำสั่ง >>**fopen(g)**

6. ขั้นตอนต่อไปจะทำการตรวจสอบ Object ที่ใช้ติดต่อกับ GPIB ทั้งหมดโดยพิมพ์คำสั่ง **>>out=instrfind(g)** สังเกตผลที่เกิดขึ้น จะคล้ายกับในขั้นตอนที่ 4 เนื่องจากมีการใช้งาน GPIB ติดต่อกับเครื่องกำเนิดสัญญาณเพียงตัวเดียว แตกต่างกันที่ช่อง Status จะแสดงคำว่า Open แสดงว่าพร้อมใช้งานแล้ว

7. ที่ Command Window ให้พิมพ์คำสั่ง >>**fprintf(g, '*RST')** เพื่อเป็นการ Reset การปรับแต่งค่าให้เป็นค่าที่ได้ตั้งไว้ตอนเปิดเครื่อง หลังจากการใช้คำสั่งนี้สังเกตว่าสัญญาณไฟที่ปุ่ม CH1 จะดับลง แสดงว่าไม่มีสัญญาณออกที่ CH1 ถ้าต้องการใช้งานอีกครั้งกดที่ให้ปุ่ม CH1

8. พิมพ์คำสั่ง >>**fprintf(g, '*IDN?')** เป็นการร้องขอรายละเอียดของเครื่องมือวัดที่ต้องการติดต่อด้วย

9. พิมพ์คำสั่ง **>>data=fscanf(g)** เป็นการนำค่ารายละเอียดของเครื่องที่ร้องขอกลับมาเก็บที่ตัวแปรชื่อ data สังเกตผลที่เกิดขึ้นที่ Command Window บันทึกผลการทดลอง

.....

.....

10. ที่หน้าจอของเครื่องกำเนิดสัญญาณบันทึกค่าต่างๆ ลงในตารางที่ 6.1 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.1 การบันทึกสัญญาณ ความถี่ และแอมพลิจูด

Function	Frequency (Hz)	Amplitude (V)

11. ขั้นตอนต่อไปจะเป็นการทดลองเปลี่ยนพารามิเตอร์ต่างๆ ของเครื่องกำเนิดสัญญาณ โดยใช้รูปแบบคำสั่ง **>>fprintf (Obj,'Command')** ซึ่งเป็นการเขียนคำสั่งที่เป็นลักษณะตัวอักษร (ASCII) ไปยังเครื่องกำเนิดสัญญาณ โดยที่ Command Window พิมพ์คำสั่ง **>>fprintf(g, 'Volt 2')** สังเกตผลที่หน้าจอเครื่องกำเนิดสัญญาณ, พิมพ์คำสั่ง **>>fprintf(g, 'Freq 10k')** และพิมพ์คำสั่ง **>>fprintf(g, 'Func sin')** สังเกตผลที่เกิดขึ้นที่หน้าจอเครื่องกำเนิดสัญญาณ บันทึกผลการทดลองในตารางที่ 6.2

ตารางที่ 6.2 การบันทึกสัญญาณ ความถี่ และแอมพลิจูด

Function	Frequency (Hz)	Amplitude (V)

คำสั่งในการเปลี่ยนแปลงฟังก์ชั่นนั้นสามารถกำหนดได้ดังนี้

- **Sin** เป็นการกำหนดให้สร้างสัญญาณ Sine Wave
- **Squ** เป็นการกำหนดให้สร้างสัญญาณ Square Wave
- **Tri** เป็นการกำหนดให้สร้างสัญญาณ Triangle Wave
- **Ramp** เป็นการกำหนดให้สร้างสัญญาณ Ramp Wave
- **Puls** เป็นการกำหนดให้สร้างสัญญาณ Pulse Wave
- **Dc** เป็นการกำหนดให้สร้างสัญญาณ DC Signal
- **Prn** เป็นการกำหนดให้สร้างสัญญาณ Noise

จากผลการทดลองเป็นการกำหนดให้เครื่องกำเนิดสัญญาณสร้าง Sine Wave โดยมีค่าแรงดันเท่ากับ 2 โวลต์ และความถี่เท่ากับ 10 kHz โดยสังเกตได้ที่หน้าจอเครื่องกำเนิดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ทดลองอ่านค่า OutputBufferSize ซึ่งเป็นจำนวนข้อมูลสูงสุด (มีหน่วยเป็นไบต์) ของกำเนิดสัญญาณที่สามารถส่งออกได้ในแต่ละครั้ง พิมพ์คำสั่ง **>>get(g, 'OutputBufferSize')** บันทึกผลการทดลอง

.....

.....

.....

13. ทดลองอ่านค่า InputBufferSize ซึ่งเป็นจำนวนข้อมูลสูงสุด (มีหน่วยเป็นไบต์) ของเครื่องกำเนิดสัญญาณที่สามารถรับได้ในแต่ละครั้ง พิมพ์คำสั่ง **>>get(g, 'InputBufferSize')** บันทึกผลการทดลอง

.....

.....

.....

14. ขั้นตอนต่อไปจะเป็นการทดลองอ่านค่าข้อมูลจากเครื่องกำเนิดสัญญาณ โดยจะใช้คำสั่ง **>>fscanf(Obj,'Command')** เป็นการอ่านค่าข้อมูลที่เป็นลักษณะตัวอักษร (ASCII) พิมพ์คำสั่ง **>>fprintf(g, 'Volt?')** และคำสั่ง **>>data=fscanf(g)** บันทึกผลการทดลอง

.....

.....

.....

15. พิมพ์คำสั่ง **>>fprintf(g, 'Freq?')** และคำสั่ง **>>data=fscanf(g)** บันทึกผลการทดลอง

.....

.....

.....

16. พิมพ์คำสั่ง **>>fprintf(g, 'Func?')** และคำสั่ง **>>data=fscanf(g)** บันทึกผลการทดลอง

.....

.....

.....

คำสั่งนั้นในการอ่านค่าข้อมูลจากเครื่องมือวัดนั้น จะมีความแตกต่างจากคำสั่งเปลี่ยนค่า คือ คำสั่งในการอ่านค่าข้อมูลนั้นจะต้องมีเครื่องหมาย '?' ต่อท้ายคำสั่ง

17. ทดลองอ่านจำนวนข้อมูลที่ส่งออกไปให้กับเครื่องมือวัดทั้งหมด (มีหน่วยเป็นไบต์) ใช้คำสั่ง **>>get(g, 'ValuesSent')** บันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....

.....

.....

18. ทดลองอ่านจำนวนข้อมูลที่รับมาจากเครื่องมือวัดทั้งหมด (มีหน่วยเป็นไบต์) ใช้คำสั่ง

.....

.....

.....

19. เมื่อจบการทำงานต้องมีการยกเลิกการติดต่อกับการ์ด GPIB และเครื่องกำเนิดสัญญาณ โดยพิมพ์คำสั่งดังนี้

>>fclose(g) เพื่อเป็นการปิดการใช้งานการ์ด GPIB

>>delete(g) เพื่อเป็นการลบตัวแปร g ที่ใช้เก็บค่าต่างๆ ในการติดต่อ

>>clear all เพื่อเป็นการลบค่าพารามิเตอร์ใน Workspace ทั้งหมดที่ใช้งานอยู่

>>clc เพื่อเป็นการลบคำสั่งทั้งหมดที่ Command Window

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....

.....

.....

คำถามท้ายการทดลอง

1. ถ้าต้องการทราบรายละเอียดของเครื่องมือวัดนั้นจะต้องใช้คำสั่งใดและมีลักษณะการใช้งานอย่างไร?

.....

.....

.....

.....

.....

.....

.....

2. ในการเริ่มต้นการติดต่อกับการ์ด GPIB เครื่องมือวัดนั้นต้องใช้คำสั่งใด และมีลักษณะการใช้งานอย่างไร?

.....

.....

.....

.....

.....

.....

.....

3. เมื่อต้องการยกเลิกการติดต่อกับการ์ด GPIB และเครื่องมือวัดเสร็จสิ้นการวัดสัญญาณแล้วต้องใช้คำสั่งอะไรบ้างในการยกเลิกการติดต่อกับการ์ด GPIB ?

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 7

เรื่อง การใช้งานโปรแกรม MATLAB ควบคุมเครื่องวัดสัญญาณ

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. อธิบายขั้นตอนการใช้งานโปรแกรม MATLAB เพื่อควบคุมเครื่องวัดสัญญาณ Tektronix TDS-340A ได้อย่างถูกต้อง
2. แสดงขั้นตอนการใช้คำสั่งในการติดต่อกับเครื่องวัดสัญญาณได้อย่างถูกต้อง
3. อธิบายผลที่เกิดขึ้นหลังจากการใช้คำสั่งควบคุมเครื่องมือวัดสัญญาณได้อย่างถูกต้อง
4. แสดงขั้นตอนการอ่านข้อมูลจากเครื่องมือวัดมาแสดงผลที่โปรแกรม MATLAB ได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0
2. เครื่องกำเนิดสัญญาณ Sony Tektronix AFG310 (Function Generator)
3. เครื่องวัดสัญญาณ Tektronix TDS 340A (Oscilloscope)
4. การ์ด GPIB ของบริษัท National Instruments รุ่น GPIB+
5. สาย GPIB ของบริษัท National Instruments จำนวน 2 เส้น

ความรู้และทักษะเดิม

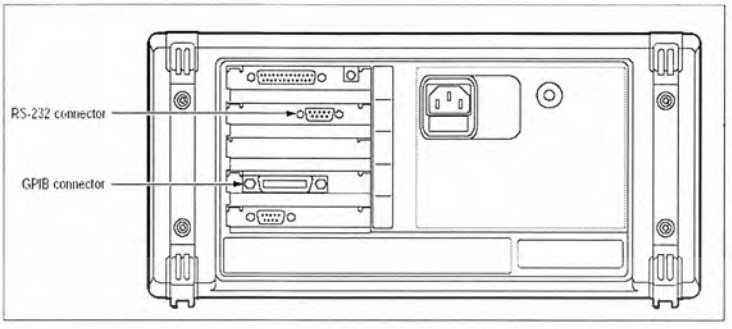
1. การใช้งานโปรแกรม MATLAB เบื้องต้น
2. การใช้งานโปรแกรม MATLAB ติดต่อกับเครื่องกำเนิดสัญญาณ

ทฤษฎีเบื้องต้น

การใช้งานการ์ด GPIB ติดต่อกับเครื่องวัดสัญญาณรุ่น Tektronix TDS 340A นั้นจะมีลักษณะที่คล้ายกับการติดต่อกับเครื่องกำเนิดสัญญาณในใบงานที่ 6 แตกต่างกันที่คำสั่งควบคุมเครื่องวัดสัญญาณนั้นจะแตกต่างกับเครื่องกำเนิดสัญญาณ ซึ่งสามารถศึกษาเพิ่มเติมจากคู่มือการใช้งานเครื่องวัดสัญญาณรุ่น Tektronix TDS 340A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของเครื่องวัดสัญญาณ Tektronix TDS 340A ดังรูปที่ 7.1



รูปที่ 7.1 ลักษณะทางด้านหลังของเครื่องวัดสัญญาณ Tektronix TDS 340A

คำสั่งในการติดต่อกับเครื่องมือวัด (MATLAB Command)

- **fread** เป็นคำสั่งในการอ่านค่าข้อมูลจากเครื่องมือวัดที่เป็น Binary
- **fwrite** เป็นคำสั่งในการเขียนข้อมูลให้กับเครื่องมือวัดที่เป็น Binary
- **fprintf** เป็นคำสั่งในการเขียนข้อมูลที่เป็นตัวอักษร (ASCII) ซึ่งเป็นลักษณะคำสั่งให้กับเครื่องมือวัด
- **fscanf** เป็นคำสั่งในการอ่านค่าข้อมูลในลักษณะตัวอักษรจากเครื่องมือวัด
- **ValuesReceive** เป็นคำสั่งในการอ่านค่าข้อมูลที่รับมาจากเครื่องมือวัดทั้งหมดในขณะนั้น
- **ValuesSent** เป็นคำสั่งในการอ่านค่าข้อมูลที่ส่งออกไปให้กับเครื่องมือวัดในขณะนั้น
- **InputBufferSize** เป็นการตรวจสอบจำนวนของข้อมูลทั้งหมดที่สามารถรับเข้ามาได้ในการติดต่อ 1 ครั้ง
- **OutputBufferSize** เป็นการตรวจสอบจำนวนของข้อมูลทั้งหมดที่สามารถส่งออกไปให้กับเครื่องมือวัดในการติดต่อ 1 ครั้ง

คำสั่งในการควบคุมเครื่องวัดสัญญาณ Tektronix TDS 340A (Instruments Command)

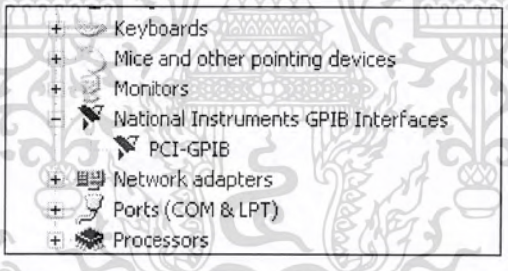
คำสั่งในการควบคุมเครื่องวัดสัญญาณ Tektronix TDS 340A จะเป็นคำสั่งเฉพาะที่จะไม่สามารถนำไปใช้กับเครื่องมือวัดรุ่นอื่นๆ ได้ แต่ก็มีบางคำสั่งที่เป็นมาตรฐานของ IEEE 488 ซึ่งจะสามารถนำไปใช้ควบคุมการทำงานของเครื่องมือวัดทั่วๆ ไปได้ สามารถศึกษาเพิ่มเติมได้จากคู่มือการใช้งานเครื่องวัดสัญญาณรุ่น Tektronix TDS 340A ตัวอย่างคำสั่งในควบคุมเครื่องวัดสัญญาณมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **:MEASU:MEAS(x):TYP FREQ** เป็นคำสั่งให้แสดงผลค่าความถี่ของสัญญาณที่ x
- **:MEASU:MEAS(x):SOU CH(y)** เป็นคำสั่งกำหนดสัญญาณที่จะวัดมาจากช่องของสัญญาณ y
- **:MEASU:MEAS(x):STATE ON** เป็นคำสั่งให้เริ่มต้นแสดงค่าสัญญาณตามที่กำหนดที่ x
- **:MEASU:MEAS(x):STATE OFF** เป็นคำสั่งให้หยุดแสดงค่าสัญญาณตามที่กำหนดที่ x
- ***IDN** เป็นคำสั่งในการบอกให้เครื่องมือวัดนั้นแจ้งรายละเอียดของเครื่องมือทั้งหมด
- ***RST** เป็นคำสั่งในการกำหนดให้เครื่องมือวัดนั้นรีเซตการกำหนดค่าต่างๆ ให้เป็นค่าเริ่มต้น

ลำดับขั้นการทดลอง

1. ตรวจสอบการติดตั้งการ์ด GPIB ว่าสมบูรณ์หรือไม่ โดยทดลองเปิดที่ Device Manager ของ Widows จะปรากฏดังรูปที่ 7.2



รูปที่ 7.2 หน้าต่าง Device Manager เมื่อติดตั้งการ์ด GPIB เรียบร้อยแล้ว

2. เชื่อมต่อระหว่างการ์ด GPIB กับเครื่องกำเนิดสัญญาณ Tektronix AFG310 และเครื่องวัดสัญญาณ Tektronix TDS340A โดยใช้สาย GPIB เปิดเครื่องกำเนิดสัญญาณและเครื่องวัดสัญญาณ

3. เชื่อมต่อสายวัดสัญญาณระหว่างเครื่องกำเนิดสัญญาณและเครื่องวัดสัญญาณ ที่ช่องของสัญญาณที่ 1 (Channel 1)

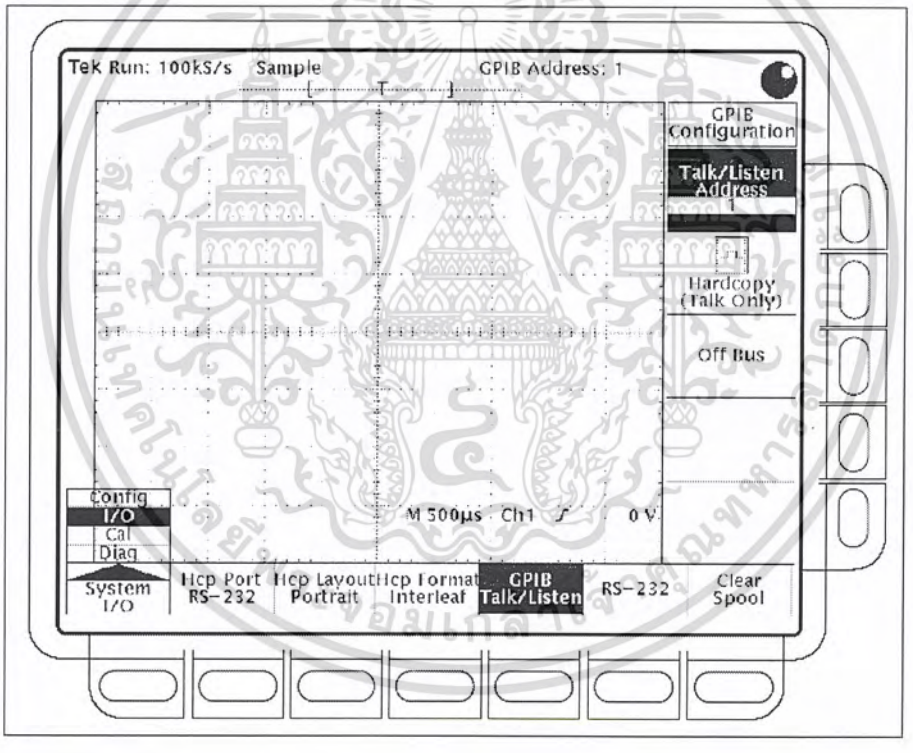
4. ขั้นตอนต่อไปนั้นจะทำการกำหนดค่า Primary Address ของเครื่องกำเนิดสัญญาณและสัญญาณ ซึ่งจากใบงานที่ 6 ได้กำหนด Primary Address ของเครื่องกำเนิดสัญญาณเป็น 2 แล้วไม่ต้องกำหนดใหม่ แต่จะกำหนดค่า Primary Address ที่เครื่องวัดสัญญาณมีขั้นตอนดังนี้กดที่ปุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UTILITY ของเครื่องวัดสัญญาณ

- กดที่ปุ่ม **Config** บริเวณหน้าจอเครื่องวัดสัญญาณ จะแสดงเมนูย่อยขึ้นมาที่หน้าจอ เลือกหัวข้อ **I/O**
- กดที่ปุ่ม **GPIB Talk/Listen** ที่เมนูย่อยทางด้านล่าง แล้วก็จะแสดงเมนูย่อยอีกครั้ง บริเวณด้านขวาของหน้าจอ
- กดที่ปุ่ม **Talk/Listen Address** ที่เมนูย่อยทางด้านขวา เพื่อที่จะได้ทำการปรับค่า Primary Address ของเครื่องวัดสัญญาณ ปรับได้โดยทำการหมุนที่ปุ่ม **Select** หมุนปรับตำแหน่งเป็น **Primary Address = 1**
- กดปุ่ม **Clear Menu** เพื่อยกเลิกการปรับค่า

การปรับค่า Primary Address นั้นสามารถแสดงได้ดังรูปที่ 7.3



รูปที่ 7.3 การกำหนดค่า Primary Address ของเครื่องวัดสัญญาณ

ดังนั้นสามารถสรุปตำแหน่งของเครื่องวัดสัญญาณและเครื่องกำเนิดสัญญาณได้ดังนี้

- เครื่องวัดสัญญาณ (Oscilloscope) ค่า Primary Address = 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องกำเนิดสัญญาณ (Function Generator) ค่า Primary Address = 2

5. เปิดโปรแกรม MATLAB เวอร์ชั่น 7.0 ที่ Command Window พิมพ์คำสั่ง **>>osc = gpib('ni',0,1)** เพื่อติดต่อเครื่องวัดสัญญาณ (Oscilloscope) ผลที่เกิดขึ้นแสดงดังรูปที่ 7.4

```
GPIB Object Using NI Adaptor : GPIB0-1

Communication Address
   BoardIndex:          0
   PrimaryAddress:      1
   SecondaryAddress:    0

Communication State
   Status:              closed
   RecordStatus:        off

Read/Write State
   TransferStatus:      idle
   BytesAvailable:      0
   ValuesReceived:      0
   ValuesSent:          0
```

รูปที่ 7.4 ผลการติดต่อกับการ์ด GPIB และเครื่องวัดสัญญาณ

6. พิมพ์คำสั่ง **>>gen = gpib('ni',0,2)** เพื่อติดต่อกับเครื่องกำเนิดสัญญาณ (Function-Generator) ผลที่เกิดขึ้นแสดงดังรูปที่ 7.5

7. พิมพ์คำสั่ง **>>fopen(osc)** และ **>>fopen(gen)** เป็นการเปิดการใช้งานของเครื่องมือวัดสัญญาณและเครื่องกำเนิดสัญญาณ ทดลองพิมพ์คำสั่ง **>>get(osc, 'Status')** และ **>>get(gen, 'Status')** เพื่อดูสถานะของการเชื่อมต่อ ผลที่เกิดขึ้นแสดงดังรูปที่ 7.6

8. พิมพ์คำสั่ง **>>fprintf(osc, '*IDN?')** และ **>>data=fscanf(osc)** เพื่อเป็นการร้องขอรายละเอียดของเครื่องกำเนิดสัญญาณที่ต้องการติดต่อด้วย บันทึกผลการทดลองที่เกิดขึ้น

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GPIB Object Using NI Adaptor : GPIB0-2

Communication Address
   BoardIndex:          0
   PrimaryAddress:      2
   SecondaryAddress:    0

Communication State
   Status:              closed
   RecordStatus:        off

Read/Write State
   TransferStatus:      idle
   BytesAvailable:      0
   ValuesReceived:      0
   ValuesSent:          0
```

รูปที่ 7.5 ผลการติดต่อกับการ์ด GPIB และเครื่องกำเนิดสัญญาณ

