เอกสารอ้างอิง

[1] http://www.lib.kmutt.ac.th/st4kid/nonFlash/services/getText.jsp?id=137

[2] ศูนย์การเรียนรู้ออนไลน์ สำนักคอมพิวเตอร์และเทคโนโลยีสารสนเทศ http://lms.icit.kmutnb.ac.th/moodle/index.php

[3] http://analogsoftware.itgo.com/2.3.htm

[4] หลักการสื่อสารและโทรคมนาคม โดย รศ. ดร. ปรีชา กอเจริญ ภาควิชา วิศวกรรมไฟฟ้าและอิเล็กทรอนิกส์ประยุกต์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีปทุม, 2557.

[5] http://www.ee.eng.cmu.ac.th/~tharadol/teach/ee442/text/radio_07.pdf

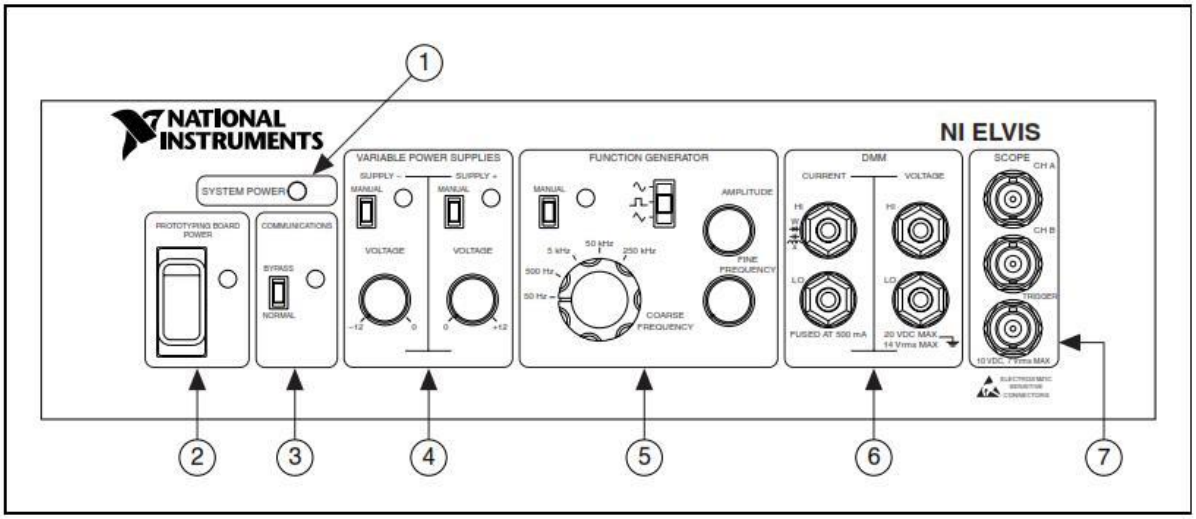
ภาคผนวก ก

การใช้งาน NI ELVIS

(NI Educational Laboratory Virtual Instrumentation Suite)

การใช้งานอุปกรณ์ NI ELVIS จะแบ่งเป็น 2 ส่วนด้วยกัน คือ

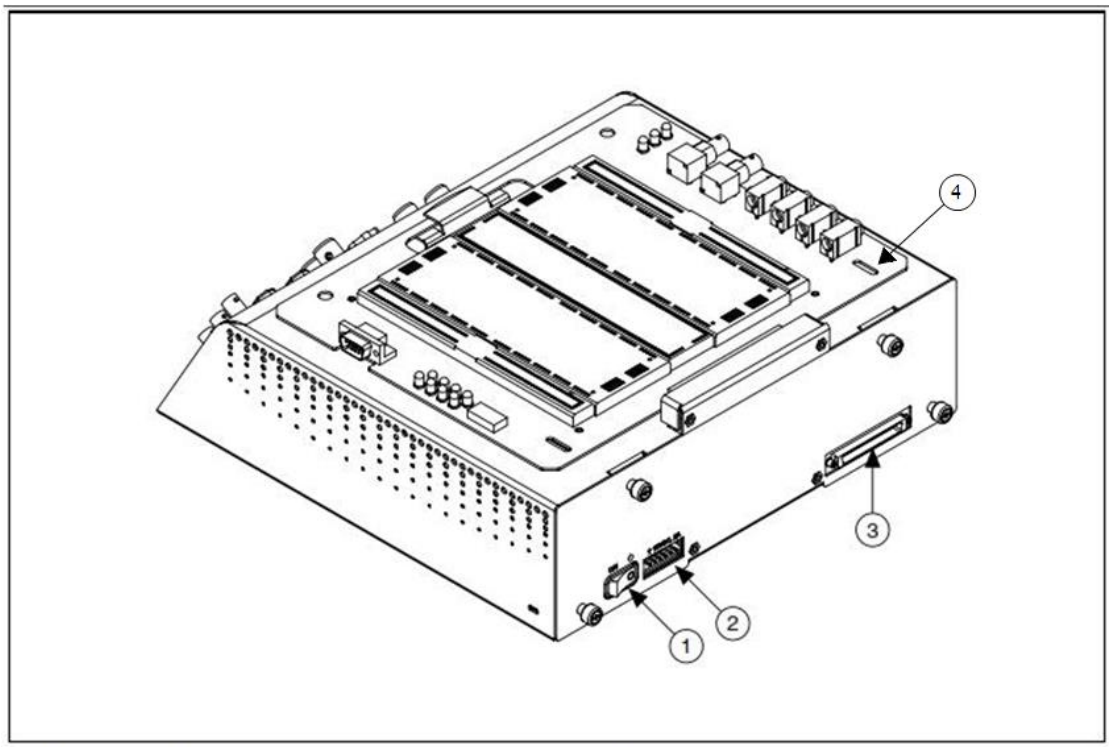
1. การปรับจากตัวอุปกรณ์ NI ELVIS หรือเรียกว่า การปรับแบบ Manual



1 System Power LED	5 Function Generator (FGEN) Controls
2 Prototyping Board Power Switch	6 DMM Connectors
3 Communications Switch	7 Oscilloscope (Scope) Connectors
4 Variable Power Supplies Controls	

ภาพที่ ก 1 แสดงตำแหน่งแผงควบคุมการใช้งานอุปกรณ์ NI ELVIS

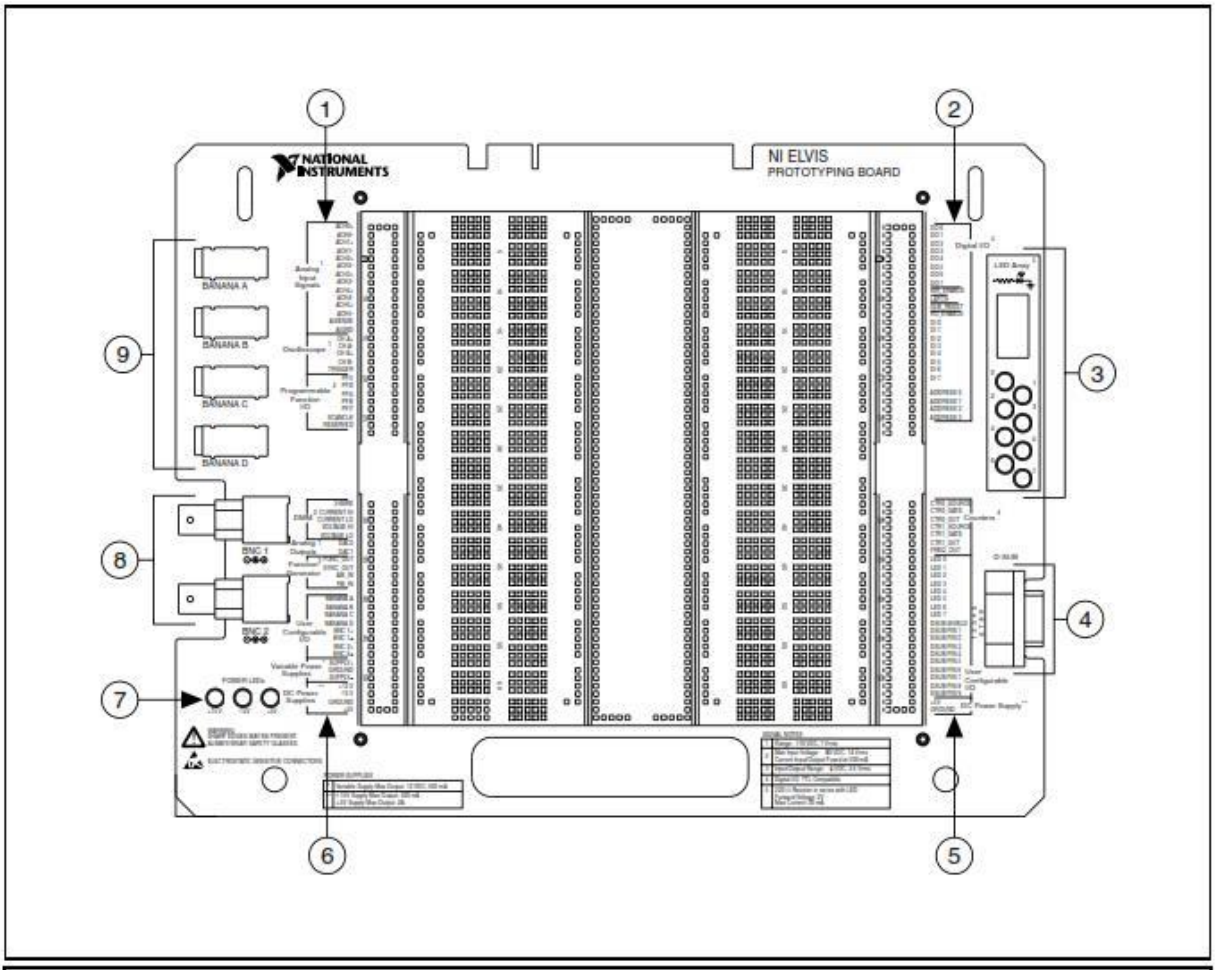
1. LED แสดงสถานะเปิด-ปิด ของอุปกรณ์ NI ELVIS
2. สวิตซ์เปิด-ปิด การใช้งานบอร์ดต่อวงจรบนอุปกรณ์ NI ELVIS
3. สวิตซ์ในการติดต่อสื่อสารระหว่างเครื่องกับโปรแกรม
4. ปุ่มปรับแรงดันของอุปกรณ์ NI ELVIS
5. ปุ่มปรับการใช้งานวงจรกำเนิดสัญญาณบนอุปกรณ์ NI ELVIS
6. คอนเนคเตอร์อินพุต ดิจิตอลมัลติมิเตอร์
7. คอนเนคเตอร์อินพุต ออสซิโลสโคป



