บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 อนุมูลอิสระ

อนุมูลอิสระคือ อะตอม โมเลกุล หรือสารประกอบ ที่มีอิเล็คตรอนไม่ครบคู่ ทำให้มีความว่องไวต่อการเกิดปฏิกิริยาออกซิเดชัน เพื่อแย่งชิงอิเลคตรอนให้ครบคู่ สารอื่นที่ถูกแย่งอิเล็กตรอนมาก็กลายเป็นสารอนุมูลอีกครั้ง เนื่องจากจะต้องไปแย่งเอาอิเล็กตรอนมาทดแทนเช่นเดียวกัน ดังนั้นจึงเกิดปฏิกิริยาลูกโซ่ขึ้นต่อเนื่อง เมื่อเกิดอนุมูลอิสระในร่างกายจะทำให้เซลล์ต่างๆ ถูกทำลาย อนุมูลอิสระมีที่มาจากทั้งภายในและภายนอกร่างกาย ภายในเกิดจากกระบวนการเมตาบอลิซึมของออกซิเจนภายในเซลล์ ส่วนภายนอกเกิดจากปัจจัยต่าง ๆ ได้แก่มลพิษจากสิ่งแวดล้อม ควันบุหรี่ รังสี หรือแสงแดด โดยปกติร่างกายของมนุษย์จะมีการรักษาสมดุลของอนุมูลอิสระให้อยู่ในปริมาณที่เหมาะสมด้วยการสร้างสารต้านอนุมูลอิสระ แต่ถ้าร่างกายเสียสมดุลโดยมีปริมาณของอนุมูลอิสระมากกว่าปริมาณของสารต้านอนุมูลอิสระแล้ว จะทำให้เกิดความเสียหายต่อเซลล์อันเป็นที่มาของโรคต่างๆ [1]

2.2 สารต้านอนุมูลอิสระ

“สารต้านอนุมูลอิสระ” ในปัจจุบันกำลังได้รับความนิยมเป็นอย่างสูงเนื่องจากเป็นสารที่มีฤทธิ์ในการทำลายอนุมูลซึ่งเป็นพิษต่อร่างกาย โดยที่ “สารต้านอนุมูลอิสระ” คือกลุ่มสารซึ่งเมื่อปรากฏในความเข้มข้นต่ำ ๆ แต่สามารถออกซิไดส์หรือยับยั้งการเกิดสภาวะ Oxidative stress ได้ [2] ซึ่งในที่นี้สารต้านอนุมูลอิสระสามารถแบ่งออกได้เป็น 2 ประเภทคือ [3]

2.2.1 สารต้านอนุมูลอิสระในร่างกาย (Endogenous antioxidants)

ในกลไกของสิ่งมีชีวิตมักจะมีระบบที่ใช้สำหรับต่อต้านอนุมูลซึ่งเป็นต้นเหตุทำให้เซลล์เกิดความเสียหาย สารต้านอนุมูลอิสระเหล่านั้นได้แก่ Glutathione peroxidase, catalase, และ superoxide dismutases (SOD) เป็นต้น ซึ่งสารต้านอนุมูลอิสระที่กล่าวมานี้เป็นเอนไซม์ที่มีฤทธิ์ในการต้านสารอนุมูลอิสระ ช่วยรักษาสมดุลระหว่างอนุมูลอิสระไม่ให้มีมากจนเกิดความเสียหายต่อเซลล์ในร่างกาย

2.2.2 สารต้านอนุมูลอิสระในร่างกาย (Exogenous antioxidants)

สารต้านอนุมูลอิสระกลุ่มนี้ได้จากการบริโภคอาหาร โดยสารที่เป็นที่รู้จักกันดีได้แก่ vitamin C, vitamin E และ β-carotene ซึ่งวิตามินซีเป็นสารต้านอนุมูลอิสระที่มีบทบาทสำคัญโดยเป็นสารต้านอนุมูลอิสระที่สามารถละลายได้ในน้ำ ทำให้สามารถกำจัดอนุมูลอิสระในส่วนที่ละลาย

น้ำ ในทางเดียวกันวิตามินอีก็มีบทบาทสำคัญในส่วนของสารที่สามารถละลายได้ในน้ำมัน นอกจากสารทั้ง 2 ตัวนี้แล้วยังมีสารต้านอนุมูลอิสระอีกมากมายซึ่งอยู่ในผักและผลไม้

2.3 ความสามารถในการต้านอนุมูลอิสระโดยรวม (Total Antioxidant Capacity)

การวัดและตรวจสอบความสามารถในการต้านอนุมูลอิสระมีอยู่หลายแบบและหลากหลายความหมายได้แก่ “capacity” (หรือ efficiency, power, parameter, potential, potency และ activity) “antioxidant activity” จะใช้สำหรับการวัดความสามารถในการต้านอนุมูลอิสระของสารที่ปรากฏอยู่เดี่ยว ๆ ซึ่งจะแตกต่างกับ Total Antioxidant Capacity [4] ที่จะวัดความสามารถในการต้านอนุมูลอิสระโดยรวมมากกว่า

เนื่องจากความซับซ้อนของสารประกอบที่มีอยู่ในอาหารที่เรารับประทานและความสามารถในการเสริมฤทธิ์กัน (synergistic) ของสารต้านอนุมูลอิสระที่มีอยู่ในอาหารเหล่านั้น นักวิจัยส่วนมากจึงสนใจความสามารถโดยรวมของตัวอย่างในการต้านอนุมูลอิสระมากกว่าที่จะไปแยกและทำการวิเคราะห์หาปริมาณของสารต้านอนุมูลอิสระในตัวอย่าง ซึ่งนอกจากจะเสียเวลา และมีความยุ่งยากแล้วยังไม่ได้บ่งบอกถึงความสามารถในการต้านอนุมูลอิสระอย่างแท้จริงตามที่ได้กล่าวไปแล้วอีกด้วย เช่น quercetin และ rutin ซึ่งเป็นสารต้านอนุมูลอิสระที่มีอยู่มากในผลไม้เช่น ทับทิม เมื่อสารทั้ง 2 ชนิดอยู่ร่วมกันจะมีความสามารถในการต้านอนุมูลอิสระมากกว่าปรากฏอยู่เดี่ยว ๆ เสียอีก [5]

ความสามารถในการต้านอนุมูลอิสระ Total antioxidant capacity (TAC) จึงเป็นการวัดจำนวนโมลของอนุมูลอิสระที่ถูกจับด้วยสารอนุมูลอิสระทั้งหมดที่มีอยู่ในสารละลายตัวอย่าง [6]

2.4 ชุดทดสอบ (Test Kit)

ชุดทดสอบหมายถึง เครื่องมือหรืออุปกรณ์ที่ได้สำหรับการตรวจวัดสารเคมีที่มีความจำเพาะเจาะจง ทั้งระดับปริมาณของสาร ส่วนประกอบ ซึ่งโดยทั่วไปแล้วชุดทดสอบมักจะประกอบไปด้วยรีเอเจนต์ ขวดสารเคมีขนาดเล็ก อุปกรณ์วัดการเปลี่ยนแปลงสีและวัสดุอื่นที่ใช้สำหรับการทดสอบ ชุดทดสอบที่มีอยู่ในปัจจุบันนี้ได้แก่ ชุดทดสอบความเป็นกรด-ด่าง ชุดทดสอบคลอรีน ชุดทสอบโลหะหนัก

ประโยชน์ของชุดทดสอบ

1. ใช้เป็นวิธีทดสอบเบื้องต้น สำหรับตรวจหาสารต่างๆ
2. วิธีใช้สะดวกและรวดเร็ว และมีประสิทธิภาพ ใช้ในห้องปฏิบัติการทั่วไปและใช้ในภาคสนามได้
3. ลดงานการตรวจวิเคราะห์ในเบื้องต้น คงไว้เฉพาะการตรวจยืนยันในห้องปฏิบัติการ

ข้อเสียของชุดทดสอบ

1. เป็นการทดสอบในระดับเบื้องต้นเท่านั้น ซึ่งต้องมีการยืนยันอีกครั้งจากห้องปฏิบัติการ

2.4.1 ชุดทดสอบความสามารถในการต้านอนุมูลอิสระ

Test kit for Potential Anti Oxidant (PAO) [7] ชุดทดสอบความสามารถในการต้านอนุมูลอิสระจากบริษัท Nikken SEIL ประเทศญี่ปุ่นพัฒนาขึ้นโดยใช้ปฏิกิริยารีดักชันของ Cu^{2+} เป็น Cu^{+} โดยสารต้านอนุมูลอิสระที่มีในตัวอย่าง ซึ่งจะสามารถตรวจสอบได้ภายใน 5 นาที สารตัวอย่างจะถูกผสมกับ Cu^{2+} หลังจากนั้นสารต้านอนุมูลอิสระในตัวอย่างจะรีดิวซ์ Cu^{2+} เป็น Cu^{+} และ Cu^{+} จะไปทำปฏิกิริยากับ Solution (Bathocuproine) ซึ่งสามารถตรวจสอบได้โดยการวัดค่าการดูดกลืนแสงที่ 480 ถึง 490 nm ความสามารถในการต้านอนุมูลอิสระจะถูกคำนวณจากปริมาณของ Cu^{+} ที่เกิดขึ้น โดยเทคนิคนี้สามารถวิเคราะห์สารต้านอนุมูลอิสระได้ทั้งสารที่ละลายในน้ำเช่น วิตามินซี และสารที่ไม่ละลายในน้ำเช่น วิตามินอี เป็นต้น รายละเอียดแสดงดังตาราง