```
>> get(osc,'status')
ans =
open
```

```
>> get(gen,'status')
ans =
open
```

รูปที่ 7.6 สถานะของการเชื่อมต่อกับเครื่องกำเนิดสัญญาณและเครื่องวัดสัญญาณ

9. พิมพ์คำสั่ง **>>fprintf(gen, '*IDN?')** และ **>>data1=fscanf(gen)** เพื่อเป็นการร้องขอรายละเอียดของเครื่องกำเนิดสัญญาณที่ต้องการติดต่อด้วย บันทึกผลการทดลองที่เกิดขึ้น

.....

.....

10. กำหนดค่าต่างๆ ที่เครื่องกำเนิดสัญญาณและเครื่องวัดสัญญาณ โดยพิมพ์คำสั่งดังนี้

>>fprintf(gen, 'Func Sin') เป็นการกำหนดให้สร้างสัญญาณ Sine Wave

>>fprintf(gen, 'Freq 20k') เป็นการกำหนดให้สร้างสัญญาณมีความถี่ 20 kHz

>>fprintf(gen, 'Volt 2') เป็นการกำหนดให้สร้างสัญญาณมีระดับสัญญาณ 2 V

>>fprintf(osc,'Autoset Execute') เป็นการกำหนดให้เครื่องวัดสัญญาณทำการปรับขนาดการแสดงผลให้เหมาะสมกับหน้าจอการแสดงผลสังเกตผลที่เกิดขึ้นที่หน้าจอเครื่องกำเนิดสัญญาณ อธิบายผลการเปลี่ยนแปลงที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

……

……

……

……

11. กำหนดค่าการวัดสัญญาณที่เครื่องวัดสัญญาณโดยพิมพ์คำสั่งดังนี้

>>fprintf(osc, ':MEASU:MEAS1:TYP FREQ') กำหนดให้วัดความถี่ของสัญญาณ

>>fprintf(osc, ':MEASU:MEAS1:SOU CH1') กำหนดว่าให้มีสัญญาณที่ต่ออยู่ CH1

>>fprintf(osc, ':MEASU:MEAS1:STATE ON') กำหนดให้มีเริ่มต้นวัดความถี่ของสัญญาณ

สังเกตผลที่เกิดขึ้นที่หน้าจอเครื่องวัดสัญญาณจะมีการวัดค่าความถี่ของสัญญาณที่ CH1 บันทึกค่าความถี่ที่วัดได้และเปรียบเทียบกับการกำหนดค่าความถี่จากเครื่องกำเนิดสัญญาณว่ามีค่าเท่ากันหรือไม่

……

……

……

……

12. กำหนดค่าการวัดสัญญาณที่เครื่องวัดสัญญาณโดยพิมพ์คำสั่งดังนี้

>>fprintf(osc, ':MEASU:MEAS2:TYP AMPL') กำหนดให้วัดระดับของสัญญาณ

>>fprintf(osc, ':MEASU:MEAS2:SOU CH1') กำหนดว่าให้ของสัญญาณที่ต่ออยู่ที่ CH1

>>fprintf(osc, ':MEASU:MEAS2:STATE ON') กำหนดให้เริ่มต้นวัดความถี่ของสัญญาณ

สังเกตผลที่เกิดขึ้นที่หน้าจอเครื่องวัดสัญญาณจะมีการวัดค่าระดับของสัญญาณที่ CH1 บันทึกค่าระดับสัญญาณที่วัดได้และเปรียบเทียบกับการกำหนดค่าความถี่จากเครื่องกำเนิดสัญญาณว่ามีค่าเท่ากันหรือไม่

……

……

……

……

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. ทดลองอ่านค่าสัญญาณที่วัดได้มาแสดงผลที่โปรแกรม MATLAB โดยพิมพ์คำสั่งดังนี้

>>fprintf(osc, 'MEASU:MEAS1:VAL?;UNI?') เป็นการร้องขอผลการวัดของข้อมูลที่ 1

>>data_measu1=fscanf(osc) เป็นการนำค่าของข้อมูลที่ 1 มาเก็บไว้ที่ตัวแปรชื่อ data_measu1

บันทึกผลการทดลองที่เกิดขึ้น

.....

.....

.....

.....

14. ทดลองการอ่านค่าสัญญาณที่วัดได้มาแสดงผลที่โปรแกรม MATLAB โดยพิมพ์คำสั่งดังนี้

>>fprintf(osc, 'MEASU:MEAS2:VAL?;UNI?') เป็นการร้องขอผลการวัดของข้อมูลที่ 2

>>data_measu2=fscanf(osc) เป็นการนำค่าของข้อมูลที่ 2 มาเก็บไว้ที่ตัวแปรชื่อ data_measu2

บันทึกผลการทดลองที่เกิดขึ้น

.....

.....

.....

.....

15. ทดลองใช้คำสั่ง **>>fprintf(osc, ':MEASU:MEAS1:STATE OFF')** และ **>>fprintf(osc, ':MEASU:MEAS2:STATE OFF')** สังเกตผลที่เกิดขึ้นที่หน้าจอเครื่องวัดสัญญาณ

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16.ขั้นตอนต่อไปจะเป็นการอ่านค่า InputBufferSize และค่า OutputBufferSize เป็นข้อมูลที่สามารถเขียนและอ่านระหว่างโปรแกรม MATLAB กับเครื่องมือวัดในการติดต่อ 1 ครั้ง โดยมีรายละเอียดดังนี้

16.1 ทดลองอ่านค่า BufferSize ที่ใช้การติดต่อกับเครื่องวัดสัญญาณ พิมพ์คำสั่งดังนี้

>>get(osc, 'InputBufferSize')

>>get(osc, 'OutputBufferSize')

บันทึกผลการทดลอง

.....

.....

.....

16.2 ทดลองอ่านค่า BufferSize ที่ใช้การติดต่อกับเครื่องกำเนิดสัญญาณ พิมพ์คำสั่งดังนี้

>>get(gen, 'InputBufferSize')

>>get(gen, 'OutputBufferSize')

บันทึกผลการทดลอง

.....

.....

.....

17. ขั้นตอนต่อไปจะเป็นการทดลองอ่านค่าสัญญาณที่เกิดขึ้นที่หน้าจอเครื่องวัดสัญญาณมาแสดงผลที่โปรแกรม MALAB พิมพ์คำสั่งดังนี้

>>fprintf(osc, 'Data:Source CH1')

>>fprintf(osc 'Data:Encdg SRPbinary')

>>fprintf(osc 'Data:Width 1')

>>fprintf(osc 'Data:Start 1')

>>fprintf(osc 'Curve?')

>>data_osc = fread(osc, 512)

โดยคำสั่ง fread(Obj,'InputBufferSize') เป็นการอ่านค่าข้อมูลจากเครื่องมือวัด ซึ่งผลที่เกิดขึ้นที่ Command Window คือจะแสดงข้อมูลของ data_osc ที่รับมาจากเครื่องวัดสัญญาณ โดยมีจำนวนทั้งหมด 512 ข้อมูล เนื่องจากค่าของ InputBuferSize ที่สามารถรับได้ทั้งหมด 1 ครั้งมีจำนวนทั้งหมด 512 ไบต์

18. พิมพ์คำสั่ง **>>max(data_osc)** และ **>>min(data_osc)** เพื่อเป็นการอ่านค่าสูงสุดและค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่ำสุดของข้อมูล data_osc บันทึกผลการทดลอง

……

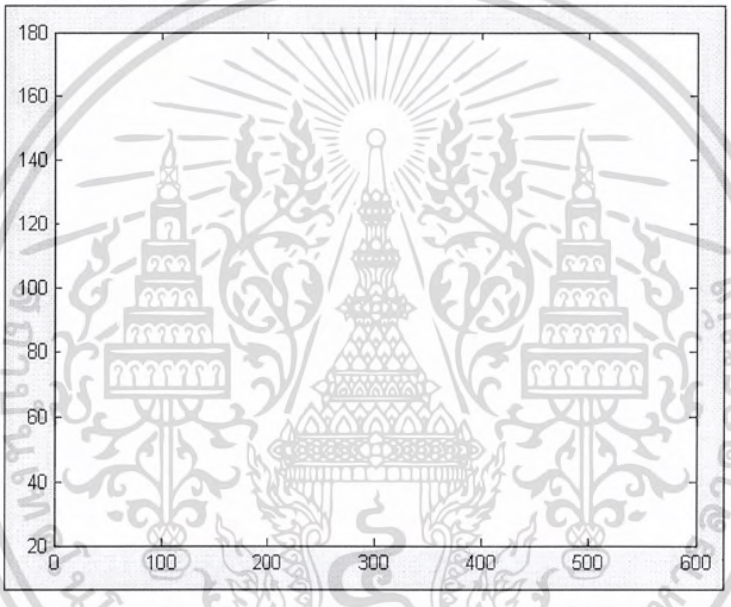
……

……

……

……

19. นำค่าข้อมูล data_osc มาทำการพล็อตกราฟ โดยพิมพ์คำสั่ง >>plot(data_osc) บันทึกผลการทดลองในรูปที่ 7.7



รูปที่ 7.7 ผลการพล็อตกราฟของค่าข้อมูล data_osc

20. พิมพ์คำสั่ง **>>fprintf(osc, '*RST')** เพื่อที่จะรีเซตการอ่านค่าของเครื่องวัดสัญญาณและพิมพ์คำสั่ง **>>fprintf(osc,'Autoset Execute')** เพื่อปรับการแสดงผลให้เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. การกำหนดค่า Primary Address ของเครื่องวัดสัญญาณ มีขั้นตอนวิธีการกำหนดอย่างไร?

.....

.....

.....

.....

.....

.....

.....

2. จากการทดลองกำหนดให้เครื่องวัดสัญญาณรายงานผลค่าความถี่และค่าระดับสัญญาณในขั้นตอนที่ 11 และ 12 นั้น ถ้าต้องการยกเลิกการรายงานผล จะต้องใช้คำสั่งใด?

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ค่า InputBuffer Size และค่า OutputBuffer Size มีความหมายว่าอย่างไร?

.....

.....

.....

.....

.....

.....

.....

4. จากผลการทดลองในขั้นตอนที่ 19 จงอธิบายความแตกต่างของข้อมูลที่พล็อตกราฟกับสัญญาณที่แสดงที่หน้าจอเครื่องวัดสัญญาณ?

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 8
เรื่อง การใช้งานโปรแกรม System Generator เบื้องต้น

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. อธิบายขั้นตอนการใช้งานกล่องเครื่องมือ System Generator ในการออกแบบร่วมกับโปรแกรม MATLAB ได้อย่างถูกต้อง
2. อธิบายขั้นตอนการเชื่อมต่อระหว่างบล็อกของ Xilinx และ Simulink ได้อย่างถูกต้อง
3. แสดงขั้นตอนการรับค่าข้อมูลจากโปรแกรม MATLAB ได้อย่างถูกต้อง
4. แสดงขั้นตอนการส่งค่าข้อมูลออกไปที่โปรแกรม MATLAB ได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0
2. กล่องเครื่องมือ Simulink เวอร์ชั่น 6.0
3. โปรแกรม Xilinx ISE Foundation 6.3i Evaluation
4. โปรแกรม System Generator 6.3i Evaluation

ความรู้และทักษะเดิม

1. การใช้งานโปรแกรม MATLAB และ Simulink เบื้องต้น

ทฤษฎีเบื้องต้น

ผู้ที่เคยใช้งานโปรแกรม MATLAB นั้นจะทราบดีว่าเป็นโปรแกรมที่มีความสามารถสูงในการแก้ปัญหาทางด้านวิศวกรรม เนื่องจากโครงสร้างการเขียนโปรแกรมนั้นไม่ยุ่งยากซับซ้อน และทำสามารถทำความเข้าใจได้ง่าย ทำให้ช่วยลดเวลาในการศึกษาปัญหาในเชิงวิจัยได้เป็นอย่างดี ประกอบกับโปรแกรม MATLAB นี้จะมีกล่องเครื่องมือในด้านสาขาวิชาต่างๆ มากมาย อาทิเช่น Digital Signal Processing, Communication System และ Control System เป็นต้น ทำให้ผู้ใช้งานสามารถพัฒนาโปรแกรมได้อย่างรวดเร็วและคุณสมบัติเด่นอีกประการหนึ่งก็คือมีกล่องเครื่องมือ Simulink ซึ่งช่วยในการออกแบบโมเดลเพื่อจำลองการทำงานของระบบก่อนที่จะนำไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

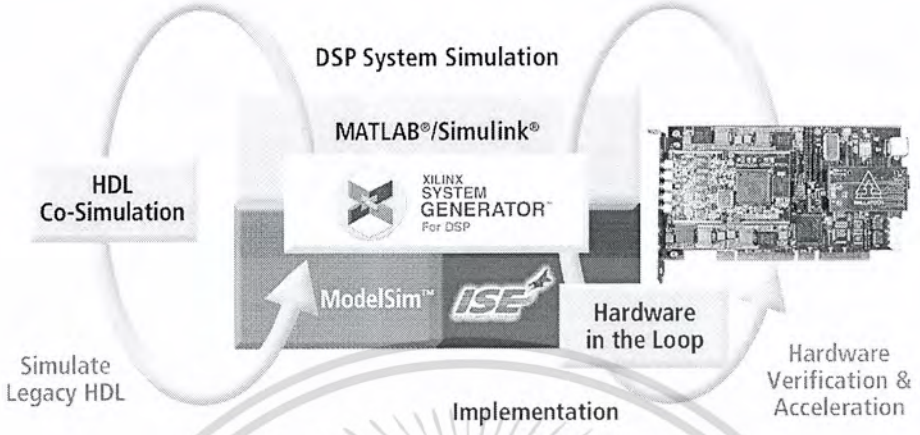
โดยลักษณะการออกแบบโมเดลจะเป็นการนำบล็อกต่างๆ มาเชื่อมต่อกันเป็นระบบทำให้ง่ายต่อการศึกษาและทำความเข้าใจภาพรวมของระบบได้ง่ายและรวดเร็วขึ้น

โดยจะเห็นว่าการใช้งานโปรแกรม MATLAB นั้นจะใช้ในการช่วยแก้ปัญหาเชิงวิจัยเป็นส่วนใหญ่ ซึ่งการใช้โปรแกรม MATLAB เชื่อมต่อกับฮาร์ดแวร์จริงนั้นยังมีผู้สนใจและพัฒนาอยู่ในระดับน้อยมากเมื่อเทียบกับแบบแรก แต่โปรแกรม MATLAB ก็พัฒนาขีดความสามารถในการเชื่อมต่อกับฮาร์ดแวร์ขึ้นไปเรื่อยๆ โดยจะเห็นได้จากเวอร์ชั่นใหม่ๆ จะมีการเพิ่มเติมกล่องเครื่องมือที่สามารถรองรับกับงานทางด้านนี้เรื่อยๆ เช่น ความสามารถในการเชื่อมต่อกับไมโครโปรเซสเซอร์ ไมโครคอนโทรลเลอร์หรือชิพ DSP เป็นต้น

ในปัจจุบันเทคโนโลยีในด้านการออกแบบวงจรรวมขนาดใหญ่มาก (VLSI : Very Large-Scale Integrated Circuit) ได้มีการพัฒนาไปอย่างรวดเร็ว จะเห็นได้จากหน่วยประมวลผลกลาง (Central Processing Unit) ในเครื่องคอมพิวเตอร์ที่ใช้งานในปัจจุบัน ซึ่งได้มีระบบการประมวลผลถึงระดับ 64 บิทและใช้เทคโนโลยีในการออกแบบ 90 nm (นาโนเมตร) หมายถึงตัวทรานซิสเตอร์มีขนาดเท่ากับ 1/1000 ล้านเมตร เป็นต้น โดยการออกแบบตัวชิพนี้จะอยู่ในรูปของ Full Custom หรือชิพเฉพาะงาน (Application Specific Integrated Circuit : ASIC) ซึ่งหลายคนอาจจะมองว่าเป็นเทคโนโลยีชั้นสูงที่ยุ่งยากซับซ้อนต่อการทำความเข้าใจและต้องใช้เครื่องมือในการออกแบบที่มีประสิทธิภาพสูง แต่ในทางกลับกันได้มีการศึกษาและพัฒนาให้การออกแบบชิพอยู่ในรูปที่สามารถโปรแกรมได้หรือที่เรียกว่า โปรแกรมเมเบิ้ลชิพ (Programmable Chip) ทำให้การออกแบบรวดเร็วขึ้นและสามารถแก้ไขได้ง่าย ซึ่งช่วยลดเวลาในการออกแบบและพัฒนาได้เป็นอย่างดี โดยโปรแกรมเมเบิ้ลชิพ ที่ได้รับความนิยมและใช้งานอย่างแพร่หลายในปัจจุบันคือ FPGA (Field-Programmable Gate Array)

ในโปรแกรม MATLAB เวอร์ชั่นใหม่ๆ ได้มีการเพิ่มขีดความสามารถในการออกแบบระบบเพื่อโปรแกรมและเชื่อมต่อกับอุปกรณ์ FPGA ได้โดยจะต้องมีซอร์ฟแวร์ที่นำมาใช้งานร่วมกับโปรแกรม MATLAB ด้วย เช่นโปรแกรม System Generator และ Xilinx ISE Foundation โดยเป็นโปรแกรมที่พัฒนาโดยบริษัท Xilinx.Inc ประเทศสหรัฐอเมริกา ซึ่งเป็นผู้ผลิตชิพ FPGA ที่มีผู้ใช้งานอย่างแพร่หลายในปัจจุบันที่รู้จักกันในชื่อ Xilinx การออกแบบชิพโดยใช้โปรแกรม MATLAB ร่วมกับโปรแกรมของบริษัท Xilinx นั้นแสดงได้ดังรูปที่ 8.1

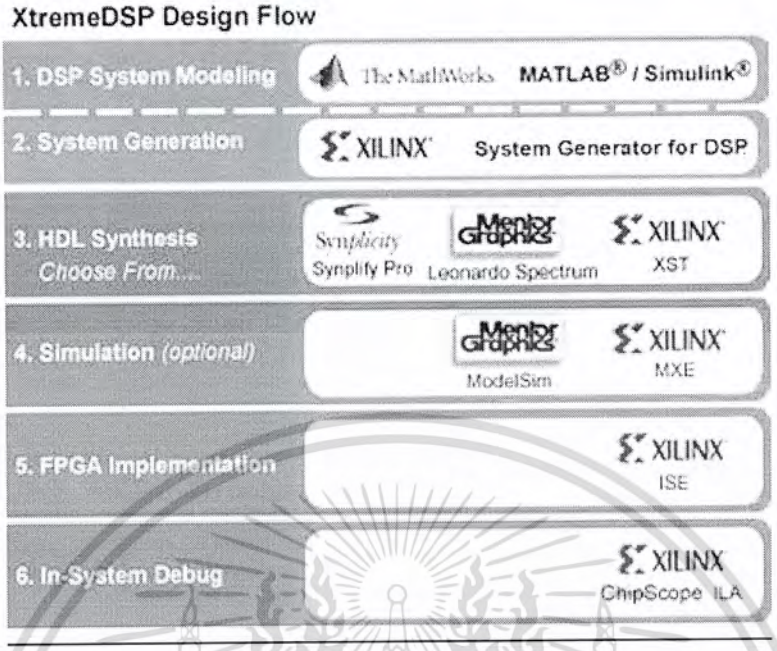
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.1 การทำงานของโปรแกรม MATLAB ร่วมกับโปรแกรม System Generator

การนำเอาโปรแกรม MATLAB มาช่วยในการออกแบบนั้นเพราะว่าโปรแกรม MATLAB นั้นมีกล่องเครื่องมือในการวิเคราะห์ในด้านวิศวกรรมที่มีประสิทธิภาพสูงอยู่แล้ว เพียงแต่เพิ่มเติมเครื่องมือที่ทำหน้าที่แปลงภาษา MATLAB เป็นภาษาที่สามารถโปรแกรมลงบนชิพ FPGA ได้ เช่น ภาษา VHDL หรือ Verilog เป็นต้น และด้วยความสามารถของโปรแกรม System Generator ยังสามารถที่จะเชื่อมต่อกับโปรแกรมจำลองการทำงานอื่นๆ เช่น ModelSim กับการจำลองการทำงานพื้นฐานของ Simulink ได้ หรือที่เรียกว่า HDL Co-Simulation และเมื่อโปรแกรมลงบน FPGA แล้วสามารถที่จะจำลองการทำงานของฮาร์ดแวร์จริงๆ แล้วมาแสดงผลที่โปรแกรม MATLAB ได้ หรือที่เรียกว่า Hardware in the Loop โดยอุปกรณ์ FPGA ที่สามารถใช้งานได้คือจะต้องเป็นชิพ FPGA ของบริษัท Xilinx เท่านั้น เช่นรุ่น Spartan3, Vertex2 หรือ Vertex4 เป็นต้น ขั้นตอนการออกแบบระบบเพื่อการโปรแกรมลงบน FPGA นั้นแสดงขั้นตอนดังรูปที่ 8.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.2 ขั้นตอนการออกแบบระบบด้วยโปรแกรม System Generator

1) การออกแบบระบบจาก MATLAB และ Simulink (DSP System Modeling)

เมื่อทำการติดตั้งโปรแกรม System Generator แล้วจะมีกล่องเครื่องมือของ Xilinx เพิ่มเข้าไปในไลบรารี่ของ Simulink การออกแบบระบบ จะเป็นการนำเอาบล็อกต่างๆ ของ Xilinx มาต่อร่วมกับบล็อกพื้นฐานของ Simulink เพื่อการจำลองการทำงาน แต่การสร้างแปลงโค๊ดจะจำกัดแค่บล็อกของ Xilinx เท่านั้น

