

## PROCESS MEASUREMENT &amp; MONITORING LABOLATORY 1

## การใช้งานโปรแกรม LabVIEW

## วัตถุประสงค์

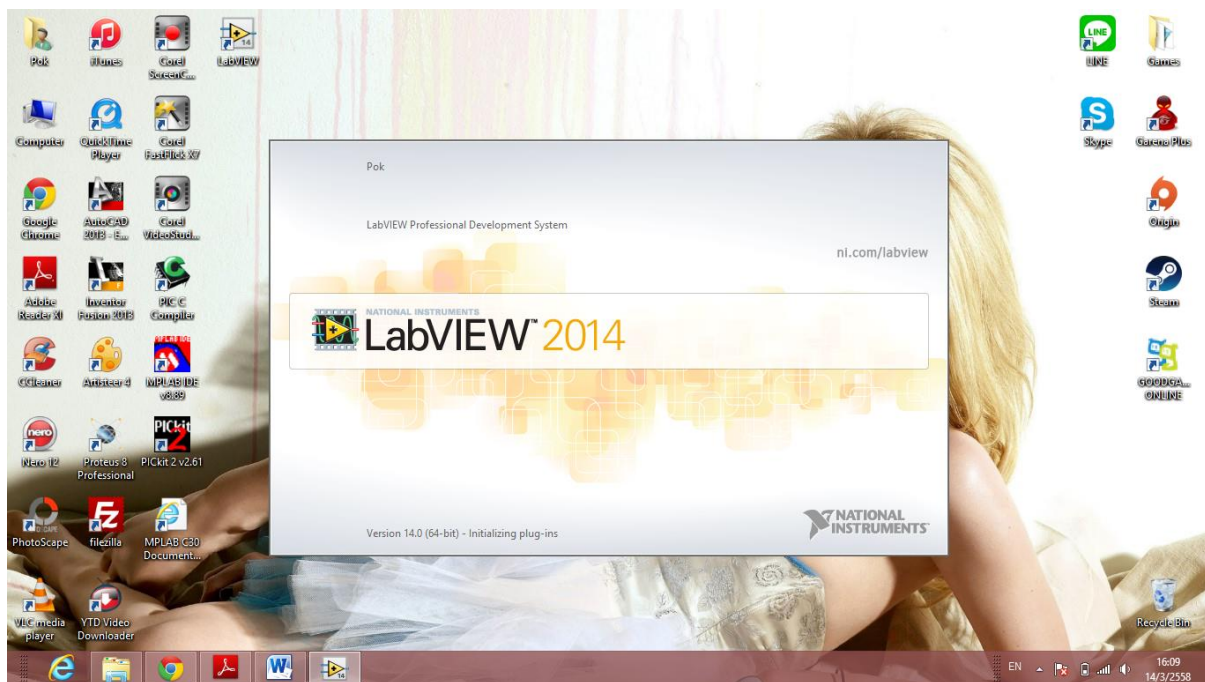
1. ศึกษาการใช้งานโปรแกรม LabVIEW 2014®
2. สามารถประยุกต์ใช้งานฟังก์ชันในโปรแกรม LabVIEW 2014® ได้

## อุปกรณ์ที่ใช้การทดลอง

1. โปรแกรม LabVIEW 2014®
2. Notebook

## ทฤษฎีที่เกี่ยวข้อง

## โปรแกรม LabVIEW 2014®



LabVIEW ย่อมาจาก Laboratory Virtual Instrument Engineering Workbench

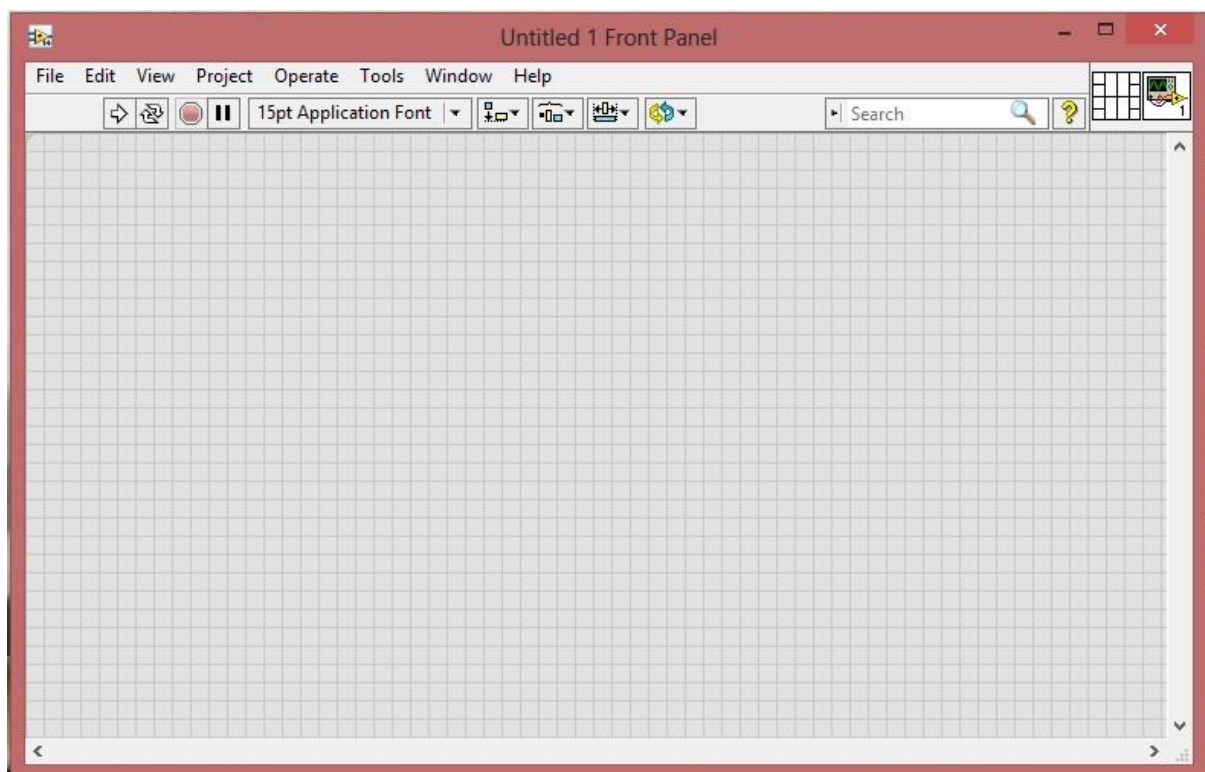
เป็นโปรแกรม คอมพิวเตอร์ที่สร้างเพื่อนำมาใช้ในด้านการจัดการวัดและเครื่องมือวัดสำหรับงานทางวิศวกรรม ซึ่งเป็นโปรแกรมประเภท GUI (Graphic User Interface) นั่นคือผู้ใช้พัฒนาโปรแกรมไม่จำเป็นต้องเขียน code หรือ

คำสั่งใดๆ ทั้งสิ้น และภาษาที่ใช้ในโปรแกรมจะเรียกว่าเป็นภาษารูปภาพหรือเรียกอีกอย่างว่าภาษา G (Graphical Language) ซึ่งจะแทนการเขียนโปรแกรมเป็นบรรทัดภาษาพื้นฐานเช่น C, BASIC หรือ FORTRAN ด้วยรูปภาพ

หรือสัญลักษณ์ทั้งหมดโดยจะช่วยอำนวยความสะดวกและสามารถลดเวลาในการเขียนโปรแกรมลงไปได้มากโดยเฉพาะในงานเขียนโปรแกรมคอมพิวเตอร์เพื่อเชื่อมต่อกับอุปกรณ์ภายนอกเช่น Port หรือ Card ต่างๆรวมถึงการจัดวางตำแหน่งในหน่วยความจำเพื่อที่จะสามารถรวบรวมข้อมูลมาใช้ในการคำนวณและเก็บข้อมูลให้ได้ประโยชน์สูงสุด

สำหรับโปรแกรมที่พัฒนาขึ้นโดย LabVIEW จะเรียกว่า Virtual Instrument (VI) เพราะลักษณะที่ปรากฏทางจอภาพเมื่อเริ่มใช้งานจะเหมือนกับเครื่องมือหรืออุปกรณ์ทางวิศวกรรมในขณะเดียวกันหลังจากของอุปกรณ์เสมือนจริงเหล่านั้นจะเป็นการทำงานของฟังก์ชันต่างๆซึ่งในหนึ่ง Virtual Instrument (VI) จะประกอบด้วย ส่วนประกอบ 2 ส่วนคือ

1. **Front Panel** เป็นส่วนตั้งค่าการวัดและอ่านค่าตัวเลขหรือกราฟที่ออกมาจากblock diagram จึงทำหน้าที่เสมือนเครื่องมือวัดจริงโดย input ที่ป้อนเข้าไปจะเป็นตัวควบคุม ส่วน output ที่ออกมาจะเป็นตัวแสดงผล