1	Benchtop Workstation Power Switch	3	68-Pin DAQ Device Connector
2	AC-DC Power Supply Connector	4	Circuit Boards

ภาพที่ ก 2 การเชื่อมต่อด้านหลังของอุปกรณ์ NI ELVIS

1. สวิตซ์เปิด-ปิดการทำงานของอุปกรณ์ NI ELVIS
2. ขั้วต่อพาวเวอร์ซัพพลายAC-DC
3. พอร์ตเชื่อมต่ออุปกรณ์ DAQ 68-Pin
4. บอร์ดต่อวงจร



1	AI, Oscilloscope, and Programmable Function I/O Signal Rows	6	DMM, AO, Function Generator, User-Configurable I/O, Variable Power Supplies, and DC Power Supplies Signal Rows
2	DIO Signal Rows	7	Power LEDs
3	LED Array	8	BNC Connectors
4	D-SUB Connector	9	Banana Jack Connectors
5	Counter/Timer, User-Configurable I/O, and DC Power Supply Signal Rows		

ภาพที่ ก 3 ตำแหน่งการใช้งานของบอร์ดต่อวงจรอุปกรณ์ NI ELVIS

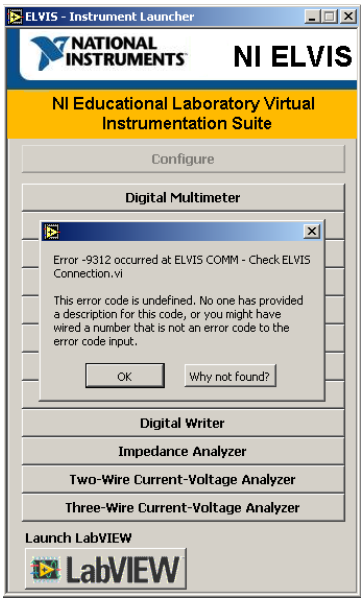
1. ช่องสำหรับการต่อใช้งานการรับสัญญาณอนาล็อค, ช่องสำหรับต่อใช้งานออส
2. ซิโลสโคป, ช่องสำหรับการต่อใช้งานฟังชั่นของโปรแกรม
3. ช่องสำหรับการต่อรับค่าสเตตัส/สั่งงานเปิดปิด ดิจิตอลอินพุต-ดิจิตอลเอาต์พุต
4. LED แสดงสถานะการทดลอง
5. พอร์ตต่ออนุกรม หรือ คอมพอร์ต D-SUB DB 9 Pin
6. ช่องสำหรับต่อใช้งานการนับ/การกำหนดเวลา, ช่องสำหรับเลือกใช้งาน อินพุต-เอาต์พุต ของ LED กับ D-SUB, ช่องสำหรับต่อใช้งานแรงดัน 5 VDC

7. ช่องสำหรับใช้งานมัลติมิเตอร์, ช่องสำหรับต่อใช้งานฟังก์ชั่นเจนเนอร์เรเตอร์, ช่องสำหรับเลือกใช้งาน อินพุต-เอาต์พุต บานานา กับ BNC, ช่องสำหรับต่อใช้งานการปรับแรงดันให้กับอุปกรณ์ +15, -15, +5 VDC
8. LED แสดงสถานะการต่อใช้งาน +15, -15, +5 VDC
9. พอร์ตเชื่อมต่อสายสัญญาณขั้วต่อแบบ BNC
10. พอร์ตเชื่อมต่อสายสัญญาณขั้วต่อแบบแจ็คบานานา

ภาคผนวก ข

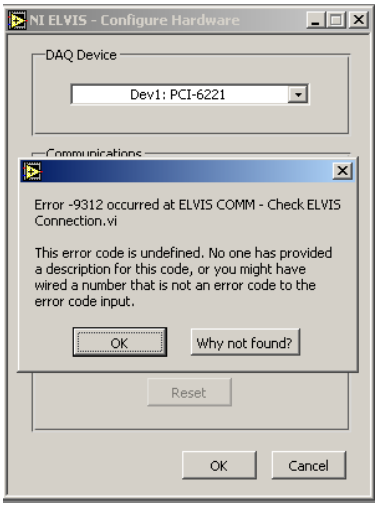
การปรับตัวโปรแกรม NI Traditional ของชุดทดลอง NI

1. เปิดโปรแกรม NI Traditional คลิกที่ ICON
2. จะปรากฏหน้าต่าง ELVIS – Instrument Launcher ขึ้นมา ให้ Click OK



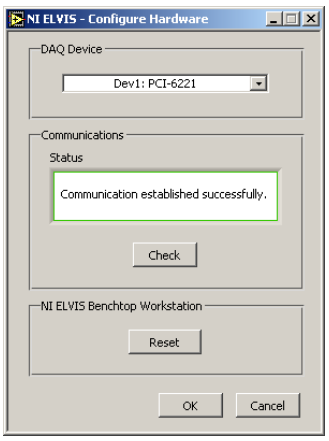
ภาพที่ ข1.1 หน้าต่าง ELVIS – Instrument Launcher ที่ยังไม่ Config กับโปรแกรม

3. จากนั้นคลิกที่ Configure จะปรากฏหน้าต่าง NI ELVIS – Configure Hardware ขึ้นมา ให้ Click OK



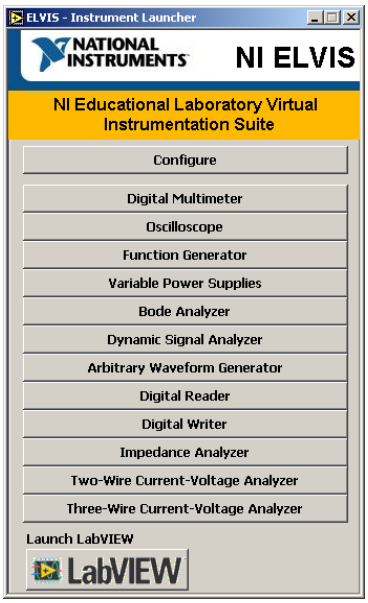
ภาพที่ ข1.2 หน้าต่าง NI ELVIS – Configure Hardware ที่ยังไม่ Config กับโปรแกรม

4. จากนั้น คลิกที่ Check รอสักครู่ โปรแกรมจะ Check Device Status โปรแกรมกับอุปกรณ์ NI ELVIS ว่าเชื่อมต่อกันหรือไม่ เมื่อเชื่อต่อเสร็จแล้วจะขึ้น Communication established successfully และ Click OK



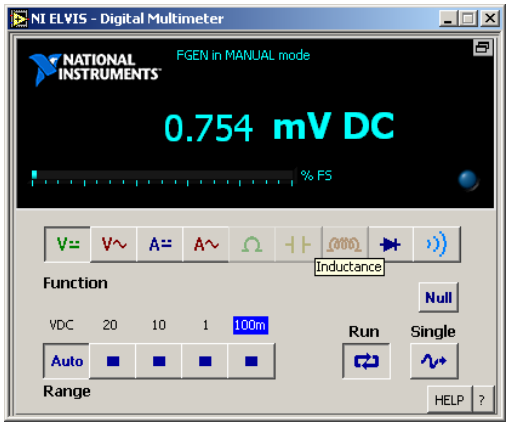
ภาพที่ ข1.3 หน้าต่าง NI ELVIS – Configure Hardware ที่ Config กับโปรแกรม

5. จากนั้นหน้าต่าง ELVIS – Instrument Launcher จะขึ้นให้เลือกฟังชั่นต่างๆ ที่สามารถใช้ได้ดังต่อไปนี้



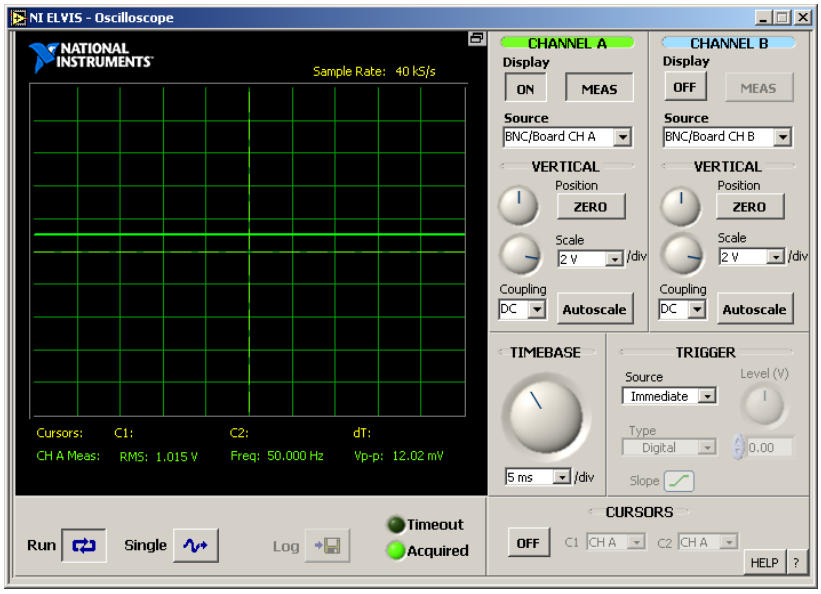
ภาพที่ ข1.4 หน้าต่าง ELVIS – Instrument Launcher ที่ Config กับโปรแกรม