2) การแปลงระบบให้เป็นไฟล์สำหรับ FPGA (System Generator)

ในขั้นตอนนี้จะใช้ความสามารถของ System Generator แปลงระบบที่ออกแบบจาก Simulink ให้เป็นไฟล์สำหรับโปรแกรมลงบน FPGA ได้เช่น HDL Netlist, NGC Netlist หรือ Bitstream เป็นต้น

3) การสังเคราะห์วงจร (HDL Synthesis)

เป็นขั้นตอนการสังเคราะห์วงจร จะเป็นการการนำไฟล์ HDL มาสังเคราะห์เป็นวงจรในระดับ RTL (Register Transfer Level) ให้สามารถที่จะนำไปโปรแกรมลงบน FPGA ได้ โดยจะใช้โปรแกรม Xilinx ISE Foundation ซึ่งมีตัวสังเคราะห์วงจรอยู่แล้วเรียกว่า XST (Xilinx Synthesis-Tool) แต่ก็สามารถเลือกตัวสังเคราะห์วงจรอื่นได้ โดยผู้ใช้งานจะต้องลงโปรแกรมเพิ่มเติมเอง เช่น Leonardo Spectrum, Synplify, Synplify Pro เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การจำลองการทำงาน (Simulation)

เป็นการทดสอบการทำงานของระบบก่อนที่จะโปรแกรมลงบน FPGA โดยการนำไฟล์ทดสอบการทำงาน (Testbench file) ที่ได้ในขั้นตอนการแปลงระบบเป็นไฟล์ FPGA มาใช้ในการจำลองการทำงาน เพื่อตรวจสอบความถูกต้อง ซึ่งสามารถใช้โปรแกรมได้หลายตัว เช่น ModelSim XE หรือ ModelSim SE เป็นต้น

5) การโปรแกรมลงบน FPGA (FPGA Implementation)

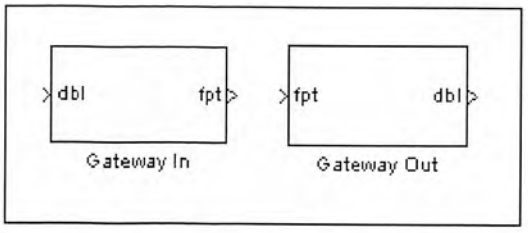
เป็นการนำวงจรที่ได้ในขั้นตอนการสังเคราะห์วงจร มาสร้างลงบนชิพ FPGA จริงโดยจะคำนึงถึงสถาปัตยกรรมของชิพที่ใช้งานเป็นหลัก ซึ่งมีขั้นตอนคือ Translate, Map, Place&Route ไฟล์ที่ได้หลังจากขั้นตอนนี้คือ ไฟล์ Bitstream ซึ่งใช้ในการโปรแกรมลงบนชิพ FPGA

6) การทดสอบการทำงานของ FPGA(In-System Debug)

เป็นขั้นตอนที่ที่เพิ่มเข้ามาเพื่อจะทดสอบการทำงานของชิพ FPGA หลังจากการโปรแกรมแล้ว และทำงานร่วมกับอุปกรณ์อื่นๆ ในวงจร โดยจะต้องมีซอฟต์แวร์ติดตั้งเพิ่มเติม โปรแกรมที่ใช้คือ Xilinx ChipScope Pro

ในการทดลองนี้จะเป็นการใช้งานโปรแกรม MATLAB ร่วมกับโปรแกรม System Generator เบื้องต้น โดยต้องออกแบบใน Simulink เป็นการนำเอาบล็อกต่างๆ ของ Xilinx มาออกแบบร่วมกับบล็อกพื้นฐานของ Simulink จุดประสงค์ที่นำเอาบล็อกทั้งสองมาเชื่อมต่อกันเพราะว่าต้องการนำเอาสัญญาณอินพุตที่มีอยู่อย่างมากมายในไลบรารี่ของ Simulink มาทดสอบระบบที่ได้ออกแบบ และนำเอาบล็อกที่เป็นเอาต์พุตมาจำลองการทำงานให้เห็นผลที่เกิดขึ้นของระบบ การเชื่อมต่อบล็อกจะมีข้อจำกัดอยู่ว่า บล็อกพื้นฐานของ Simulink จะประมวลผลแบบระบบเลขทศนิยม (Floating Point) แต่การประมวลผลของบล็อก Xilinx จะเป็นระบบเลขที่สามารถใช้ได้จริงเมื่อสร้างเป็นฮาร์ดแวร์ เช่น Unsigned, Boolean หรือ Sign 2's Complement ซึ่งเป็นคุณสมบัติสำคัญเพราะจะคำนึงถึงการทำงานของระบบเป็นฮาร์ดแวร์จริงไม่ใช่ซอร์ฟแวร์ ดังนั้นการเชื่อมต่อบล็อกทั้งสอง จะต้องมีบล็อกที่ชื่อว่า Gateway In และ Gateway Out ทำหน้าที่แปลงระบบจำนวนเลขของ Simulink ไปเป็นระบบจำนวนเลขที่สามารถทำงานได้จริงในฮาร์ดแวร์ เสมือนว่าบล็อกทั้งสองเป็นภาค Quantization ของวงจรแปลงแอนะลอกเป็นดิจิตอล (Analog to Digital Converter) และวงจรแปลงดิจิตอลเป็นแอนะลอก (Digital to Analog Converter) นั่นเอง ซึ่งในการทดลองนี้ได้ศึกษาการกำหนดค่าพารามิเตอร์ต่างๆ และสังเกตผลที่เกิดขึ้นหลังจากการกำหนดค่าพารามิเตอร์ของบล็อกทั้งสอง โดยบล็อก Gateway In และบล็อก Gateway ดังรูปที่ 8.3

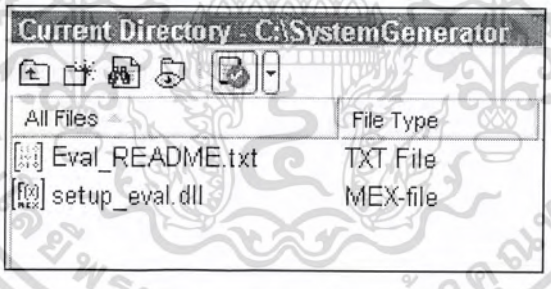
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.3 บล็อก Gateway In และบล็อก Gateway Out

ลำดับขั้นการทดลอง

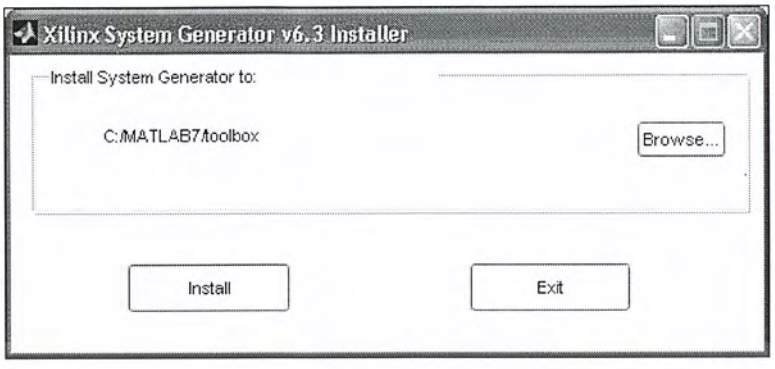
1. ขั้นตอนการติดตั้งโปรแกรม System Generator ต้องทำในโปรแกรม MATLAB เท่านั้น โดยก่อนการติดตั้งต้องลงโปรแกรม Xilinx ISE Foundation ก่อน การติดตั้งต้องคัดลอกไฟล์โปรแกรม System Generator มาเก็บในไดเร็กทอรี่ของไดร์ฟ C: ในขั้นตอนนี้เก็บไว้ที่ **C:\ System Generator** เปิดโปรแกรม MATLAB เลือกไดเร็กทอรี่ใช้งานเป็น C:\ System Generator จะแสดงไฟล์ของโปรแกรม 2 ไฟล์ แสดงดังรูปที่ 8.4 ใช้คำสั่ง **>>setup_eval** ที่ MATLAB Command คลิกที่ **Install** เพื่อเริ่มการติดตั้งแสดงดังรูปที่ 8.5



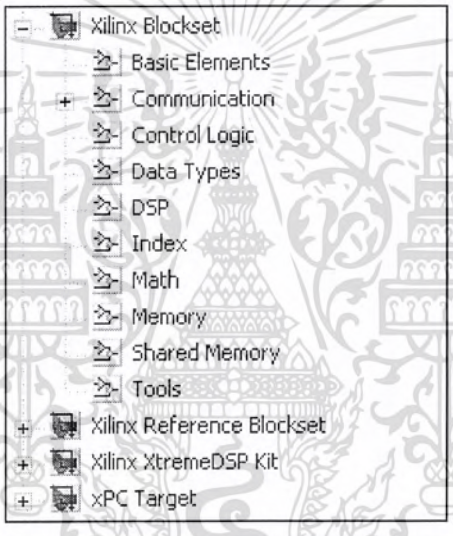
รูปที่ 8.4 ไฟล์โปรแกรม System Generator

2. เมื่อทำการติดตั้งเสร็จสิ้นแล้ว เปิดกล่องเครื่องมือ Simulink โดยคลิกที่ หรือใช้คำสั่ง **>>Simulink** ที่ Command Window จะพบไลบารี่ของ **Xilinx** เพิ่มเข้ามาแสดงว่าถูกต้องแสดงดังรูปที่ 8.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.5 การติดตั้งโปรแกรม System Generator



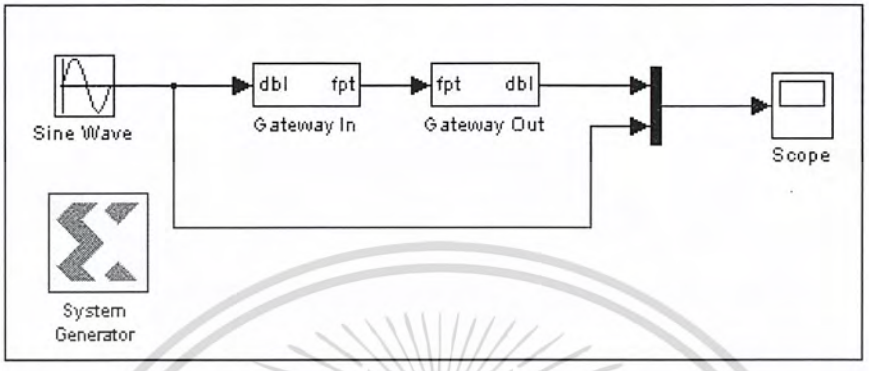
รูปที่ 8.6 บล็อกต่างๆ ของ Xilinx

3. สร้างโมเดลใหม่ (New Model) โดยคลิกที่ ให้สร้างบล็อกต่างๆ มีรายละเอียดดังนี้

- คลิกที่ไลบารี่ **Simulink** คลิกที่หัวข้อ **Source** เลือก **Sine Wave**
- คลิกที่ไลบารี่ **Simulink** คลิกที่หัวข้อ **Sink** เลือก **Scope**
- คลิกที่ไลบารี่ **Simulink** คลิกที่หัวข้อ **Signal Routing** เลือก **Mux**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **System Generator**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **Gateway In**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **Gateway Out**

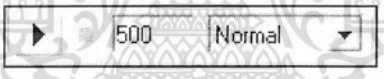
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ลากเส้นเชื่อมต่อและตั้งชื่อบล็อกต่างๆ เสร็จแล้วบันทึกโมเดลตั้งชื่อเป็น **lab8** ในไดเร็ก-ทอรี่ **C:\MATLAB7\LAB8** แสดงดังรูปที่ 8.7



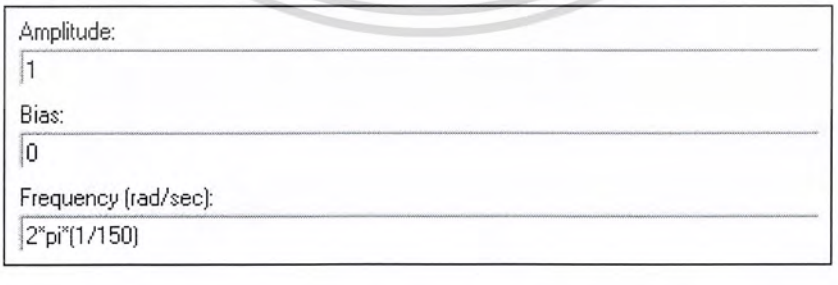
รูปที่ 8.7 การลากเส้นเชื่อมต่อของโมเดล lab8

5. กำหนดเวลาในการจำลองที่แถบเมนูด้านบนเป็น **500** แสดงดังรูปที่ 8.8



รูปที่ 8.8 การกำหนดค่าเวลาในการจำลองการทำงาน

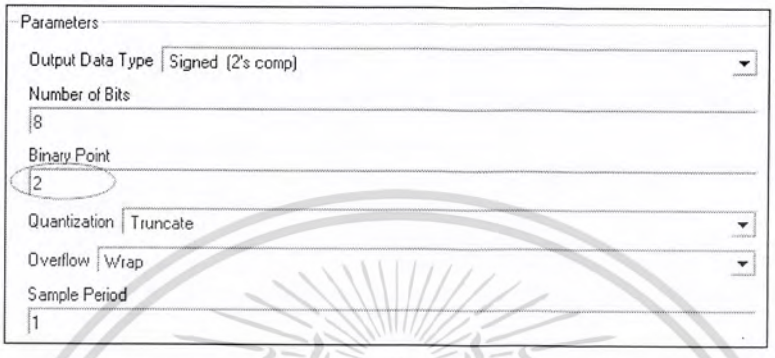
6. ดับเบิ้ลคลิกที่ **Sine Wave** กำหนดค่าความถี่เป็น **2*pi*(1/150)** ซึ่งมีความถี่เป็น (1/150) Hz เพื่อให้เห็นลักษณะการ Quantization ของสัญญาณที่ชัดเจน แสดงดังรูปที่ 8.9



รูปที่ 8.9 การกำหนดค่าความถี่ของบล็อก Sine Wave

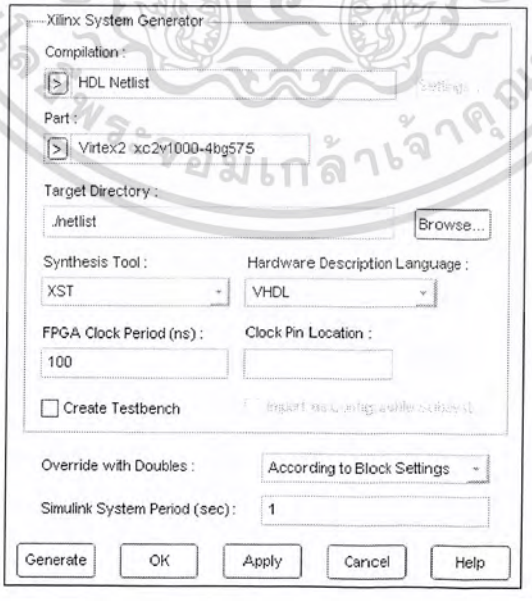
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ดับเบิ้ลคลิกที่บล็อก **Gateway In** กำหนดค่า **Number of bit** เป็น **8** และ **Binary Point** เป็น **2** แสดงดังรูปที่ 8.10 หมายความว่ากำหนดระบบจำนวนเลขเป็น 2's Complement จำนวน 8 บิต โดยมีข้อมูลทศนิยม 2 บิตจากทั้งหมด 8 บิต ดังนั้นจะเหลือข้อมูลที่เป็นจำนวนเต็ม 6 บิต



รูปที่ 8.10 การกำหนดค่าระบบจำนวนเลขของบล็อก Gateway In

8. ดับเบิ้ลคลิกที่บล็อก **System Generator** จะปรากฏหน้าต่าง System Generator แสดงดังรูปที่ 8.11 เป็นการกำหนดค่าพารามิเตอร์ต่างๆ ที่ใช้ในการสร้างโค๊ด VHDL หรือ Verilog จากโมเดลที่ออกแบบ แต่ในใบงานการทดลองนี้จะไม่มีการสร้างโค๊ดให้กำหนดตามค่าที่โปรแกรมตั้งไว้คลิก **Ok** ดังรูปที่ 8.11



รูปที่ 8.11 หน้าต่าง System Generator

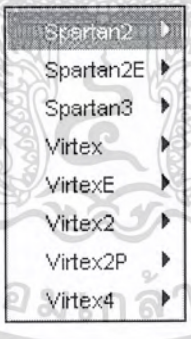
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของค่าพารามิเตอร์ในหน้าต่าง System Generator มีรายละเอียดดังนี้

ในหัวข้อ **Compilation** เป็นการกำหนดไฟล์ที่ต้องการหลังจากการสร้างโค๊ด ค่าเริ่มต้นของโปรแกรมจะตั้งไว้ที่ HDL Netlist ซึ่งจะสร้างเป็นไฟล์ HDL และสร้างเป็นไฟล์โปรเจ็คของโปรแกรม Xilinx ISE Foundation แสดงดังรูปที่ 8.12

รูปที่ 8.12 การกำหนดไฟล์ที่ต้องการสร้างโค๊ด

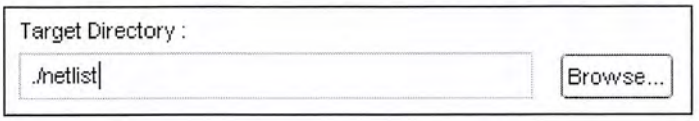
ในหัวข้อ **Part** เป็นการกำหนดชิพ FPGA ที่ต้องการโปรแกรมหลังการสร้างโค๊ด โดยจะต้องกำหนดรุ่นของชิพ เบอร์ของชิพและขนาดของตัวถังของชิพให้ถูกต้อง ซึ่งมีรายการชิพที่สามารถใช้ได้ดังรูปที่ 8.13



รูปที่ 8.13 การเลือกชิพ FPGA ที่ต้องการใช้งาน

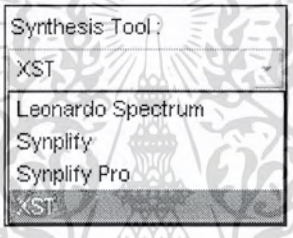
ในหัวข้อ **Target Directory** เป็นการกำหนดโฟล์เดอร์ปลายทางที่เก็บไฟล์ทั้งหมดหลังการสร้างโค๊ด ซึ่งโปรแกรมจะตั้งค่าเริ่มต้นเป็น ./netlist แสดงดังรูปที่ 8.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



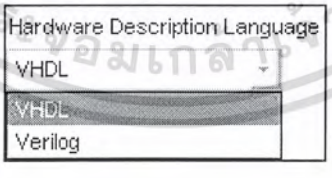
รูปที่ 8.14 โฟล์เดอร์ปลายทางที่เก็บไฟล์หลังการสร้างโค๊ด

ในหัวข้อ **Synthesis Tool** เป็นการกำหนดโปรแกรมที่ใช้สำหรับสังเคราะห์วงจร โดยจะนำโค๊ดที่ได้จากการสร้างโค๊ดมาสังเคราะห์เป็นวงจรในระดับลอจิก (RTL) ซึ่งการใช้งานต้องติดตั้งโปรแกรมสังเคราะห์วงจรเอง ซึ่งโปรแกรมจะตั้งค่าเริ่มต้นเป็น XST (Xilinx Synthesis Tool) เป็นตัวสังเคราะห์วงจรของโปรแกรม Xilinx ISE Foundation แสดงดังรูปที่ 8.15



รูปที่ 8.15 การเลือกโปรแกรมสังเคราะห์วงจร

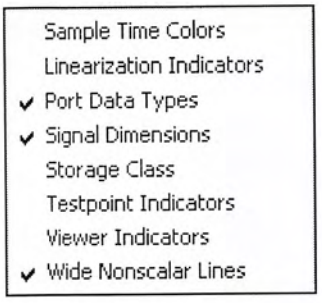
ในหัวข้อ **Hard Ware Description Language** เป็นการกำหนดภาษาที่ต้องการสร้างโค๊ด โดยสามารถกำหนดได้ 2 ภาษาคือ VHDL และ Verilog แสดงดังรูปที่ 8.16



รูปที่ 8.16 การกำหนดภาษาที่ต้องการสร้างโค๊ด

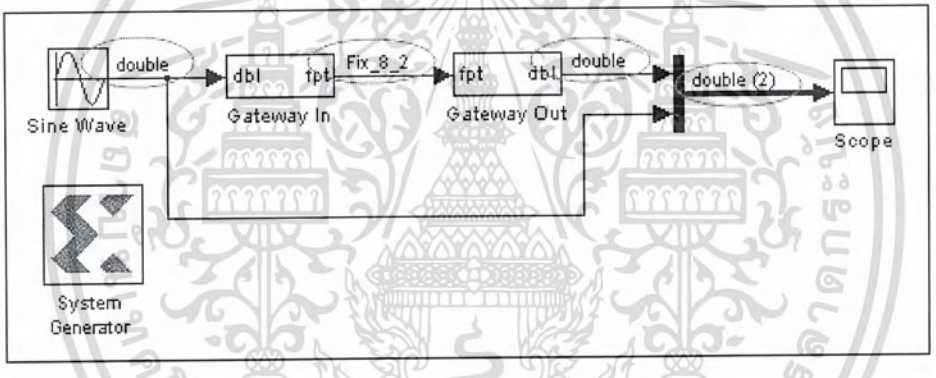
9. บนเมนูบาร์ด้านบน คลิกที่ **Format > Port/Signal Display** กำหนดค่าดังรูปที่ 8.17 ซึ่งเป็นการกำหนดให้แสดงผลของระบบจำนวนเลขที่ผ่านเข้าและออกบล็อกต่างๆ ดังรูปที่8.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.17 การกำหนดค่า Port/Signal Display

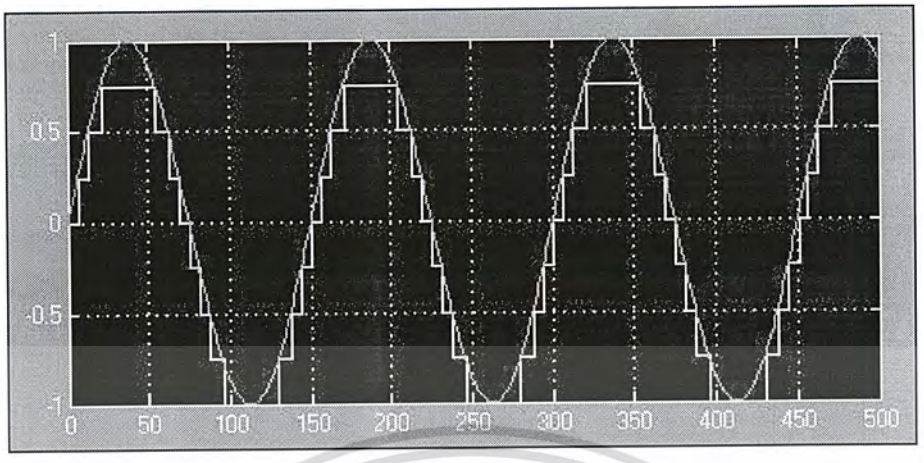
10. สังเกตการเปลี่ยนแปลงที่บล็อกต่างๆ ถ้าไม่มีการเปลี่ยนแปลงให้คลิกที่ **Edit > Update Diagram** จะแสดงผลที่เกิดขึ้นที่บล็อกต่างๆ ดังรูปที่ 8.18



รูปที่ 8.18 ผลการเปลี่ยนแปลงของบล็อกต่างๆ หลังการกำหนดค่า Port/Signal Display

11. กลับมาที่หน้าต่าง lab8.mdl จำลองการทำงานของระบบโดยคลิกที่ปุ่ม ▶ และ**ดับเบิ้ลคลิก**ที่บล็อก **Scope** ขึ้นมาจะแสดงผลดังรูปที่ 8.19 ถ้าไม่แสดงดังรูปให้คลิกขวาเลือก **AutoScale**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

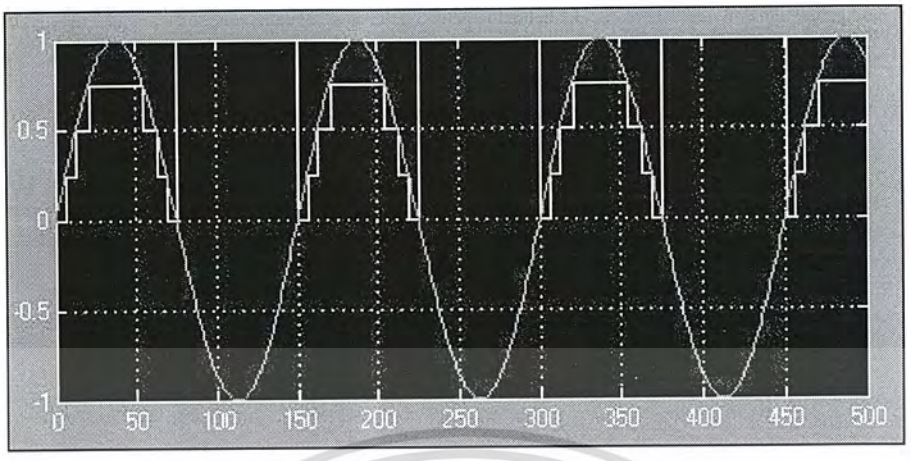


รูปที่ 8.19 ผลการจำลองการทำงานเมื่อกำหนดค่าเป็น 2's Complement, Overflow เป็น Truncate และ Quantization เป็น Wrap

จะเห็นว่าขอบของสัญญาณจะมีลักษณะหยาบกว่าสัญญาณที่รับเข้ามาเนื่องจาก สัญญาณที่รับเข้ามาจะเป็น Double precision floating point แต่สัญญาณที่ใช้ในการประมวลผลจะเป็น fixed-point 8_2 เมื่อผ่านบล็อก Gateway Out ที่ทำการแปลงกลับเป็น Double precision floating point อีกครั้งหนึ่งจะมีค่าผิดพลาดจากการจัดระดับสัญญาณ เรียกว่า Quantization Noise การลดค่าความผิดพลาดนี้สามารถทำได้โดยกำหนดลักษณะการ Quantization แบบอื่น หรือกำหนดจำนวนบิตของข้อมูลให้มากขึ้น ซึ่งจะแสดงขั้นตอนการปรับแต่งในขั้นตอนต่อไป

12. เปลี่ยนแปลงค่าพารามิเตอร์ของบล็อก Gateway In ในหัวข้อ **Output data Type** กำหนดเป็น **Unsigned** คลิกที่ปุ่ม ▶ สังเกตผลการจำลองการทำงานที่บล็อก Scope แสดงดังรูปที่ 8.20 ถ้าไม่แสดงดังรูปที่ 8.20 ให้คลิกขวาเลือก **Axes Properties** กำหนดค่า **Ymin = -1 , Ymax = 1**

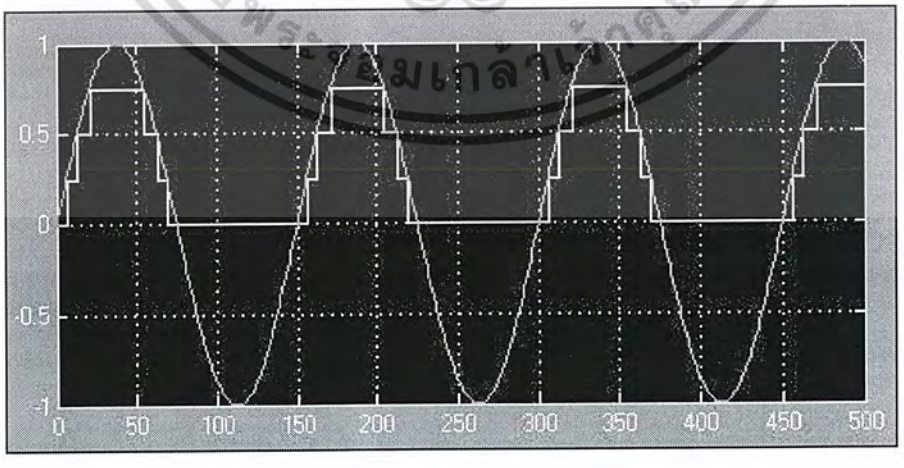
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.20 ผลการจำลองการทำงานเมื่อกำหนดค่าเป็น Unsigned, Overflow เป็น Truncate และ Quantization เป็น Wrap

ผลการจำลองการทำงานจะเห็นว่าที่สัญญาณเอาต์พุตจะไม่มีข้อมูลทางด้านลบ เนื่องจากได้มีการกำหนดระบบจำนวนในการประมวลเป็น Unsigned จึงไม่มีการคิดเครื่องหมายของสัญญาณทางด้านลบ ผลที่ได้คือจะจัดระดับสัญญาณทางด้านบวกเท่านั้น เมื่อผ่านบล็อก Gateway Out จึงแสดงดังรูปที่ 8.20