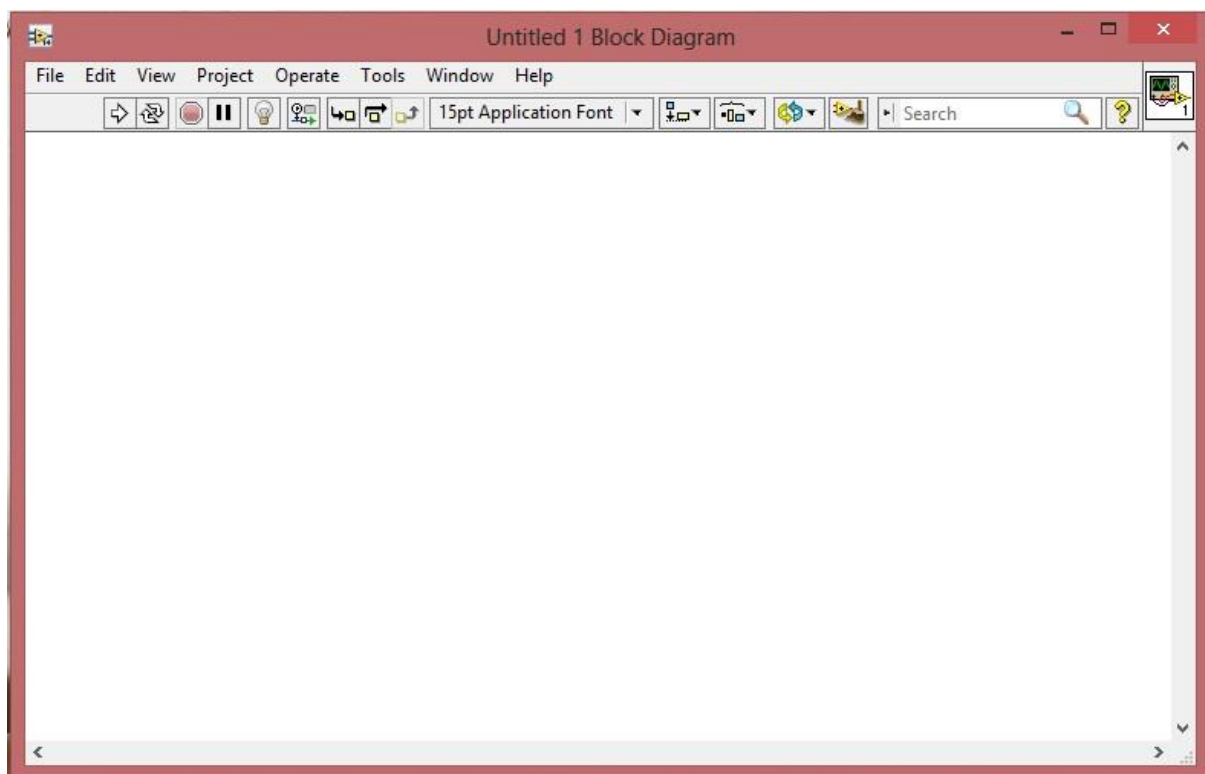
Front Panel

สำหรับหน้า **Front Panel** ของ **LabVIEW** จะมีส่วนประกอบที่สำคัญ 2 แบบคือ

1.) **Controls** มีหน้าที่เป็นตัวควบคุมคือการใส่ค่า **Input** จากผู้ใช้ลักษณะของ **Controls** เช่น ปุ่มปรับค่า, สะพานเปิด-เปิดไฟ, แท่งเลื่อนเพื่อปรับค่า, การให้ค่าด้วยตัวเลข **Digital** หรืออื่นๆ ดังนั้น **Controls** คือการกำหนดค่าหรือแหล่ง (source)

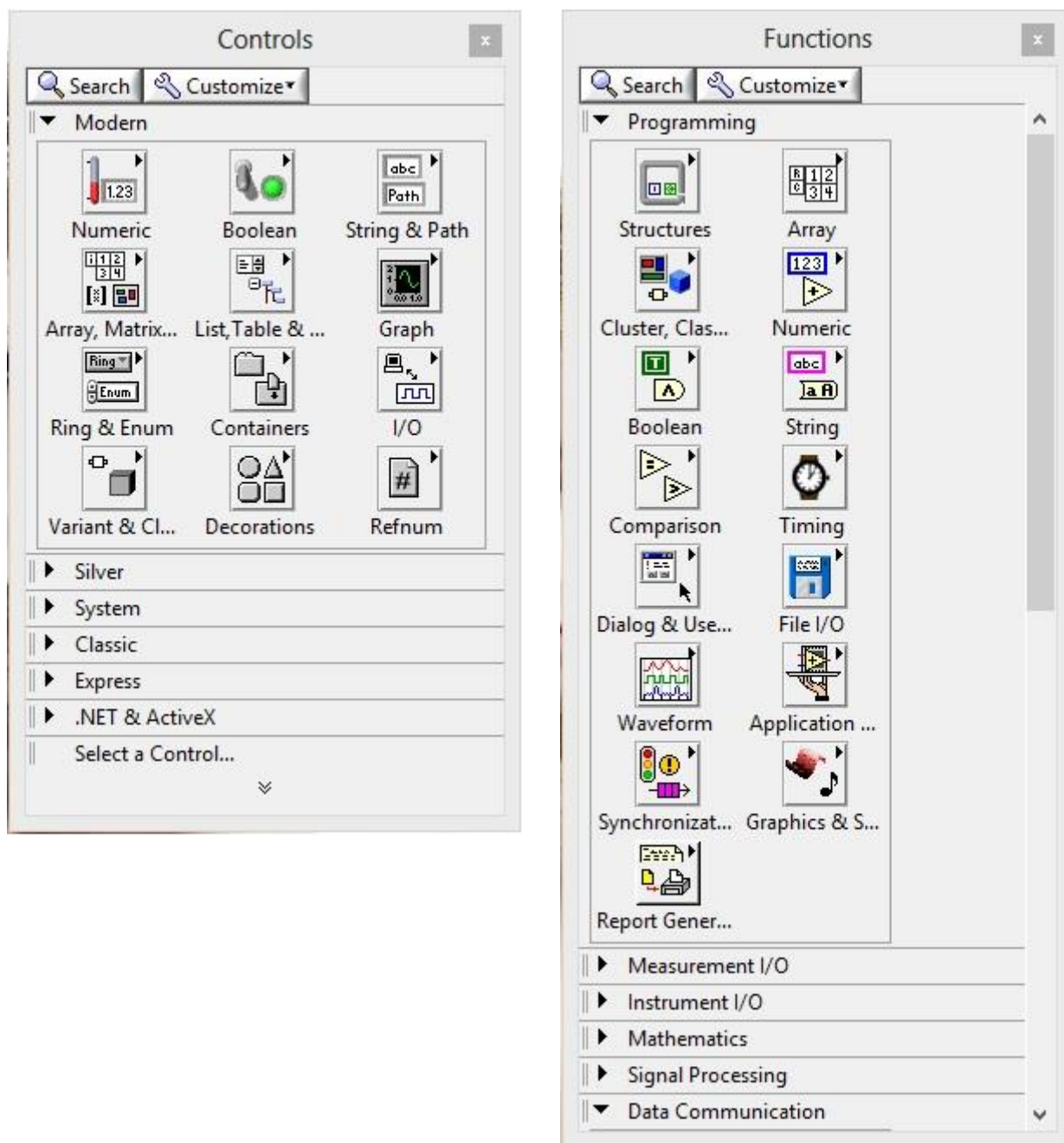
2.) **Indicators** มีหน้าที่เป็นตัวแสดงผลเพียงอย่างเดียวโดยจะรับค่าที่ได้จากแหล่งข้อมูลมาแสดงผลซึ่งอาจปรากฏในรูปของกราฟ, เข็มชี้, ระดับของเหลวหรืออื่นๆ **Indicators** นี้เปรียบเสมือน **output** เพื่อให้ผู้ใช้ได้ทราบค่าสิ่งที่เรากำลังวิเคราะห์ห้อยู่และผู้ใช้งานจะไม่สามารถปรับค่าต่างๆบน **Indicators** ได้โดยตรงแต่จะต้องมีแหล่งข้อมูลที่ส่งให้กับ **Indicators** เหล่านี้

2. **Block Diagram** ทำหน้าที่เสมือนเป็น **Source Code** โดยใช้โปรแกรมภาษากากราฟฟิก องค์ประกอบของ **block diagram** นี้จะแทนโปรแกรม **Node** เช่น **For Loop**, **Case Structure** และฟังก์ชันทางคณิตศาสตร์ เป็นต้น



Block Diagram

นอกจากนี้ยังมีส่วนอื่นๆ เช่น ไอคอน เครื่องมือ และฟังก์ชันต่างๆ เพื่อการออกแบบในสองส่วนด้านบนดังกล่าวข้างต้น



ไอคอนเครื่องมือ และฟังก์ชันต่างๆ

## ความสามารถของโปรแกรม LabVIEW

เนื่องจากบริษัท National Instrument (NI) ซึ่งเป็นผู้พัฒนาโปรแกรม LabVIEW มี Product ในการพัฒนาอยู่มากมายทั้ง Hardware และ Software จึงทำให้โปรแกรม LabVIEW มีความสามารถในการติดต่อ Hardware อย่างหลากหลายเช่น

### Hardware

การใช้ โปรแกรม LabVIEW เพื่อเชื่อมต่อกับฮาร์ดแวร์ภายนอกทำได้โดยผ่านทางการ์ด DAQ (Data Acquisition) การเชื่อมต่อสามารถเชื่อมต่อกับพอร์ต (Port) ได้หลายชนิด เช่น พอร์ตขนาน (Parallel Port), พอร์ตอนุกรม (serial port), GPIB, และ HPIB เป็นต้น จึงมีแนวความคิดในการออกแบบวงจรขึ้นมา โดยกำหนดคุณสมบัติให้เป็นบอร์ดแบบภายนอกเชื่อมต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม (RS-232) มีจำนวนอินพุต-เอาต์พุต 16 ช่อง (Channel) อินพุตทำงานได้ทั้งโหมดดิจิตอลอินพุตและอนาล็อกอินพุต สำหรับเอาต์พุตกำหนดให้เป็นแบบดิจิตอลเอาต์พุต ออกแบบให้สร้างง่ายและต้นทุนต้องไม่สูงมากจนเกินไป