ภาพที่ 2.1 ชุดทดสอบสารต้านอนุมูลอิสระจาก Nikken SEIL ประเทศญี่ปุ่น

วิธีวิเคราะห์	วัดการดูดกลืนแสง (480 - 492 nm)
ช่วงในการวิเคราะห์	21.9 - 4378 micro mol/L (cupric ion reducing power)
รูปแบบการวิเคราะห์	96 wells
การเก็บรักษา	อุณหภูมิห้อง (10 - 25°C)
ตัวอย่าง	ซีรัมของสัตว์หรือมนุษย์ ตัวอย่างอาหารหรือเครื่องดื่ม

Free Radical Test Kit [8] เป็นชุดทดสอบเพื่อวัดความสามารถในการต้านอนุมูลอิสระจากบริษัท Osumex Flax Hulls ผลิตขึ้นโดยอาศัยหลักการวัดค่า ORAC (Oxygen Radical Absorbance Capacity) โดยมี Malondialdehyde (MDA) เป็นสารผลิตภัณฑ์ที่ถูกตรวจวัด โดยในชุดทดสอบประกอบไปด้วย

- หลอดทดลอง – สำหรับบรรจุสารตัวอย่างได้แก่ ตัวอย่างปัสสาวะ
- ampoule – สำหรับบรรจุสารเคมีซึ่งจะถูกตรวจวัดในรูปของ MDA
- ปิเปต - ใช้สำหรับถ่ายสารตัวอย่าง

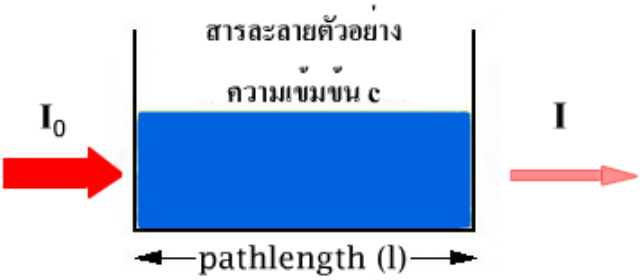
ภาพที่ 2.2 ชุดทดสอบสารต้านอนุมูลอิสระจาก Osumex Flax Hulls

3rd Gen.Total Antioxidant Status (TAS) Assay Kit [9] ผลิตขึ้นโดยมีหลักการคือสารต้านอนุมูลอิสระในตัวอย่างจะรีดิวซ์ อนุมูลอิสระ ABTS ที่มีสีฟ้าไปเป็นสารที่ไม่มีสี โดยการเปลี่ยนแปลงค่าการดูดกลืนแสงที่ 660 nm จะสัมพันธ์กับความสามารถในการต้านอนุมูลอิสระโดยรวม โดยเทคนิคนี้จะใช้ Trolox เป็นตัวเปรียบเทียบ ในชุดทดสอบจะประกอบไปด้วย R1 (Buffer), R2 (ABTS Radical Cation)

ภาพที่ 2.3 ชุดทดสอบความสามารถในการต้านอนุมูลอิสระโดยรวมจาก baranmedikal

2.5 การวัดค่าการดูดกลืนแสงของสารตัวอย่าง [10]

การวัดค่าการดูดกลืนแสงของสารตัวอย่างเราสามารถทำได้โดยให้ลำแสงผ่านเข้าไปในตัวอย่าง (Incident Light : I_0) แล้ววัดปริมาณแสงที่เหลือผ่านออกมา (I) โดยเทียบกับแสงที่ผ่านออกมาเมื่อไม่มีสารตัวอย่าง ดังภาพที่ 2.4



ภาพที่ 2.4 ลำแสงที่ผ่านเข้าออกสารละลายความเข้มข้น C เป็นระยะทาง *l*

Transmittance (T) เป็นสัดส่วนปริมาณแสงที่ผ่านออกมา (I) ต่อปริมาณแสงที่ผ่านเข้าไปในตัวอย่าง (I_0) เขียนสมการได้ว่า

$$T = \frac{I}{I_o} \qquad \text{สมการที่ 2.1}$$

Absorbance (A) นิยามสมการได้เป็น

$$A = \log \frac{I_o}{I} = -\log T \qquad \text{สมการที่ 2.2}$$

โดยทั่วไปจะรายงานค่า Transmittance เป็นเปอร์เซ็นต์ (%T) ดังนั้น

$$\%T = 100 \frac{I}{I_o} \qquad \text{สมการที่ 2.3}$$

$$\log \%T = \log 100 \frac{I}{I_o}$$

$$\log \%T = 2 + \log \frac{I}{I_o}$$

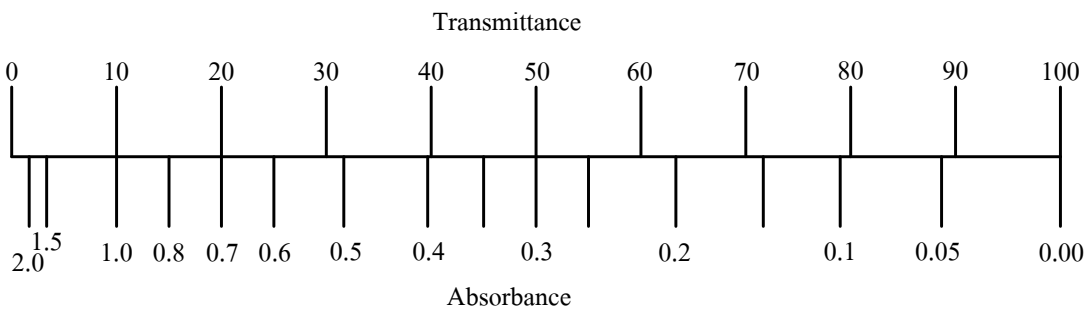
$$\log \%T = 2 - A \quad \text{หรือ}$$

$$A = 2 - \log \%T$$

ดังนั้น ค่า T มีค่าอยู่ในช่วง 0-1 และ %T มีค่าตั้งแต่ 0 - 100 ส่วน A = 0 เมื่อแสงที่ผ่านเข้าไปในตัวอย่างไม่ถูกดูดกลืนไว้ และผ่านออกมา 100%, A = 1 เมื่อแสงผ่านออกมาเพียง 10% และ A = 2 ถ้าแสงผ่านออกมาน้อยมากเพียง 1% ดังตารางที่ 2.1

ตารางที่ 2.1 ค่า T มีค่าอยู่ในช่วง 0-1 และ %T

Trasmittance(I/Io)	%T(log I/Io)	Log%T	Absorbonce(-log T)
1	100	2	0
0.1	10	1	1
0.01	1	0	2
0.001	0.1	-1	3



ภาพที่ 2.5 ความสัมพันธ์ระหว่างค่าการดูดกลืน กับ %Transmittance

2.5.1 ความสัมพันธ์ของค่าการดูดกลืนแสงและความเข้มข้น (Absorbance and Concentration)

ค่าการดูดกลืนแสงของสารมีความสำคัญอย่างยิ่งในเชิงปริมาณวิเคราะห์เนื่องจากค่าการดูดกลืนจะเป็นสัดส่วนโดยตรงกับความเข้มข้นตามกฎของเบียร์-แลมเบิร์ต (Beer-Lambert Law) ดังสมการที่ 2.4

$$A = \varepsilon cl$$ สมการที่ 2.4

A = ค่าการดูดกลืนแสงของสาร (Absorbance)

ε = เป็นสมบัติจำเพาะของสารที่ดูดกลืนและวัดที่ความยาวค่าหนึ่ง เรียกว่า Molar Absorptivity(L mol^{-1} cm^{-1})

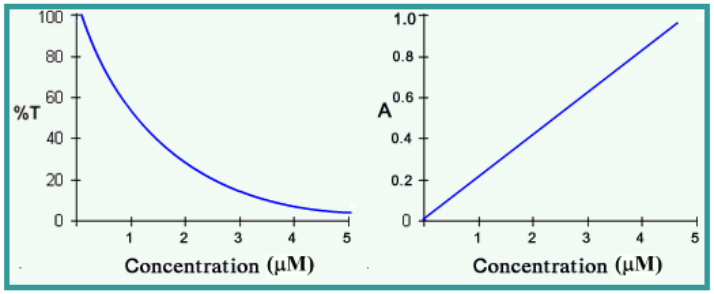
l = ระยะทางที่แสงผ่านตัวอย่าง หรือความกว้างของเซลล์(cm)

c = ความเข้มข้นเป็น โมล/ลิตร หรือโมล่าร์(M)