13. เปลี่ยนแปลงค่าพารามิเตอร์ของบล็อก Gateway In ในหัวข้อ **Output data Type** กำหนดเป็น **Unsigned** ในหัวข้อ **Overflow** เป็น **Saturate** คลิกที่ปุ่ม ▶ สังเกตผลการจำลองการทำงานที่บล็อก Scope แสดงดังรูปที่ 8.21

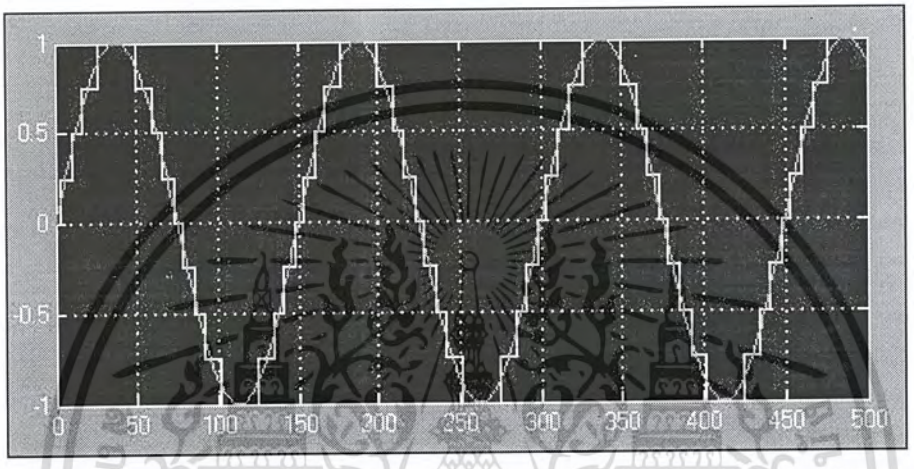


รูปที่ 8.21 ผลการจำลองการทำงานเมื่อกำหนดค่าเป็น Unsigned, Overflow เป็น Saturate และQuantization เป็น Wrap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่า Over flow เป็น Saturate นั้นจะทำให้ค่าของข้อมูลที่เกิดการ Over flow จะมีค่าเฉลี่ยลดลงเป็น 0 จะสังเกตว่าสัญญาณในด้านลบจะมีค่าเป็น 0 ทั้งหมด

14. เปลี่ยนแปลงค่าพารามิเตอร์ของบล็อก Gateway In ในหัวข้อ **Output data Type** กำหนดเป็น **2's Complement** ในหัวข้อ **Overflow** เป็น **Saturate** และในหัวข้อ **Quantization** เป็น **Round** คลิกที่ปุ่ม ▶ สังเกตผลการจำลองการทำงานที่บล็อก Scope แสดงดังรูปที่ 8.22

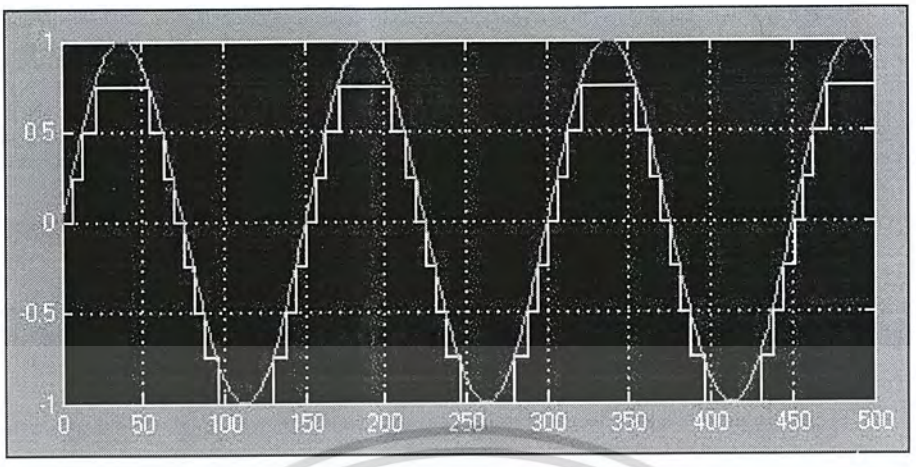


รูปที่ 8.22 ผลการจำลองการทำงานเมื่อกำหนดค่าเป็น 2's Complement, Overflow เป็น Saturate และ Quantization เป็น Round

จากผลการทดลองจะเห็นได้ว่าสัญญาณมีคุณภาพมากขึ้นเนื่องจากได้มีการกำหนดลักษณะการ Quantization เป็น Round ดังนั้นจะมีการปัดเศษของระบบจำนวนให้เหมาะสมกับสัญญาณที่ได้รับเข้ามา

15. เปลี่ยนแปลงค่าพารามิเตอร์จากขั้นตอนที่ 13 ในหัวข้อ **Quantization** เป็น **Truncate** คลิกที่ปุ่ม ▶ สังเกตผลการจำลองการทำงานที่บล็อก Scope แสดงดังรูปที่ 8.23

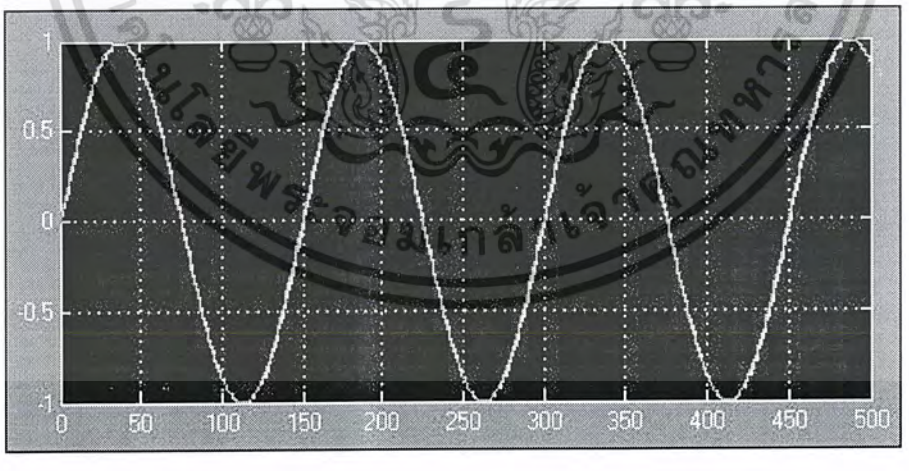
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.23 ผลการจำลองการทำงานเมื่อกำหนดค่าเป็น 2's Complement, Overflow เป็น Saturate และ Quantization เป็น Truncate

เมื่อปรับเปลี่ยนค่า Quantization เป็น Truncate สัญญาณทางด้านบวกจะลดระดับการ Quantization ลงมา แต่ก็เป็นค่าที่ยอมรับได้

16. เปลี่ยนแปลงค่าพารามิเตอร์ของบล็อก Gateway In ในหัวข้อ **Binary Point** เปลี่ยนค่าเป็น **6** แล้วจำลองการทำงาน จะแสดงผลดังรูปที่ 8.24

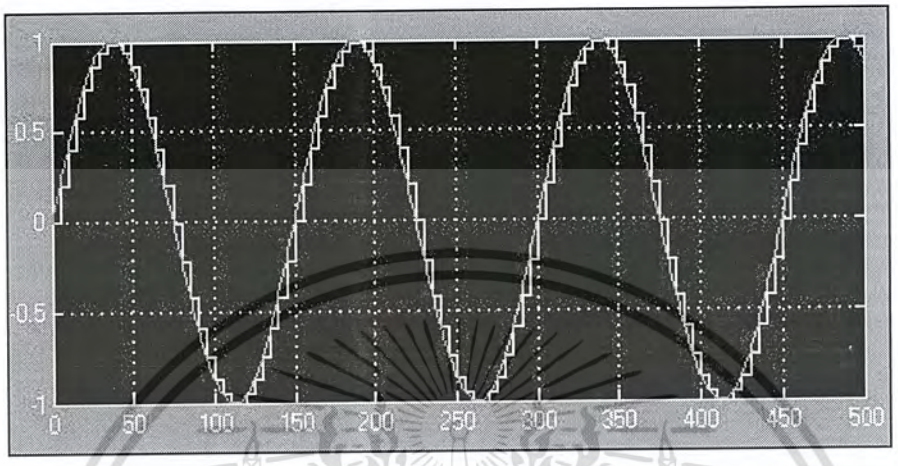


รูปที่ 8.24 ผลการจำลองการทำงานเมื่อกำหนดค่าจำนวนบิตของข้อมูลเป็น 8_6

จากผลการทดลองจะเห็นว่าการเพิ่มจำนวนบิตของข้อมูลจะทำให้สัญญาณมีคุณภาพมากขึ้น เนื่องจากจำนวนบิตที่ใช้มีผลต่อการเก็บค่าที่เป็นทศนิยมมากขึ้น จึงทำให้ความผิดพลาดของ Quantization Noise ลดลงมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. จากขั้นตอนที่ 16 กำหนดค่าเพิ่มเติมลงในหัวข้อ **Sample Period** กำหนดค่าใหม่เป็น **5** แล้วจำลองการทำงาน จะแสดงผลดังรูปที่ 8.25



รูปที่ 8.25 ผลการจำลองการทำงานเมื่อกำหนดค่า Sample Period เป็น 5

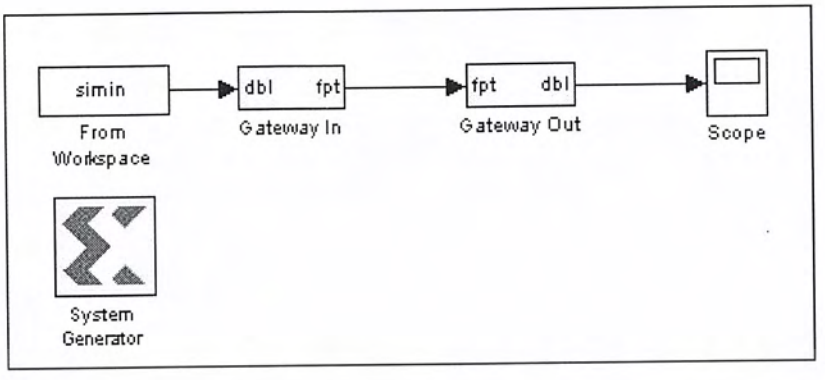
การกำหนดค่าช่วงเวลาการสุ่มสัญญาณนั้นก็มีผลกับคุณภาพของสัญญาณที่ได้สังเกตว่าจากเดิมกำหนดที่กำหนดเป็น 1 แล้วเปลี่ยนแปลงค่าเป็น 5 ผลที่ได้คือสัญญาณจะมีความหยาบลง 5 เท่าจากเดิม

18. ขั้นตอนต่อไปจะเป็นการทดลองการรับค่าข้อมูลจาก MATLAB Workspace ให้เป็นสัญญาณอินพุตให้กับโมเดล สร้างโมเดลใหม่ (New Model) โดยคลิกที่ ให้สร้างบล็อกต่างๆ มีรายละเอียดดังนี้

- คลิกที่ไลบารี่ **Simulink** คลิกที่หัวข้อ **Source** เลือก **From Workspace**
- คลิกที่ไลบารี่ **Simulink** คลิกที่หัวข้อ **Sink** เลือก **Scope**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **System Generator**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **Gateway In**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **Gateway Out**

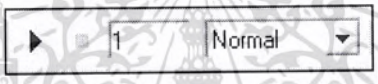
19. ลากเส้นเชื่อมต่อและตั้งชื่อบล็อกต่างๆ เสร็จแล้วบันทึกโมเดลตั้งชื่อเป็น **lab8_2** แสดงดังรูปที่ 8.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



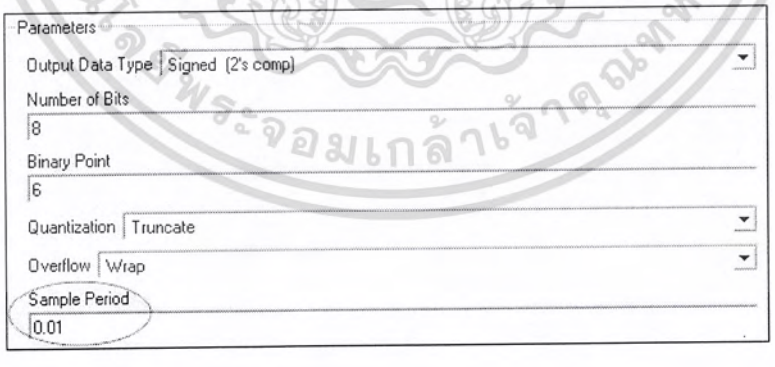
รูปที่ 8.26 การเชื่อมต่อโมเดล lab8_2.mdl

20. กำหนดเวลาในการจำลองที่แถบเมนูด้านบนเป็น **1** แสดงดังรูปที่ 8.27



รูปที่ 8.27 กำหนดค่า Stop Time เป็น 1

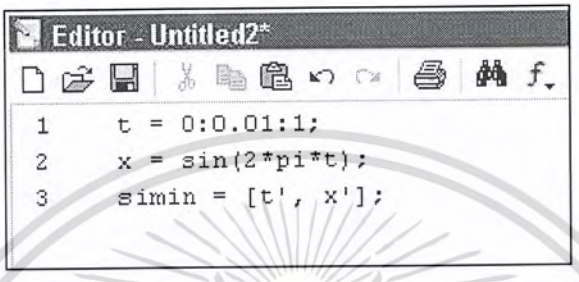
21. ดับเบิ้ลคลิกที่บล็อก **Gateway In** กำหนดค่า **Sample Period** เป็น **0.01** ค่าอื่นๆ กำหนดตามที่โปรแกรมตั้งไว้ แสดงดังรูปที่ 8.28



รูปที่ 8.28 การกำหนดค่าพารามิเตอร์ของบล็อก Gateway In

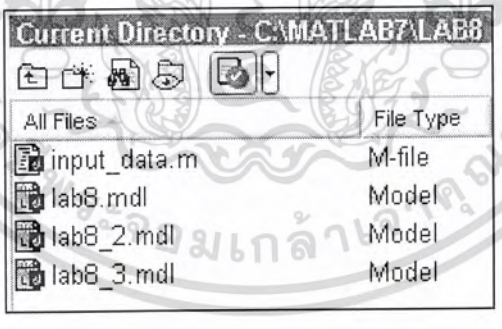
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

22. กลับมาที่หน้าต่าง MATLAB จะสร้าง M-file ที่เก็บค่าสัญญาณอินพุตที่มีชื่อว่า simin โดยเป็นสัญญาณ Sine Wave ที่มีความถี่เท่ากับ 1 Hz บนเมนูบาร์ด้านบน คลิกที่ **File >New > M-file** จะปรากฏหน้าต่าง Editor ขึ้นมาเป็นหน้าต่างที่เอาไว้เขียนโปรแกรมที่เป็น M-File เขียนโปรแกรมดังรูปที่ 8.29



รูปที่ 8.29 การเขียน M-file กำเนิดสัญญาณ Sine Wave

เมื่อเขียนโปรแกรมเสร็จสิ้น บนเมนูบาร์ด้านบนคลิกที่ **File > Save As** ตั้งชื่อเป็น **input_data** บันทึกที่ **C:\MATLAB7\LAB8** เมื่อเสร็จสิ้นจะแสดงดังรูปที่ 8.30 และทำการปิดหน้าต่าง Editor



รูปที่ 8.30 ผลจากการบันทึกไฟล์ input_data.m

23. พิมพ์คำสั่ง **>>clear all** เพื่อลบข้อมูลทั้งหมดที่มีอยู่ใน Workspace พิมพ์คำสั่ง **>>whos** เพื่อดูข้อมูลใน Workspace จะสังเกตเห็นว่าจะไม่มีข้อมูลใดๆ อยู่

24. พิมพ์คำสั่ง **>>input_data** เพื่อเป็นการเรียกไฟล์ input_data.m โดยเมื่อไฟล์นี้ทำงานจะทำการสร้างค่าพารามิเตอร์ที่เป็นสัญญาณ Sine Wave ทดลองพิมพ์คำสั่ง **>>whos บันทึกผลในตารางที่ 8.1** ที่เกิดขึ้นที่ MATLAB Command

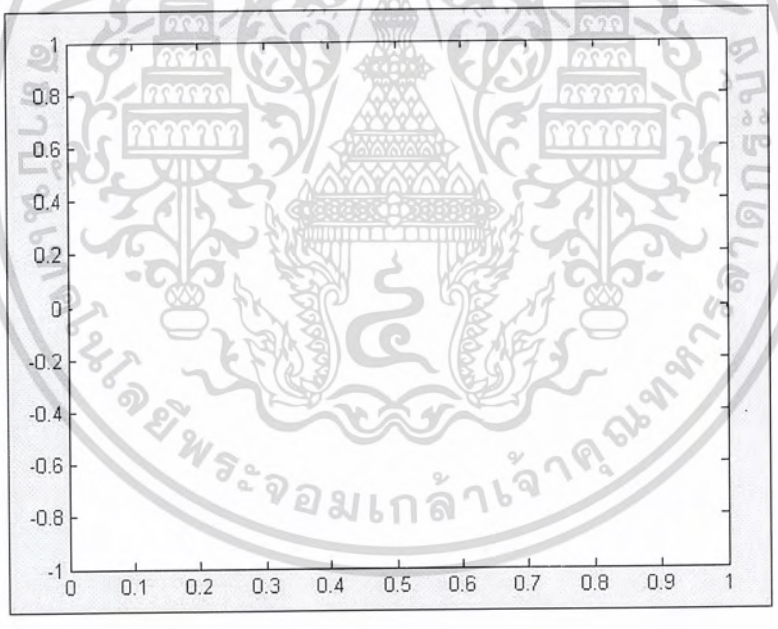
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 8.1 ค่าพารามิเตอร์ที่ใช้งานอยู่ใน MATLAB Workspace

Name	Size	Bytes	Class

Grand total is _____ elements using _____ bytes

25. พิมพ์คำสั่ง **>>plot(t, x)** ที่ MATLAB Command เพื่อเป็นการนำค่าพารามิเตอร์ที่สร้างขึ้นมาทำการพล็อตกราฟดูผลที่เกิดขึ้นก่อนที่จะส่งให้กับ Simulink บันทึกผลการทดลอง ดังรูปที่ 8.31



รูปที่ 8.31 กราฟดูผลที่เกิดขึ้นก่อนที่จะส่งให้กับ Simulink

26. พิมพ์คำสั่ง **>>clear all** อีกครั้งหนึ่งเพื่อให้ลบข้อมูลทั้งหมดที่มีอยู่ใน Workspace และพิมพ์คำสั่ง **>>whos** เพื่อดูข้อมูลใน Workspace จะสังเกตเห็นว่าจะไม่มีข้อมูลใดๆ อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

27. เปิดโมเดล **lab8_2.mdl** คลิกที่ปุ่ม ▶ สังเกตผลที่เกิดขึ้น บันทึกผลการทดลอง

.....

.....

.....

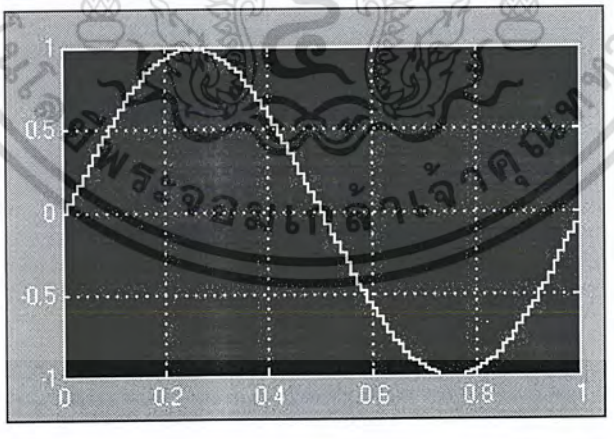
.....

.....

28. ผลที่เกิดขึ้นในขั้นตอนที่ 27 นั้นเนื่องจากโมเดล lab8_2.mdl จะหาข้อมูลที่ชื่อ simin จาก MATLAB Workspace แต่ได้มีการใช้คำสั่ง clear จึงไม่มีข้อมูลใดๆ อยู่จึงรายงานผลความผิดพลาดขึ้น ในขั้นตอนนี้จะแก้ไขปัญหาที่เกิดขึ้น โดยการกำหนดว่าเมื่อมีการเรียกใช้งานโมเดล lab8_2 นั้นให้เรียกไฟล์ input_data.m ให้ทำงาน ทำได้โดยเปิดโมเดล lab8_2 กลับมาที่หน้าต่าง MATLAB พิมพ์คำสั่ง

>>set_param(gcs, 'PreLoadFcn','input_data')

29. คลิก **Save** ในหน้าต่าง lab8_2.mdl ทดลองปิดโมเดลและเปิดขึ้นมาใหม่ทดลองคลิกที่ปุ่ม ▶ จะไม่มีการรายงานผลความผิดพลาด ดับเบิ้ลคลิกที่ Scope จะแสดงผลดังรูปที่ 8.32



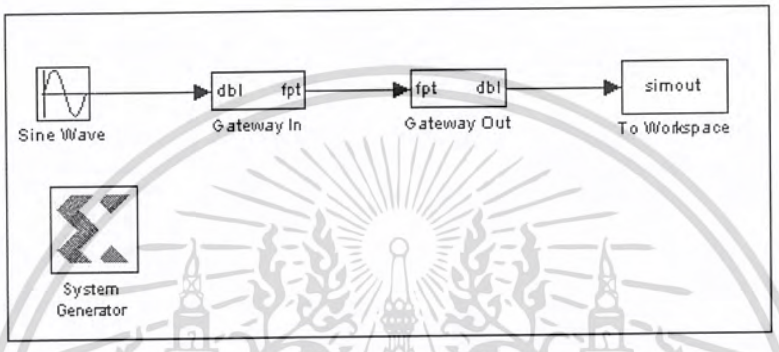
รูปที่ 8.32 ผลที่เกิดขึ้นที่ Scope ของ lab8_2.mdl

จากผลการทดลองที่ Scope จะเห็นว่าสามารถรับค่าพารามิเตอร์สัญญาณจาก MATLAB Workspace แล้วนำมาจำลองการทำงานที่หน้าต่าง Simulink ได้ แต่สัญญาณที่ได้จากหน้าจอ Scope จะมีลักษณะหยาบกว่าสัญญาณที่สร้างจากโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

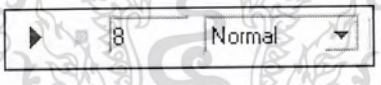
30. ขั้นตอนต่อไปนั้นจะเป็นการทดลองการส่งผลการจำลองการทำงานจาก Simulink ให้กับ MATLAB Workspace โดยทำการแก้ไขโมเดล lab8_2 โดยเปลี่ยนแปลงบล็อกต่างๆ และบันทึกเป็น **Save As** ในชื่อ **lab8_3.mdl** แสดงดังรูปที่ 8.33

- คลิกที่ไลบารี่ **Simulink** คลิกที่หัวข้อ **Source** เลือก **Sine Wave**
- คลิกที่ไลบารี่ **Simulink** คลิกที่หัวข้อ **Sink** เลือก **To Workspace**



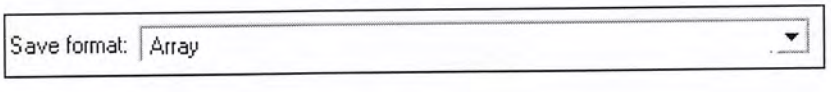
รูปที่ 8.33 โมเดล lab8_3.mdl

31. กำหนดเวลาในการจำลองที่แถบเมนูด้านบนเป็น **8** แสดงดังรูปที่ 8.34



รูปที่ 8.34 การกำหนดเวลาในการจำลองที่แถบเมนูด้านบน

32. ดับเบิ้ลคลิกที่บล็อก **To Workspace** กำหนดค่า ในช่อง **Save Format** เป็น **Array** ส่วนค่าอื่นกำหนดตามที่โปรแกรมตั้งไว้ ดังรูปที่ 8.35



รูปที่ 8.35 การกำหนดค่าของบล็อก To Workspace

33. คลิกที่ปุ่ม ▶ โมเดลจะทำงานและส่งค่าไปที่ MATLAB Workspace แล้วกลับมายังที่หน้าต่าง MATLAB ที่ Command Window พิมพ์คำสั่ง **>>whos** บันทึกผลการทดลองในตารางที่ 8.2

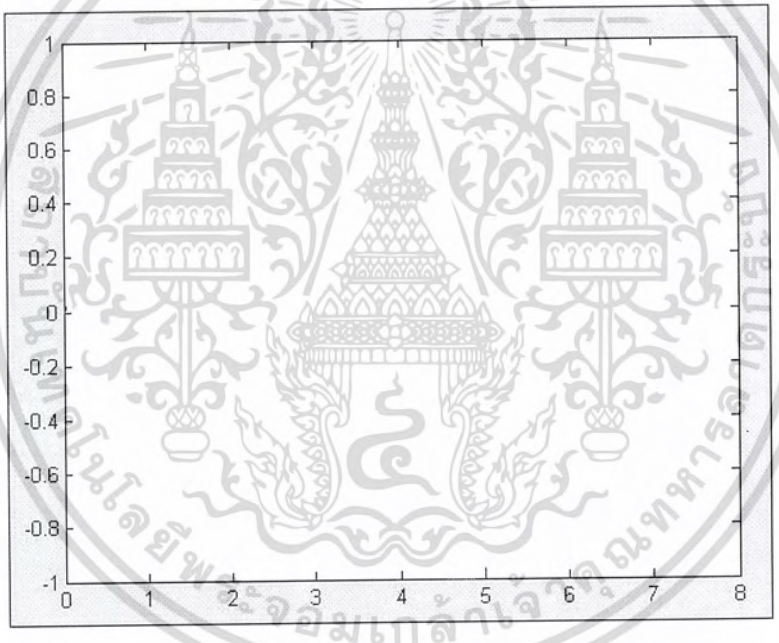
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 8.2 การกำหนดค่าของบล็อก To Workspace

Name	Size	Bytes	Class

Grand total is _____ elements using _____ bytes

34. นำค่าพารามิเตอร์ที่ได้มาทำการพล็อตกราฟ โดยพิมพ์คำสั่ง >>**plot(tout,simout)** จะแสดงกราฟที่เกิดขึ้น บันทึกผลการทดลอง ดังรูปที่ 8.36



รูปที่ 8.36 กราฟที่เกิดขึ้นของข้อมูล tout กับ Simout

สรุปผลการทดลอง

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. บล็อก Gateway In และบล็อก Gateway Out มีหน้าที่อะไร?

.....

.....

.....

.....

.....

2. การกำหนดพารามิเตอร์ต่างๆ ของบล็อก Gateway In มีผลต่อสัญญาณที่ได้อย่างไร?

.....

.....

.....

.....

.....

3. การนำเอาข้อมูลในโปรแกรม MATLAB Workspace มาเป็นสัญญาณอินพุตให้กับโมเดลมีวิธีกำหนดอย่างไร?

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การส่งค่าข้อมูลหลังจากการจำลองการทำงานใน Simulink ให้กับโปรแกรม MATLAB นั้นมีวิธีกำหนดอย่างไร?

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 9
เรื่อง การออกแบบวงจรกรองดิจิตอลโดยใช้ โปรแกรม System Generator

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. แสดงการออกแบบสัมประสิทธิ์ของวงจรกรองโดยใช้ FDATool ได้อย่างถูกต้อง
2. ออกแบบวงจรกรองดิจิตอลโดยใช้โปรแกรม System Generator ได้อย่างถูกต้อง
3. แสดงวิธีการคำนวณขนาดพื้นที่ของวงจร โดยใช้โปรแกรม System Generator ได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม MATLAB เวอร์ชั่น 7.0
2. กล่องเครื่องมือ Simulink เวอร์ชั่น 6.0
3. โปรแกรม System Generator 6.3i Evaluation

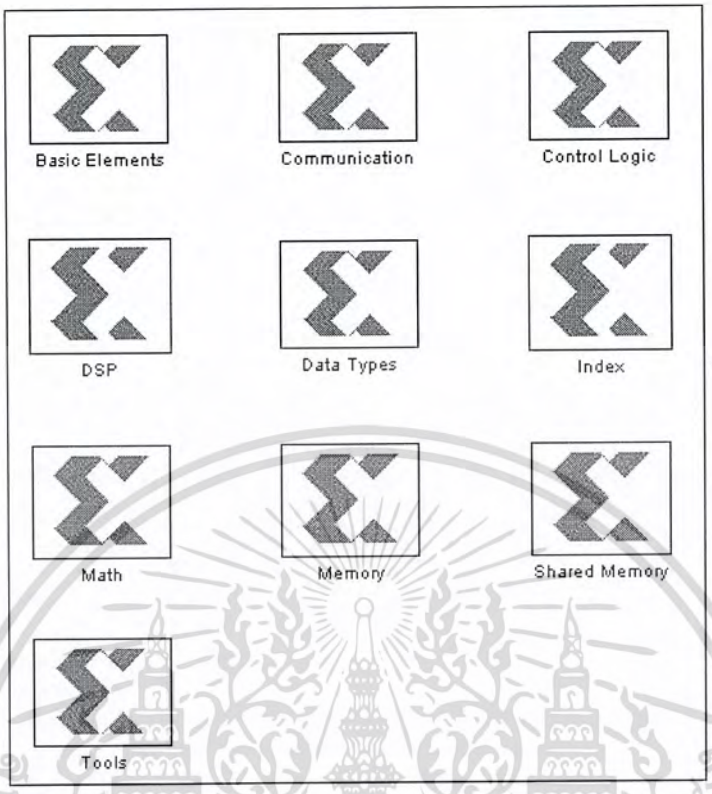
ความรู้และทักษะเดิม

1. การใช้งานโปรแกรม MATLAB และ Simulink เบื้องต้น
2. การใช้งานโปรแกรม System Generator เบื้องต้น

ทฤษฎีเบื้องต้น

การใช้งานโปรแกรม System Generator ออกแบบโมเดลร่วมกับ Simulink นั้นสามารถออกแบบระบบต่างๆ ได้อย่างง่ายและรวดเร็วเนื่องจากมีบล็อกสำเร็จรูปต่างๆ มากมาย แสดงดังรูปที่ 9.1 การใช้งานเพียงเชื่อมต่อบล็อกต่างๆ ให้เป็นโมเดลใหญ่ แล้วจำลองการทำงานของโมเดล โดยใช้บล็อกพื้นฐานของ Simulink หลังจากการจำลองการทำงานสามารถที่จะสร้างโค๊ดภาษา HDL ที่สร้างเป็นวงจรที่สามารถโปรแกรมลงบนชิพ FPGA ได้ ขั้นตอนการสังเคราะห์วงจรจากโค๊ด HDL นั้นจะต้องใช้โปรแกรม Xilinx ISE ในการสังเคราะห์วงจรและโปรแกรมลงบนชิพ FPGA

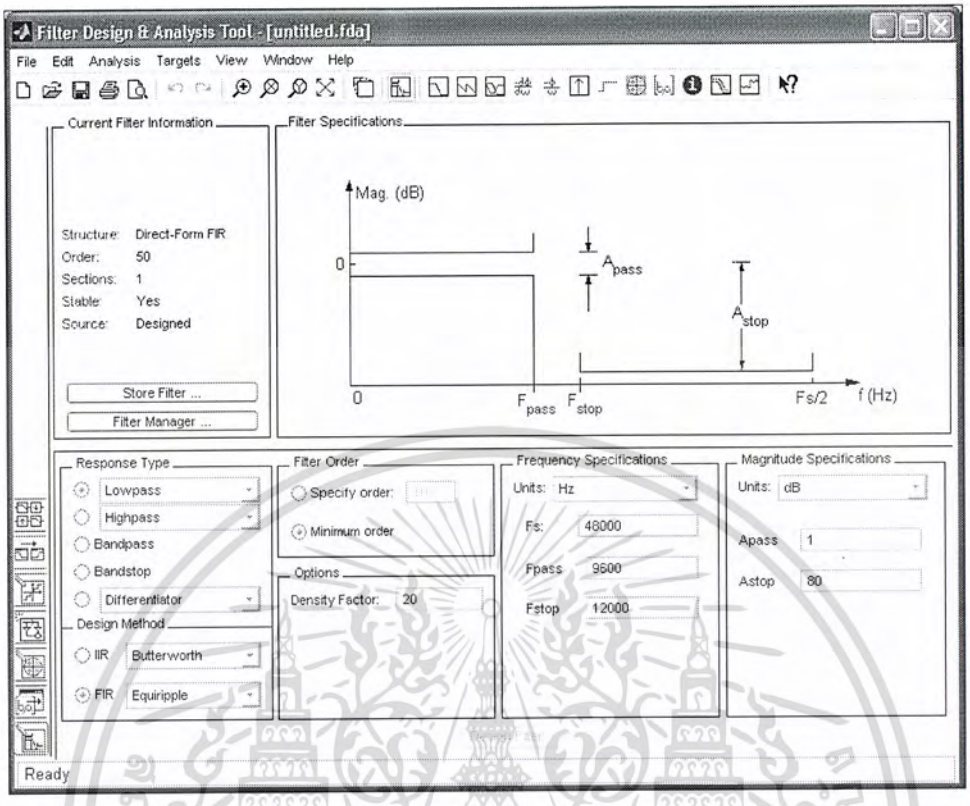
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.1 บล็อกสำเร็จรูปของ Xilinx