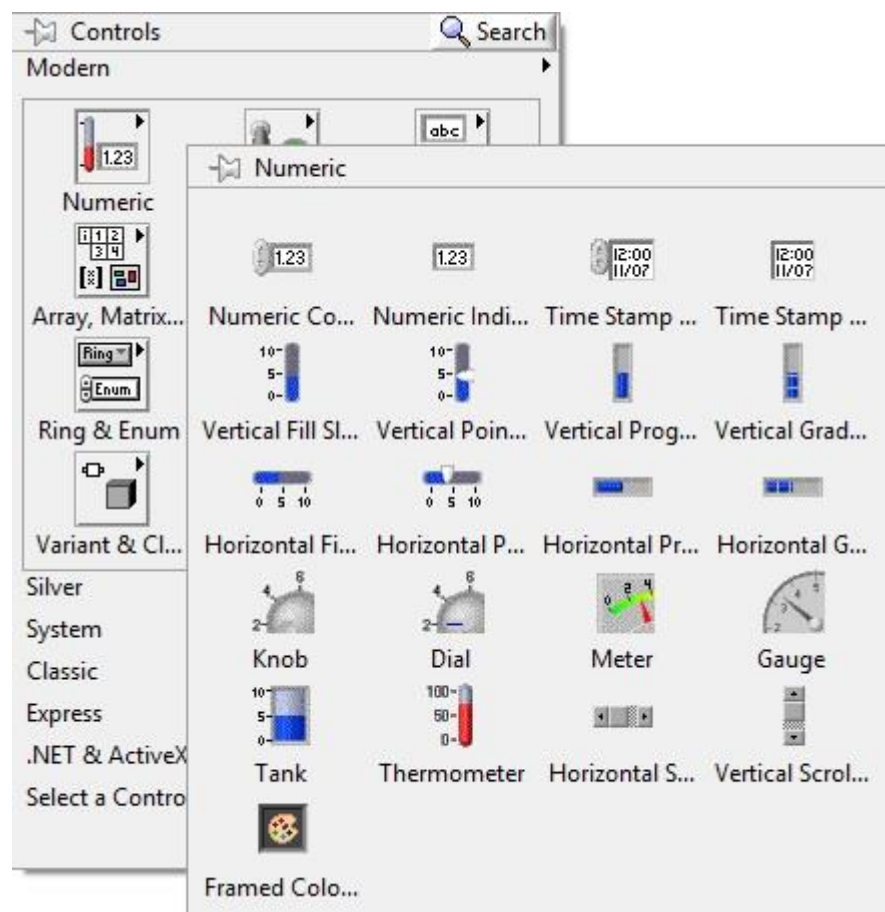
### Software

- Protocol ต่างๆในทางอุตสาหกรรม LabVIEW ก็สามารถติดต่อสื่อสารได้รวมทั้ง PLC ยี่ห้อต่างๆ และงาน SCADA LabVIEW ก็สามารถทำได้เหมือนโปรแกรม SCADA ทั่วไป และบริษัท NI ยังมี PLC ของตนเองขายอีก
- ความสามารถในการทำ Image Processing ก็ทำได้ไม่แพ้ Image Processing ในท้องตลาด
- สามารถติดต่อกับ Database มาตรฐานรวมทั้งการควบคุมการทำงานกับโปรแกรม MS-OFFICE และอื่นๆ ใน windows

ตัวอย่างการใช้งานโปรแกรม

## 1. การใช้ฟังก์ชัน Numeric

ไปหน้า Front Panel >> คลิกเมาส์ขวา >> Numeric

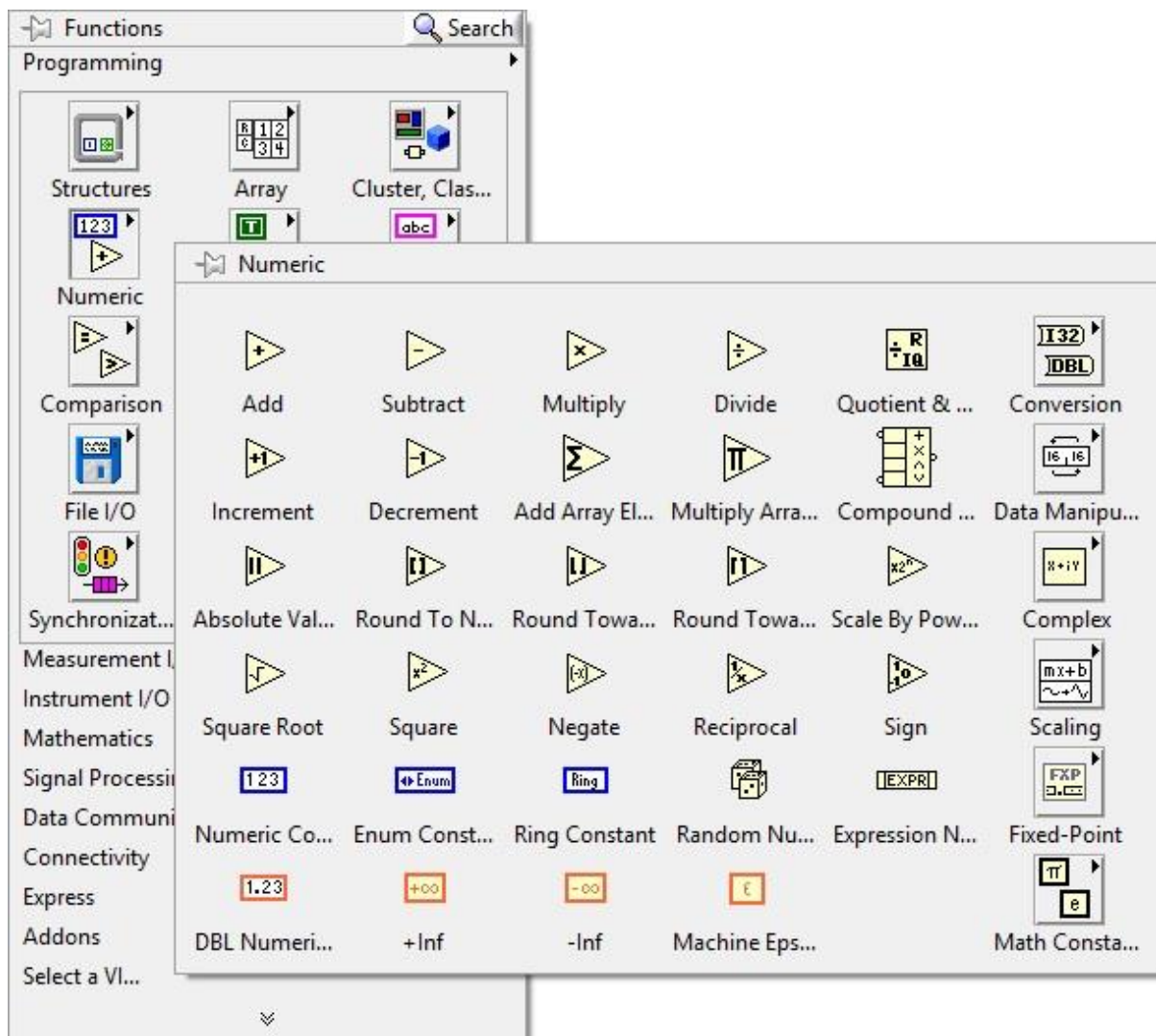


มีหลากหลายแบบสามารถเลือกได้ตามความต้องการของผู้ใช้



ฟังก์ชัน GATE Numeric การบวก ลบเลข

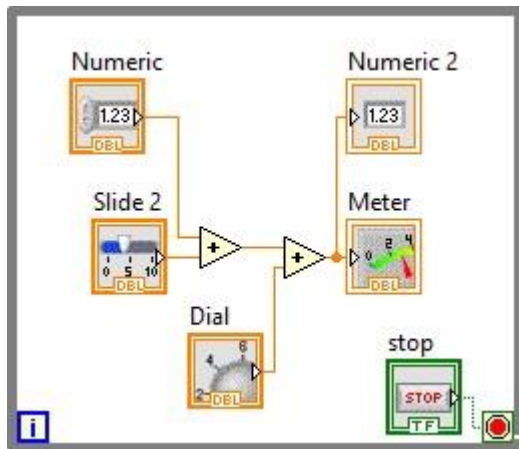
ไปหน้า Block Diagram >> คลิกเมาส์ขวา >> Numeric