5.1 Digital Mutimeter



ภาพที่ ข1.5 หน้าต่าง Digital Mutimeter

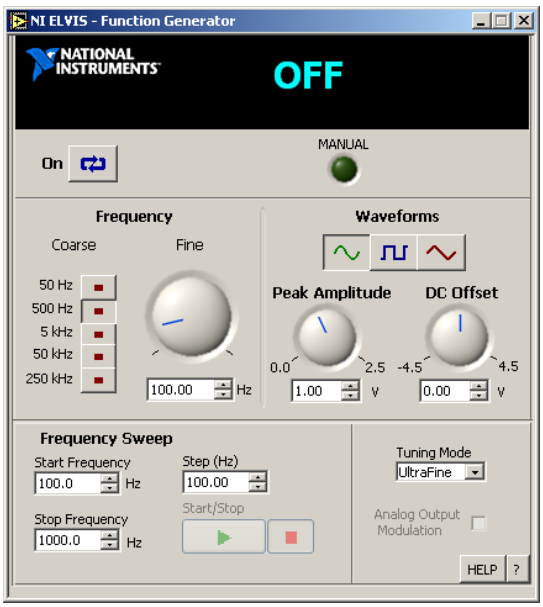
5.2 Oscilloscope



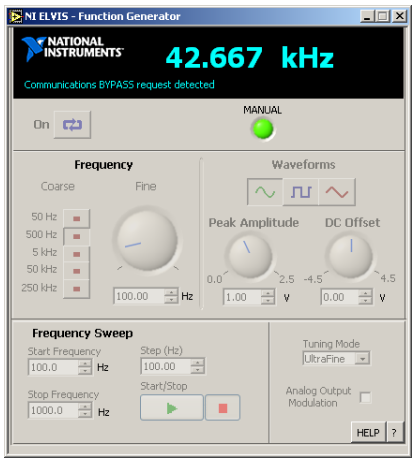
ภาพที่ ข1.6 หน้าต่าง Oscilloscope

5.2.1 Function Generator

สามารถเลือกกันปรับค่าได้ทั้งแบบ Auto และ Manual



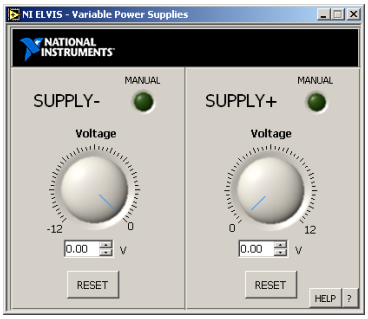
ภาพที่ ข1.7 หน้าต่าง Function Generator แบบ Auto



ภาพที่ ข1.8 หน้าต่าง Function Generator แบบ Manual

5.2.2 Variable Power Supplies

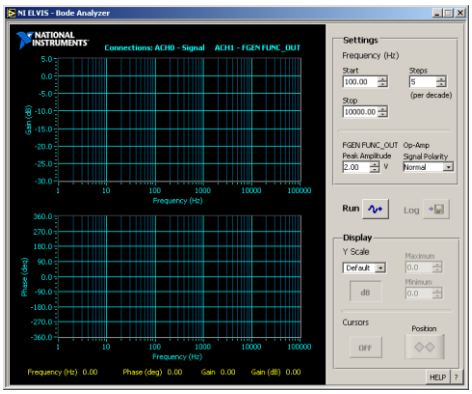
สามารถเลือกกันปรับค่าได้ทั้งแบบ Auto และ Manual



ภาพที่ ข1.9 หน้าต่าง Variable Power Supplies แบบ Auto

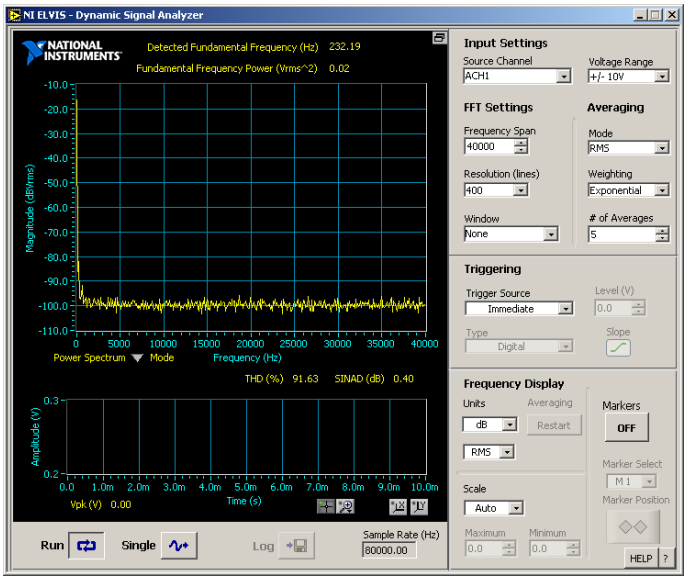
ภาพที่ ข1.10 หน้าต่าง Variable Power Supplies แบบ Manual

5.2.3 Bode Analyzer



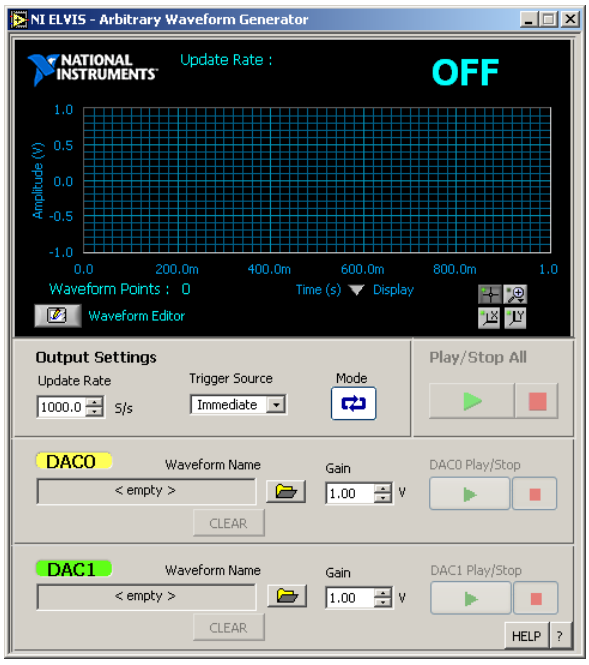
ภาพที่ ข1.11 หน้าต่าง Bode Analyzer

5.2.4 Dynamic Signal Analyzer



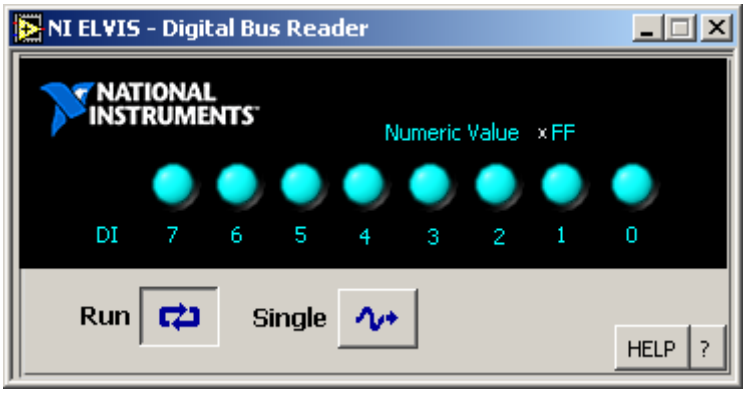
ภาพที่ ข1.12 หน้าต่าง Bode Analyzer

5.2.5 Arbitrary Waveform Generator



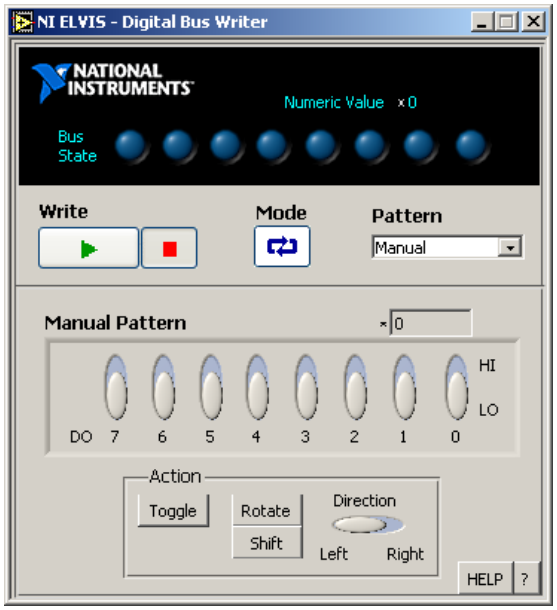
ภาพที่ ข1.13 หน้าต่าง Arbitrary Waveform Generator

5.2.6 Digital Bus Reader



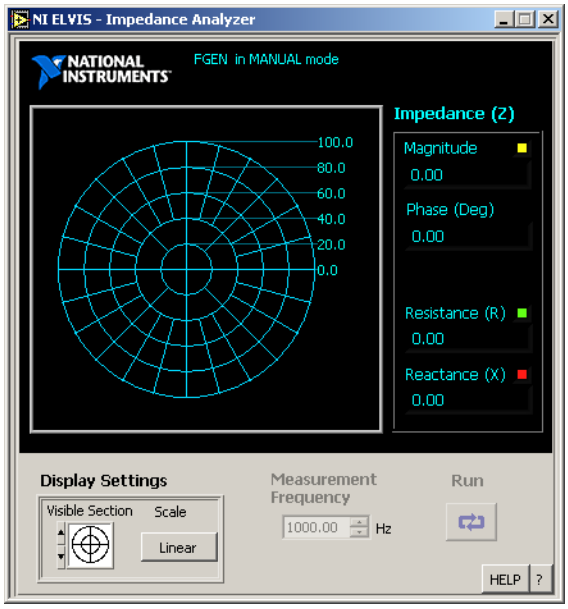
ภาพที่ ข1.14 หน้าต่าง Digital Bus Reader