ถ้าความเข้มข้นของสารอยู่ในหน่วยอื่นจะเขียนสมการเป็น

$$A = acl$$ สมการที่ 2.5

โดย a = Absorptivity เป็นค่าคงที่ขึ้นกับชนิดของสาร และความยาวคลื่น



ภาพที่ 2.6 ความสัมพันธ์ระหว่างความเข้มข้นกับ %T และ Absorbance

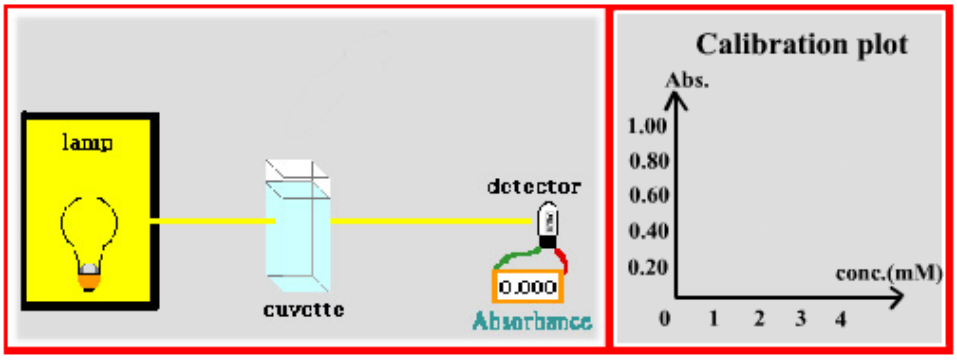
2.5.2 กราฟความเข้มข้นมาตรฐาน (Calibration Curve) [11]

จากภาพที่ 2.11 เราทราบแล้วว่า ค่าการดูดกลืนแสงของสารมีความสัมพันธ์กับค่าความเข้มข้นของสาร หรือปริมาณของเนื้อสารนั้นตามกฎของเบียร์ และแลมเบิร์ต ดังนั้นถ้าเรานำความสัมพันธ์นี้มาสร้างกราฟ และได้ความสัมพันธ์เชิงเส้นตรง เราจะเรียกกราฟนี้ว่า กราฟความเข้มข้นมาตรฐาน (Calibration Curve)

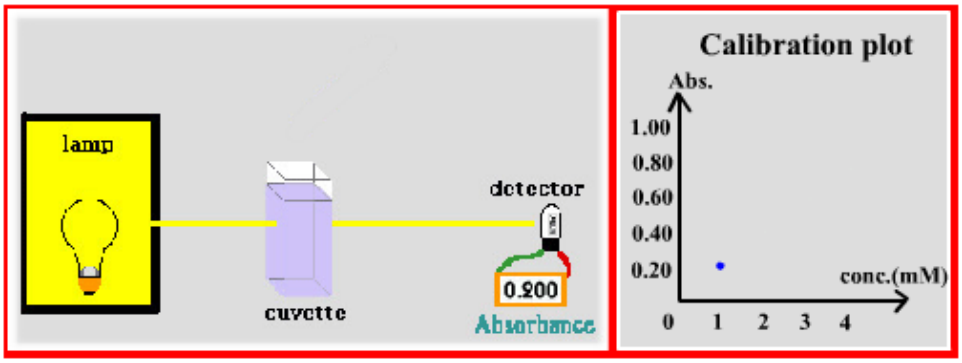
กราฟความเข้มข้นมาตรฐานนี้มีประโยชน์มากในเชิงปริมาณวิเคราะห์ เพราะสามารถใช้เทียบเพื่อหาความเข้มข้นของสารที่ไม่ทราบค่าได้ โดยที่สารที่ไม่ทราบค่าความเข้มข้นนั้นจะต้องอยู่ในช่วงความเข้มข้นมาตรฐานที่ทราบค่าแล้ว และเส้นกราฟความเข้มข้นมาตรฐานจะต้องเป็นเส้นตรงเสมอ

วิธีการสร้างกราฟความเข้มข้นมาตรฐาน คือ นำสารละลายมาตรฐาน(Standard Solution) ที่ทราบค่าความเข้มข้นที่แน่นอนอย่างน้อย 3-4 ความเข้มข้น มาวัดค่าการดูดกลืนแสง จากนั้นก็นำความสัมพันธ์ที่ได้ไปสร้างกราฟ ส่วนความเข้มข้นที่ไม่ทราบค่าก็นำไปวัดค่าการ

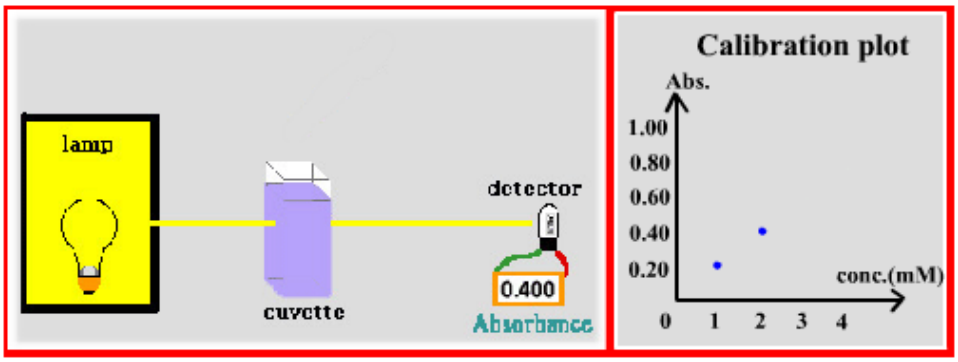
ดูดกลืนเช่นเดียวกัน แล้วนำค่าการดูดกลืนนั้นไปเทียบกับกราฟความเข้มข้นมาตรฐาน หรือคำนวณจากสมการเชิงเส้นของกราฟความเข้มข้นมาตรฐาน เพื่อย้อนกลับมาเป็นความเข้มข้น เราก็จะทราบค่าความเข้มข้นของสารนั้นได้ ดังภาพที่ 2.6 แสดงการวัดค่าการดูดกลืนแสง เพื่อนำมาสร้างกราฟความเข้มข้นมาตรฐาน



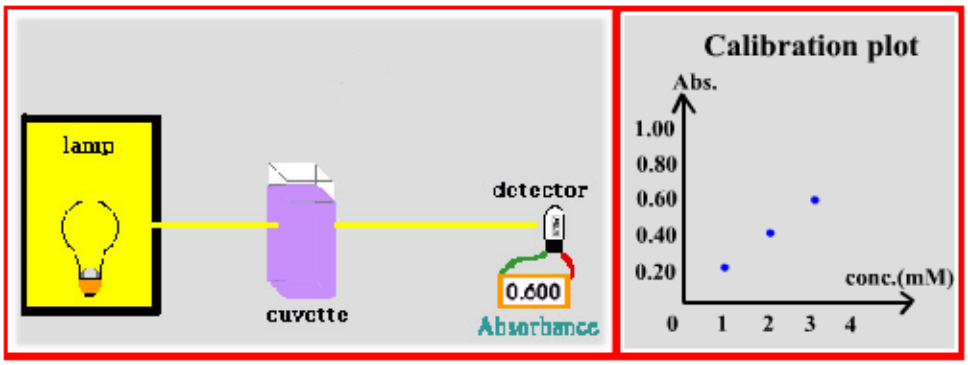
ก) วัดค่าการดูดกลืนแสงของน้ำเปล่า



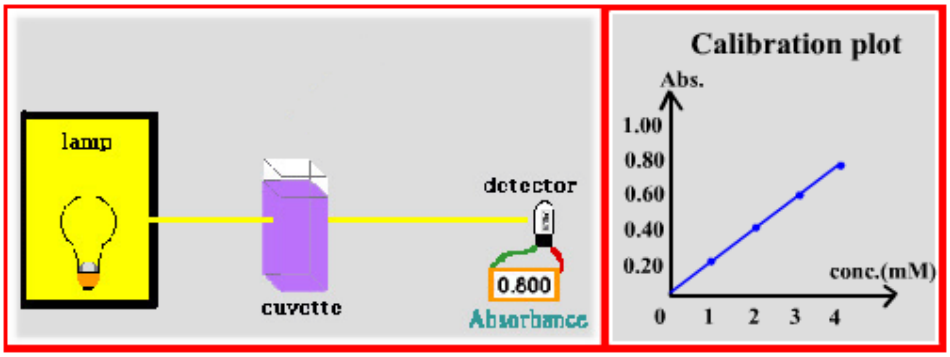
ข) วัดค่าการดูดกลืนแสงของสารละลายความเข้มที่ 1