โมเดลหนึ่งที่นิยมนำมาใช้เป็นพื้นฐานในการศึกษาการออกแบบและโปรแกรมลงบนชิพ FPGA คือ วงจรกรองความถี่ดิจิตอล (Digital Filter) ซึ่งโปรแกรม System Generator จะมีบล็อกของวงจรกรองสำเร็จรูปคือ CIC และ FIR เพียงแต่ผู้ใช้จะต้องออกแบบสัมประสิทธิ์ (Coefficients) ของวงจรกรองเอง ในโปรแกรม MATLAB เวอร์ชั่น 7.0 จะมีกล่องเครื่องมือที่ใช้ในการออกแบบชื่อว่า FDATool (Filter Design & Analysis Tool) จะมีลักษณะเป็น GUI (Graphic Users Interface) ซึ่งทำให้การออกแบบง่ายขึ้น จากเดิมที่ผู้ออกแบบต้องเขียนโปรแกรม M-File ที่สร้างสัมประสิทธิ์ของวงจรกรองขึ้นมา หน้าต่าง FDATool แสดงดังรูปที่ 9.2 และวิธีการออกแบบแสดงดังตารางที่ 9.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



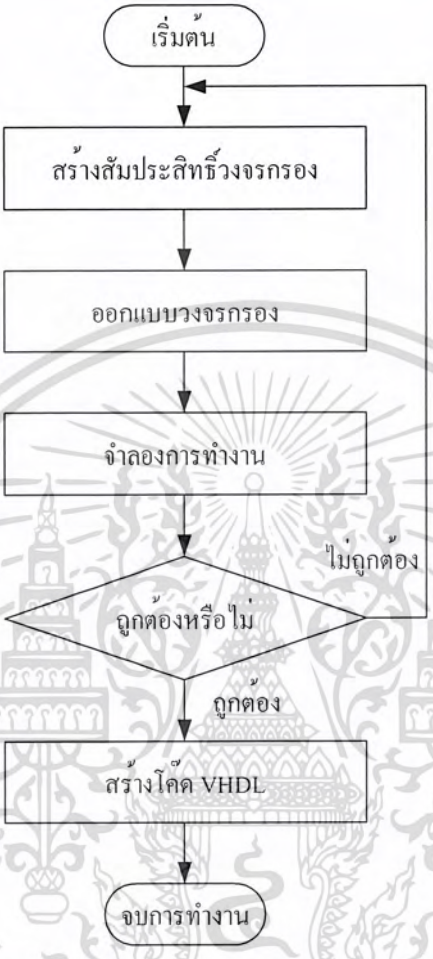
รูปที่ 9.2 หน้าต่างกล่องเครื่องมือ FDATool

ตารางที่ 9.1 วิธีการออกแบบวงจรกรองที่ FDATool สามารถออกแบบได้

Butterworth	Chebyshev Type I&II	Tukey (tapered cosine) window
Elliptic	Maximally Flat	Equiripple
Least squares	Constrained least squares	Complex equiripple
Bartlett window	Bartlett-Hanning window	Blackman window
Chebyshev window	Bohman window	Blackman-Harris window
Flat top window	Gaussian window	Hamming window
Hann window	Kaiser window	Nuttall's Blackman-Harris window
Triangular window	Rectangular window	Parzen or de la Vallé-Poussin window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออกแบบวงจรกรองด้วยโปรแกรม System Generator แสดงดังรูปที่ 9.3



รูปที่ 9.3 ผังงานการออกแบบวงจรกรองความถี่ด้วยโปรแกรม System Generator

1. สร้างสัมประสิทธิ์วงจรกรอง (Coefficients Design)

เป็นการออกแบบวงจรกรองเพื่อสร้างสัมประสิทธิ์ โดยจะออกแบบโดยใช้กล่องเครื่องมือ FDATool ซึ่งจะต้องกำหนดค่าพารามิเตอร์ต่างๆ ของวงจรกรอง เช่น แถบความถี่ที่ต้องการให้ผ่านหรือหยุด วิธีการออกแบบวงจรกรอง หรือจำนวน Order ที่ต้องการ

2. ออกแบบวงจรกรองความถี่ (Digital Filter Design)

เป็นการออกแบบโมเดลของวงจรกรองจากหน้าต่าง Simulink โดยเป็นการนำบล็อก FIR ของ Xilinx มาเชื่อมต่อกับบล็อกพื้นฐานของ Simulink ซึ่งกำหนดให้นำค่าสัมประสิทธิ์ที่ได้ออกแบบไว้มาใช้ในโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การจำลองการทำงานของโมเดล (Simulation)

เป็นการสร้างบล็อกที่เป็นสัญญาณอินพุตและเครื่องมือวัดสัญญาณ มาทดสอบการทำงานของโมเดล โดยบล็อกสัญญาณอินพุตเช่น Random Signal และบล็อกเครื่องมือวัดสัญญาณ เช่น Spectrum Scope เป็นต้น

4. การสร้างโค๊ด HDL (HDL Generator)

เมื่อผลการจำลองการทำงานถูกต้องตามที่ออกแบบไว้ จะสามารถสร้างโค๊ด HDL ได้ ซึ่งต้องกำหนดค่าพารามิเตอร์ต่างๆ ที่สำคัญเช่น ไฟล์ HDL ที่ต้องการ และเลือกชิพ FPGA ที่ใช้งาน เป็นต้น

ลำดับขั้นการทดลอง

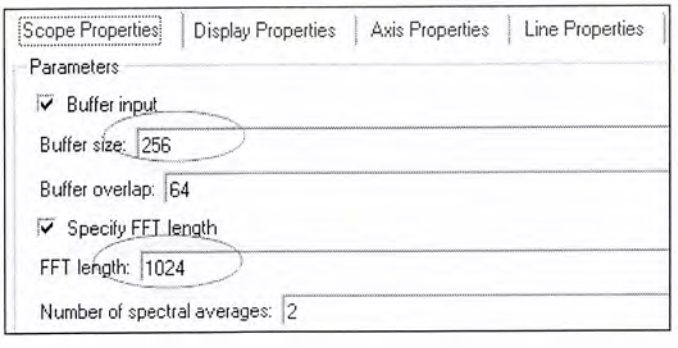
การทดลองนี้จะเป็นการออกแบบวงจรกรองแถบความถี่ผ่าน ชนิด FIR โดยจะมีกำหนดให้ช่วงความถี่ที่ต้องการอยู่ระหว่าง **300KHz** ถึง **450KHz** กำหนดความถี่ในการสุ่มตัวอย่างเป็น **1.5MHz** เมื่อขั้นตอนการออกแบบเสร็จสิ้น จะทำการสร้างโค๊ดภาษา VHDL เพื่อสังเคราะห์วงจรต่อไป

1. เปิดโปรแกรม MATLAB เวอร์ชั่น7.0 เลือกไดเร็กทอรี่ที่ใช้งานเป็น **C:\MATLAB7\LAB9**

2. พิมพ์คำสั่ง >>**Simulink** ที่ Command Windows แล้วสร้างโมเดลใหม่มีรายละเอียดดังนี้

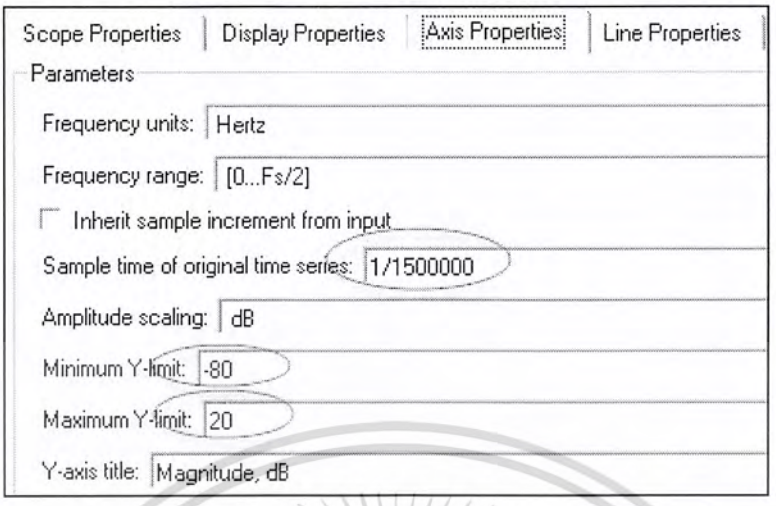
- คลิกที่ไลบารี่ **Simulink** คลิกที่หัวข้อ **Signal Routing** เลือก **Manual Switch** และ **Mux**
- คลิกที่ไลบารี่ **Signal Processing** คลิกที่หัวข้อ **DSP Sinks** เลือก **Spectrum Scope**

กำหนดค่าพารามิเตอร์แสดงดังรูปที่ 9.4 รูปที่ 9.5 และรูปที่ 9.6

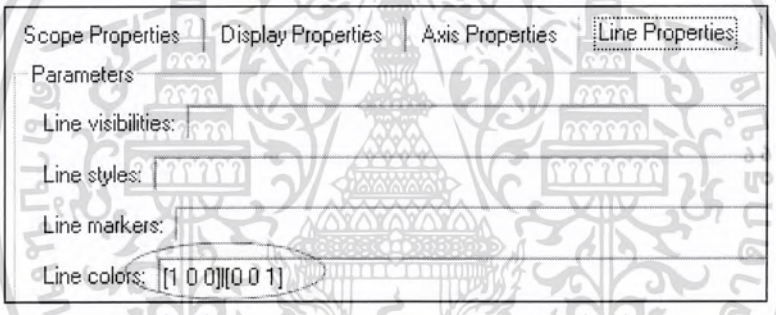


รูปที่ 9.4 การกำหนดค่าในบล็อก Spectrum Scope ในหัวข้อ Scope properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.5 การกำหนดค่าในบล็อก Spectrum Scope ในหัวข้อ Axis Properties

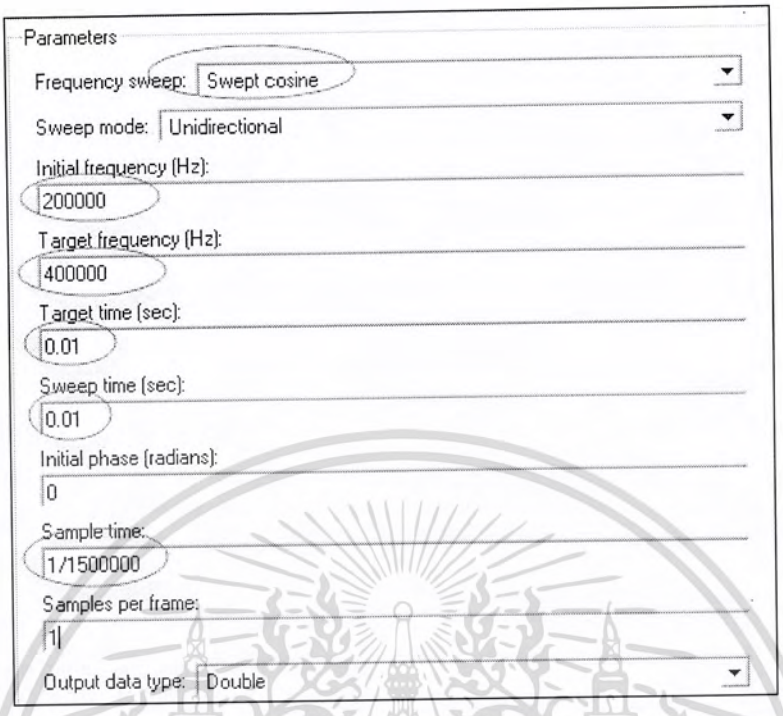


รูปที่ 9.6 การกำหนดค่าในบล็อก Spectrum Scope ในหัวข้อ Line Properties

Line colors จะเป็นการกำหนดสีของกราฟ รูปแบบใช้งานในกรณีที่มี 2 CH คือ [R G B]| - [R G B]

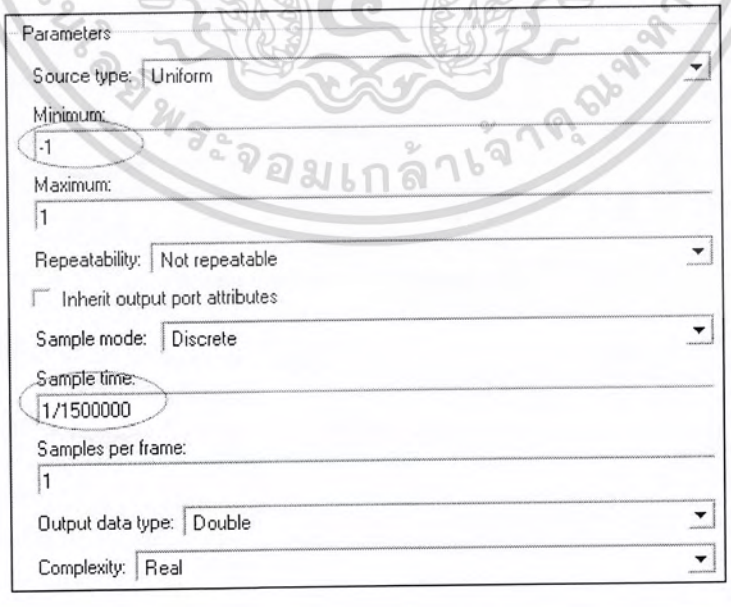
- คลิกที่ไลบารี่ **Signal Processing** คลิกที่หัวข้อ **DSP Source** เลือก **Chirp** และ **Random Source**
- กำหนดค่าพารามิเตอร์ของบล็อก Chirp .ให้สร้างสัญญาณที่มีความถี่ตั้งแต่ 200 kHz – 400 kHz ความถี่ในการสุ่มสัญญาณเป็น 1.5 MHz กำหนดค่าอื่นๆ แสดงดังรูปที่ 9.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.7 การกำหนดค่าของบล็อก Chirp

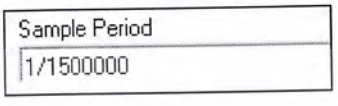
- กำหนดค่าพารามิเตอร์ของบล็อก Random Source ให้สร้างสัญญาณทุกๆ ความถี่ที่มีระดับของสัญญาณอยู่ระหว่าง -1 ถึง 1 V กำหนดค่าอื่นๆ แสดงดังรูปที่ 9.8



รูปที่ 9.8 การกำหนดค่าของบล็อก Random Source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

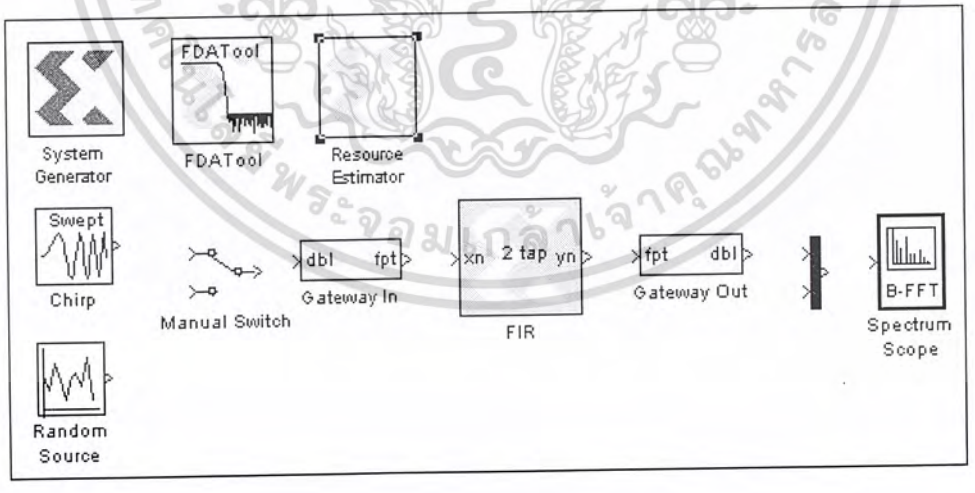
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **System Generator**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **Gateway In** แสดงดังรูปที่ 9.9



รูปที่ 9.9 การกำหนดค่า Sample Period

- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **Gateway Out**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **DSP** เลือก **FIR**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **DSP** เลือก **Resource Estimator**
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Tool** เลือก **FDATool**

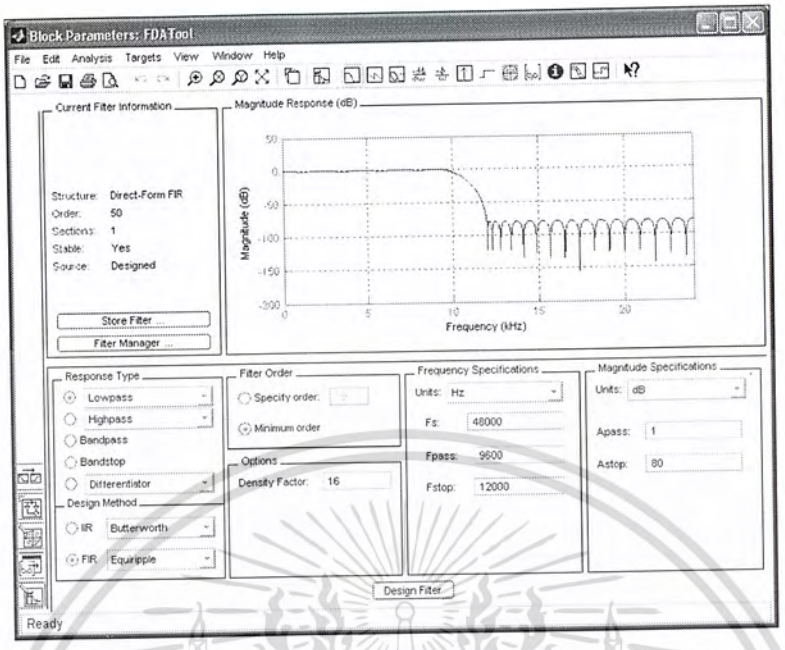
แสดงการเลือกบล็อกทั้งหมดดังรูปที่ **บันทึกเป็น lab9.mdl** แสดงดังรูปที่ 9.10



รูปที่ 9.10 บล็อกต่างๆ ของโมเดล lab9.mdl

3. ดับเบิ้ลคลิกที่บล็อก FDATool จะแสดงหน้าต่าง FDA Tool เป็นการออกแบบวงจรกรอง แสดงดังรูปที่ 9.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



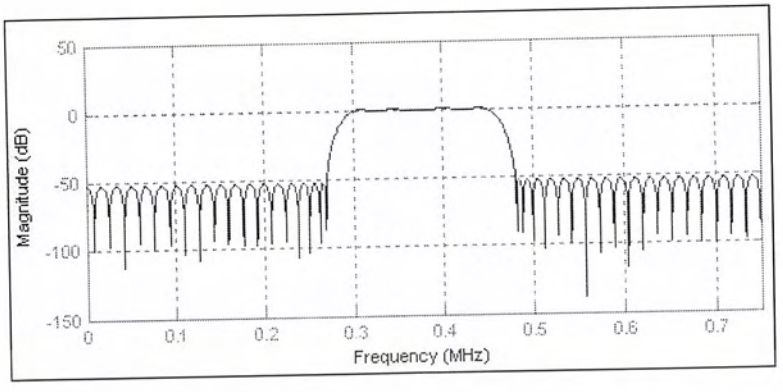
รูปที่ 9.11 หน้าต่าง FDA Tool

4. ออกแบบวงจรกรองโดยกำหนดค่าพารามิเตอร์ในหน้าต่าง FDA Tool ดังนี้

- Response Type : **Bandpass**
- Design Method : **FIR Equiripple**
- Units: **kHz**
- Sampling Frequency (Fs) = **1500 kHz**
- Fstop 1 = **270 kHz**
- Fpass 1 = **300 kHz**
- Fpass 2 = **450 kHz**
- Fstop 2 = **480 kHz**
- Attenuation on both sides of the passband = **54 dB** (Astop1 and Astop2-parameters)
- Pass band ripple = **1** (Apass)

5. เมื่อกำหนดค่าต่างๆ เสร็จสิ้นแล้วคลิกที่ปุ่ม Design Filter จะแสดงผลตอบสนองของระบบดังรูปที่ 9.12 สังเกตข้อมูลในหัวข้อ Current Filter Information บันทึกผลการทดลองที่ได้ในตารางที่ 9.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

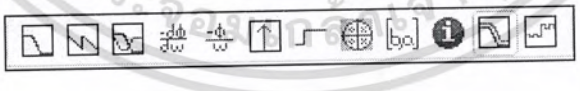


รูปที่ 9.12 ผลตอบสนองของระบบ

ตารางที่ 9.2 ข้อมูลของวงจรกรองความถี่ที่ออกแบบ

Structure	
Orders	
Sections	
Stable	
Source	

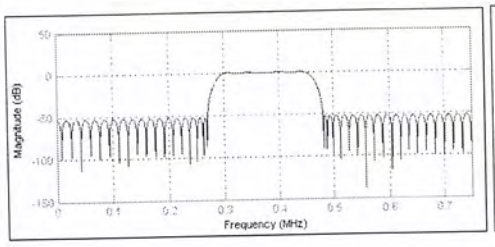
6. ทดลองคลิกที่ปุ่มเมนูบาร์ด้านบนเพื่อแสดงค่าพารามิเตอร์ของวงจรกรองที่ได้ออกแบบ เมนูบาร์แสดงดังรูปที่ 9.13



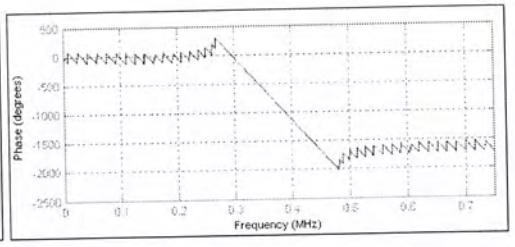
รูปที่ 9.13 เมนูแสดงค่าพารามิเตอร์ต่างๆ

ค่าพารามิเตอร์ที่สำคัญของวงจรกรองสามารถแสดงได้ดังรูปที่ 9.14

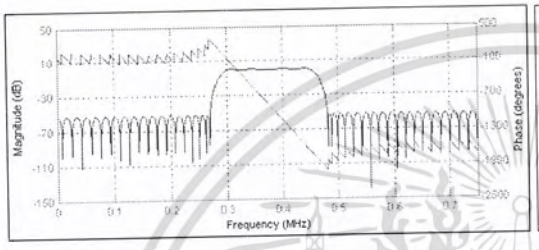
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



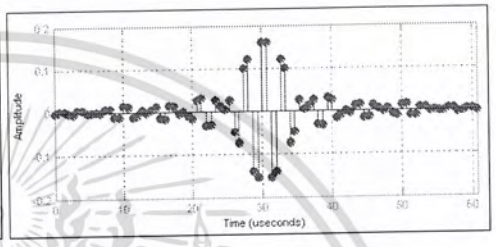
(ก) Magnitude Response (dB)



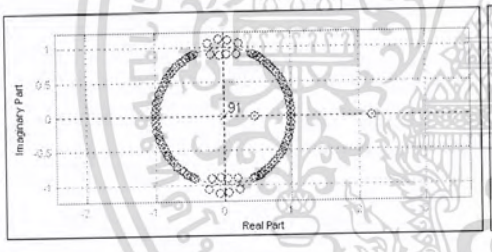
(ข) Phase Response



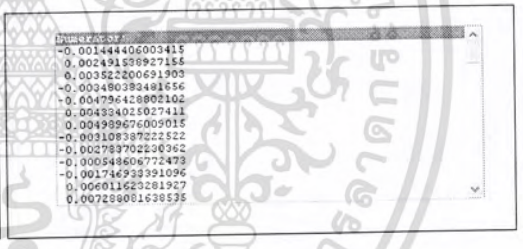
(ค) Magnitude (dB) & Phase Responses



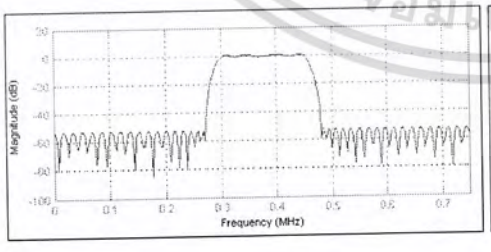
(ง) Impulse Response



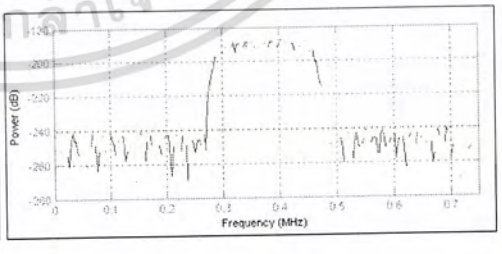
(จ) Pole/Zero Plot



(ฉ) Filter Coefficients



(ช) Magnitude Response Estimate

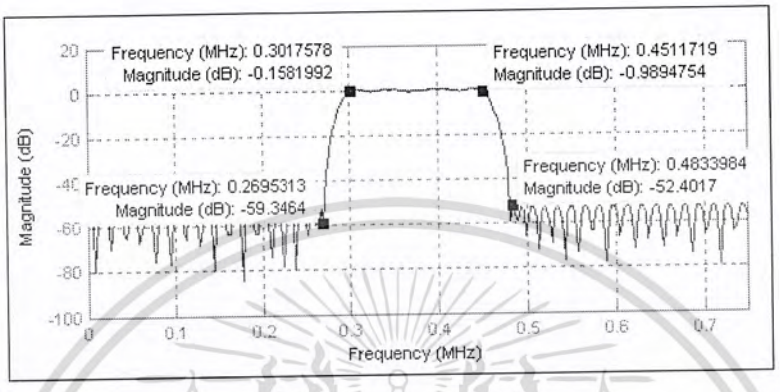


(ซ) Round-off Noise Power Spectrum

รูปที่ 9.14 ค่าพารามิเตอร์ต่างๆ ของวงจรกรองที่ออกแบบ

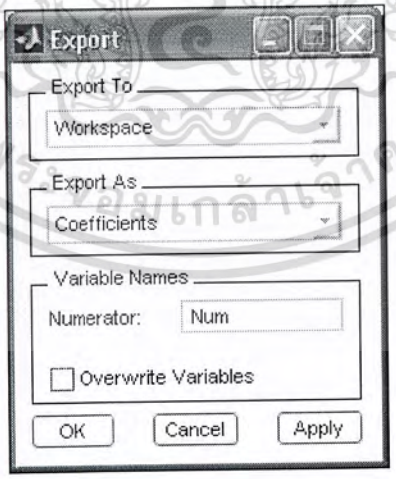
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. จากค่าพารามิเตอร์ในหัวข้อ Magnitude Response Estimate ลองทดสอบคลิกที่จุด Pass Band และ Stop Band ดังรูปที่ 9.15 จะเห็นว่าที่แถบผ่านจะอยู่ระหว่าง 300 kHz ถึง 450 kHz แสดงว่าถูกต้อง



รูปที่ 9.15 การทดสอบแถบความถี่ผ่านของวงจรกรอง

8. ขั้นตอนต่อไปจะแปลงค่าสัมประสิทธิ์ของวงจรกรองเป็นค่าตัวแปรของโปรแกรม MATLAB โดยคลิกที่ **File > Export** จะแสดงหน้าต่างดังรูปที่ 9.16 สังเกตว่าค่าสัมประสิทธิ์จะถูกกำหนดชื่อเป็น **Num** คลิก **Ok**



รูปที่ 9.16 หน้าต่างการส่งค่าสัมประสิทธิ์ไปที่ MATLAB Workspace

9. บันทึกวงจรกรองที่ได้ออกแบบไว้ คลิกที่ **File > Save Session** ตั้งชื่อเป็น **lab9.fda** แล้วปิดหน้าต่าง FDATool

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. กลับมาที่หน้าต่างโปรแกรม MATLAB พิมพ์คำสั่ง >> **whos** บันทึกผลการทดลองในตารางที่ 9.3

ตารางที่ 9.3 คุณสมบัติของตัวแปร Num

Name	Size	Byte	Class

Grand total is _____ elements using _____ bytes

11.พิมพ์คำสั่ง >> **max(Num)** บันทึกผลการทดลองและพิมพ์คำสั่ง >> **min(Num)** บันทึกผลการทดลองในตารางที่ 9.4

ตารางที่ 9.4 ค่าสูงสุดและค่าต่ำสุดของตัวแปร Num

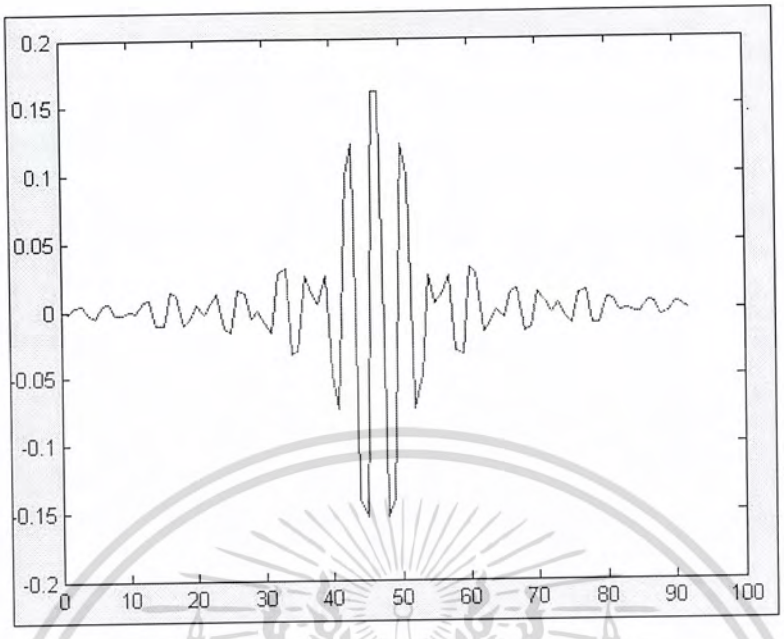
Max Num =	
Min Num =	

12. พิมพ์คำสั่ง >> **plot(Num)** จะแสดงผลของค่า Num เมื่อนำมาพล็อตกราฟ แสดงดังรูปที่ 9.17 โดยลักษณะของกราฟที่ได้จะเป็นรูป Sinc Pulse แสดงว่าถูกต้อง

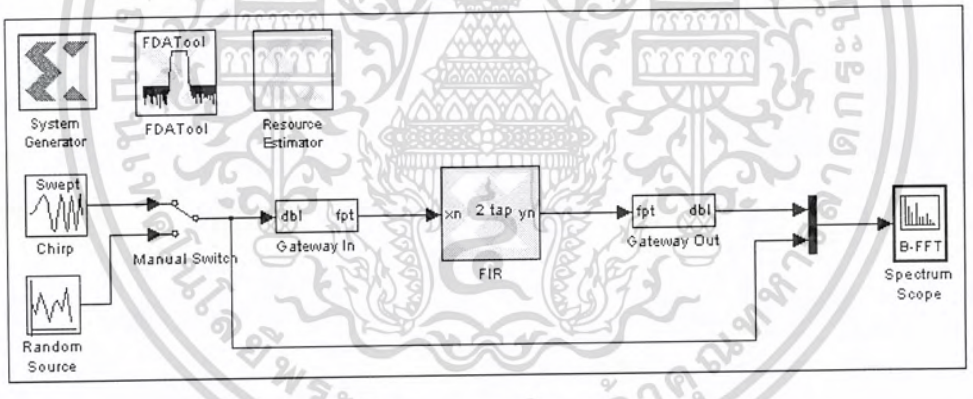
13. กลับมาที่หน้าต่าง **lab9.mdl** ลากเส้นเชื่อมต่อบล็อกทั้งหมดแสดงดังรูปที่ 9.18 คลิกปุ่ม **Save**

14. ดับเบิ้ลคลิกที่บล็อก FIR กำหนดค่าในช่อง Coefficients เป็น **xlfda_numerator ('FDAtool')** เพื่อให้เรียกสัมประสิทธิ์วงจรกรองจาก FDATool และกำหนดค่าอื่นๆ แสดงดังรูปที่ 9.19 คลิก **OK** และบันทึก **lab9.mdl** โดยคลิกที่ปุ่ม **Save**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.17 ผลการพล็อตกราฟของค่าสัมประสิทธิ์ที่ได้ออกแบบไว้



รูปที่ 9.18 การลากเส้นเชื่อมต่อบล็อกทั้งหมดของ lab9.mdl

15. สังเกตการเปลี่ยนแปลงที่บล็อก FIR บันทึกผลการทดลอง

.....

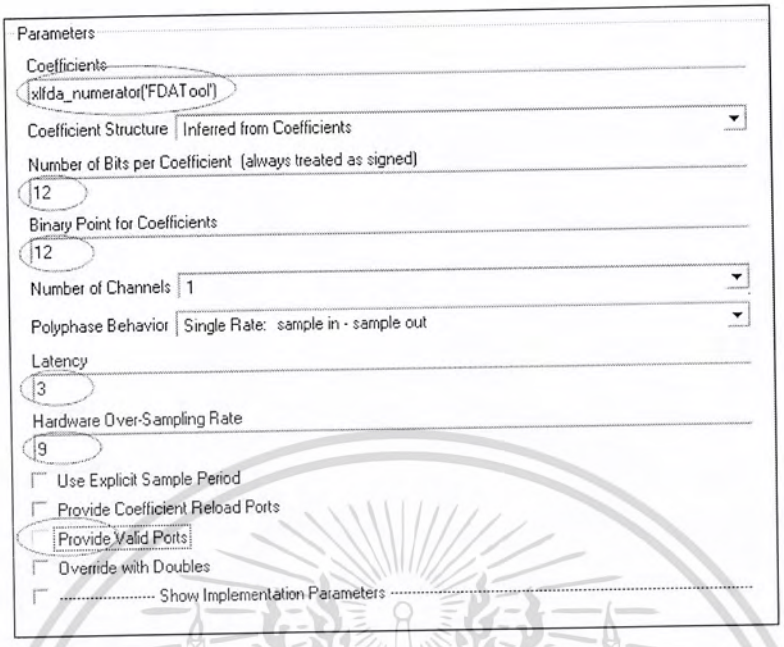
.....

.....

.....

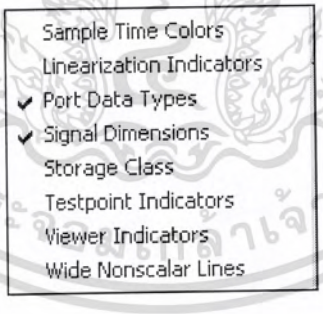
.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.19 การกำหนดค่าของบล็อก FIR

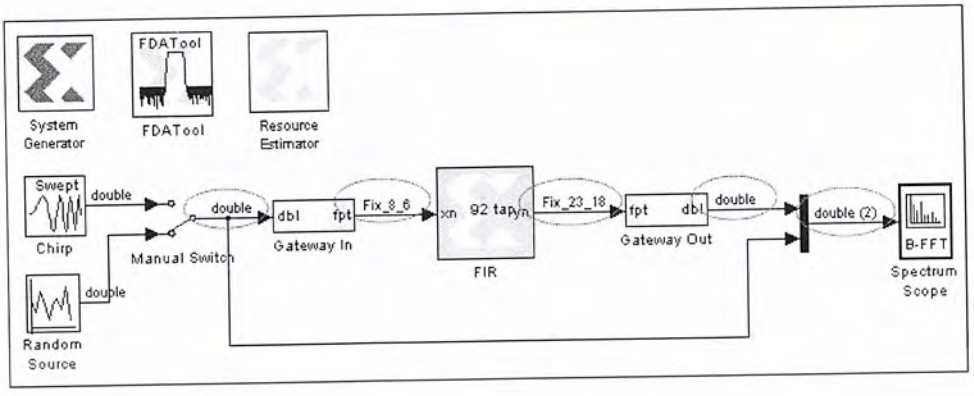
16. คลิกที่ **Format > Port/Signal Display** เป็นการกำหนดลักษณะการแสดงผลของโมเดล กำหนดค่าดังรูปที่ 9.20



รูปที่ 9.20 การกำหนดรูปแบบการแสดงผลของโมเดล

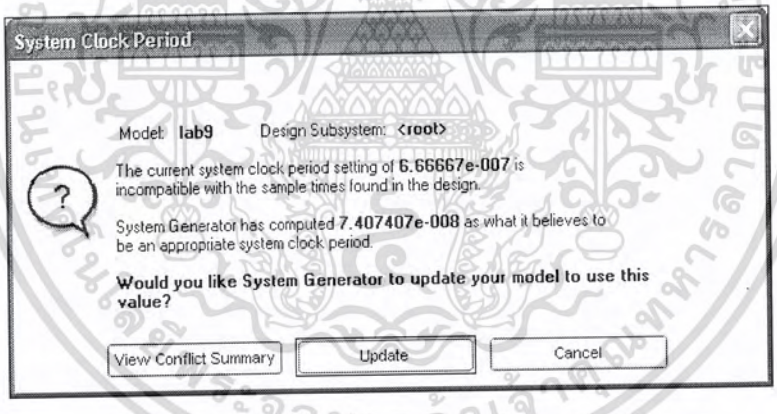
17. **คลิกขวา**บริเวณพื้นที่โมเดล เลือก **Update Diagram** โมเดลจะเกิดการเปลี่ยนแปลงคือ จะแสดงลักษณะข้อมูลเข้าและออกบล็อกทั้งหมดแสดงดังรูปที่ 9.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.21 โมเดลที่แสดงลักษณะข้อมูลเข้าและออกบล็อกทั้งหมด

18. คลิกที่ปุ่ม เพื่อจำลองการทำงาน จะปรากฏหน้าต่าง System Clock Period เป็นการบอกว่าได้มีการเปลี่ยนเวลาการทำงานของโมเดลเป็น 1/1500000 ให้คลิกที่ **Update** แสดงดังรูปที่ 9.22

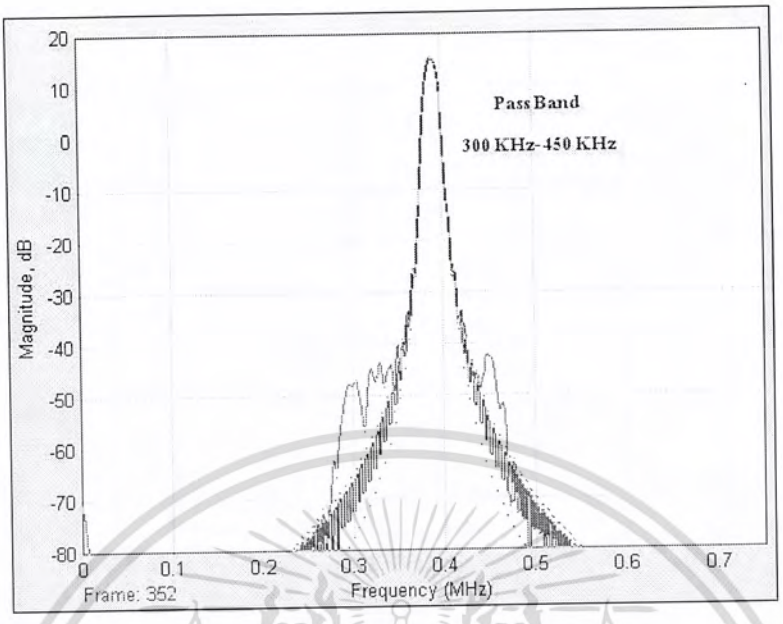


รูปที่ 9.22 หน้าต่าง System Clock Period

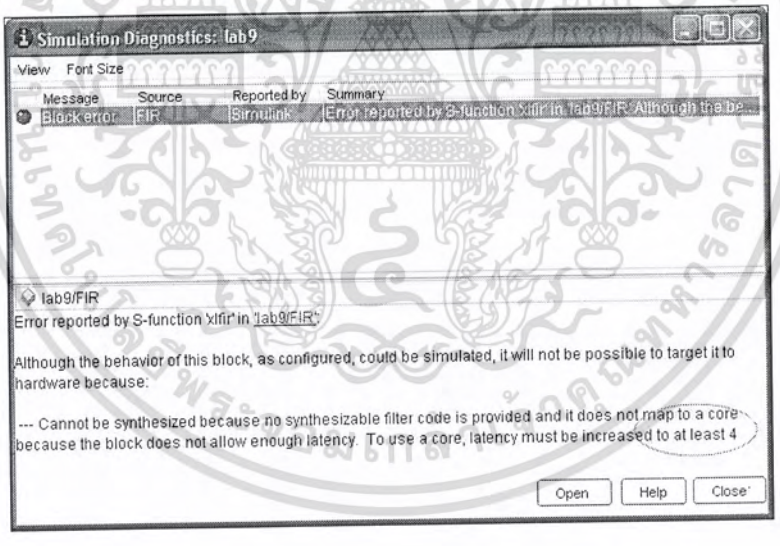
19. สังเกตการเปลี่ยนแปลงของ Spectrum Scope แสดงดังรูปที่ 9.23