5.2.7 Digital Bus Writer



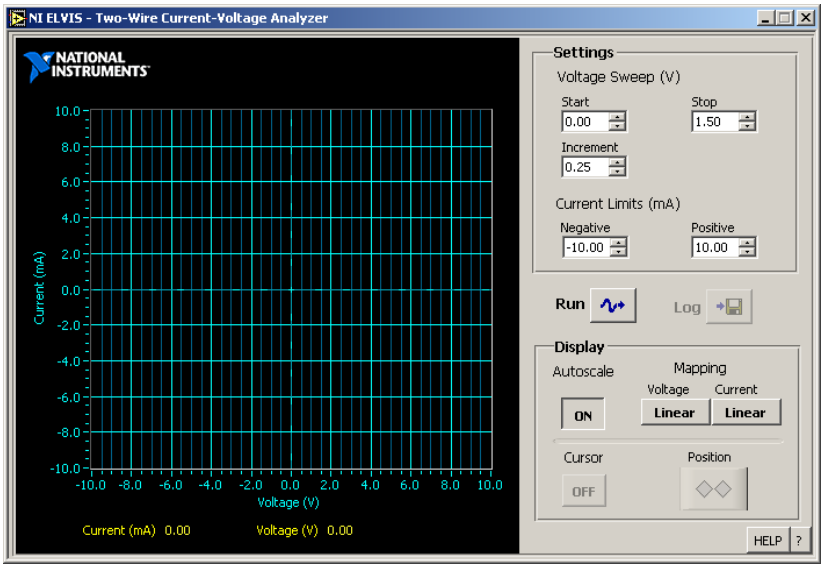
ภาพที่ ข1.15 หน้าต่าง Digital Bus Writer

5.2.8 Impedance Analyzer



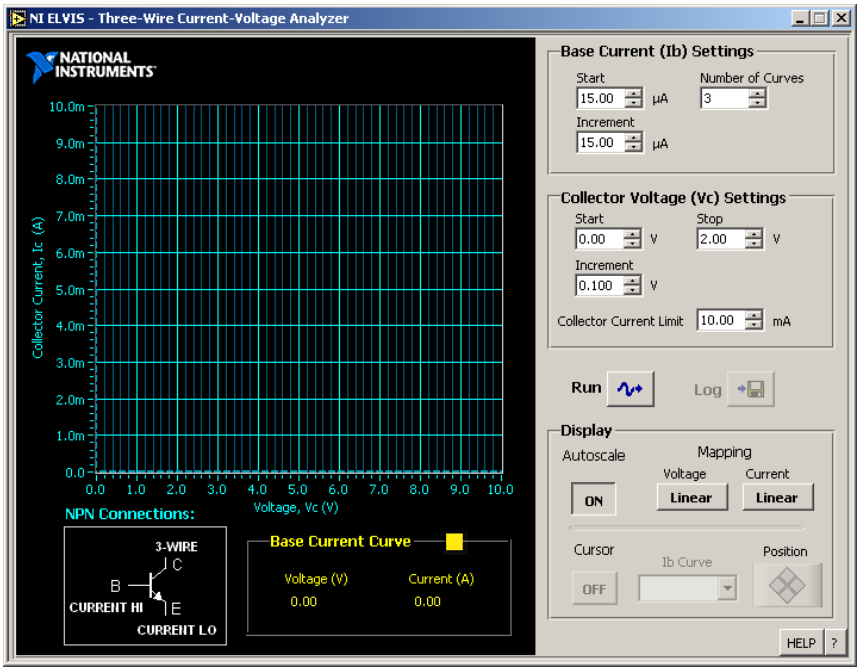
ภาพที่ ข1.16 หน้าต่าง Impedance Analyzer

5.2.9 Two-Wire Current-Voltage Analyzer



ภาพที่ ข1.17 หน้าต่าง Two-Wire Current_Voltage Analyzer

5.2.10 Three-Wire Current_Voltage Analyzer



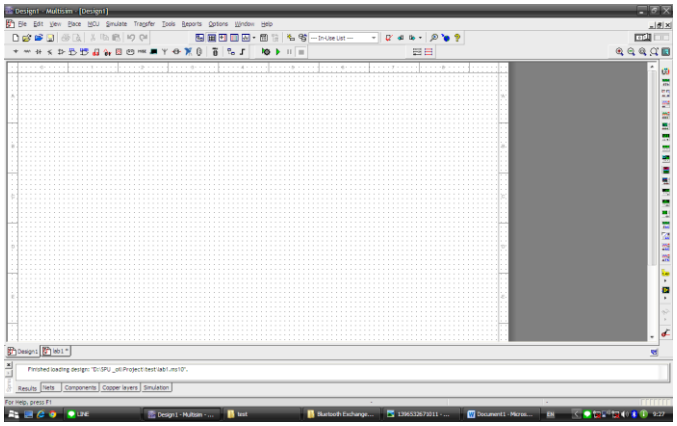
ภาพที่ ข1.18 หน้าต่าง Three-Wire Current-Voltage Analyzer

ภาคผนวก ค
การใช้งาน โปรแกรม NI Multisimเบื้องต้น

ตัวอย่างการใช้งานโปรแกรม NI Multisim แสดงด้วยการทดลองต่อวงจรเพื่อการแสดงผลลัพธ์ของวงจร ซึ่งมีขั้นตอนดังนี้

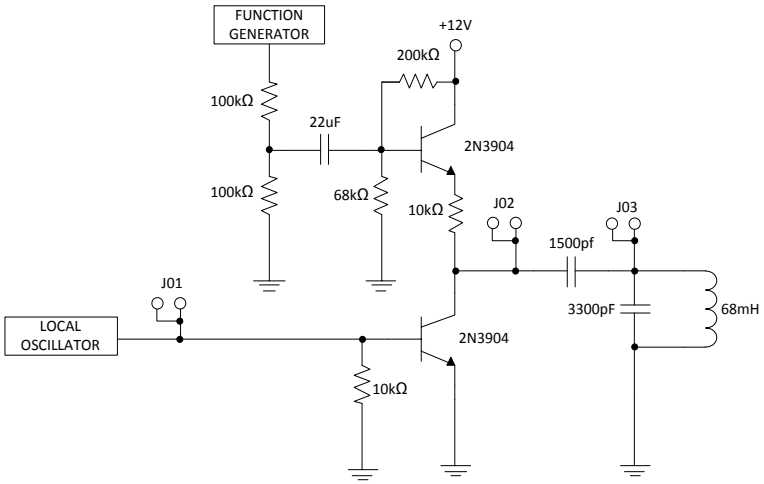
1. สร้างวงจรจำลองบนโปรแกรม NI Multisim
2. เปิดโปรแกรม NI Multisim คลิกที่ ICON

3. จะปรากฏหน้าต่าง Multisim ขึ้นมา



ภาพที่ ค1.1 หน้าต่างโปรแกรม NI Multisim

4. วาดวงจรแอมปลิจูดมอดูเลเตอร์ โดยมีค่าพารามิเตอร์ดังตารางที่ 1 ลงในโปรแกรม NI Multisim



ภาพที่ ค1.2 วงจรจำลอง

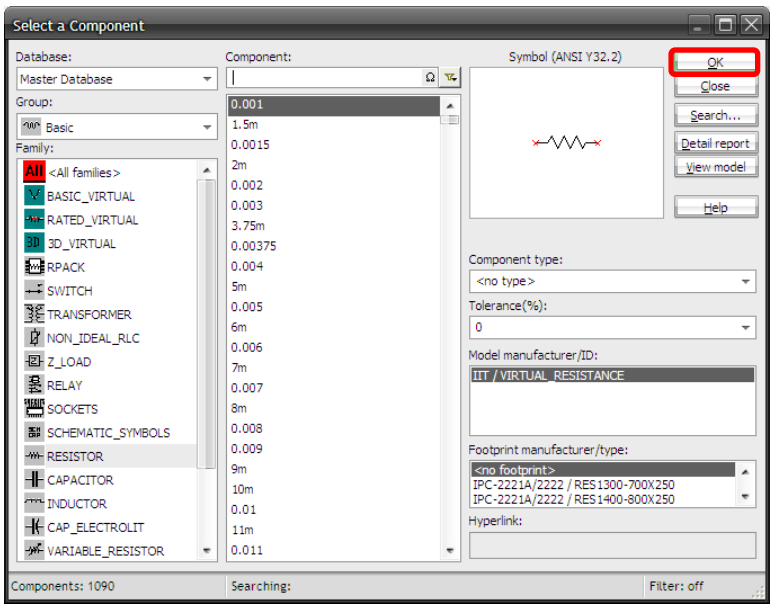
ตารางที่ ค1 รายการส่วนรายชื่ออุปกรณ์วงจรแอมปลิจูดมอดูเลเตอร์

Group	Family	ชื่ออุปกรณ์
Sources	POWER_SOURCES	GROUND
		VCC 12V
Basic	RESISTORS	100 kΩ 2 ตัว, 200 kΩ, 68 kΩ, 10 kΩ 2 ตัว
	CAPACITOR	22 uF, 1500 pF, 3300 pF
	INDUCTOR	50 mH
Transistors	All families	2N3904

5. คลิกที่ปุ่ม เพื่อเข้าไปเลือกอุปกรณ์ จะพบหน้าต่าง Select a Component Pick Devices