ค) วัดค่าการดูดกลืนแสงของสารละลายความเข้มที่ 2



ง) วัดค่าการดูดกลืนแสงของสารละลายความเข้มที่ 3

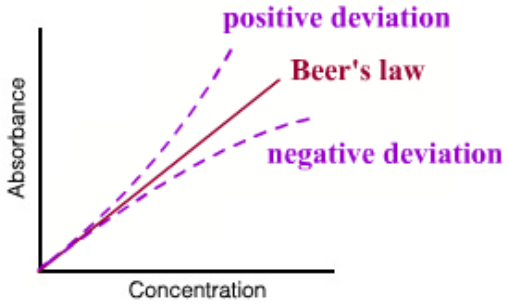


จ) วัดค่าการดูดกลืนแสงของสารละลายความเข้มที่ 4

ภาพที่ 2.7 แสดงการสร้างกราฟมาตรฐาน

2.5.3 การเบี่ยงเบนจากกฎของเบียร์ (Deviation from Beer's Law)

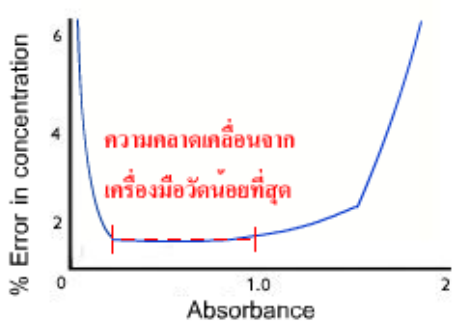
เราทราบมาแล้วจากกฎของเบียร์ว่า ค่าการดูดกลืนจะแปรผันโดยตรงกับความเข้มข้น ดังนั้นเมื่อเราพล็อตกราฟระหว่างค่าการดูดกลืน(A) กับความเข้มข้น(c) เมื่อความหนาของเซลล์คงที่ จะได้กราฟเส้นตรง แต่ในบางครั้งพบว่าอาจจะเกิดการเบี่ยงเบนได้ความสัมพันธ์ที่ไม่เป็นเส้นตรง ซึ่งอาจเป็นการเบี่ยงเบนในเชิงบวก (Positive Deviation) คือ ค่าการดูดกลืนให้แนวโน้มสูงกว่าปกติ ทำให้กราฟที่ได้โค้งขึ้น หรือเป็นการเบี่ยงเบนเชิงลบ (Negative Deviation) คือ ค่าการดูดกลืนมีแนวโน้มต่ำลงทำให้กราฟโค้งลง ดังภาพที่ 2.8



ภาพที่ 2.8 แสดงการเบี่ยงเบนที่ไม่เป็นไปตามกฎของเบียร์

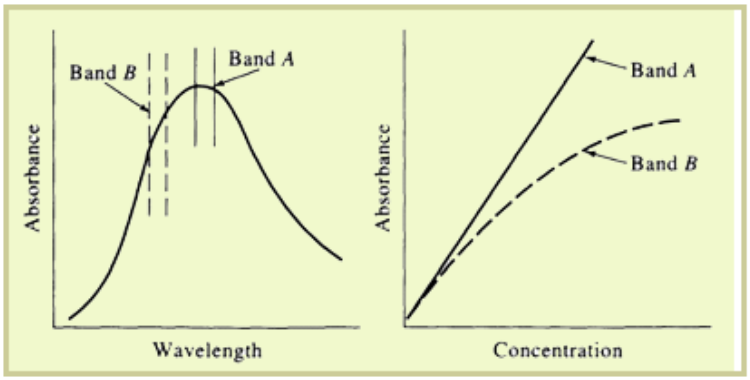
ซึ่งการเบี่ยงเบนดังกล่าวมาจากสาเหตุการเบี่ยงเบนจากเครื่องมือ(Instrumental Deviation) เกิดจาก

1) ความไม่แม่นยำของเครื่องมือวัด ในทางปฏิบัติพบว่าการวัดค่าการดูดกลืนของสารให้มีค่าอยู่ในช่วง A = 0.1-1.0 จะมีความคลาดเคลื่อนของการวัดความเข้มข้นน้อยมาก ประมาณ 1-2% ดังนั้นในปฏิบัติการจริงจะมีการปรับความเข้มข้นของสารละลายให้อยู่ในช่วง A = 0.1-1.0 ถ้าค่าการดูดกลืน อยู่ในช่วงที่มากกว่า หรือน้อยกว่านี้แสงที่ผ่านออกมายังเครื่องตรวจวัดจะน้อย หรือมากเกินไป ดังนั้นความคลาดเคลื่อนจึงมีสูง ดังภาพที่ 2.9



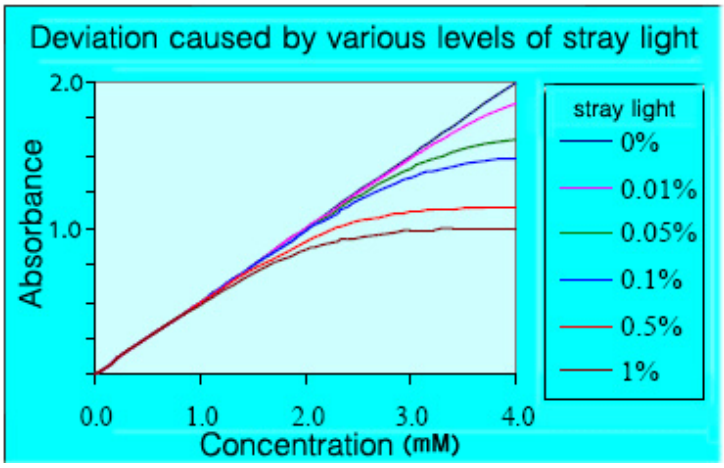
ภาพที่ 2.9 %ความเข้มข้นที่คลาดเคลื่อน (% Error in Concentration) อันเนื่องมาจากความคลาดเคลื่อนในการอ่านค่าการดูดกลืน (Absorbance) ของเครื่องมือ

2) แสงที่มีหลายความยาวคลื่น (Polychromatic Light) จากกฎของเบียร์อนุมานว่าความยาวคลื่นที่ผ่านสารละลายเป็นความยาวคลื่นเดี่ยว แต่ในทางปฏิบัติ ตัวแยกความยาวคลื่น (Monochromator) อาจแยกความยาวคลื่นให้เป็นคลื่นเดี่ยวไม่ได้ มีความยาวคลื่นอื่นปนมาด้วย ดังนั้นการวัดค่าการดูดกลืน อาจคลาดเคลื่อนได้ ถ้าความยาวคลื่นอื่นที่ปนมามีค่า Molar (ε) ของสารเปลี่ยนแปลงไปจากความยาวคลื่นที่เราเลือกมาก ดังภาพที่ 2.15



ภาพที่ 2.10 ผลของแสงที่มีหลายความยาวคลื่นต่อการเบี่ยงเบนจากกฎของเบียร์
แถบ A เป็นไปตามกฎของเบียร์เพราะ ε เปลี่ยนแปลงไม่มาก
แถบ B ไม่เป็นไปตามกฎของเบียร์ เพราะ ε เปลี่ยนแปลงอย่างมาก

3) คลื่นแสงรบกวน(Stray Light) คือการรบกวนของความยาวคลื่นที่เราไม่ได้เลือกผ่านเข้าไปตกกระทบบนตัวตรวจจับสัญญาณ (Detector) อาจเกิดจากการกระเจิงแสง (Scattering) หรือการสะท้อน(Reflection) หลายๆ ครั้งบนผิวของเกรตติ้งเลนส์ หรือฟิลเตอร์ เป็นต้น ผลของแสงรบกวนทำให้ค่าการดูดกลืนลดลง โดยเฉพาะที่ความเข้มข้นสูงๆ ถ้าค่าการดูดกลืนมีค่ามาก แสงที่ผ่านออกมาตกบนตัววัดมีน้อย ดังนั้นแสงรบกวนจึงส่งผลได้มากขึ้น ทำให้ได้กราฟที่ไม่เป็นเส้นตรง ดังภาพที่ 2.10



ภาพที่ 2.11 อิทธิพลของคลื่นแสงรบกวน(Stray Light)