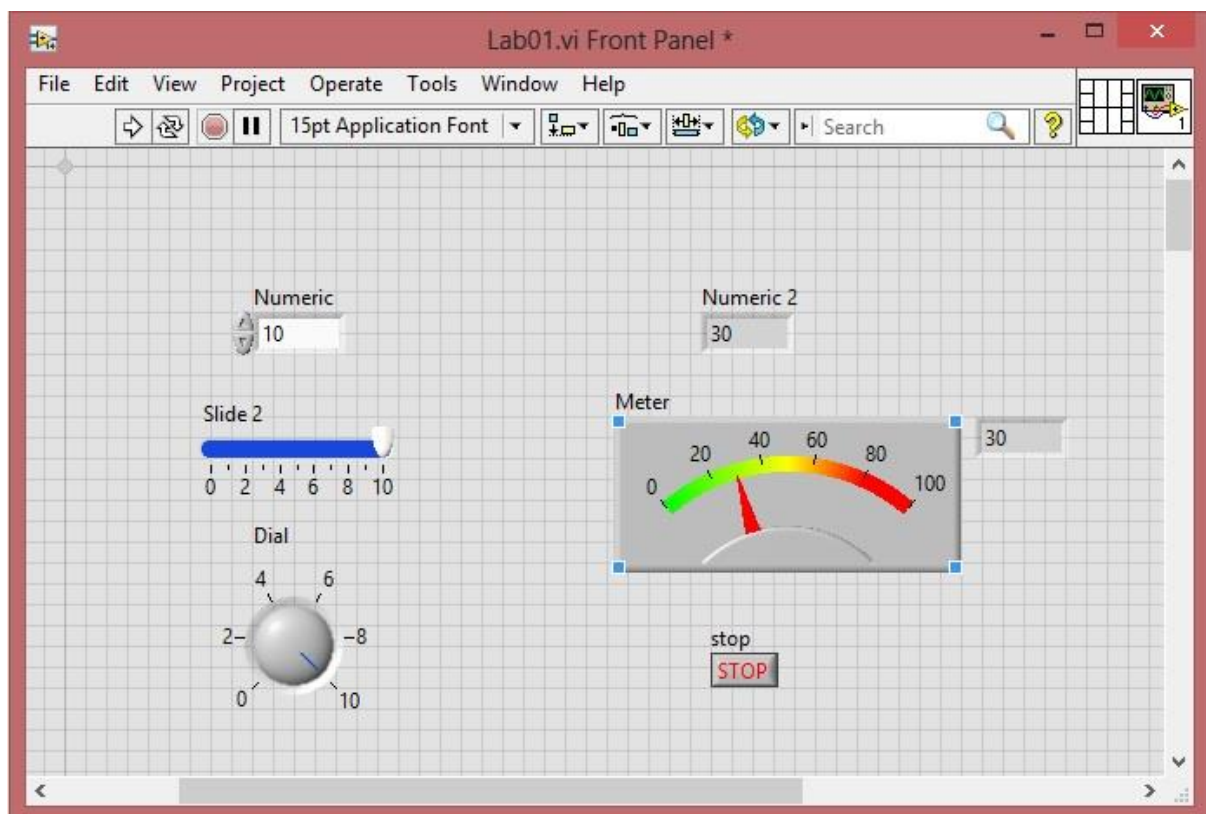
สามารถเลือกได้ตามความต้องการของผู้ใช้

ตัวอย่างการใช้ฟังก์ชัน Numeric ในการบวกเลขหา Output

ส่วน Block Diagram



ส่วน Front Panel



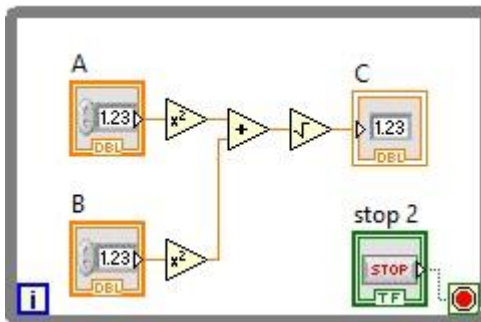


**Example 1:** จงคำนวณสูตรพีทาโกรัส  $A^2+B^2=C^2$

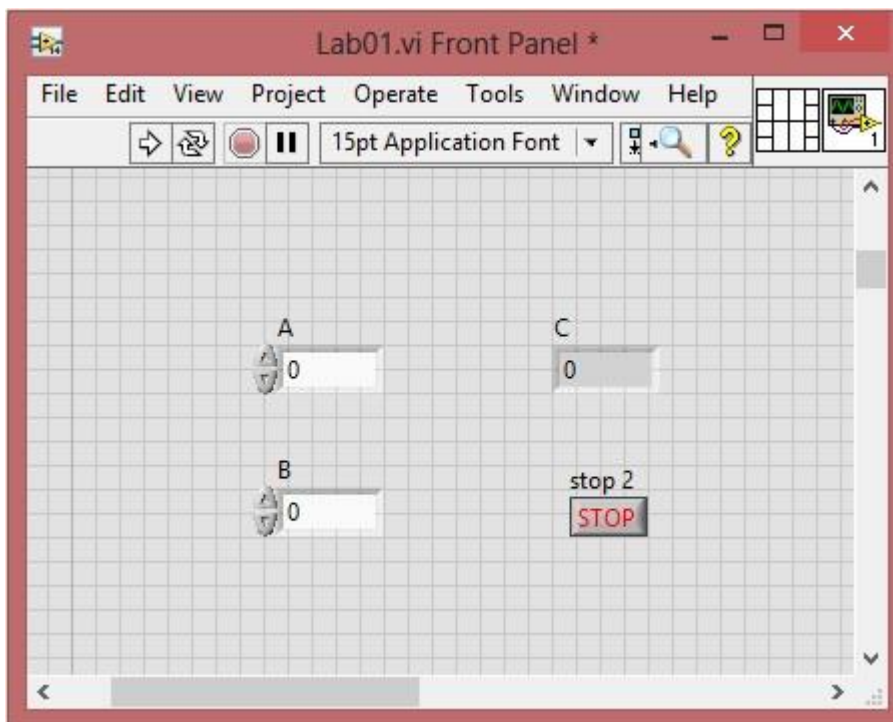
ให้ A, B เป็น Input และ C เป็น Output

จะได้  $C = \sqrt{(A^2 + B^2)}$

ส่วน Block Diagram

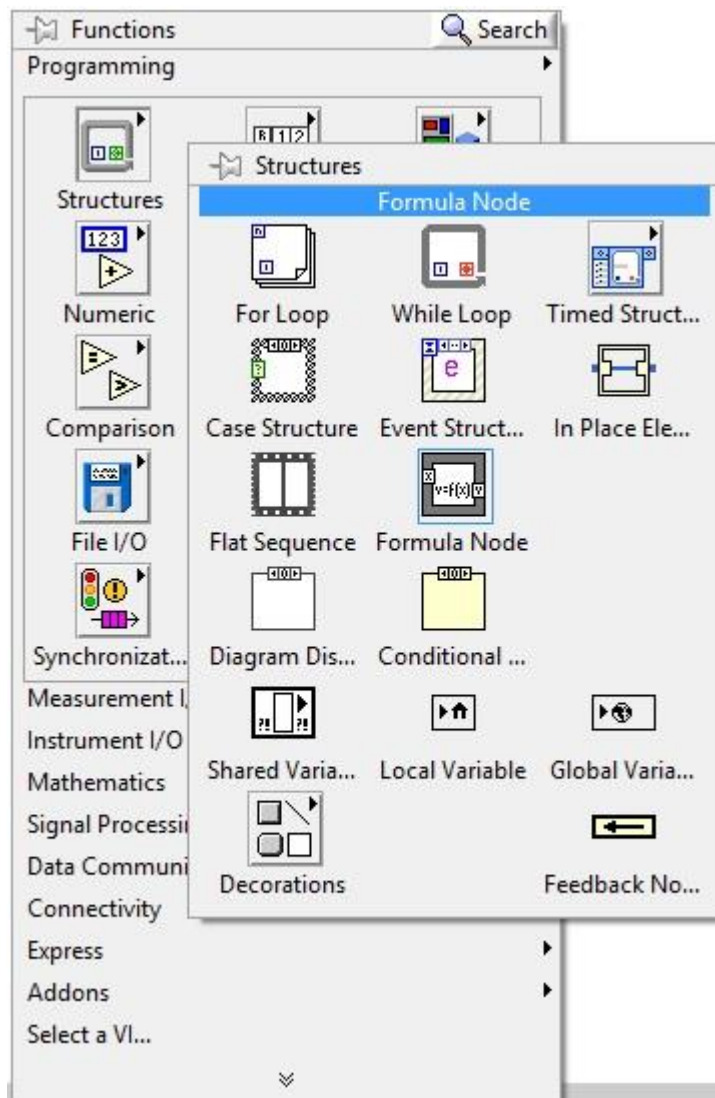


ส่วน Front Panel



## 2. การใช้ฟังก์ชัน Formula Node

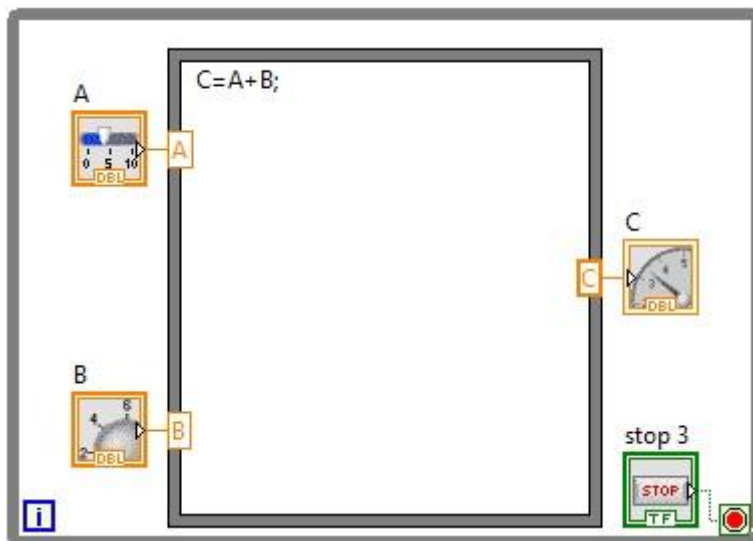
ไปหน้า Block Diagram >> คลิกเมาส์ขวา >> Structures >> Formula Node



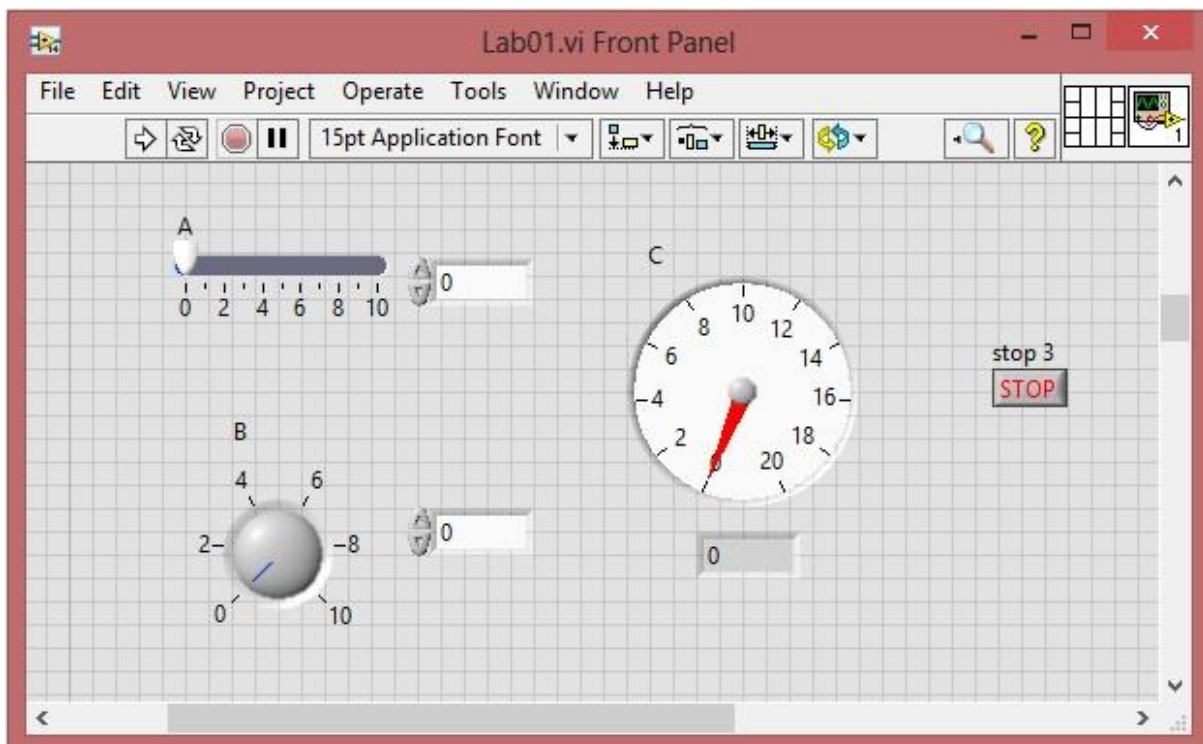
ซึ่ง Output ที่ได้จะเป็นไปตามที่เราโปรแกรมไว้ในกรอบ Formula Node ตัวอย่างเช่น  $C = A+B$

ตัวอย่างการใช้ฟังก์ชัน Formula Node ( $C = A+B$ )