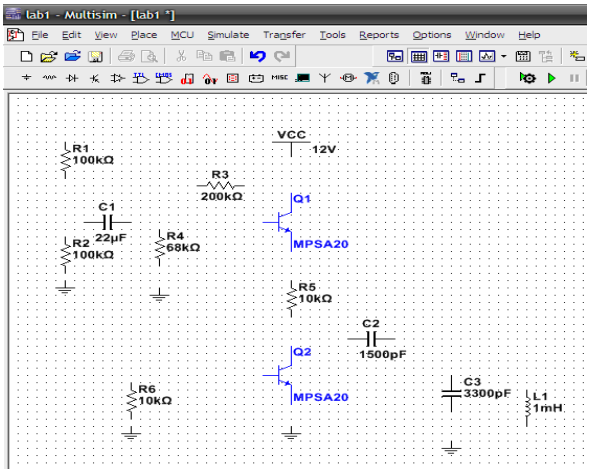
6. เลือกอุปกรณ์ หรือพิมพ์ชื่ออุปกรณ์ลงในช่อง Component และอุปกรณ์ก็จะแสดงขึ้นมา

7. คลิกปุ่ม OK



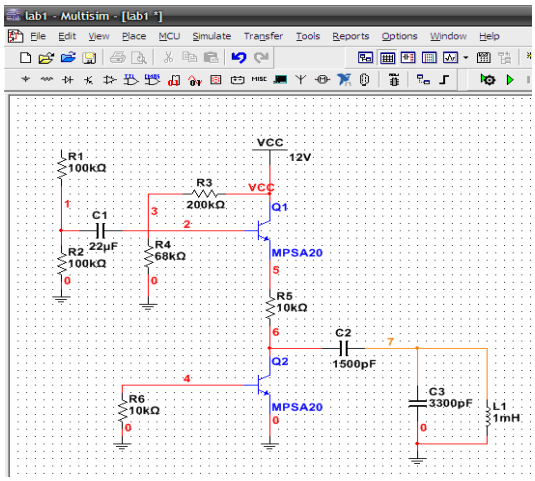
ภาพที่ ค1.3 หน้าต่าง Select a Component Pick Devices

8. คลิกเมาส์หนึ่งครั้งบนพื้นที่ทำงาน เพื่อวางอุปกรณ์ลงไปในตำแหน่งที่ต้องการ วางอุปกรณ์ทุกตัวลงไป แล้วจัดเรียงตามที่ต้องการ ดังภาพที่ ค1.4



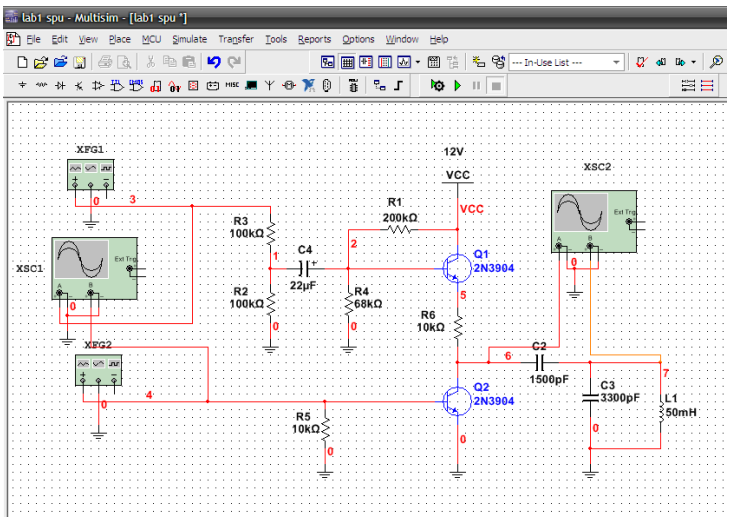
ภาพที่ ค1.4 การวางอุปกรณ์ต่างๆ บนพื้นที่ทำงาน

9. นำเมาส์มาชี้ที่ขาที่อุปกรณ์จะเห็นว่า มีจุดวงกลมสีดำเล็กๆ เกิดขึ้น ให้คลิกลงไปหนึ่งครั้ง เพื่อทำการเชื่อมต่อลากสายสัญญาณมาเชื่อมต่อยังอีกขาหนึ่ง ก็จะปรากฏจุดวงกลมสีแดงเล็กๆ ขึ้นมา ให้คลิกลงไปหนึ่งครั้ง จะปรากฏเส้นสีแดง ก็จะทำให้ขาอุปกรณ์ทั้งสองเชื่อมต่อถึงกันโดยสมบูรณ์ จากนั้นทำการเชื่อมต่อสายสัญญาณ และกำหนดค่าอุปกรณ์ให้ครบตามวงจร ก็จะได้ดังภาพที่ ค1.5



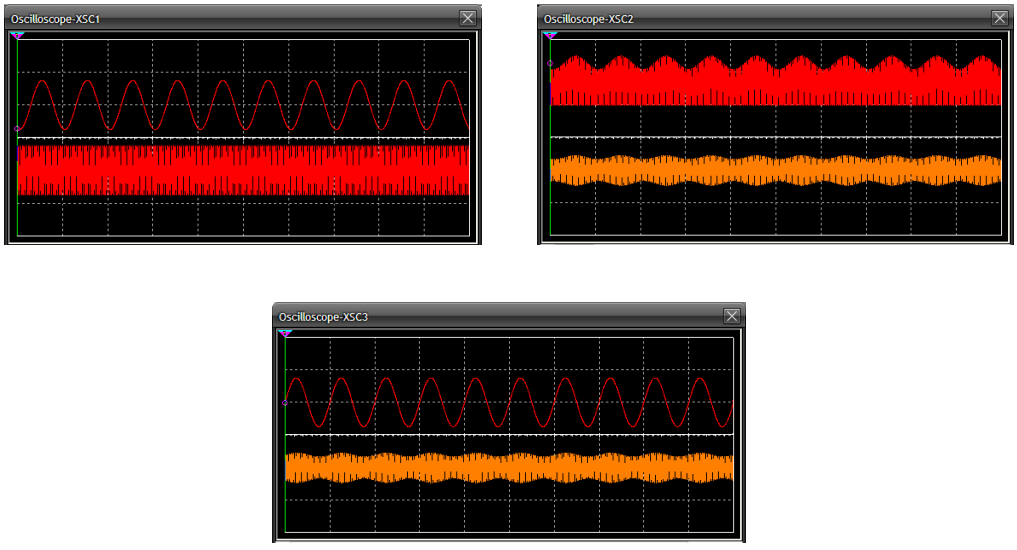
ภาพที่ ค1.5 การเชื่อมต่อสายสัญญาณ

10. เมื่อลากสายสัญญาณเสร็จแล้ว ให้คลิกที่ปุ่ม Function Generator และ Oscilloscope ทางด้านขวา และเชื่อมต่อสายสัญญาณ ดังภาพที่ ค1.6



ภาพที่ ค1.6 การเชื่อมต่อสายสัญญาณ Function Generator และ Oscilloscope

11. คลิกที่ปุ่ม Run เพื่อ Run Program
12. คลิกที่ Function Generator เพื่อปรับค่าต่างๆ ตามต้องการ
13. คลิกที่ Oscilloscope จะปรากฏสัญญาณ ดังภาพที่ ค1.7



ภาพที่ ค1.7 สัญญาณต่างๆ ที่ Oscilloscope

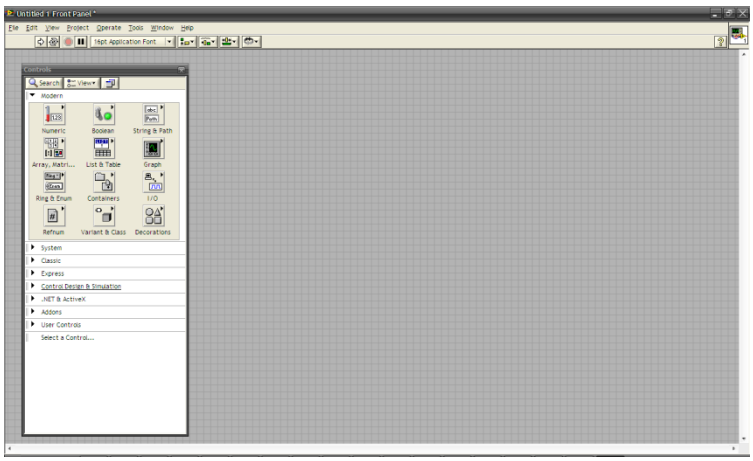
14. จากนั้นคลิกที่ปุ่ม STOP เพื่อหยุด Program
15. คลิกที่ปุ่ม เพื่อบันทึกเก็บไฟล์ตามต้องการ

ภาคผนวก ง
การใช้งานโปรแกรม LabVIEW เบื้องต้น

LabVIEW เป็นโปรแกรมที่ใช้ เพื่อสร้างโปรแกรมสำหรับการเก็บข้อมูล , การเชื่อมต่อกับอุปกรณ์หรือเครื่องมือวัดต่างๆ และการควบคุมระบบต่างๆ LabVIEW จะใช้ภาษากราฟฟิกในการสร้างโปรแกรม ซึ่งแตกต่างจากโปรแกรมอื่นๆ ที่ใช้ตัวอักษรเพื่อสร้างโปรแกรม เช่น C/C++ , Visual C++ , Visual Basic , etc.