ยกตัวอย่างเช่น สมมติว่าเครื่องสเป็กโตรโฟโต้มิเตอร์เครื่องหนึ่ง จะมีผลของแสงรบกวน 1% ของทุกๆ ครั้งที่ทำการวัด จากตารางเปรียบเทียบปริมาณของแสงรบกวนกับค่าการดูดกลืนแสงต่างๆ พบว่า ถ้าค่าการดูดกลืนแสงยังมีค่าน้อยหรือแสงที่ผ่านออกมา(%T) ยังมีค่ามากผลของแสงรบกวนจะถือว่ารบกวนน้อย เช่นแสงรบกวน 1% มีค่าน้อยมากเมื่อเทียบกับแสง100%T(แสงรบกวนคิดเป็น 0.01%ของค่า%T) แต่เมื่อไรก็ตามที่ค่าการดูดกลืนมีค่ามากจนกระทั่งแสงที่ผ่านออกมาเหลือน้อยมาก เช่น แสงผ่านออกมาเพียง 1%T ดังนั้นแสงรบกวน ซึ่งมี 1% จะมีค่าเท่ากันกับแสง %T เลยทีเดียว หรือแสงรบกวนคิดเป็น 50 % ของแสง %T

ตารางที่ 2.2 เปรียบเทียบปริมาณของแสงรบกวนกับค่าการดูดกลืนแสงต่างๆ

Absorbance	%T	%Stray Light	Ratio of %(Stray Light / T)
0	100	1	1/100
1	10	1	1/10
2	1	1	1/1

2.6 การเขียนโฟล์วชาร์ตเบื้องต้น [12]

การเขียนโฟล์วชาร์ต(Flowchart) เป็นเทคนิคหรือวิธีการอย่างหนึ่งสำหรับใช้เขียนแสดงอัลกอริทึม การเขียนโฟล์วชาร์ตจะใช้รูปภาพที่เป็นสัญลักษณ์มาตรฐานสากลนำไปเขียนแทนกิจกรรมต่างๆ ในแต่ละขั้นตอนของอัลกอริทึม ซึ่งกิจกรรมที่แตกต่างกันก็จะใช้สัญลักษณ์รูปภาพที่แตกต่างกัน โดยสัญลักษณ์ภาพนี้กำหนดตามมาตรฐานของ ANSI (American National Standards Institute) และ ISO (International Standard Organization)

การเขียนแสดงอัลกอริทึมด้วยโฟล์วชาร์ตสามารถออกแบบได้ง่าย ผู้อื่นสามารถเข้าใจโฟล์วชาร์ตได้ง่ายเพราะเป็นมาตรฐานสากล และเมื่อนำโฟล์วชาร์ตไปเขียนโปรแกรมก็จะทำได้สะดวกรวดเร็ว

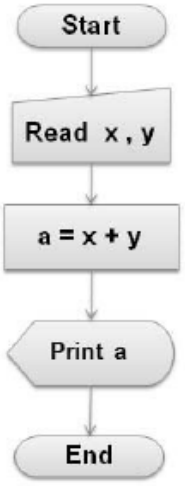
2.6.1 ประเภทของโฟล์วชาร์ต

โฟล์วชาร์ตทางคอมพิวเตอร์มี 2 ประเภท คือ โฟล์วชาร์ตระบบ(System Flowchart) และโฟล์วชาร์ตโปรแกรม(Program Flowchart)

1) โฟล์วชาร์ตระบบ(System Flowchart) เป็นการแสดงให้เราเห็นว่า ในระบบหนึ่งนั้นมีขั้นตอนในการทำงานอย่างไร ซึ่งจะมองเห็นในลักษณะภาพกว้างๆ ของระบบ แต่จะไม่เจาะลึกลงไปว่าในระบบว่าในแต่ละงานนั้นมีการทำงานอย่างไร คือ จะให้เห็นว่าจุดเริ่มต้นของงานเริ่มจากส่วนใด เป็นข้อมูลแบบใด มีการประมวลผลอย่างไร และจะได้ผลลัพธ์เป็นอย่างไรและเก็บอยู่ที่ใด ดังภาพที่ 2.47

ภาพที่ 2.12 ตัวอย่างโฟล์วชาร์ตระบบ (System Flowchart)

2) โฟล์วชาร์ตโปรแกรม (Program Flowchart) เป็นโฟล์วชาร์ตที่แสดงถึงขั้นตอนในการทำงานของโปรแกรมซึ่งจะแสดงการทำงาน ตั้งแต่เริ่มต้นในส่วนของการรับข้อมูล การคำนวณหรือการประมวลผล จนถึงการแสดงผลลัพธ์ จะทำให้การเขียนโปรแกรมสะดวกขึ้น ซึ่งโฟล์วชาร์ตชนิดนี้อาจสร้างมาจากโฟล์วชาร์ตระบบ โดยดึงเอาจุดที่เกี่ยวข้องกับคอมพิวเตอร์มาวิเคราะห์ว่าจะใช้ทำงานส่วนใดเพื่อที่จะให้ได้มาซึ่งผลลัพธ์ที่ต้องการ ดังภาพที่ 2.48



ภาพที่ 2.13 ตัวอย่างโฟล์วชาร์ตโปรแกรม (Program Flowchart)

2.6.2 ประโยชน์ของโฟล์วชาร์ต

1) ช่วยลำดับขั้นตอนการทำงานโปรแกรม และสามารถนำไปเขียนโปรแกรมโดยไม่สับสน

2) หากมีข้อผิดพลาดสามารถตรวจสอบ และแก้ไขโปรแกรมได้ง่าย

3) หากมีการพัฒนาระบบงานสามารถดัดแปลง แก้ไขได้อย่างรวดเร็ว

4) ทำให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมได้ง่ายและรวดเร็ว

5) การเขียนโฟล์วชาร์ตเป็นสากล สามารถนำไปเขียนได้ทุกภาษา

2.6.3 หลักการเขียนโฟล์วชาร์ตที่ดี

1) ใช้สัญลักษณ์ตามที่กำหนดไว้

2) ใช้สัญลักษณ์ที่มีขนาดเหมาะสมกับคำสั่ง

3) มีทางเข้าหรือจุดเริ่มต้น และทางออกหรือจุดสิ้นสุดเพียงทางเดียวเท่านั้น

4) ลำดับขั้นตอนการทำงานควรจะเริ่มจากบนลงล่าง หรือจากซ้ายไปขวา

5) ในสัญลักษณ์ใดๆ มีทางออกเพียงทางเดียว ยกเว้นสัญลักษณ์แสดงการตัดสินใจหรือ ทางเลือกสามารถมีทางออกได้อย่างน้อยสองทาง

6) เส้นทางเดินในโฟล์วชาร์ตควรชัดเจน เป็นระเบียบ

7) การกำหนดทิศทางการทำงานด้วยลูกศร ควรจะมีทิศทางจากบนลงล่าง หรือขวาไปซ้ายเท่านั้น

8) ในกระบวนการทำงานที่ต้องการเพิ่มคำอธิบายเข้าไปเพื่อให้เกิดความเข้าใจและจะสามารถทำได้โดยการใช้สัญลักษณ์หมายเหตุประกอบ

9) ไม่ควรโยงเส้นเชื่อมโฟล์วชาร์ตที่อยู่ไกลมากๆ ควรใช้สัญลักษณ์จุดเชื่อมต่อแทน

10) โฟล์วชาร์ตควรมีการทดสอบความถูกต้องของการทำงานก่อนนำไปเขียนโปรแกรม

2.6.4 สัญลักษณ์ของโฟล์วชาร์ต

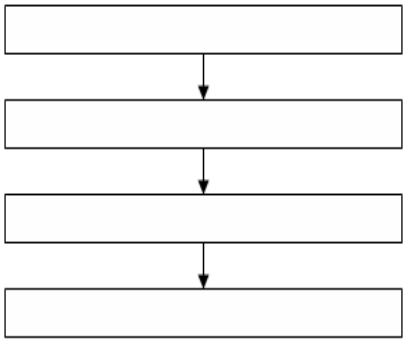
สัญลักษณ์ของโฟล์วชาร์ตมีรูปแบบและความหมายที่ต่างกันไป ดังภาพที่ 2.14

	ป้อนข้อมูลด้วยคีย์บอร์ด	Read a	ป้อนค่า a ทางคีย์บอร์ด
	แสดงผลลัพธ์ทางจอภาพ	Show a	แสดงค่า a ทางจอภาพ
	การคำนวณ หรือ การกำหนดค่า	A = B + C Num = 5	1.คำนวณค่า B+C แล้ว เก็บผลลัพธ์ไว้ที่ A 2. กำหนดค่า Num = 5
	แสดงข้อมูลหรือผลลัพธ์ ทางเครื่องพิมพ์ลงบน กระดาษ	Write a	พิมพ์ค่า a ทางกระดาษ
	การเปรียบเทียบ การตัดสินใจ	X>5	เปรียบเทียบ X > 5 หรือไม่
	การเตรียมการ การกำหนดค่าล่วงหน้า	For i=1 to 10	กำหนดให้ i=1 แล้ว เพิ่มขึ้นทีละ 1 จนถึง 10
	โปรแกรมย่อย	Call sub1	เรียกโปรแกรมย่อยชื่อ sub1
	รับหรือแสดงข้อมูลโดย ใช้เทปแม่เหล็ก	-	-
	รับหรือแสดงข้อมูลโดย ใช้จานแม่เหล็ก	-	-
	การเก็บข้อมูล	-	-
	เส้นแสดงทิศทางการ ทำงาน		ประมวลผลแล้ว ต่อด้วยการแสดงผล
	จุดเชื่อมต่อในหน้า เดียวกัน	X = 3 1	หลังกำหนด a=3 แล้วให้ ไปทำงานต่อที่จุด ต่อเนื่อง 1 หน้าเดียวกัน