ส่วน Block Diagram

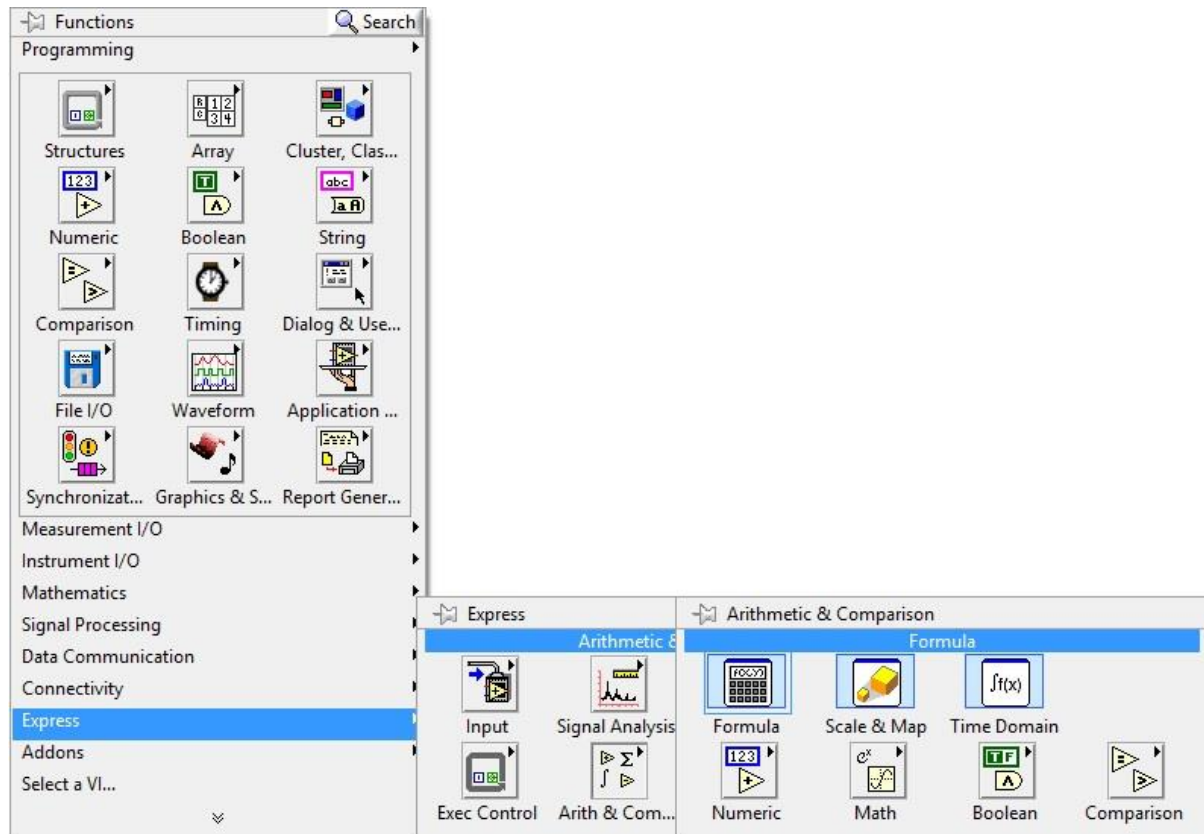


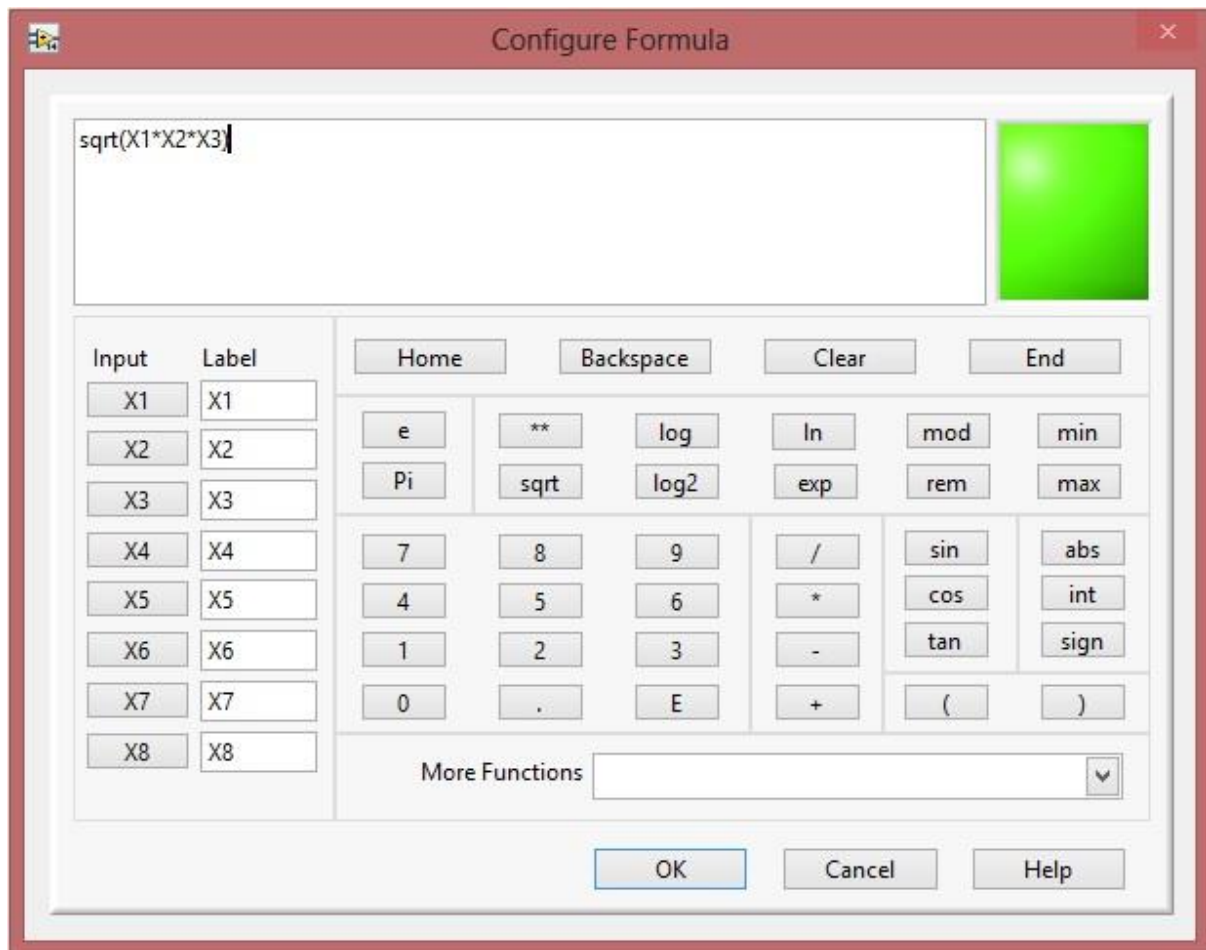
ส่วน Front Panel



### 3. การใช้ฟังก์ชัน Formula

ไปหน้า Block Diagram >> คลิกเมาส์ขวา >> Express >> Arithmetic & Comparison >> Formula

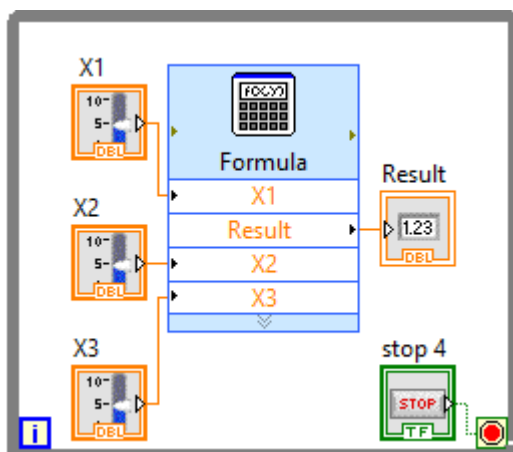




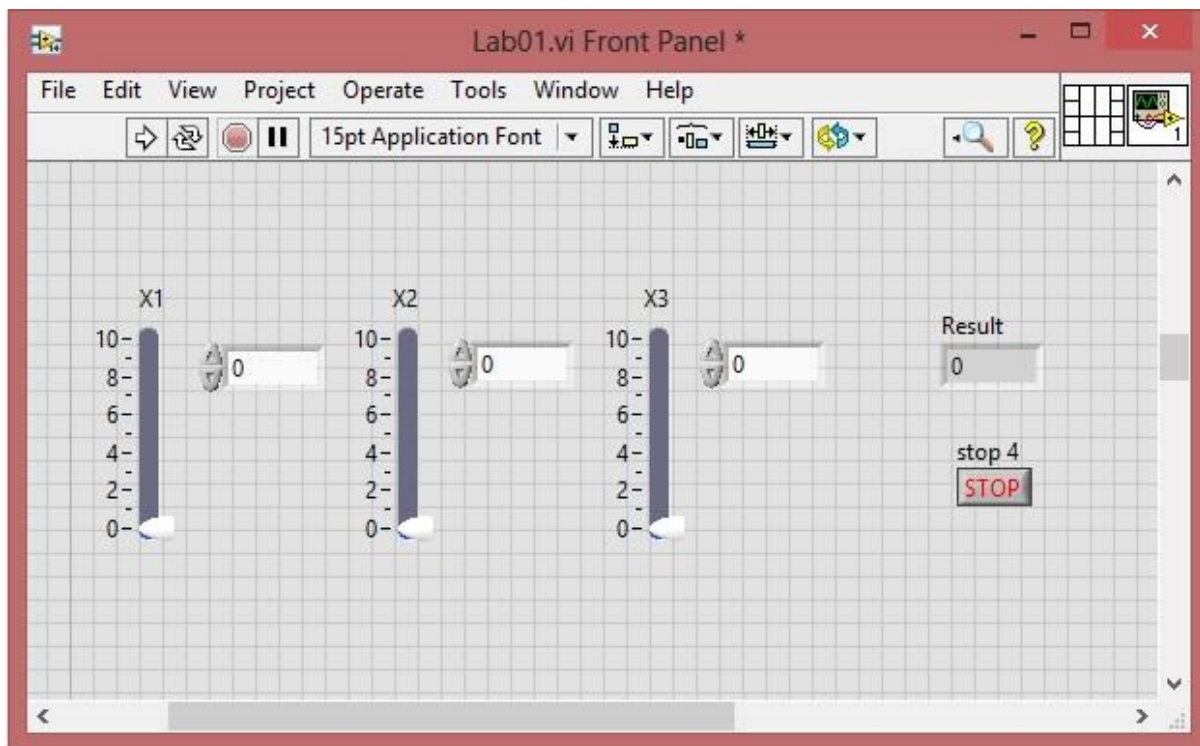
ซึ่งOutput ที่ได้จะเป็นไปตามที่เราเขียนไว้ในตัว Formula