โปรแกรม LabVIEW มีฟังก์ชัน (Functions) และเครื่องมือ (Tools) ดังภาพที่ 1 เพื่อช่วยให้ผู้ใช้สามารถนำมาสร้างโปรแกรมสำหรับงานประยุกต์ต่างๆ เช่น loops , case statements , arrays , string , file I/O , data acquisition , instrument control , analysis tools เป็นต้น โปรแกรม LabVIEW โดยปกติจะเรียกว่า Virtual Instrument (VI) ซึ่งประกอบด้วย 2 ส่วนคือ Front Panel และ Block Diagram โดยแต่ละส่วนจะใช้สำหรับวัตถุประสงค์ที่แตกต่างกัน

1. Front Panel

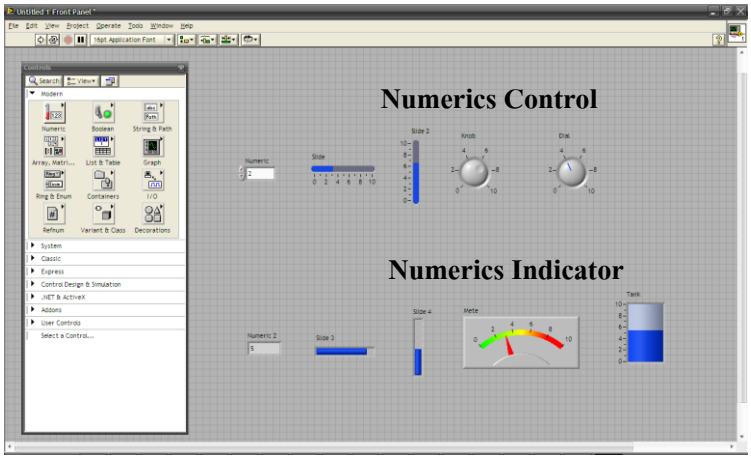


ภาพที่ ง1.1 ภาพที่ 1 Front Panel

Front Panel เป็นส่วนที่ใช้สำหรับเชื่อมต่อกับผู้ใช้งานใช้สำหรับใส่ค่า (input) และแสดงผล (output) ของตัวโปรแกรมที่สร้างขึ้นมา โดยส่วน input จะถูกเรียกว่า control และส่วน output จะเรียกว่า indicator ตัว control และ indicator ที่ถูกนำมาใช้ใน Front Panel ซึ่งจะมีจุดต่อเชื่อมปรากฏอยู่ที่ Block Diagram ด้วย เมื่อโปรแกรมเริ่มทำงานตัว control ที่ Front Panel จะทำการส่งข้อมูลผ่านไปยัง Block Diagram และตัว output ก็จะส่งค่าจาก Block Diagram กลับมาแสดงผลที่ Front Panel ผ่านตัว indicator ที่กำหนดไว้ ข้อมูลที่ใช้ในตัว control และ indicator มีอยู่ด้วยกันหลายรูปแบบ เช่น ตัวเลข (Numerics), เงื่อนไข (Booleans) และตัวอักษร (Strings)

1.1 Numerics

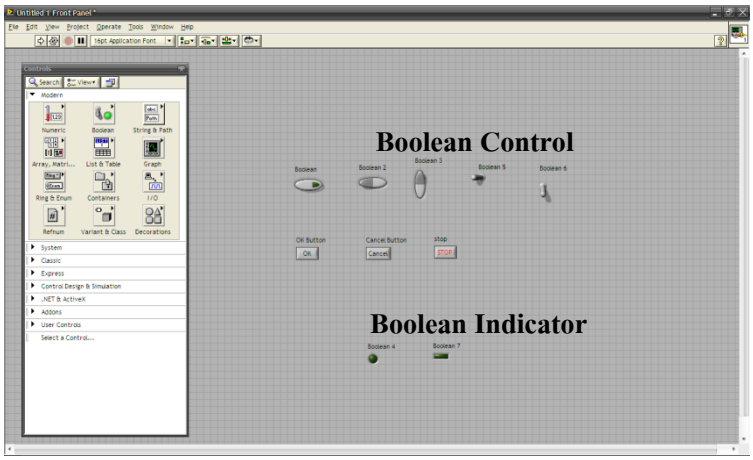
มีอยู่ด้วยกันหลายรูปแบบ เช่น ตัวเลขดิจิตอล, เกจ, หรือสเกลแบบต่างๆ ซึ่งฟังก์ชันสามารถเข้าได้จากไอคอน Numerics ดังภาพที่ ง1.2



ภาพที่ ง1.2 ตัวอย่างฟังก์ชัน Control และ Indicator แบบ Numerics ใน Front Panel

1.2 Booleans

เป็นเงื่อนไขมี 2 สถานะคือ on/off หรือ True/False รูปแบบของ Boolean จะเป็น ปุ่มสี , ปุ่มกดหยุด หรือปุ่มเลื่อนเปิดหรือปิด ซึ่งฟังก์ชันสามารถเข้าได้จากไอคอน Boolean ดังภาพที่ ง1.3

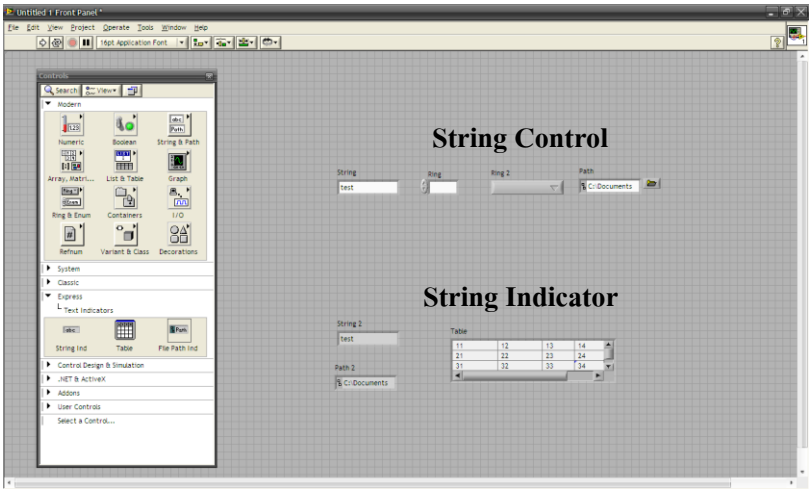


ภาพที่ ง1.3 ตัวอย่างฟังก์ชัน Control และ Indicator แบบ Boolean ใน Front Panel

1.3 String

เป็นตัวอักษรที่ใช้สำหรับป้อนค่า , แสดงผล หรือจัดเก็บไว้ในรูปของไฟล์ ซึ่งฟังก์ชันสามารถเข้าได้จากไอคอน String

ดังภาพที่ ง1.4

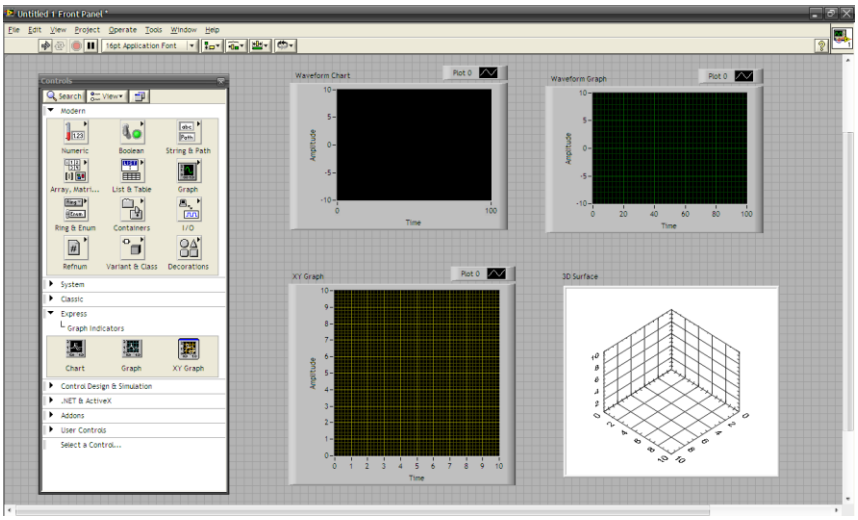


ภาพที่ ง1.4 ตัวอย่างฟังก์ชัน Control และ Indicator แบบ Strings ใน Front Panel

1.4 Charts/Graphs

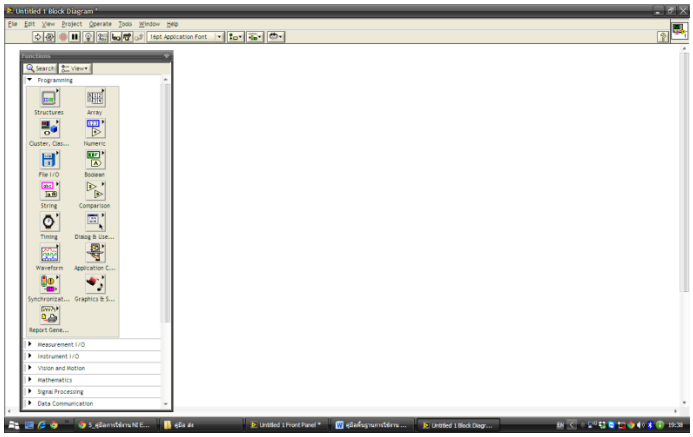
เป็น indicator ที่ใช้แสดงข้อมูลเทียบกับเวลาซึ่งมีทั้งแบบ 2 มิติ และ 3 มิติ ซึ่งฟังก์ชันสามารถเข้าได้จากไอคอน Graphs

ดังภาพที่ ง1.5



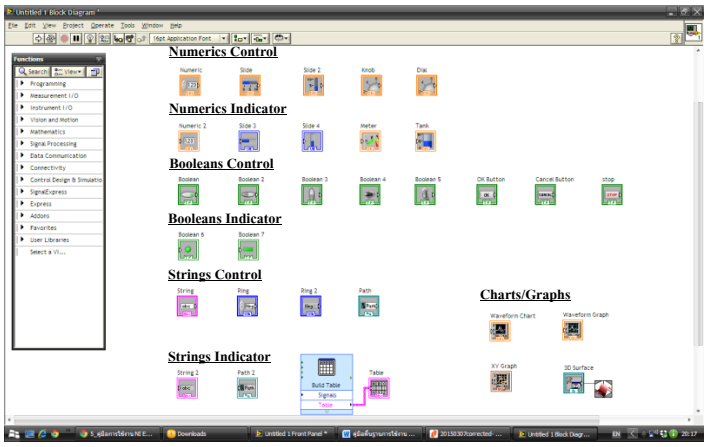
ภาพที่ ง1.5 ตัวอย่างฟังก์ชัน Charts/Graphs แบบต่างๆ ใน Front Panel