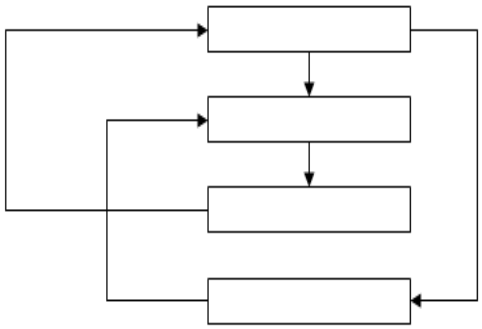
ภาพที่ 2.14 สัญลักษณ์โฟล์วชาร์ตในแบบต่างๆ

2.6.5 การเขียนโฟล์วชาร์ตแบบโครงสร้าง การเขียนแบบนี้มีประโยชน์ คือ ทำให้การไล่ขั้นตอนการทำงานทำได้ง่ายและเป็นระเบียบ ซึ่งมีหลักการเขียนอยู่ 3 ข้อ คือ

1) Sequence คือ การเขียนให้เป็นลำดับ ตั้งแต่ต้นจนจบ เป็นรูปแบบง่ายๆ ไม่มีการเปรียบเทียบใด ๆ มีทิศทางการไหลของข้อมูลเพียงทางเดียว ซึ่งอาจจะเป็นแบบบนลงล่าง หรือจากซ้ายไปขวาก็ได้ เช่น การให้คำนวณหาพื้นที่ของสี่เหลี่ยมผืนผ้า จะเขียนเป็นโฟล์วชาร์ตได้

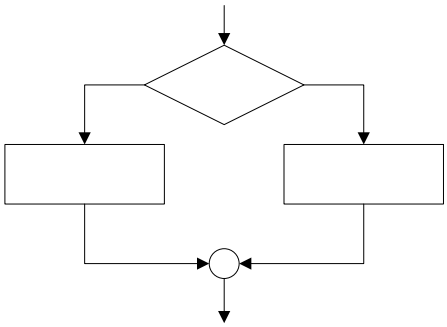


ภาพที่ 2.15 การเขียนแบบ Sequence



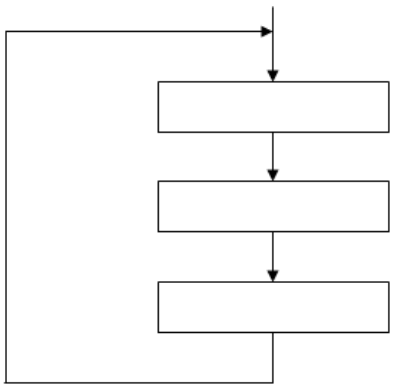
ภาพที่ 2.16 การเขียนแบบไม่เป็น Sequence

2) Selection เป็นทางเลือกของโปรแกรมซึ่ง จะต้องมีเพียงสองทางเลือกเท่านั้น ดังภาพที่ 2.52 และ หลังจากนั้นทางเลือกทั้งสองต้องมาพบกัน และทำงานในขั้นตอนต่อไป



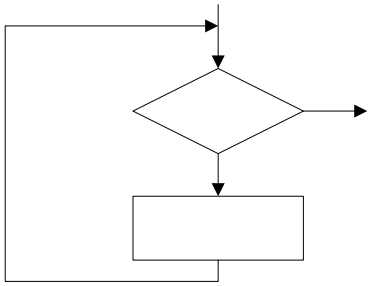
ภาพที่ 2.17 การเขียนแบบ Selection

3) Iteration คือการทำซ้ำเป็นการเขียนโฟล์วชาร์ตให้กลับมาทำงานในขั้นตอนอย่างเก่า จะเห็นว่าโฟล์วชาร์ต มีลักษณะวน ซึ่งเรียกว่า ลูป(Loop) และจะสังเกตว่า การวนลูป ดังภาพที่ 2.53 จะไม่มีทางออกไปทำงานในขั้นตอนต่อไปได้เลย เพื่อที่จะทำให้ออกจากลูปได้จะต้องมีการเช็คเงื่อนไขก่อนถึงจะออกจากลูปได้



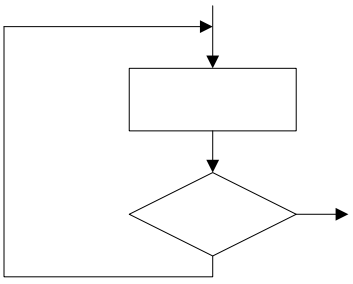
ภาพที่ 2.18 การเขียนแบบ Iteration

ในการเขียนโฟล์วชาร์ตจะมีลูปให้เลือกใช้ได้สองประเภทคือ Do While และ Do Until Do While จะทำการเช็คเพื่อที่จะออกจากลูปก่อนที่จะทำงานตามคำสั่งในลูป และเงื่อนไขเพื่อที่จะออกจากลูปจะต้องเป็นเท็จ ดังภาพที่ 2.54



ภาพที่ 2.19 การทำงานของคำสั่ง Do While

Do Until จะทำการเช็คเพื่อที่จะออกจากลูป ณ ตำแหน่งสุดท้ายของลูป และเงื่อนไขเพื่อที่จะออกจากลูปจะต้องเป็นจริง ดังภาพที่ 2.55



ภาพที่ 2.20 การทำงานของ คำสั่ง Do Until

2.7 โปรแกรมเกี่ยวข้องกับโครงงาน [13,14]

ในการทำโครงงานนั้น สิ่งที่ขาดไม่ได้คือ โปรแกรม ซึ่งจะเป็นสิ่งอำนวยความสะดวกและให้โครงงานสามารถก้าวหน้าได้อย่างรวดเร็ว และเป็นระเบียบร้อย โดยโปรแกรมที่ใช้ในการทำโครงงานชิ้นนี้ ได้แก่ โปรแกรม LabVIEW 2009 โปรแกรม Keil C51 โปรแกรม WSD โปรแกรม Protel โปรแกรม Solid Works และโปรแกรมภาษาซี เป็นต้น

2.7.1 โปรแกรม LabVIEW 2009

LabView ย่อมาจาก (Laboratory Virtual Instrument Engineering Workbench) เป็นซอฟต์แวร์ที่พัฒนาขึ้นโดยบริษัทเนชั่นเนล อิสทรูเมนตส์(National Instruments : NI) ซึ่งเป็นบริษัทผู้พัฒนาอุปกรณ์ และซอฟต์แวร์ที่ใช้ในงานวัดและงานควบคุมอัตโนมัติชั้นนำบริษัทหนึ่งโดยที่ LabView เป็นซอฟต์แวร์ตัวหนึ่งในบรรดาซอฟต์แวร์ทางด้านนี้ได้รับการพัฒนามาตั้งแต่ปีค.ศ. 1983 และเปิดตัว LabView 1.0 สำหรับเครื่องแมคอินทอชเป็นครั้งแรกเมื่อปี ค.ศ.1986 จนถึงปัจจุบันคือ LabView 2009