20. คลิกที่ปุ่ม เพื่อหยุดการจำลองการทำงาน จะปรากฏหน้าต่าง **ERROR** เป็นการบอกว่าโมเดลนี้ไม่สามารถสร้างโค๊ดได้ ซึ่งต้องกำหนดค่า Latency อย่างน้อย = 4 จึงสร้างสามารถสร้างโค๊ดได้ จะกำหนดค่าในขั้นตอนอื่นๆ ให้คลิกที่ปุ่ม **Close** แสดงดังรูปที่ 9.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



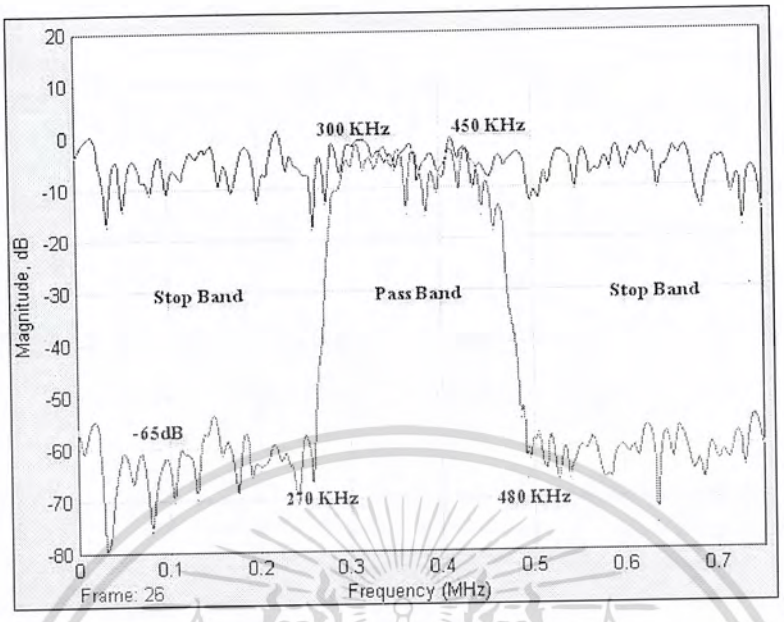
รูปที่ 9.23 หน้าต่าง Spectrum เมื่อกำหนดสัญญาณอินพุตเป็น Chirp



รูปที่ 9.24 หน้าต่างการบอกว่าโมเดลไม่สามารถสร้างโค๊ดได้

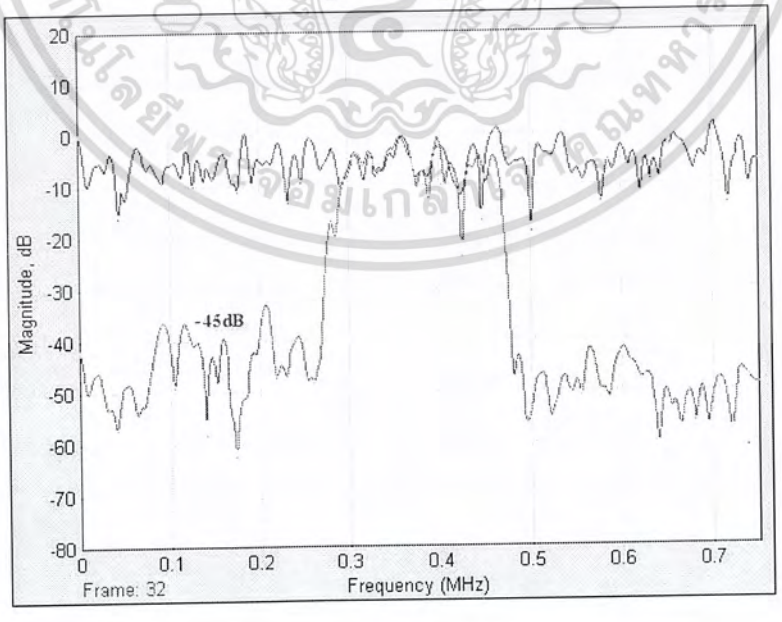
21. ดับเบิ้ลคลิกที่ **Manual Switch** เพื่อเลือกสัญญาณอินพุตเป็น **Random Source** คลิกที่ปุ่ม ▶ อีกครั้ง จะแสดงหน้าต่าง Spectrum Scope ดังรูปที่ 9.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.25 หน้าต่าง Spectrum เมื่อกำหนดสัญญาณอินพุตเป็น Random Source

22. ทดลองเปลี่ยนจำนวนบิตของสัมประสิทธิ์ของบล็อก FIR เป็น **Number of bit** เป็น **8** และ **Binary Point** เป็น **8** คลิกที่ปุ่ม ▶ อีกครั้งสังเกตผลที่เกิดขึ้นที่หน้าจอ Spectrum Scope แสดงดังรูปที่ 9.26



รูปที่ 9.26 หน้าต่าง Spectrum เมื่อกำหนดจำนวนบิตของสัมประสิทธิ์เป็น 8_8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

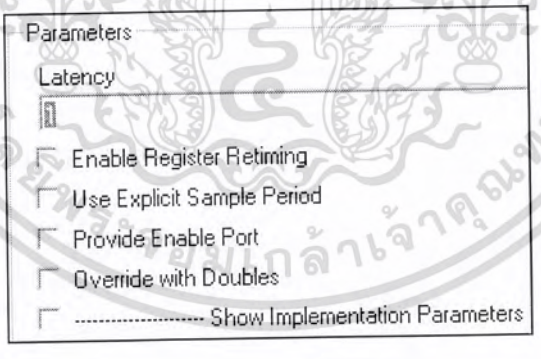
จากผลที่เกิดขึ้นที่หน้าต่าง Spectrum Scope เมื่อลดจำนวนบิตข้อมูลของสัมประสิทธิ์วงจรกรองเป็น 8_8 บิต จะเห็นว่าช่วง Stop Band ที่ต้องการให้มีระดับของสัญญาณต่ำที่สุดนั้น จะเพิ่มขึ้นจาก 65dB เป็น -45 dB

23. ขั้นตอนต่อไปจะเป็นการกำหนดค่าพารามิเตอร์เพิ่มเติม เพื่อให้สามารถสร้างโค๊ดจากโมเดลได้ดับเบิ้ลคลิกที่บล็อก FIR กำหนดค่าดังนี้

- **Number of bit = 12**
- **Binary Point = 12**
- **Latency = 4**
- **Hardware Over - Sampling Rate = 9**

24. ขั้นตอนต่อไปจะทำการเพิ่มเติมบล็อกต่างๆ ให้วงจรมีประสิทธิภาพมากขึ้นเมื่อสร้างเป็นวงจรทางฮาร์ดแวร์ โดยต้องทำให้จำนวนบิตของเอาต์พุตเท่ากับอินพุต และเพิ่มส่วนที่หน่วงสัญญาณให้วงจรทำงานอย่างต่อเนื่อง โดยคลิกที่ปุ่ม เพื่อเปิดหน้าต่าง Simulink Library Browser เพิ่มบล็อกต่างๆ มีรายละเอียดดังนี้

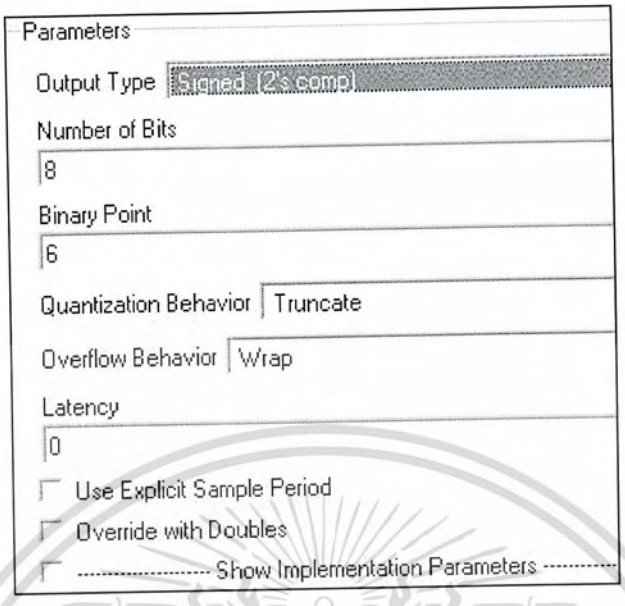
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Basic Elements** เลือก **Delay** กำหนดค่าต่างๆ แสดงดังรูปที่ 9.27



รูปที่ 9.27 การกำหนดค่าของบล็อก Delay

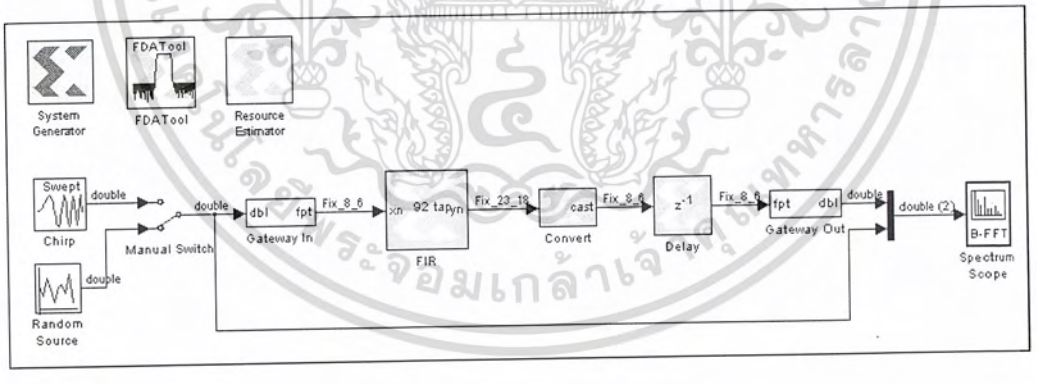
- คลิกที่ไลบารี่ **Xilinx Blockset** คลิกที่หัวข้อ **Math** เลือก **Convert** กำหนดต่างๆ แสดงดังรูปที่ 9.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.28 การกำหนดค่าของบล็อก Convert

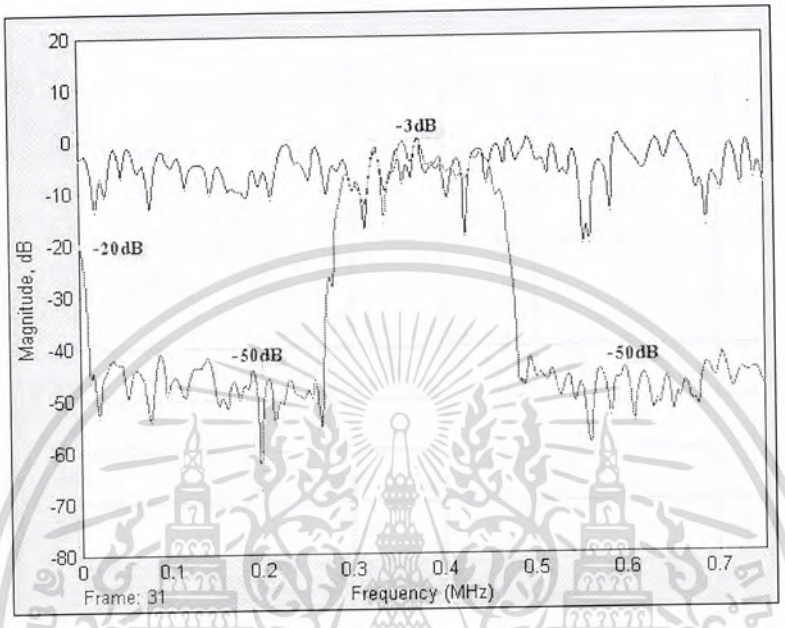
25. ลากเส้นเชื่อมต่อกับบล็อกทั้งหมด คลิกขวาที่พื้นที่โมเดล เลือก **Update Diagram** และคลิกที่ปุ่ม **Save** แสดงโมเดลที่สมบูรณ์ดังรูปที่ 9.29



รูปที่ 9.29 การลากเส้นเชื่อมต่อของโมเดล lab9.mdl ที่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

26. เลือกสัญญาณอินพุตเป็น **Random Source** แล้วคลิกที่ปุ่ม ▶ ซึ่งผลการจำลองการทำงานที่ Spectrum Scope แสดงดังรูปที่ 9.30

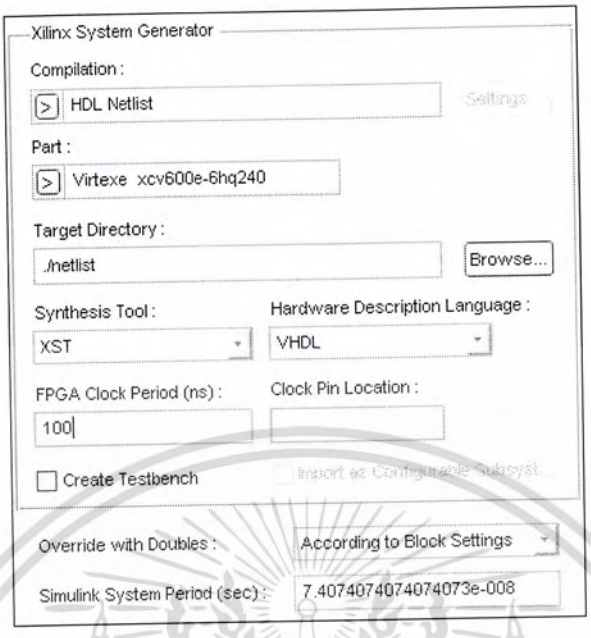


รูปที่ 9.30 ผลการจำลองการทำงานที่ Spectrum Scope ของโมเดล lab9.mdl ที่สมบูรณ์

จากผลการจำลองการทำงานจะเห็นว่าคุณภาพของวงจรกรองจะลดต่ำลง สังเกตได้จากระดับของสัญญาณในช่วง Stop Band จะมีค่าเพิ่มขึ้น เนื่องได้มีการลดจำนวนบิตของข้อมูลทางเอาต์พุตให้มีค่าเท่ากับเอาต์พุต เพราะระบบประมวลผลของฮาร์ดแวร์จะได้มีจำนวนบิตที่เท่ากันและง่ายต่อการออกแบบอุปกรณ์ทางด้านฮาร์ดแวร์เพิ่มเติม และในช่วงเวลาที่วงจรกรองเริ่มทำงานจะมีระดับสัญญาณช่วง Stop Band มีค่าสูงสุด ที่ -20dB และจะลดต่ำลงเรื่อยๆ จนถึงค่าคงที่ค่าหนึ่ง เป็นผลมาจากที่ได้มีการเพิ่มบล็อก Delay ดังนั้นจึงทำให้ช่วงแรกของการทำงานจึงเกิดความผิดพลาดขึ้น และระดับสัญญาณจะคงที่ในช่วงเวลาถัดไป

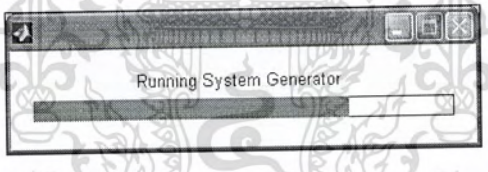
27. ดับเบิ้ลคลิกที่บล็อก **System Generator** เลือกชิพ FPGA เป็น **VirtexE > xcv600E > -6 > hq240** และกำหนดค่าอื่นๆ ดังรูปที่ 9.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



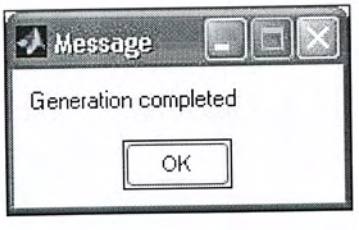
รูปที่ 9.31 การกำหนดค่าของบล็อก System Generator

28. คลิกที่ปุ่ม **Generate** เพื่อสร้างโค๊ด VHDL จะแสดงขั้นตอนการสร้างโค๊ดดังรูปที่ 9.32



รูปที่ 9.32 ขั้นตอนการสร้างโค๊ด VHDL

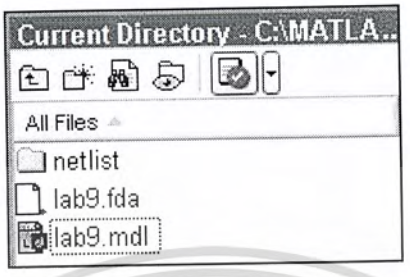
เมื่อขั้นตอนการสร้างโค๊ดเสร็จสิ้นจะแสดงดังรูปที่ 9.33



รูปที่ 9.33 หน้าต่างเมื่อขั้นตอนการสร้างโค๊ดเสร็จสิ้น

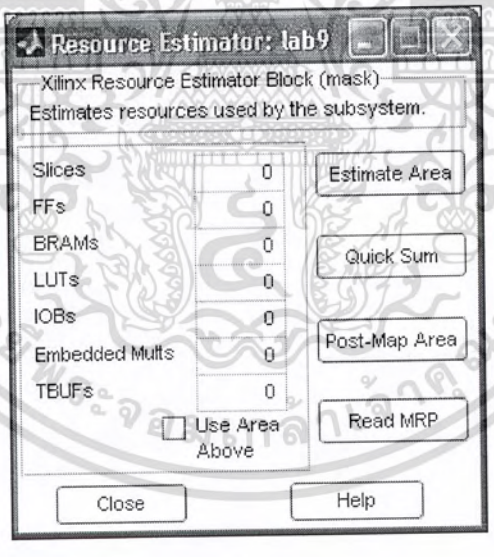
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

29. กลับมาที่หน้าต่าง Current Directory จะแสดงโฟลเดอร์ **Netlist** เพิ่มขึ้นมาแสดงดังรูปที่ 9.34



รูปที่ 9.34 หน้าต่าง Current Directory แสดงโฟลเดอร์ Netlist

30. กลับมายังที่หน้าต่าง lab9.mdl ดับเบิ้ลคลิกที่บล็อก **Resource Estimator** เพื่อคำนวณขนาดพื้นที่ใช้งานของวงจร แสดงดังรูปที่ 9.35



รูปที่ 9.35 หน้าต่าง Resource Estimator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

31. คลิกที่ปุ่ม **Post-Map Area** จะแสดงขั้นตอนการทำงานดังรูปที่ 9.36

รูปที่ 9.36 ขั้นตอนการคำนวณค่าพารามิเตอร์ในการโปรแกรมลงบนชิพ FPGA

32. เมื่อเสร็จสิ้นจะปรากฏหน้าต่าง Resource Estimator อีกครั้ง บันทึกผลการทดลองในตารางที่ 9.5

ตารางที่ 9.5 รายงานผลขนาดพื้นที่ของวงจร

Slices	
FFs	
BRAMs	
LUTs	
IOBs	
Embedded Mults	
TBUFs	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1.วงจรกรองที่ออกแบบมีจำนวน Order สูงสุดเท่ากับเท่าใด?

.....

.....

2. จำนวนบิตที่เก็บค่าข้อมูลของสัมประสิทธิ์ นั้นมีผลต่อคุณภาพของวงจรกรองหรือไม่ จงอธิบาย?

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 10
เรื่อง การใช้งานโปรแกรม Xilinx ISE Foundation ในการสังเคราะห์วงจร

จุดประสงค์เชิงพฤติกรรม

เพื่อให้นักศึกษาสามารถ

1. แสดงขั้นตอนการสังเคราะห์วงจรได้อย่างถูกต้อง
2. แสดงขั้นตอนการตรวจสอบขนาดพื้นที่ที่ใช้งานได้อย่างถูกต้อง
3. แสดงขั้นตอนการตรวจสอบพอร์ตอินพุตและเอาต์พุตที่ใช้งานได้อย่างถูกต้อง

เครื่องมือและอุปกรณ์

1. โปรแกรม Xilinx ISE Foundation 6.3i Evaluation

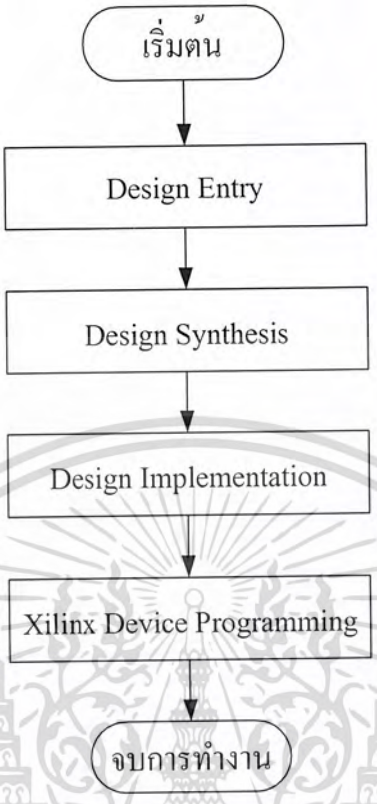
ความรู้และทักษะเดิม

1. การใช้งานโปรแกรม Xilinx ISE เบื้องต้น

ทฤษฎีเบื้องต้น

โปรแกรม Xilinx ISE Foundation เป็นโปรแกรมที่ออกแบบมาใช้งานกับ FPGA ของบริษัท Xilinx เท่านั้นโดยภายในโปรแกรมจะมีโปรแกรมย่อยมากมายที่ช่วยให้ผู้ออกแบบใช้งานได้สะดวกมากขึ้น เช่น โปรแกรมสังเคราะห์วงจร มีชื่อว่า Xilinx Synthesis Tool (XST) ซึ่งโปรแกรมนี้ถือว่ามีประสิทธิภาพมากเพราะสามารถใช้งานได้ตั้งแต่ขั้นตอนออกแบบวงจร จนถึงโปรแกรมลงบนชิพ FPGA โดยมีผังงานแสดงขั้นตอนการทำงานของโปรแกรมดังรูปที่ 10.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.1 ผังงานการทำงานของโปรแกรม Xilinx ISE Foundation

1. Design Entry

เป็นการออกแบบวงจรโดยสามารถที่จะออกแบบโดยใช้การเขียนภาษาบรรยายพฤติกรรมการออกแบบเป็นลักษณะวงจร หรืออาจใช้ Core Generator สร้างขึ้นมาก็ได้

2. Design Synthesis

เป็นกระบวนการสังเคราะห์วงจรจากการออกแบบ โดยจะแปลงเป็นวงจรที่สามารถโปรแกรมลงบนชิพ FPGA ได้ซึ่งจะอยู่ในระดับเกต ซึ่งจะใช้โปรแกรมสังเคราะห์วงจรที่มีชื่อว่า Xilinx Systhesis Tool

3. Design Implementation

เป็นการนำวงจรที่ได้มาสร้างลงบนโครงสร้างของชิพ FPGA จริง โดยจะทำการลากเส้นเชื่อมต่อภายในชิพและเชื่อมต่อกับพอร์ตอินพุตและพอร์ตเอาต์พุตของตัวชิพ FPGA

4. Xilink Device Programming

ในขั้นตอนการ Implementation จะได้ไฟล์ที่มีนามสกุล .bit ที่ใช้สำหรับการโปรแกรมลงบน FPGA ซึ่งจะต้องตรวจสอบว่าพอร์ตต่างๆ ได้มีการกำหนดถูกต้องแล้ว

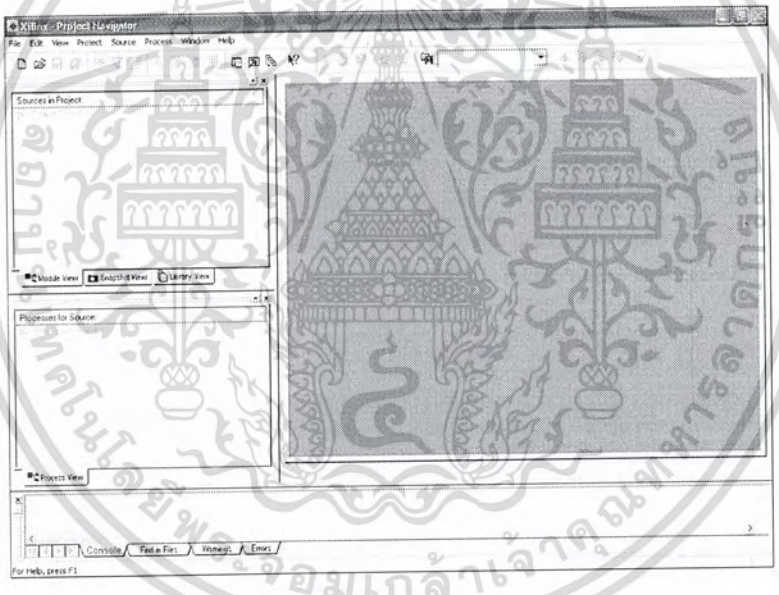
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

ในการทดลองนี้จะเป็นการนำโค๊ดภาษา VHDL ที่สร้างขึ้นจากโปรแกรม MATLAB และโปรแกรม System Generator มาสังเคราะห์เป็นวงจรที่สามารถโปรแกรมลงบนชิฟ FPGA ได้จริง ซึ่งจะแสดงขั้นตอนการสังเคราะห์วงจร การแบ่งวงจรเป็นส่วนๆ (Translate) และการเชื่อมต่อสายสัญญาณ (Place&Route) โดยจะแสดงวงจรในระดับ RTL (Register Transfer Level) ซึ่งสามารถดูผลการรายงานของความเร็วสูงสุดของวงจรและขนาดของพื้นที่ที่ใช้เมื่อโปรแกรมลงบนชิพ FPGA

1. เปิดโปรแกรม Xilinx ISE โดยดับเบิ้ลคลิกที่ไอคอน

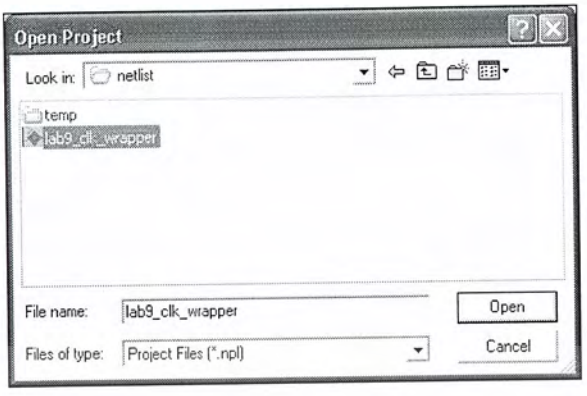
(Project Navigator) ก็จะปรากฏหน้าต่างโปรแกรมแสดงดังรูปที่ 10.2



รูปที่ 10.2 หน้าต่างโปรแกรม Xilinx ISE Foundation

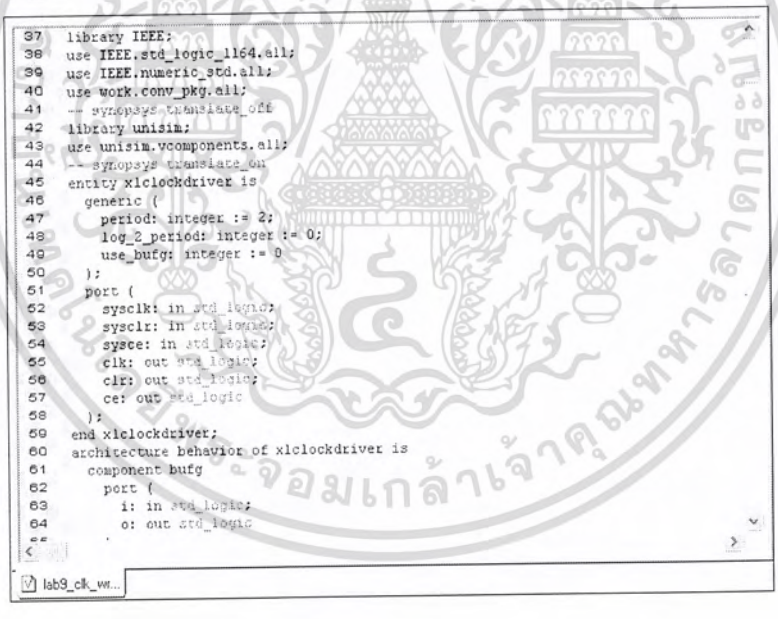
2. คลิกที่ **File > Open Project** เลือกไดเร็กทอรี่เป็น **C:\MATLAB7\LAB9\Netlist** เลือกไฟล์เป็น **lab9_clk_wrapper** แสดงรูปที่ 10.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.3 การเลือกไฟล์ lab9_clk_wrapper

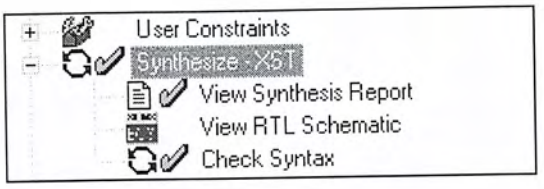
3. ที่หน้าต่าง Project Navigator แล้วคลิกขวาที่ไฟล์ **lab9_clk_wrapper** เลือก **Open** จะแสดงไฟล์ VHDL ดังรูปที่ 10.4



รูปที่ 10.4 โค๊ด VHDL ของไฟล์ lab9_clk_wrapper

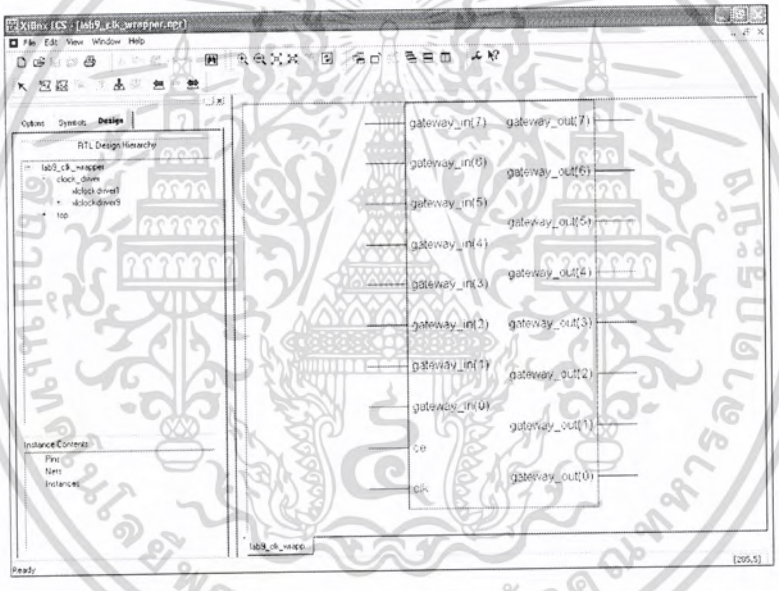
4. ที่หน้าต่าง Process for Source แล้วดับเบิ้ลคลิกที่ **Synthesize XST** จะแสดงขั้นตอนการ Check Syntax และ Synthesis เมื่อเสร็จสิ้นจะแสดงเครื่องหมาย ✔ แสดงผลดังรูปที่ 10.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.5 การสังเคราะห์วงจรเสร็จสิ้น

5. ดับเบิ้ลคลิกที่หัวข้อ **View Synthesis Report** จะแสดงหน้าต่าง Xilinx ECS ซึ่งจะแสดงวงจรที่ผ่านการสังเคราะห์แล้ว และสามารถที่จะดูวงจรย่อยของบล็อกได้ด้วยการดับเบิ้ลคลิกที่บล็อก แสดงดังรูปที่ 10.6



รูปที่ 10.6 วงจรกรองความถี่ที่ผ่านการสังเคราะห์แล้ว

6. ดับเบิ้ลคลิกที่ View Synthesis Report จะรายงานผลการสังเคราะห์วงจร ดูผลขนาดพื้นที่และความเร็วในการประมวลผล

- สังเกตผลใน**บรรทัดที่ 702** บันทึกผลการทดลองในตารางที่ 10.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 10.1 ขนาดพื้นที่และจำนวน I/O ที่ใช้งาน

Number of Slices:		out of		
Number of Slice Flip Flops:		out of		
Number of 4 input LUTs:		out of		
Number of bonded IOBs:		out of		
Number of GCLKs:		out of		

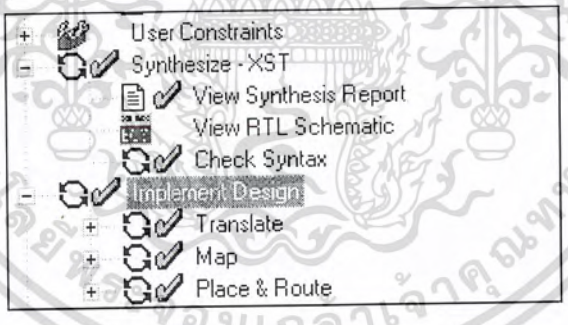
- สังเกตผลใน**บรรทัดที่ 733** บันทึกผลการทดลอง

  Minimum Period:_____

  Maximum Frequency:_____

ผลของ Maximum Frequency เป็นความเร็วสูงสุดของวงจรที่สามารถทำงานได้

7. ทำการดับเบิ้ลคลิกที่ **Implement Design** ก็จะแสดงขั้นตอนการ Translate, Map และ Place&Route เมื่อเสร็จสิ้นจะแสดงเครื่องหมาย ☑ แสดงผลดังรูปที่ 10.7



รูปที่ 10.7 การ Implement วงจรกรองเสร็จสิ้น

8. คลิกที่หัวข้อ Place & Route ดับเบิ้ลคลิกที่หัวข้อ Place & Route Report เพื่อดูผลการรายงานการเชื่อมต่อสายสัญญาณภายในชิพและทรัพยากรทั้งหมดที่ใช้งาน สังเกตผลใน**บรรทัดที่ 28** บันทึกผลการทดลองลงในตารางที่ 102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 10.2 รายงานผลจำนวน SLICE ที่ใช้งาน

Number of SLICEs		out of	

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จากขั้นตอนการทดลองที่ 8 นั้นก็จะเห็นได้ว่าจำนวน Slices ลดลง จงนำผล Slices ที่ได้เปรียบเทียบกับผลการทดลองที่ 39 ในใบงานที่ 9 ว่ามีค่าเท่ากันหรือไม่เพราะเหตุใด?

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