2. Block Diagram



ภาพที่ ง1.6 Block Diagram

Block Diagram เป็นส่วนที่เก็บ Source Code ของโปรแกรม LabVIEW ซึ่งตัวโปรแกรมใน LabVIEW จะเรียกว่า VI ตัว Code ในโปรแกรม LabVIEW เป็นกราฟฟิกที่เรียกกันว่า G (Graphical) programming หลักการของโปรแกรมจะเชื่อมต่อตัวจุดเชื่อมต่างๆ เข้าด้วยกัน แทนที่จะเขียนโดยใช้คำสั่งต่างๆ ดังที่ใช้ทั่วไปในโปรแกรมอื่นๆ เช่น C/C++ , Visual C++ ซึ่งอาจจะกล่าวได้ว่า LabVIEW ใช้หลักการเดียวกันกับการเขียน flow chart ตัวอย่างของ Numerics , Booleans , Strings และCharts/Graphs ใน Block Diagram มีรูปแบบแตกต่างกันดังภาพที่ ง1.7



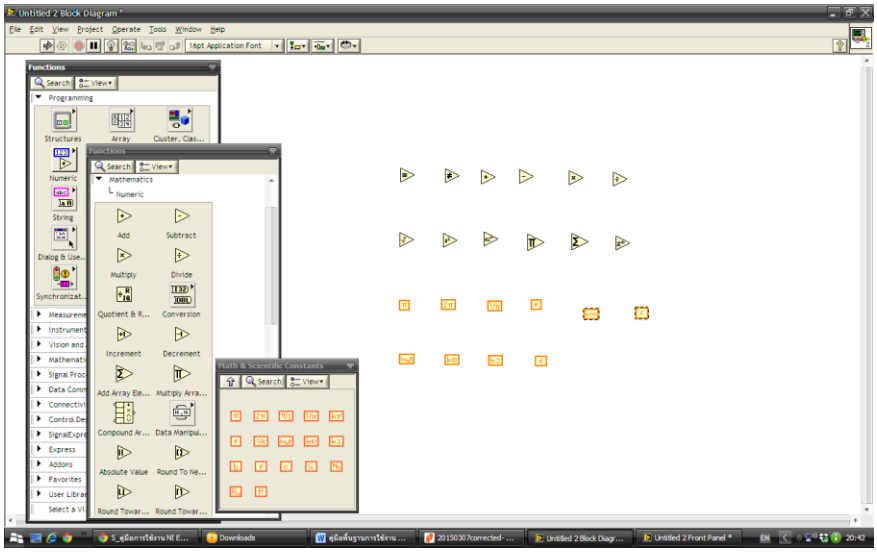
ภาพที่ ง1.7 ตัวอย่างฟังก์ชัน Numerics, Booleans, Strings และ Charts/Graphs ใน Block Diagram

โปรแกรม LabVIEW มีฟังก์ชั่นต่างๆ ที่ใช้ช่วยในการพัฒนาโปรแกรมที่มีความซ้ำซ้อน เช่น ตัว operation (+ , - , * . / , ^ เป็นต้น) ค่าคงที่ (π , e , ln2 , log10 เป็นต้น) ฟังก์ชันมาตรฐานต่างๆ (sin , cos , tan , atan เป็นต้น) Loop & Case Structures (while loop , for loop , case เป็นต้น) และฟังก์ชัน File input/output (File I/O) ฟังก์ชันต่างๆ ที่กล่าวถึงจะปรากฏอยู่ใน Block diagram เท่านั้น

2.1 Operation

เป็นฟังก์ชันที่ใช้ช่วยเพื่อสร้างสมการหรือคำสั่งในโปรแกรม LabVIEW ตามวัตถุประสงค์ที่กำหนดไว้ ซึ่งฟังก์ชันสามารถเข้าได้จากไอคอน Numeric

ดังภาพที่ ง1.8

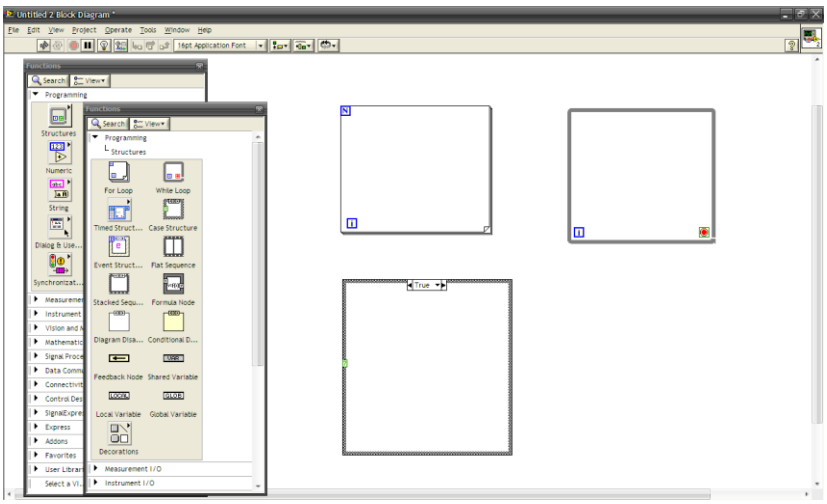


ภาพที่ ง1.8 ตัวอย่างฟังก์ชัน Operation ต่างๆ ใน Block Diagram

2.2 Loop & Case Structures

เป็นฟังก์ชันพื้นฐานสำหรับการเขียนโปรแกรม ซึ่งมีไว้ช่วยสำหรับการเขียนโปรแกรมที่ต้องมีขั้นตอนการทำงานซ้ำหลายๆ ครั้ง โดยรูปแบบของฟังก์ชันขึ้นอยู่กับลักษณะของกระบวนการ เช่น While Loop กระบวนการจะทำซ้ำจนกระทั่งบรรลุตามเงื่อนไขที่กำหนดไว้ เป็น ForLoop ตัวโปรแกรมจะกระทำซ้ำตามจำนวนครั้งที่กำหนด และในกรณีของ Case Structure ตัวโปรแกรมจะกระทำเมื่อค่าเริ่มต้นตรงตาม ข้อกำหนดของเงื่อนไขนั้นๆ ซึ่งฟังก์ชันสามารถเข้าได้จากไอคอน Structures

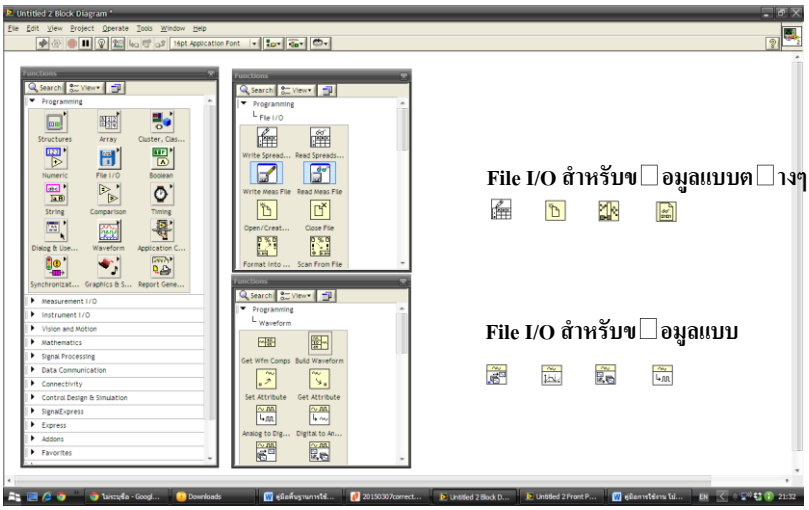
ดังภาพที่ ง1.9



ภาพที่ ง1.9 ตัวอย่างฟังก์ชัน Loop & Case Structures ใน Block Diagram

2.3 File Input/Output (I/O)

เป็นฟังก์ชัน ทีสามารถนำมาใช้เพื่อจัดเก็บข้อมูลในรูปของไฟล์ ทั้งแบบ Binary และ ASCII (text) Format โดยสามารถนำมาวิเคราะห์ และจัดทำรายงานหรือกราฟในภายหลังได้ ฟังก์ชัน File I/O ประกอบด้วยฟังก์ชันอ่าน (Read) ข้อมูลจากไฟล์ และจัดเก็บ (Write) ข้อมูลในรูปของไฟล์ ส่วนวิธีการใช้ฟังก์ชันขึ้นอยู่กับชนิดของข้อมูล (String , Array , Integer , Cluster เป็นต้น) และ format ของไฟล์ (binary หรือ ASCII) ที่ ต้องการอ่านหรือจัดเก็บ ซึ่งฟังก์ชันสามารถเข้าได้จากไอคอน File I/O และ Waveform ดังภาพที่ ง1.10



ภาพที่ ง1.10 ตัวอย่างฟังก์ชัน File I/O ใน Block Diagram

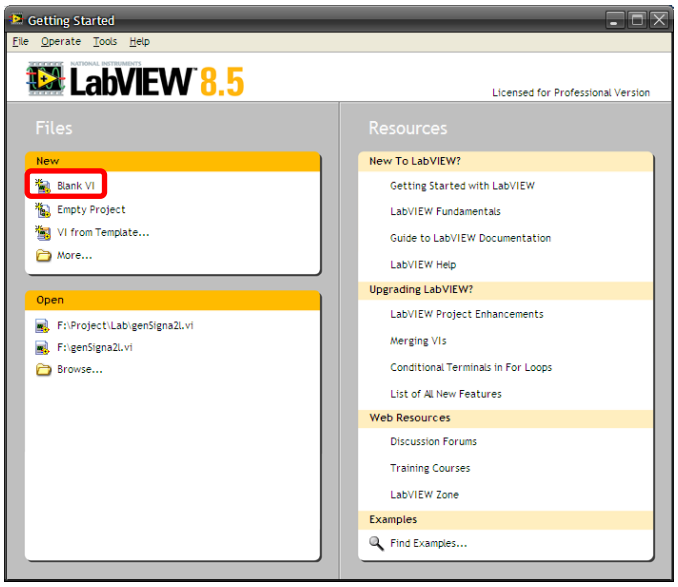
3. ตัวอย่างการเขียนโปรแกรม LabVIEW

การเขียนโปรแกรม LabVIEW นั้นขึ้นอยู่กับวัตถุประสงค์ที่ต้องการใช้งานซึ่ง Front Panel และ Block Diagram จะถูกใช้ร่วมกันเพื่อเขียนโปรแกรมโดย Front Panel จะใช้เพื่อแสดงผล และติดต่อกับผู้ใช้ ส่วน Block Diagram จะใช้เพื่อเขียนและแสดงเส้นทางเดินของข้อมูลรวมทั้งเป็นตัวกำหนดค่าตัวแปรเริ่มต้นหรือตัวแปรควบคุมต่างๆ ที่จำเป็นต้องใช้ในโปรแกรม