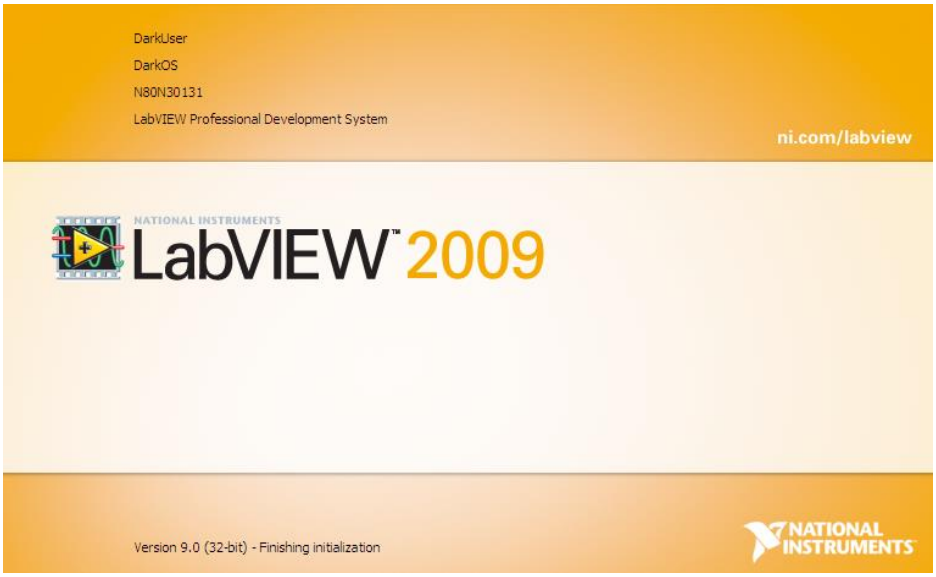
LabView คือ เครื่องมือที่ใช้พัฒนา Application (โปรแกรมประยุกต์) ชนิดหนึ่งเช่นเดียวกับ Visual Basic, Visual C++ แต่จะเป็นการเขียนโปรแกรมโดยใช้รูปในการพัฒนา (Graphical-Based Programming) ซึ่งจะแต่งต่างจากแนวคิด Text Based Programming เช่น Text-Based จะทำงานจากบนลงล่าง แต่ LabVIEW จะทำงานแบบ Dataflow คือ จะใช้บล็อกฟังก์ชั่นซึ่งแทนด้วยรูปไอคอนแทนการเขียนโปรแกรมย่อย(Subroutine) และใช้เส้นเชื่อมต่อระหว่างบล็อกฟังก์ชั่นแทนการไหลของข้อมูลระหว่างโปรแกรมย่อยนั้นๆ คล้ายกับการเขียนโฟล์วชาร์ต หรือบล็อกไดอะแกรมของโปรแกรม สิ่งที่ทำให้ LabView ต่างจากโปรแกรมอื่นทั่วไปคือ ในการพัฒนาโปรแกรมใช้งานทางด้านงานวัดและงานควบคุมอัตโนมัติ ซึ่งถือเป็นเป้าหมายสำคัญของ LabView โดยจะมีเครื่องมือ และไลบรารีที่สนับสนุนการใช้งานทางด้านนี้ไว้อย่างมากมายให้ผลลัพธ์ออกมาในรูปแบบของเครื่องมือเสมือนจริง(Virtual Instrument หรือ VI)

1) การใช้งานโปรแกรม LabVIEW 2009

เมื่อต้องการเข้าสู่โปรแกรม LabVIEW 2009 ให้ดับเบิ้ลคลิ๊กที่ไอคอน LabVIEW 2009 ดังภาพที่ 2.56

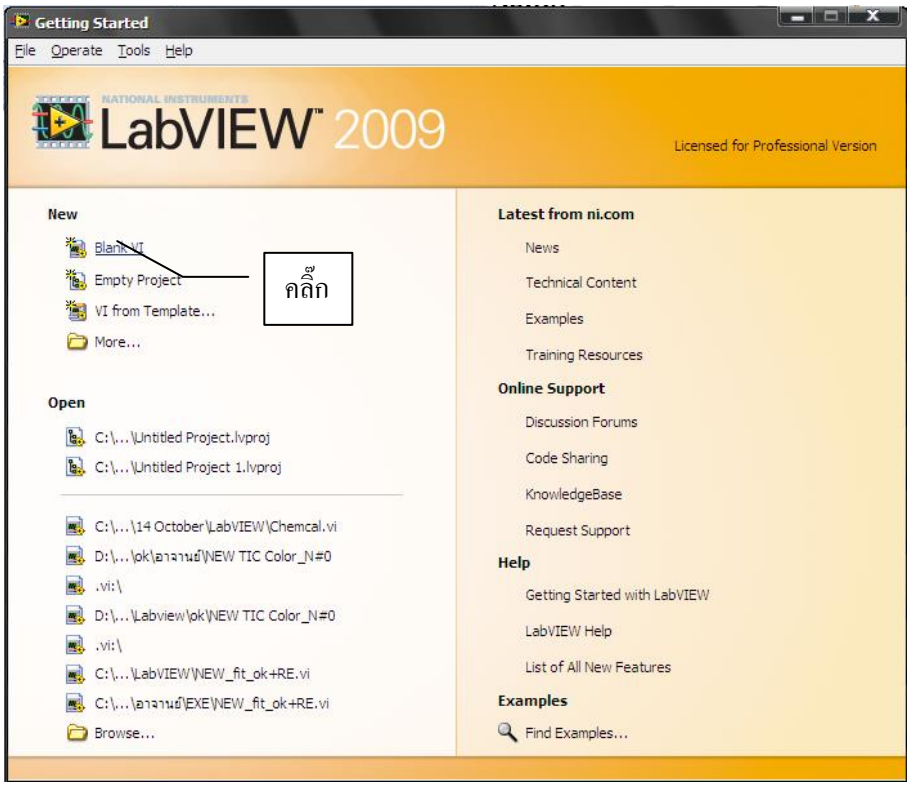
ภาพที่ 2.21 ไอคอน LabVIEW 2009

เมื่อเข้าสู่โปรแกรม LabVIEW 2009 จะขึ้นหน้าต่างแรก ดังภาพที่ 2.22



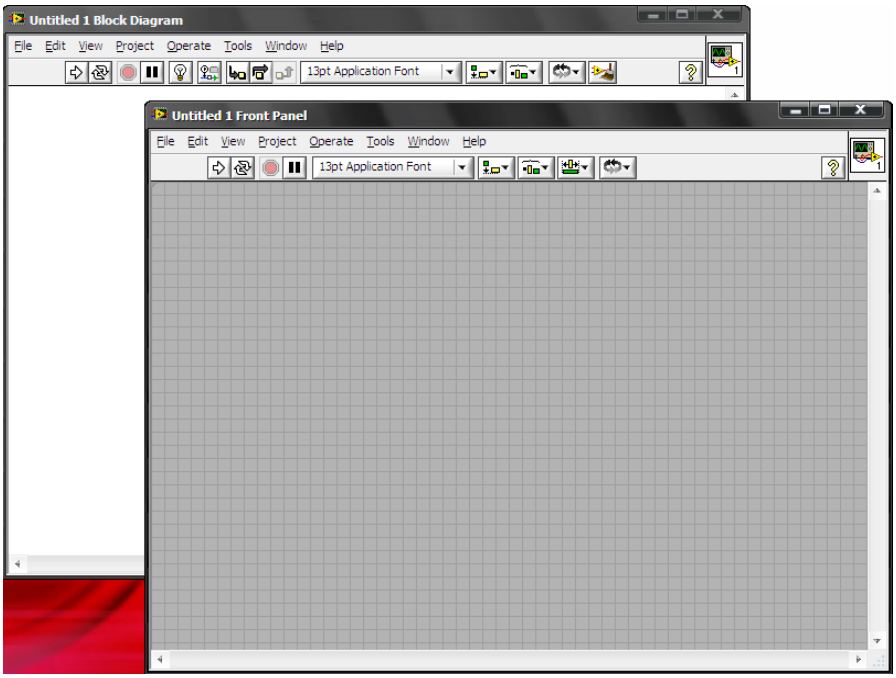
ภาพที่ 2.22 หน้าต่างการเข้าโปรแกรม LabVIEW 2009

นั้นก็จะเข้าสู่เมนูหลักของโปรแกรม ดังภาพที่ 2.23



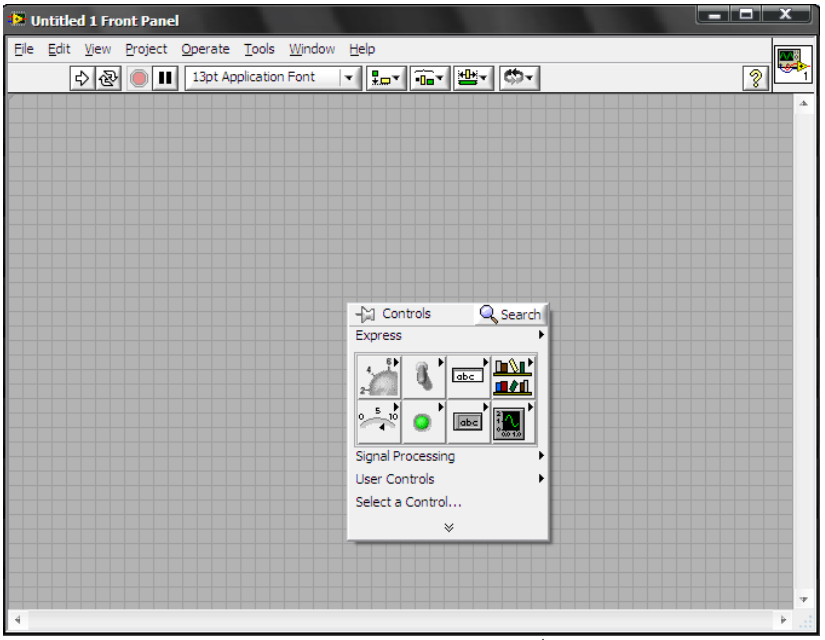
ภาพที่ 2.23 เมนูหลักของโปรแกรม LabVIEW 2009