รูปแบบคำสั่งการติดต่อกับเครื่องวัดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบคำสั่งการติดต่อกับเครื่องวัดสัญญาณ

เราสามารถควบคุม Oscilloscope ได้โดยผ่านพอร์ต GPIB หรือ RS-232 ซึ่งเราจะต้องใช้ชุดคำสั่งและการตรวจสอบ และในที่นี้จะเป็นการบอกรูปแบบของคำสั่งและการตรวจสอบ รวมทั้งสัญลักษณ์ของ Oscilloscope ที่ใช้ในกระบวนการนี้

เราจะส่งคำสั่งไปควบคุม Oscilloscope โดยใช้รหัส ASCII ซึ่งจะแสดงรูปแบบของรหัสอยู่ในภาคผนวก คู่มือที่ใช้ในการ BNF (Baclus-Naur Form) แสดงดังในตารางที่ ข.1

ตารางที่ ข.1 สัญลักษณ์ของ BNF และความหมาย

สัญลักษณ์	ความหมาย
< >	ขอบเขต
::=	กำหนดค่า
|	Exclusive OR
{ }	กลุ่ม, หนึ่งประเภทที่ต้องการ
[]	ทางเลือก, สามารถละเว้นได้
( )	หมายเหตุ

โครงสร้างของคำสั่งและการตรวจสอบ

ชุดคำสั่งนั้นจะมีทั้งคำสั่งในการตั้งค่าและคำสั่งที่ใช้ในการตรวจจะสอบ ซึ่งคำสั่งทั้งสองแบบนี้จะมีรูปแบบของคำสั่งเหมือนกัน แต่คำสั่งในการตรวจสอบค่าจะต้องมีเครื่องหมาย “?” ต่อท้ายคำสั่งด้วย ตัวอย่างเช่น

คำสั่งในการตั้งค่าคือ

ACQuire : MODe

คำสั่งในการตรวจสอบก็คือ

ACQuire : MODeM?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ชุดคำสั่งทุกชุดนั้นไม่จำเป็นจะต้องมีทั้งคำสั่งในการตั้งค่าคำสั่งในการตรวจสอบเสมอไปบางคำสั่งก็มีแต่คำสั่งในการตั้งค่าเพียงอย่างเดียวและบางคำสั่งก็มีคำสั่งในการตรวจสอบอย่างเดียวเช่นกัน

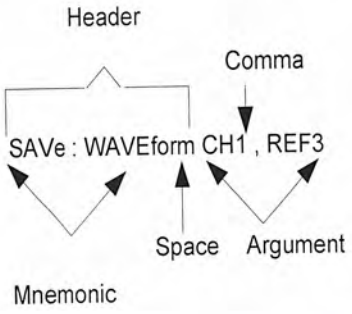
ประโยคคำสั่งคือ ชื่อคำสั่งหรือการตรวจสอบ ตามด้วยข้อมูลที่ Oscilloscope ต้องการที่จะนำมาเพื่อปฏิบัติงานตามสั่งนั้นๆ

ประโยคคำสั่ง ประกอบด้วยส่วนสำคัญที่แตกต่างกัน 5 ส่วน ดังที่แสดงในตารางที่ ข.2 จะเป็นคำจำกัดความของแต่ละส่วนและในรูปที่ 1 เป็นตัวอย่างคำสั่งที่ใช้งาน

ตารางที่ ข.2 ส่วนประกอบของประโยคคำสั่ง

สัญลักษณ์	ความหมาย
<Header>	ชื่อคำสั่งพื้นฐาน ถ้าตอนท้ายของ Header มีเครื่องหมาย "?" จะเป็นคำสั่งที่ใช้ในการตรวจสอบ Header จะเริ่มต้นด้วยเครื่องหมาย ":" ถ้าคำสั่งที่เชื่อมโยงอยู่กับคำสั่งอื่นจะต้องมีเครื่องหมาย ":" คั่นก่อนที่จะเริ่มต้นคำสั่งต่อไปในการเริ่มต้นคำสั่งถ้า Header ของคำสั่งขึ้นต้นด้วยเครื่องหมาย "*" แล้วก็ไม่จำเป็นต้องใช้เครื่องหมาย ':' ก็ได้
<Mnemonic>	หัวฟังก์ชันย่อย ปกติ Header ของคำสั่งจะมี Mnemonic จะต้องแยกออกทีละทางจากแต่ละที่อื่นๆ โดยเครื่องหมาย ":"
<Argument>	ปริมาณ < คุณภาพ ข้อจำกัด หรือขอบเขตจำกัดของ Header ซึ่งไม่จำเป็นที่ทุกคำสั่งจะต้องมี Argument และในขณะเดียวกัน 1 คำสั่งก็สามารถมีหลายๆ Argument ได้ โดยที่ Argument จะแยกจาก Header ด้วยการเคาะ Space Bar และจะแยกจากกันโดยเครื่องหมาย ","
<Common>	Comma (,) 1 ตัวจะอยู่ระหว่าง 1 Argument ของหลายๆ Argument ซึ่งจะต้องทำการกด Space ทั้งก่อนและหลัง Comma
<Space>	ช่องว่างระหว่าง Header และ Argument ซึ่งอาจจะมีหลายๆ ช่องว่างก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.1 ส่วนประกอบสำคัญของประโยคคำสั่ง

คำสั่งที่ใช้ในกรณีของ Oscilloscope จะแสดงข้อกำหนดของ Function หรือเปลี่ยนแปลงค่าใดค่าหนึ่งของตัวมัน ซึ่งมีโครงสร้างของคำสั่ง ดังนี้

[:]<Header>[<Space><Argument>[<comma><Argument>]…]

หัวคำสั่งเป็นการรวมกันของ Mnemonic 1 ตัว หรือมากกว่า ซึ่งจะจัดระเบียบเป็นแบบลำดับชั้นหรือโครงสร้างแบบต้นไม้ Mnemonic ตัวแรกคือพื้นฐานหรือรากของต้นไม้ และแต่ละลำดับถัดไปของ Mnemonic คือ ระดับหรือกิ่งปีดของระดับชั้นก่อนหน้านั้น คำสั่งที่ระดับสูงขึ้นไปในโครงสร้างแบบต้นไม้นี้ จะส่งผลกระทบต่อคำสั่งระดับต่ำ เครื่องหมาย ":" ที่นำหน้าจะส่งค่ากลับไปยังฐานหรือรากของต้นไม้เสมอ ซึ่งนั่นก็คือ Mnemonic ตัวแรกนั้นเอง

การตรวจสอบ การตรวจสอบในกรณีของ Oscilloscope

จะต้องมีการส่งกลับข้อมูลเกี่ยวกับและการตั้งค่า

- [:]<Header>]?
- [:]<Header>?[<Space><Argument>[<Comma><Argument>]…]

เราสามารถกำหนดรายละเอียดของคำสั่งตรวจสอบได้ที่ทุกระดับชั้น ภายในโครงสร้างแบบต้นไม้เว้นเสียแต่ว่าจะเป็นโครงสร้างสัญลักษณ์แบบอื่น ที่กิ่งของโครงสร้างจะทำการตรวจสอบการส่งกลับข้อมูลเกี่ยวกับ Mnemonic ทั้งหมดที่ระดับเดียวกันหรือต่ำกว่า ตัวอย่างเช่น

DISplay : INTENSITY : CONtast?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการส่งกลับค่าความแรงจากจุดที่แรงที่สุดของ Waveform ขณะที่ DISplay : INTENSITY? จะเป็นการส่งกลับค่าความแรงที่เราตั้งค่าไว้จากทุกส่วนของตัวแสดงผล

เราสามารถควบคุม Oscilloscope ให้ส่งค่ากลับไปยัง Header ที่ส่วนของผลตอบสนองในการตรวจสอบ ได้สองกรณีโดยใช้คำสั่ง HEADer ไปควบคุม ดังนี้ ถ้า Header ทำงานอยู่ Oscilloscope จะทำการส่งคำสั่งกลับไปยัง Header ที่ส่วนของการตรวจสอบและรูปแบบของผลตอบสนองที่ใช้บังคับในการ Set คำสั่ง และเมื่อ Header ไม่ทำงานหรือหยุดทำงาน Oscilloscope จะส่งค่าทุกค่าในการตอบสนองกลับไปยัง Header ซึ่งมันจะทำให้ง่ายต่อการกระจายและรวบรวมข้อมูลจากการตอบสนอง ตารางที่ ข.3 แสดงความแตกต่างในการตอบสนอง

ตารางที่ ข.3 การเปรียบเทียบระหว่าง Header ที่มีผลตอบสนองกับไม่มี

Query	Header off Response	Header on Response
CURSor: VBArs:	1.064E-3	:CURSor: VBArs:DEL Ta 1.064E-3
ACQuire: NUMAVg?	16	:ACQuire:NUMAVG 16

Clearing the Oscilloscope

ทำการ Clear เอาต์พุตและ Reset Oscilloscope เพื่อรับคำสั่งใหม่ โดยใช้คำสั่ง Device Clear (DCL) GPIB หรือใช้สัญญาณ RS-232 BREAK

Command Entry

กฎทั่วไปในการเข้าถึงคำสั่ง

- เราสามารถเข้าถึงคำสั่งที่มีระดับสูงกว่าและในระดับที่ต่ำกว่าได้
- เราสามารถขึ้นต้นคำสั่งด้วย White Space ได้ โดยที่ White Space เป็นการรวมกันของ ASCII Control Characters 00.09 หรือ 20 Decimal
- Oscilloscope จะไม่ทำงานตามคำสั่งที่ประกอบด้วยการรวมกัน White Space และ Line Feeds

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งย่อ

เราสามารถทำการย่อคำสั่งต่างๆ ได้ ตัวอย่างเช่น

คำสั่ง ACQuire:NUMAVg

เราสามารถทำการย่อคำสั่งได้ดังนี้

ACQuire:NUM หรือ acq:numa

การเชื่อมต่อคำสั่ง

เราสามารถเชื่อมต่อและรวมคำสั่งในการตั้งค่าและการตรวจสอบโดยใช้เครื่องหมาย (;) และ Oscilloscope จะทำงานตามลำดับที่เชื่อมต่อ

เมื่อเราต้องการเชื่อมต่อคำสั่งเราต้องปฏิบัติตามกฎ ดังนี้

1) Header ที่แตกต่างกันจะถูกแยกออกจากกันโดยเครื่องหมาย (;) และจะเริ่มต้นด้วยเครื่องหมาย (:) ทุกคำสั่ง ยกเว้น คำสั่งแรก ตัวอย่างเช่น

คำสั่ง TRIGger:MODe NORMal และ ACQuire:NUMAVg 8

สามารถเชื่อมต่อให้เป็นคำสั่งเดียวกันได้ดังนี้

TRIGger:MODe NORMal ;: AQuire:NUMAVg 8

2) ถ้าแต่ละคำสั่งที่ต้องการเชื่อมต่อกันมี Headers ที่ Mnemonic ตัวสุดท้ายแตกต่างกัน เราสามารถย่อคำสั่งที่ 2 และตัดเครื่องหมาย (:) ที่เริ่มต้นทิ้งไปได้ ตัวอย่างเช่น

คำสั่ง ACQuire : MODe ENVelopeและACQuire : NUMAVg 4

สามารถเชื่อมต่อให้เป็นคำสั่งเดียวกันได้ดังนี้

ACQuire : MODe ENVelope ; NUMAVg 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสามารถทำงานได้ดีเท่ากับการเขียนแบบยาวๆ

ACQuire : MODe ENVelope ;: ACQuire : NUMAVg 4

3) เครื่องหมาย (*) จะถูกให้ความสำคัญมากที่สุด

ACQuire : MODe ENVelope ;*TRG

Oscilloscope จะทำงานตามกระบวนการของคำสั่งไปเรื่อยๆ ถ้าคำสั่งนั้นไม่มีเครื่องหมาย (*) ซึ่งจะเป็นไปตามตัวอย่างที่ใช้บังคับดังนี้

ACQuire : MODe ENVelope ;*TRG;NUMAVg 2

4) เมื่อใดที่เราทำการเชื่อมต่อการตรวจสอบ Oscilloscope จะทำการเชื่อมต่อผลตอบสนองของทุกการตรวจสอบเป็นผลตอบสนองเดียวกัน ตัวอย่างเช่น ถ้าความเข้มของการแสดงผล text คือ “bright” และของ waveform คือ “dim” การเชื่อมต่อการตรวจสอบเป็นดังนี้

DISplay:INTENsity:TEXt?;WAVEform?

จะส่งกลับแต่ละส่วน

:DISPLAY:INTENSITY”TEXTBRI;:DISPLAY:INTENSITY:WAVEFORM DIM ถ้า Header off

5) เราสามารถเชื่อมต่อ ตั้งค่าคำสั่งและการตรวจสอบไว้ในประโยคเดียวกัน ดังเช่นตัวอย่างนี้

ACQuire : MODe NORMal;NUMAVg?STATE?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเชื่อมต่อที่ไม่ถูกต้อง

DISPlay:INTENsity:TEXt BRI;ACQuire:NUMAVg 16
(ไม่มี(:)หน้าACQuire)

- DISPlay:INTENsity:TEXt DIM;:WAVEform BRI
(เครื่องหมาย Extra (:) ก่อน WAVEform สามารถใช้ ISPlay:INTENsity:WAVEform แทนที่ได้)
- DISPlay:INTENsity:TEXt DIM;:*TRG
(เครื่องหมาย Extra (:) ก่อน เครื่องหมาย *)

การสิ้นสุดคำสั่ง ในคู่มือนี้ใช้ <EOM>(End of Massage) เพื่อแทนการสิ้นสุดคำสั่ง

GPIB End of Massage Terminal GPIB EOM Terminal สามารถเป็นตัวจบคำสั่งได้ (EOI - จะเป็นตัวยืนยันพร้อมด้วยข้อมูลไบท์สุดท้าย) รหัส ASCII ของ Line Feed (LF) จะดูตามข้อมูลไบท์สุดท้ายหรือดูทั้งสองอย่าง Oscilloscope จะต้องจบคำสั่งด้วย LF และ EOI เสมอ และจะต้องเว้นช่องว่างก่อนการสิ้นสุดคำสั่งด้วย

การสร้าง Mnemonic

บาง Header Mnemonics จะระบุเป็น 1 ในขอบเขตของ Mnemonics ตัว Mnemonic อันใดอันหนึ่งตามคำสั่ง ตัวอย่างเช่น นี่คือคำสั่ง CH1 หรือ CH2 ก็ได้เหมือนกัน เราจะใช้ Mnemonic อันใดอันหนึ่งตามคำสั่ง ตัวอย่างเช่น นี่คือคำสั่ง CH1:VOLts และนี่ก็คือ คำสั่ง CH2:VOLts เช่นเดียวกัน ในลักษณะคำสั่งนี้คือการแสดงตัวเลือกเขียนย่อได้ดังนี้ CH<x> ตำแหน่ง Cursor ของ Mnemonic เมื่อ Oscilloscope แสดง Cursor คำสั่งจะต้องระบุที่ Corsor ด้วยข้อมูลทั้งคู่ที่นำไปใช้งานดังตารางที่ ข.4

ตารางที่ ข. 4 การแสดงตัวเลือกเขียนย่อ ตำแหน่ง Cursor ของ Mnemonic เมื่อ Oscilloscope แสดง Cursor

สัญลักษณ์	ความหมาย
ตำแหน่ง<x>	Cursor ให้เลือกที่ตำแหน่ง <x> เป็น 1 หรือ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การระบุเครื่องวัดให้ Mnemonic

คำสั่งสามารถระบุ เครื่องมือวัดที่เราต้องการตั้งค่าหรือหรือตรวจสอบที่ Mnemonic ใน Header Oscilloscope สามารถแสดงเครื่องมือวัดอัตโนมัติได้ 4 ตัว ด้วยการแสดง Waveform ของแต่ละตัว การแสดงเครื่องมือวัดสามารถระบุได้ดังตารางที่ ข.5, ตารางที่ ข.6, ตารางที่ ข.7, ตารางที่ ข.8 และตารางที่ ข.9

ตารางที่ ข. 5 การแสดงเครื่องมือวัดอัตโนมัติได้ 4 ตัว ด้วยการแสดง Waveform ของแต่ละตัว

สัญลักษณ์	ความหมาย
MEAS<x>	ระบุเครื่องมือวัด; <x> เป็น 1, 2, 3 หรือ 4

ตารางที่ ข.6 Channel Mnemonic เป็นคำสั่งระบุ channel ที่ใช้งานที่ Mnemonic ใน Header

สัญลักษณ์	ความหมาย
CH<x>	ระบุ Channel;<x>เป็น 1 หรือ 2

ตารางที่ ข.7 Math Waveform Mnemonic เป็นคำสั่งระบุ Waveform ในทางคณิตศาสตร์ที่ใช้งาน

สัญลักษณ์	ความหมาย
MATH<x>	ระบุ Math Waveform;<x>เป็น 1

ตารางที่ ข.8 Reference Waveform Mnemonic เป็นคำสั่งระบุ Reference Waveform ที่ใช้งานได้

สัญลักษณ์	ความหมาย
REF<x>	ระบุ Reference Waveform;<x>เป็น 1 หรือ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.9 Waveform Mnemonic ในบางคำสั่งสามารถระบุ Waveform คือ Channel Mnemonic, Math Waveform หรือ Reference Waveform สามารถระบุได้ตามนี้

สัญลักษณ์	ความหมาย
<wfm>	สามารถระบุได้เป็น CH<x>, MATH<x>, MATH12<x> หรือ REF<x>

ชนิดของ Argument

คำสั่ง Argument สามารถเป็น 1 ในหลายๆ รูปแบบ แต่ละลักษณะของคำสั่งแต่ละลักษณะของ Argument แต่ละชนิดสามารถใช้งานได้กับคำสั่งนั้นๆ

Numeric Argument ส่วนมากแล้วคำสั่งของ Oscilloscope ต้องการ Numeric Argument ในคู่มือนี้จะเสนอ Argument ตามตารางที่ ข.10

ตารางที่ ข.10 Numeric Argument

สัญลักษณ์	ความหมาย
<NR1>	สัญลักษณ์แสดงค่าของเลขจำนวนเต็ม
<NR2>	ค่าของเลขทศนิยมที่ไม่มีเลขยกกำลัง
<NR3>	ค่าของเลขทศนิยมที่มีเลขยกกำลัง