ตัวอย่างการใช้ Formula

ส่วน Block Diagram

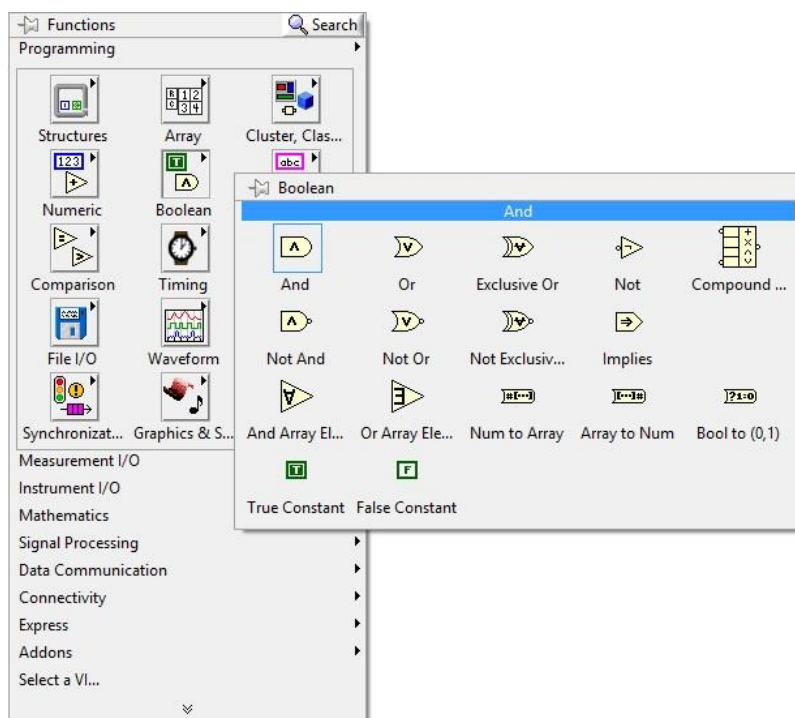


## ส่วน Front Panel



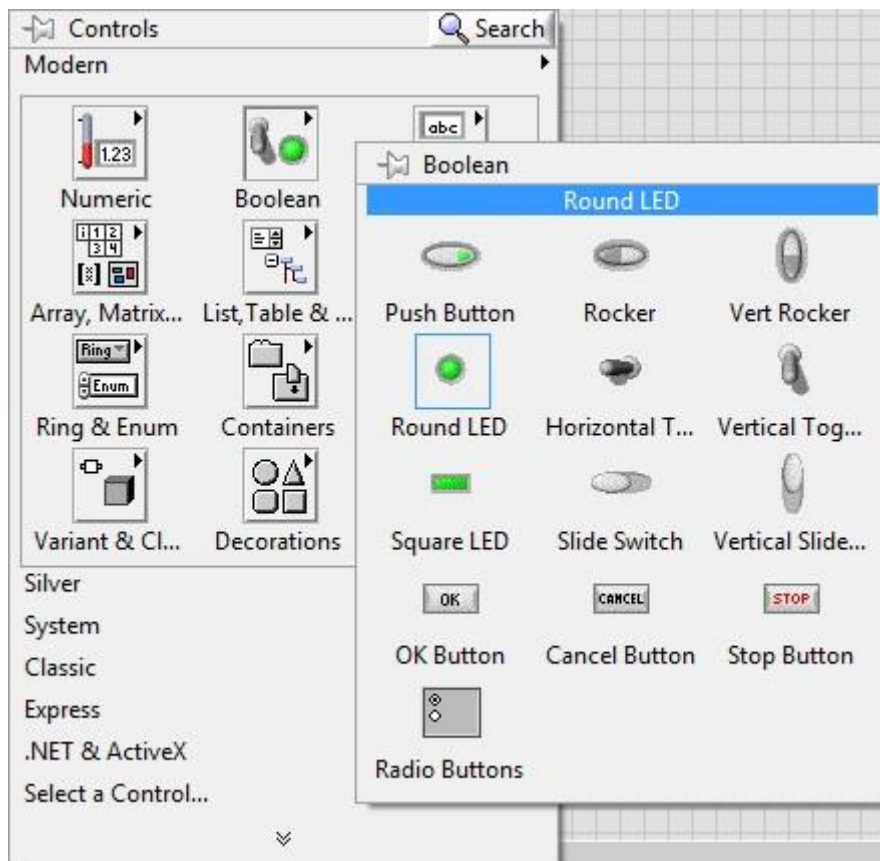
## 4. การใช้ AND GATE, OR GATE

เลือกใช้ AND GATE, OR GATE โดยคลิกเมาส์ขวาที่หน้า Block Diagram >> Boolean



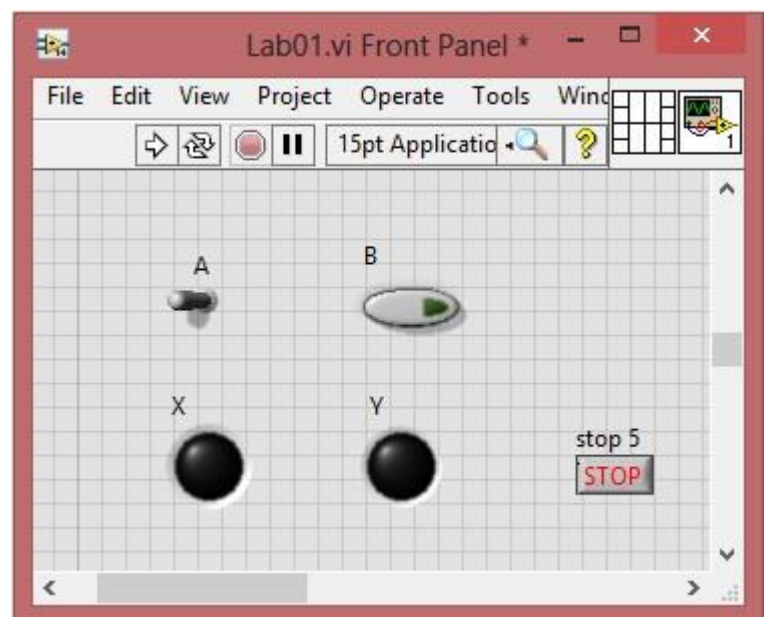
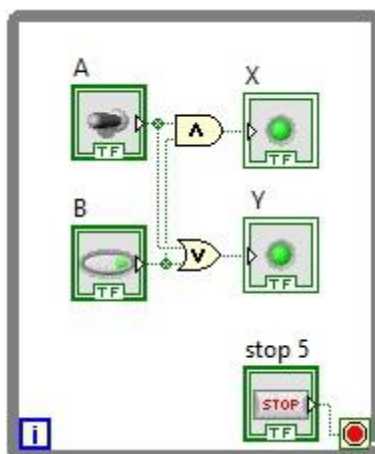


เลือกใช้ LED



ตัวอย่างการใช้ AND GATE, OR GATE

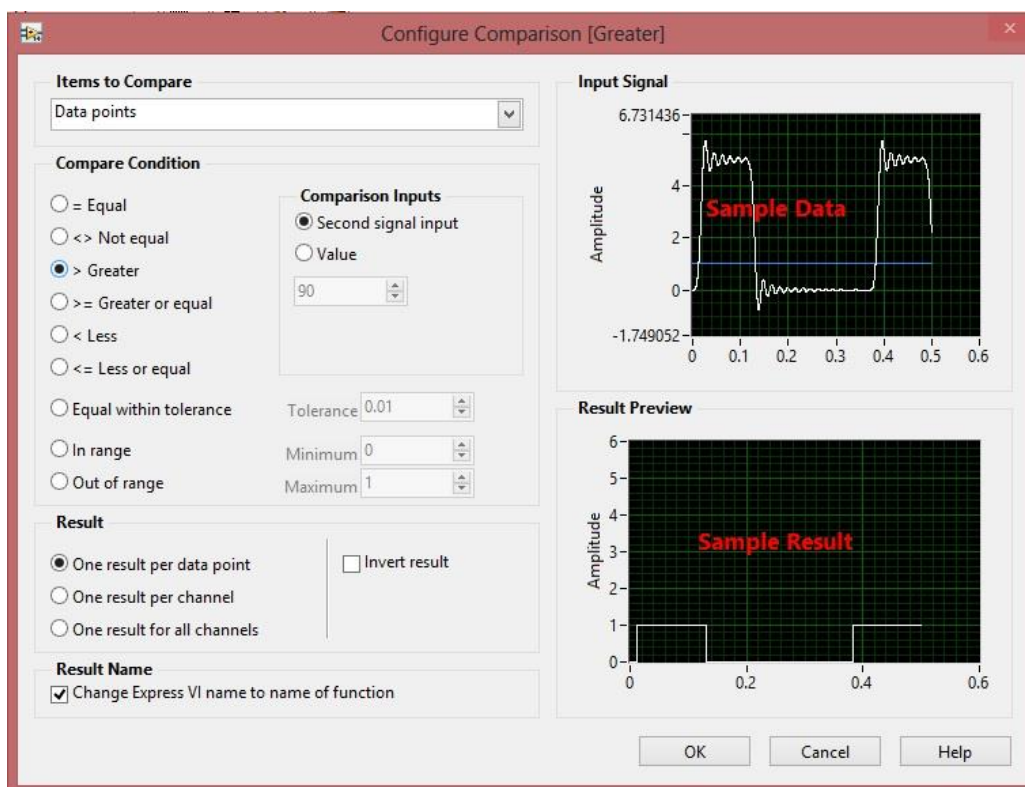
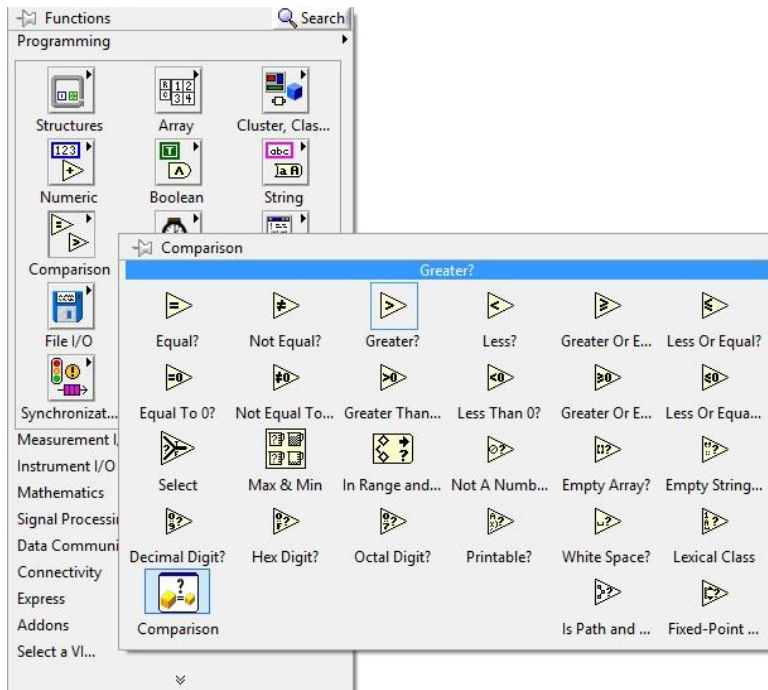
ส่วน Block Diagram และ Front Panel