ตัวอย่างการใช้งานโปรแกรม LabVIEW เบื้องต้น แสดงด้วยการทดลองฟังก์ชันเพื่อการแสดงผลลัพธ์ของฟังก์ชัน ซึ่งมีขั้นตอนดังนี้

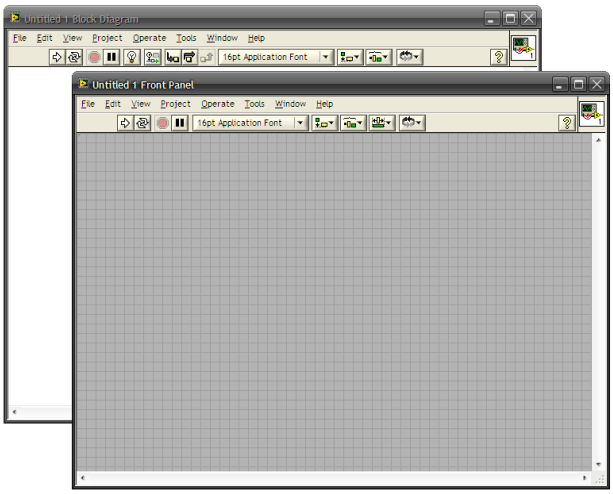
3.1 เปิดโปรแกรม LabVIEW 8.5 คลิกที่ ICON

3.2 จากนั้นจะปรากฏหน้าต่าง Getting Started ขึ้นมา ให้คลิกที่ Blank VI



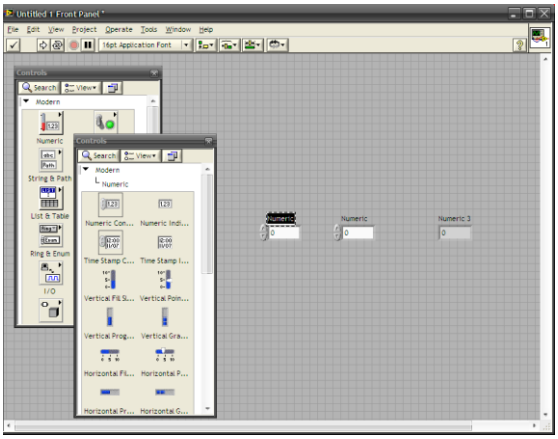
ภาพที่ ง1.11 หน้าต่างโปรแกรม LabVIEW 8.5

3.3 จากนั้นจะปรากฏหน้าต่าง Front Panel และ Block Diagram



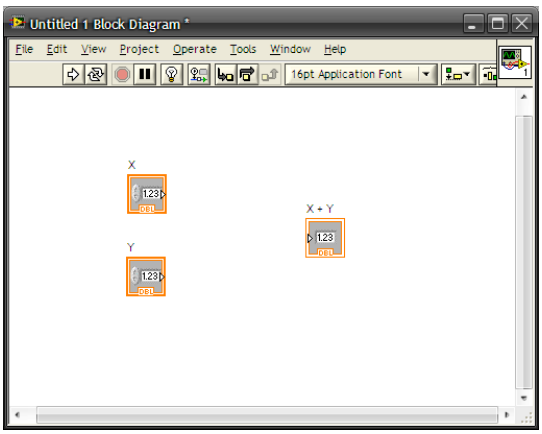
ภาพที่ ง1.12 หน้าต่าง Front Panel และ Block Diagram ของโปรแกรม LabVIEW 8.5

3.4 จากนั้นคลิกขวาที่หน้าต่าง Front Panel จะปรากฏหน้าต่าง Tool Bar Controls จากนั้นให้คลิกขวาที่ไอคอน Numeric จะปรากฏหน้าต่าง Tool Bar Controls ขึ้นมาอีก จากนั้นให้คลิกซ้ายค้างไว้ที่ไอคอน Numerics Control , Numerics Indicator และลากไปวางไว้ที่หน้า Front Panel จากนั้นเปลี่ยนชื่อที่จะใช้แสดงโดยการดับเบิ้ลคลิกที่ชื่อของฟังก์ชัน ดังภาพที่ ง1.13



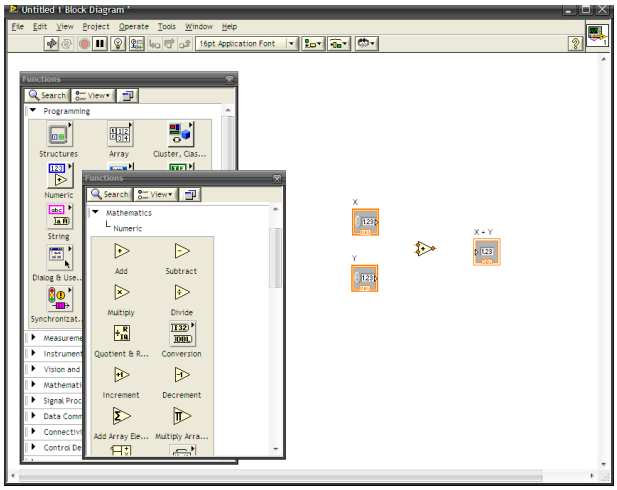
ภาพที่ ง1.13 แสดงการเรียกฟังก์ชัน ที่ใช้ในการเขียนโปรแกรมบวกเลข ใน Front Panel

3.5 จากนั้นให้ดับเบิ้ลคลิกที่ตัวฟังก์ชัน ตัวใดตัวหนึ่ง แล้วโปรแกรมจะเด้งไปที่หน้าต่างของ Block Diagram ดังภาพที่ 14



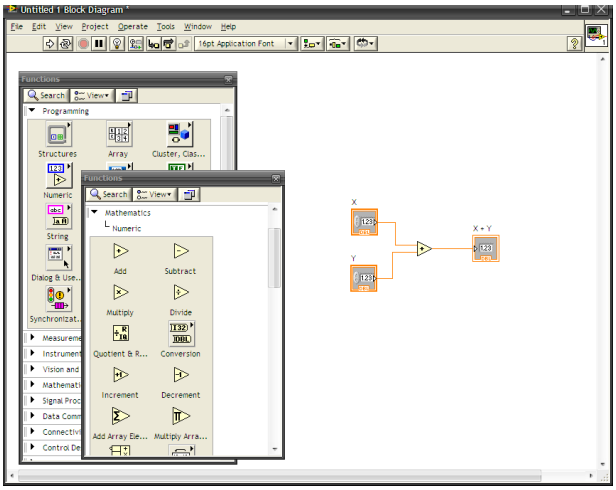
ภาพที่ ง1.14 แสดงฟังก์ชัน ที่ใช้ในการเขียนโปรแกรมบวกเลข ใน Block Diagram

3.6 จากนั้นคลิกขวาที่หน้าต่าง Block Diagram จะปรากฏหน้าต่าง Tool Bar Funtion จากนั้นให้คลิกขวาที่ไอคอน Numeric จะปรากฏหน้าต่าง Tool Bar Numeric ขึ้นมา แล้วคลิกซ้ายค้างไว้ที่ไอคอน Add และลากไปวางไว้ที่หน้า Block Diagram และลองใช้เมาส์ชี้ที่ตัวฟังก์ชัน จะเห็นว่ามีจุดโหนดแสดงขึ้นมา ดังภาพที่ ง1.15



ภาพที่ ง1.15 แสดงฟังก์ชัน ที่ใช้ในการเขียนโปรแกรมบวกเลข ใน Block Diagram

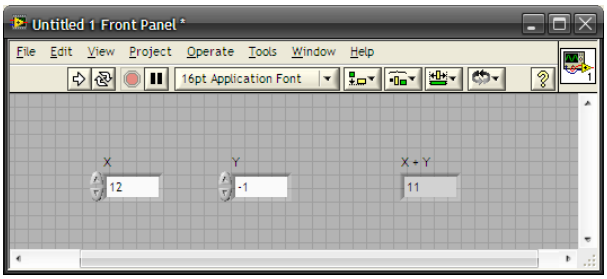
3.7 จากนั้นคลิกซ้ายที่จุดโหนดต่างๆ เพื่อเชื่อมต่อฟังก์ชันเข้าด้วยกัน ดังภาพที่ ง1.16



ภาพที่ ง1.16 แสดงการเชื่อมต่อจุดโหนด ของฟังก์ชันเข้าด้วยกันใช้ในการเขียนโปรแกรมบวกเลข ใน Block Diagram

3.8 เมื่อทำการเชื่อมต่อจุดต่างๆ เสร็จเรียบร้อย ให้คลิกที่ เพื่อทำการรันโปรแกรม

3.9 จากนั้นลองทำการปรับลูกศรขึ้นลงที่ฟังก์ชัน สังเกตที่ผลลัพธ์ว่าเป็นไปตามที่ได้เขียนโปรแกรมหรือไม่ ดังภาพที่ ง1.17



ภาพที่ ง1.17 ตัวอย่างโปรแกรมบวกเลข

ภาคผนวก จ

DATA SHEET