Quoted String Argument บางคำสั่งจะรับหรือส่งกลับข้อมูลในรูปแบบของการอ้างข้อความ ซึ่งมันจะทำให้ง่ายขึ้น กลุ่มของ อักขระ ASCII ส่งโดยเครื่องหมาย (‘) (Single Quote) หรือเครื่องหมาย (“) (Double quote) ตัวอย่างเช่น “This is a quote string” ดังตารางที่ ข.11

ตารางที่ ข.11 คำสั่งของ Quoted String Argument

สัญลักษณ์	ความหมาย
<Qstring>	การอ้างถึงข้อความของ ข้อความ ASCII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อจะใช้ Quote String เราต้องปฏิบัติตามกฎดังต่อไปนี้

- Quote String สามารถเรียกใช้อักขระได้ตามที่กำหนดไว้ในอักขระ ASCII 7 bit
- ต้องใช้เครื่องหมาย Quote Character (“”)หรือ(‘’) ชนิดเดียวกันในการเปิดและปิดข้อความเช่น “This is a valid string”
- เราสามารถใช้เครื่องหมายทั้งสองชนิดร่วมกันได้ในข้อความที่ยาวมากได้ เราสามารถทำได้ดังนี้ “This is an ‘acceptable’string”
- เราสามารถเรียกใช้ Quote character เพื่อทำให้ข้อความง่ายขึ้นโดยการใช้ Quote character ซ้ำๆ กัน ตัวอย่างเช่น “Here is a” “mark”
- ข้อความสามารถมี upper หรือ lower Case Character
- เราไม่สามารถที่จะจบข้อความด้วย END Massage ได้ก่อนปิดเขตข้อความ
- ความยาวสูงสุดของข้อความที่ส่งกลับจากการตรวจสอบคือ 1000 อักขระ

ตัวอย่างข้อความที่ไม่ถูกต้อง

“Invalid string Argument’

(เครื่องหมาย Quote ต่างชนิดกัน)

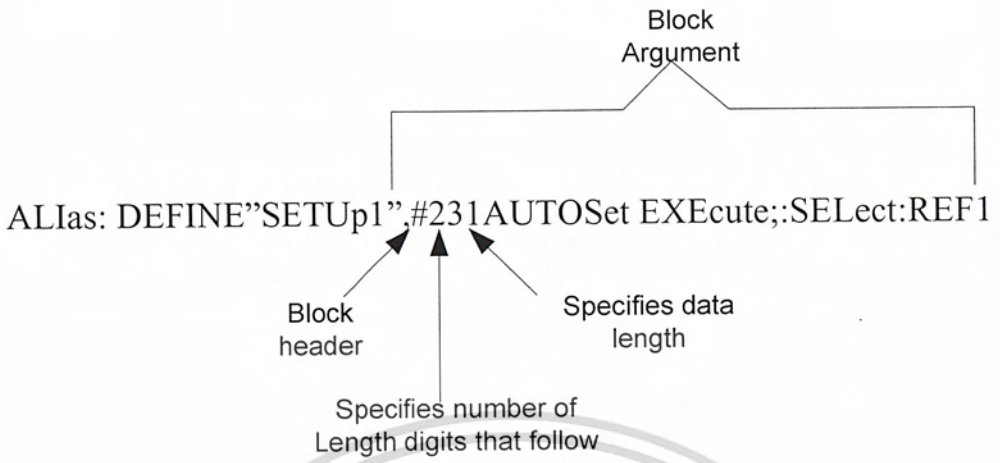
“Test<EOI>”

(EOI ไม่สามารถใช้ในการสิ้นสุดข้อความได้)

Block Argument คำสั่งหลายคำสั่งใช้ block Argument

ด้วยพอร์ต GPIB, เส้นสัญญาณ EOI ในไบท์สุดท้าย รูปที่ ข.2 แสดงตัวอย่างของ block Argument

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2 ตัวอย่างของ block Argument

Block header=#

Specifies data length = **31**

Specifies number of Length digits that follow=**2**

Block Argument =**#231 AUTOSet EXEcute;:SELect:REF**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

แบบประเมินคุณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบประเมินคุณภาพใบงานการทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB โดยผู้ทรงคุณวุฒิ

คำชี้แจง โปรดทำเครื่องหมาย √ ลงในช่องตามความคิดเห็นของท่าน

คะแนนระดับความคิดเห็น คือ ดีมาก = 5, ดี = 4, ปานกลาง = 3, น้อย = 2, ควรปรับปรุง = 1

ด้านที่ประเมิน	ระดับความคิดเห็น				
	5	4	3	2	1
1. ด้านวิชาการ					
1.1 เนื้อหามีความสอดคล้องกับวัตถุประสงค์					
1.2 ความถูกต้องของภาษาที่ใช้					
1.3 ความเหมาะสมในการจัดลำดับความสำคัญของเนื้อหา					
1.4 ความถูกต้องของใบงานการทดลอง					
1.5 ใบงานการทดลองสามารถนำไปใช้ในการเรียนการสอนได้					
1.6 มีความชัดเจนของขั้นตอนการทำงาน					
2. ด้านสื่อการเรียนการสอน					
2.1 การวางรูปแบบของใบงานการทดลอง					
2.2 ความเหมาะสมของภาพที่ประกอบใบงานการทดลอง					
2.3 มีฮาร์ดแวร์ประกอบการทดลอง					

ข้อเสนอแนะอื่นๆ

.....

.....

ลงชื่อ.....ผู้ประเมิน

(.....)

...../...../.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบประเมินคุณภาพใบงานการทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB โดยผู้ทรงคุณวุฒิ

คำชี้แจง โปรดทำเครื่องหมาย √ ลงในช่องตามความคิดเห็นของท่าน

คะแนนระดับความคิดเห็น คือ ดีมาก = 5, ดี = 4, ปานกลาง = 3, น้อย = 2, ควรปรับปรุง = 1

ด้านที่ประเมิน	ระดับความคิดเห็น				
	5	4	3	2	1
1. ด้านวิชาการ					
1.1 เนื้อหามีความสอดคล้องกับวัตถุประสงค์	✓				
1.2 ความถูกต้องของภาษาที่ใช้		✓			
1.3 ความเหมาะสมในการจัดลำดับความสำคัญของเนื้อหา	✓				
1.4 ความถูกต้องของใบงานการทดลอง	✓				
1.5 ใบงานการทดลองสามารถนำไปใช้ในการเรียนการสอนได้	✓				
1.6 มีความชัดเจนของขั้นตอนการทำงาน		✓			
2. ด้านสื่อการเรียนการสอน	✓				
2.1 การวางรูปแบบของใบงานการทดลอง	✓				
2.2 ความเหมาะสมของภาพที่ประกอบใบงานการทดลอง	✓				
2.3 มีฮาร์ดแวร์ประกอบการทดลอง		✓			

ข้อเสนอแนะอื่นๆ

.....

.....

ลงชื่อ.....ผู้ประเมิน

(.....)

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบประเมินคุณภาพใบงานการทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB โดยผู้ทรงคุณวุฒิ

คำชี้แจง โปรดทำเครื่องหมาย √ ลงในช่องตามความคิดเห็นของท่าน

คะแนนระดับความคิดเห็น คือ ดีมาก = 5, ดี = 4, ปานกลาง = 3, น้อย = 2, ควรปรับปรุง = 1

ด้านที่ประเมิน	ระดับความคิดเห็น				
	5	4	3	2	1
1. ด้านวิชาการ					
1.1 เนื้อหามีความสอดคล้องกับวัตถุประสงค์	/				
1.2 ความถูกต้องของภาษาที่ใช้	/				
1.3 ความเหมาะสมในการจัดลำดับความสำคัญของเนื้อหา	/				
1.4 ความถูกต้องของใบงานการทดลอง	/				
1.5 ใบงานการทดลองสามารถนำไปใช้ในการเรียนการสอนได้	/				
1.6 มีความชัดเจนของขั้นตอนการทำงาน	/				
2. ด้านสื่อการเรียนการสอน					
2.1 การวางรูปแบบของใบงานการทดลอง	/				
2.2 ความเหมาะสมของภาพที่ประกอบใบงานการทดลอง		/			
2.3 มีฮาร์ดแวร์ประกอบการทดลอง	/				

ข้อเสนอแนะอื่นๆ

.....

.....

ลงชื่อ.....ผู้ประเมิน

(นาย.....)

...../...../.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบประเมินคุณภาพใบงานการทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB โดยผู้ทรงคุณวุฒิ

คำชี้แจง โปรดทำเครื่องหมาย √ ลงในช่องตามความคิดเห็นของท่าน

คะแนนระดับความคิดเห็น คือ ดีมาก = 5, ดี = 4, ปานกลาง = 3, น้อย = 2, ควรปรับปรุง = 1

ด้านที่ประเมิน	ระดับความคิดเห็น				
	5	4	3	2	1
1. ด้านวิชาการ	/				
1.1 เนื้อหามีความสอดคล้องกับวัตถุประสงค์	/				
1.2 ความถูกต้องของภาษาที่ใช้		/			
1.3 ความเหมาะสมในการจัดลำดับความสำคัญของเนื้อหา	/				
1.4 ความถูกต้องของใบงานการทดลอง	/				
1.5 ใบงานการทดลองสามารถนำไปใช้ในการเรียนการสอนได้		/			
1.6 มีความชัดเจนของขั้นตอนการทำงาน	/				
2. ด้านสื่อการเรียนการสอน		/			
2.1 การวางรูปแบบของใบงานการทดลอง		/			
2.2 ความเหมาะสมของภาพที่ประกอบใบงานการทดลอง	/				
2.3 มีฮาร์ดแวร์ประกอบการทดลอง	/				

ข้อเสนอแนะอื่นๆ การให้นักศึกษาได้ลองเปลี่ยนค่าพารามิเตอร์ต่างๆ ในการทดลองเพื่อเรียนรู้ด้วยตัวเอง

ลงชื่อ ..... ผู้ประเมิน

(.....)

4 / 4 / 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
หนังสือแต่งตั้งผู้ทรงคุณวุฒิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกข้อความ

ส่วนราชการ ภาควิชาครุศาสตร์วิศวกรรม โทร. 3702-3

ที่ ศธ 0524.04(5)/ 143 **วันที่** 31 มีนาคม 2548

เรื่อง ขอเชิญเป็นอาจารย์ผู้ทรงคุณวุฒิทางด้านวิชาการและสื่อการเรียนการสอน

เรียน อาจารย์วรวิทย์ สมหา

ด้วยภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล. พิจารณาแล้วเห็นว่าท่านเป็นผู้มีความรู้ ความสามารถ และประสบการณ์ ที่เป็นประโยชน์ต่อการจัดทำโครงงานการสร้างอุปกรณ์เพื่อการสอนของนักศึกษาเป็นอย่างยิ่ง จึงมีความประสงค์เรียนเชิญเป็นอาจารย์ผู้ทรงคุณวุฒิทางด้านวิชาการและสื่อการเรียนการสอน ในวิชาโครงงานการสร้างอุปกรณ์เพื่อการสอน เรื่อง "ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB" ของนักศึกษาชั้นปีที่ 2 สาขาวิชาวิศวกรรมโทรคมนาคม โดยมีนักศึกษาดำเนินการจัดทำดังนี้

1. นายอมรเทพ ผันสิน
2. นางสาวอุทัยวรรณ มานะมุ่งประเสริฐ
3. นายสมชาย เสนาปิน

จึงเรียนมาเพื่อโปรดพิจารณา และหวังเป็นอย่างยิ่งว่าจะได้รับความร่วมมือจากท่าน และขอขอบคุณมา ณ โอกาสนี้ด้วย

(ผู้ช่วยศาสตราจารย์สุรสิทธ์ ราตรี)
หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกข้อความ

ส่วนราชการ ภาควิชาครุศาสตร์วิศวกรรม โทร. 3702-3

ที่ ศธ 0524.04(5)/ ๒๖ **วันที่** 31 มีนาคม 2548

เรื่อง ขอเชิญเป็นอาจารย์ผู้ทรงคุณวุฒิทางด้านวิชาการและสื่อการเรียนการสอน

เรียน อาจารย์พิชญ์สินี มะโน

ด้วยภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล. พิจารณาแล้วเห็นว่าท่านเป็นผู้มีความรู้ ความสามารถ และประสบการณ์ ที่เป็นประโยชน์ต่อการจัดทำโครงงานการสร้างอุปกรณ์เพื่อการสอนของนักศึกษาเป็นอย่างยิ่ง จึงมีความประสงค์เรียนเชิญเป็นอาจารย์ผู้ทรงคุณวุฒิทางด้านวิชาการและสื่อการเรียนการสอน ในวิชาโครงงานการสร้างอุปกรณ์เพื่อการสอน เรื่อง "ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB" ของนักศึกษาชั้นปีที่ 2 สาขาวิชาวิศวกรรมโทรคมนาคม โดยมีนักศึกษาดำเนินการจัดทำดังนี้

1. นายอมรเทพ ผันสิน
2. นางสาวอุทัยวรรณ มานะมุ่งประเสริฐ
3. นายสมชาย เสนาปิน

จึงเรียนมาเพื่อโปรดพิจารณา และหวังเป็นอย่างยิ่งว่าจะได้รับความร่วมมือจากท่าน และขอขอบคุณมา ณ โอกาสนี้ด้วย

(ผู้ช่วยศาสตราจารย์สุรสิทธ์ ราตรี)
หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกข้อความ

ส่วนราชการ ภาควิชาครุศาสตร์วิศวกรรม โทร. 3702-3

ที่ ศธ 0524.04(5)/ 142 **วันที่** 31 มีนาคม 2548

เรื่อง ขอเชิญเป็นอาจารย์ผู้ทรงคุณวุฒิทางด้านวิชาการและสื่อการเรียนการสอน

เรียน ดร.พรพิมล ฉายรัศมี

ด้วยภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล. พิจารณาแล้วเห็นว่าท่านเป็นผู้มีความรู้ ความสามารถ และประสบการณ์ ที่เป็นประโยชน์ต่อการจัดทำโครงงานการสร้างอุปกรณ์เพื่อการสอนของนักศึกษาเป็นอย่างยิ่ง จึงมีความประสงค์เรียนเชิญเป็นอาจารย์ผู้ทรงคุณวุฒิทางด้านวิชาการและสื่อการเรียนการสอน ในวิชาโครงงานการสร้างอุปกรณ์เพื่อการสอน เรื่อง "ชุดทดลองการปฏิบัติงานที่เวลาจริงด้วย MATLAB" ของนักศึกษาชั้นปีที่ 2 สาขาวิชาวิศวกรรมโทรคมนาคม โดยมีนักศึกษาดำเนินการจัดทำดังนี้

1. นายอมรเทพ ผันสิน
2. นางสาวอุทัยวรรณ มานะมุ่งประเสริฐ
3. นายสมชาย เสนาปีน

จึงเรียนมาเพื่อโปรดพิจารณา และหวังเป็นอย่างยิ่งว่าจะได้รับความร่วมมือจากท่าน และขอขอบคุณมา ณ โอกาสนี้ด้วย

(ผู้ช่วยศาสตราจารย์สุรสิทธ์ ราตรี)
หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อ-สกุล	นายอมรเทพ ผันสิน
วัน เดือน ปีเกิด	21 มิถุนายน พ.ศ. 2526
ภูมิลำเนา	65/2 หมู่ที่ 1 ตำบลบึง อำเภอศรีราชา
	จังหวัดชลบุรี 20230

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนบ้านบางพระ
มัธยมศึกษาตอนต้น	โรงเรียนศรีราชา
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคสัตหีบ
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคสัตหีบ
ปริญญาตรี	สาขาวิชาวิศวกรรมโทรคมนาคม
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	ไม่มีอะไรยากเกินความสามารถ ถ้าเราตั้งใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อ-สกุล	นางสาวอุทัยวรรณ มานะมุ่งประเสริฐ
วัน เดือน ปีเกิด	17 พฤษภาคม พ.ศ. 2525
ภูมิลำเนา	441/1 หมู่ที่ 13 ตำบลห้วยใหญ่ อำเภอบางละมุง จังหวัดชลบุรี 20260

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนวัดชากหมาก
มัธยมศึกษาตอนต้น	โรงเรียนวัดชากหมาก
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคสัตหีบ
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคสัตหีบ
ปริญญาตรี	สาขาวิชาวิศวกรรมโทรคมนาคม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	ทำทุกอย่างเพื่อครอบครัวมีความสุข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อ-สกุล	นายสมชาย เสนาปิน
วัน เดือน ปีเกิด	23 พฤศจิกายน พ.ศ. 2525
ภูมิลำเนา	36 หมู่ที่ 1 ตำบลสองคอน อำเภอแก่งคอย
	จังหวัดสระบุรี 18110

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนวัดสองคอนกลาง
มัธยมศึกษาตอนต้น	โรงเรียนสองคอนวิทยาคม
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคสระบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง	สถาบันเทคโนโลยีราชมงคล วิทยาเขตพระนครเหนือ
ปริญญาตรี	สาขาวิชาวิศวกรรมโทรคมนาคม
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	ทำวันนี้ให้ดีที่สุด เพื่อสิ่งที่ดีในวันข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้