## 5. การใช้ Greater (มากกว่า)

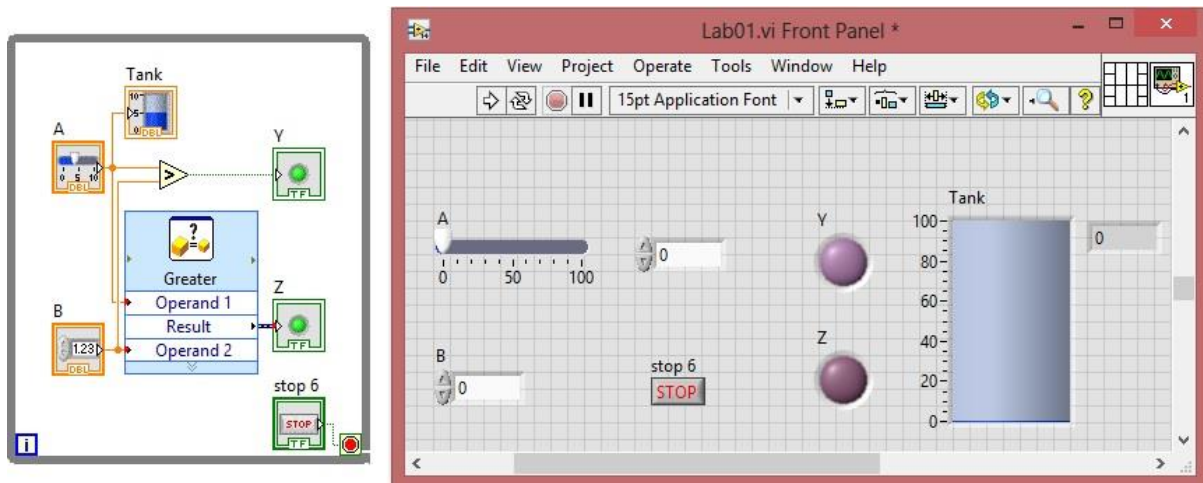
คลิกเมาส์ขวาที่หน้า Block Diagram >> Comparison >> Greater?

หรือเลือกใช้ฟังก์ชัน Comparison แล้วเลือก Greater ก็ได้



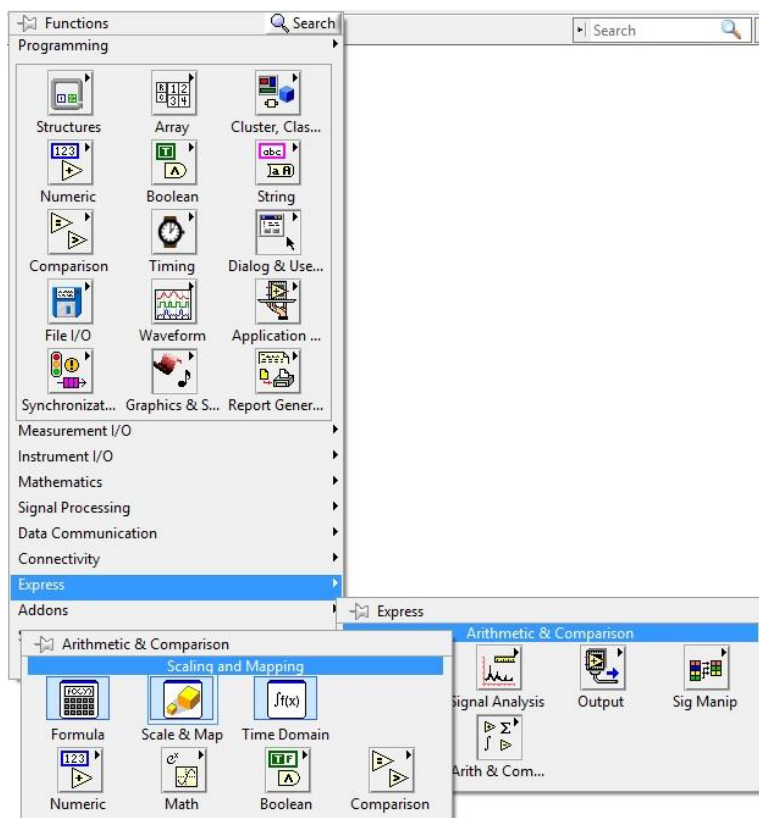
ตัวอย่างการใช้ **Greater** (มากกว่า)

ส่วน **Block Diagram** และ **Front Panel**



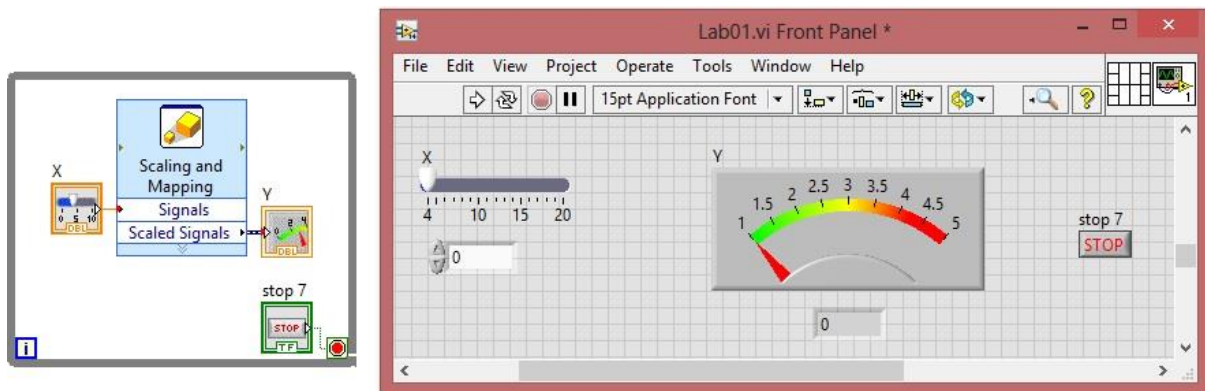
## 6. การใช้ Scaling and Mapping

คลิกเมาส์ขวาที่หน้า **Block Diagram** >> **Express** >> **Arithmetic & Comparison** >> **Scaling and Mapping**



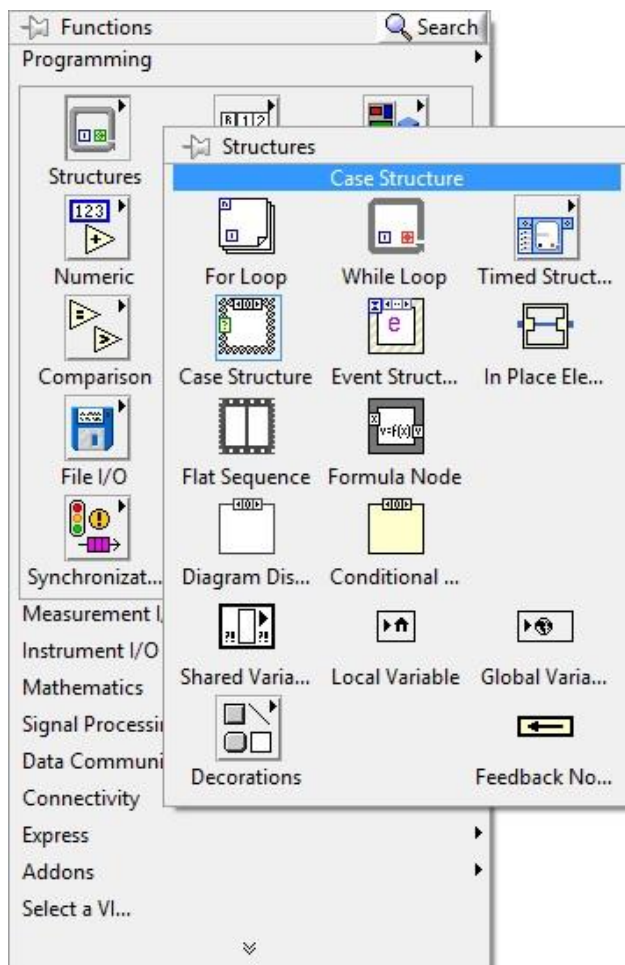
ตัวอย่างการใช้ **Scaling and Mapping** แปลงสัญญาณ 4-20 mA to 1-5 V

ส่วน **Block Diagram** และ **Front Panel**



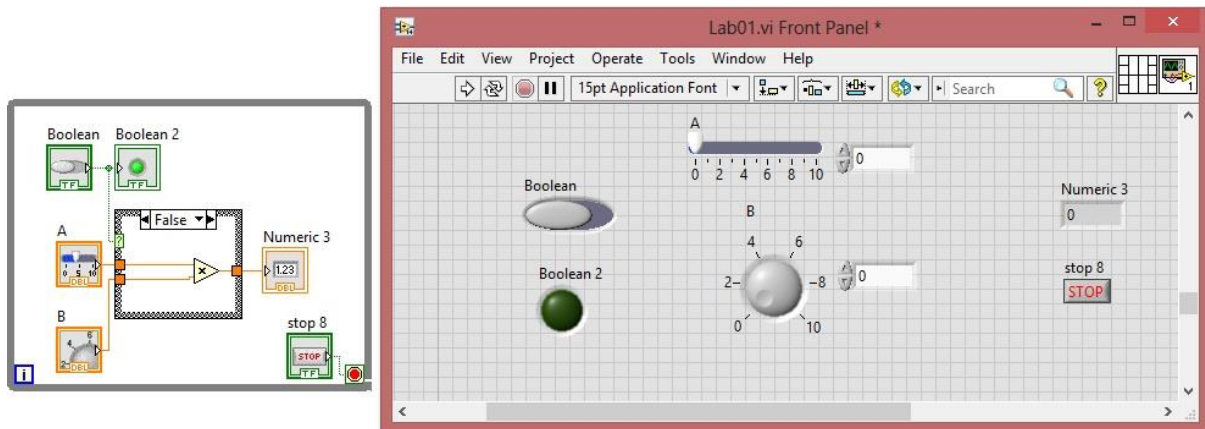
## 7. การใช้ Case Structures

คลิกเมาส์ขวาที่หน้า **Block Diagram >> Structures >> Case Structures**



## ตัวอย่างการใช้ Case Structures

## ส่วน Block Diagram และ Front Panel



## Example 2: วัดระดับน้ำในแทงค์น้ำโดยให้ LED แสดงผลดังนี้

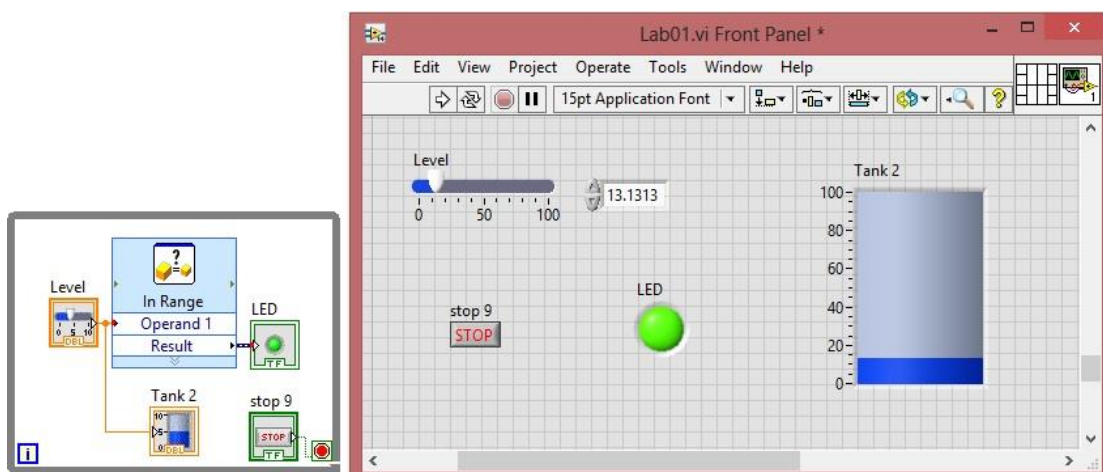
0-11% LED แสดงผลเป็นสีแดง

11-90% LED แสดงผลเป็นสีเขียว

91-100% LED แสดงผลเป็นสีแดง

วิธีทำ: เลือกใช้ฟังก์ชัน Comparison แล้วเลือก In Range ตั้งไว้ที่ 11 – 90 เพื่อให้ LED แสดงผลเป็นสีเขียว (ON) นอกนั้นแสดงผลเป็นสีแดง(OFF)

## ส่วน Block Diagram และ Front Panel





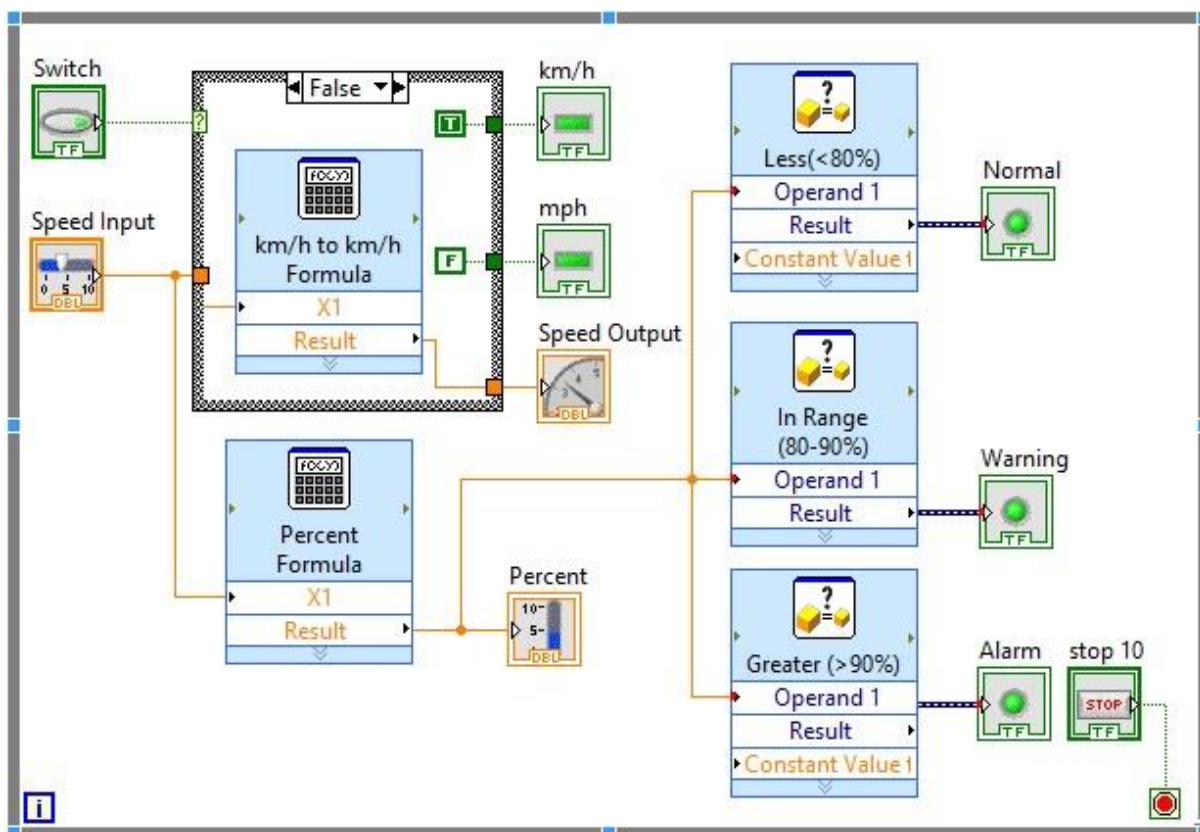
**Example 3:** วัดความเร็วหน่วย km/h โดยใช้สวิตช์เป็นตัวเปลี่ยนค่าแสดงผลความเร็วเป็น km/h กับ mph พร้อมแสดงผลเป็น % และแสดงผล LED เป็น Normal, Warning และ Alarm ดังนี้

- < 80% LED แสดงผลเป็นสีเขียว (Normal)
- 80-90% LED แสดงผลเป็นสีเหลือง (Warning)
- > 90% LED แสดงผลเป็นสีแดง (Alarm)

\* Max speed = รหัสนักศึกษา 3 ตัวท้าย

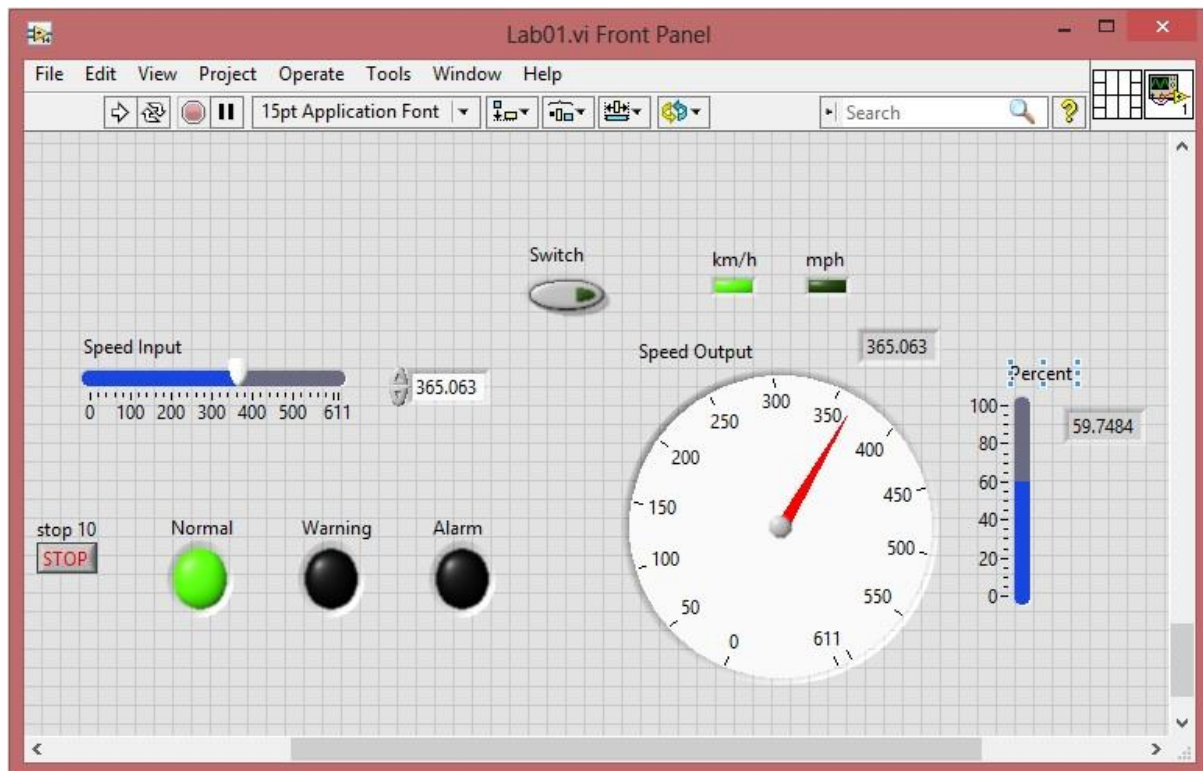
\*\* 1 mph = 1.609 km/h

ส่วน Block Diagram





## ส่วน Front Panel



## สรุปผลการทดลอง

จากการทดลองทำให้ได้รับความรู้เกี่ยวกับการใช้ฟังก์ชันต่างๆในโปรแกรม LabVIEW 2014 ทำให้เราได้ฝึกใช้โปรแกรมจริง และฝึกคิด แก้ปัญหาต่างๆที่เกิดขึ้นในแต่ละการทดลอง เพื่อคำนวณหา Output ตามที่เราต้องการ ซึ่งสามารถนำไปประยุกต์ใช้เพื่อให้เกิดประโยชน์สูงสุดต่อไปในภายหน้าได้