เริ่มต้นการเขียนโปรแกรมโดยทำการเลือกคลิ๊กที่ Blank VI ซึ่งจะได้หน้าเปล่า ดังภาพที่ 2.24



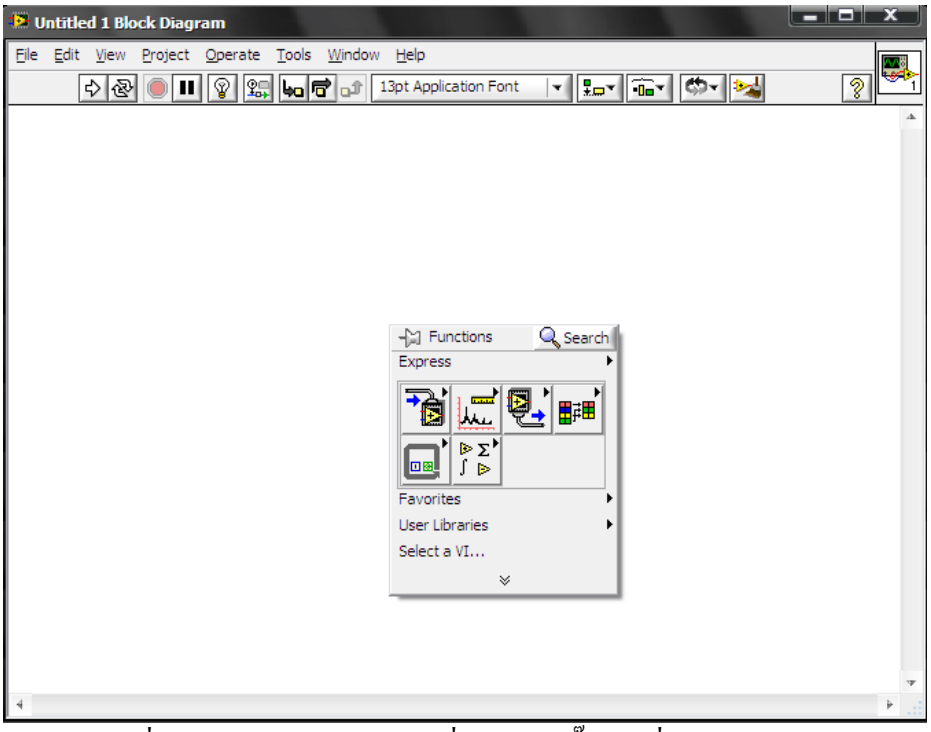
ภาพที่ 2.24 Front Panel และ Block Diagram

ไฟล์ LabVIEW มีนามสกุลเป็น *.vi ซึ่งไฟล์นี้จะประกอบด้วยหน้าต่าง User Interface พื้นที่เป็นตารางสีเทาซึ่งเรียกว่า Front Panel และหน้าต่างพื้นที่สีขาวสำหรับเขียนโค๊ดรูปภาพที่เรียกว่า Block Diagram ดังภาพที่ 2.26 จากนั้นเมื่อจะทำการเขียนโปรแกรมก็ทำการคลิ๊กขวาที่พื้นที่ของ Front Panel จะปรากฏอุปกรณ์สำหรับการสร้าง User Interface ที่เรียกว่า Controls Palette ซึ่งจะถูกแบ่งหมวดหมู่ตามประเภทของข้อมูล ดังภาพที่ 2.25



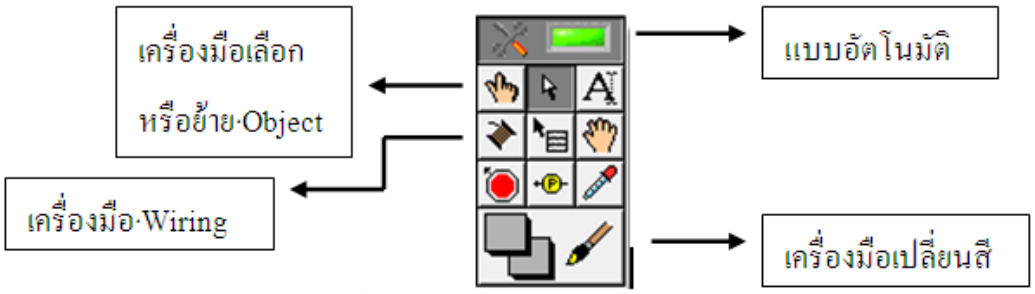
ภาพที่ 2.25 Controls Palette เมื่อทำการคลิ๊กขวาที่ Front Panel

คลิ๊กขวาที่พื้นที่ของ Block Diagram ก็จะได้ Functions Palette ซึ่งจะเป็นเครื่องมือสำหรับเขียนโค๊ด ดังภาพที่ 2.26



ภาพที่ 2.27 Functions Palette เมื่อทำการคลิ๊กขวาที่ Block Diagram

ทำการกด Shift และทำการคลิ๊กขวาหรือเลือกเมนู View >> Tools Palette ก็จะได้ Tools Palette ซึ่งเป็นเครื่องมือสำหรับเปลี่ยนรูปแบบเคอร์เซอร์(Cursor) ของเมาส์โดยปกติจะถูกเซ็ตให้เป็นแบบอัตโนมัติอยู่แล้ว(ช่องบนสุด) คือเคอร์เซอร์จะเปลี่ยนจะเปลี่ยนไปเองตามตำแหน่งที่เอาเมาส์ไปวาง เช่น วางใกล้กับ Terminal บน Block Diagram เมาส์จะเปลี่ยนรูปเป็นเครื่องมือต่อสาย(Wiring) เป็นต้น ดังภาพที่ 2.28



ภาพที่ 2.28 ส่วนต่าง ๆ ของ Tools Palette

วัตถุหรือที่เรียกว่า Object ที่อยู่บน Front Panel จะมีอยู่สามประเภทคือ

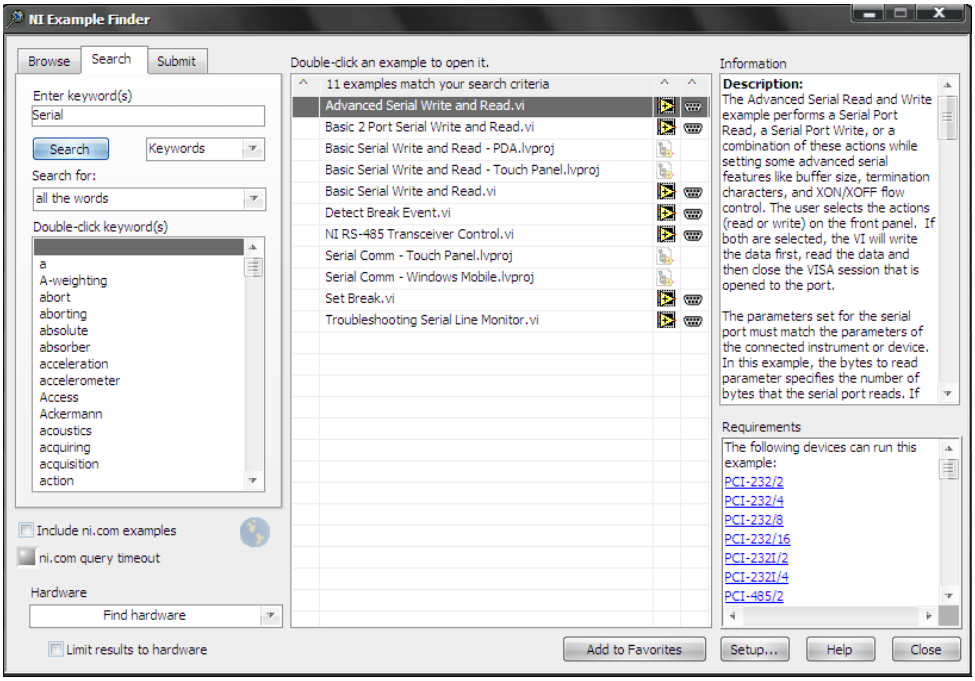
ก) Controls คือ ประเภทที่รับค่าจากผู้ใช้(Input) ซึ่งผู้ใช้จะสามารถพิมพ์ลงไปหรือใช้เมาส์คลิ๊กเพื่อทำการเปลี่ยนค่าก็ได้ เช่น เปลี่ยนมุม ปุ่มเลื่อน สวิตช์ เป็นต้น

ข) Indicators คือประเภทที่ใช้แสดงค่าเท่านั้น(Output) ผู้ใช้ไม่สามารถแก้ไขค่าบน Front Panel ได้ เช่น กราฟ มิเตอร์ หลอดไฟ เป็นต้น

ค) Decorations ซึ่งเป็น Object ที่ไม่เกี่ยวข้องกับโปรแกรม และในส่วนโค๊ดบน Block Diagram เลยแต่มีไว้เพื่อความสวยงามเป็นระเบียบของ Front Panel เท่านั้น

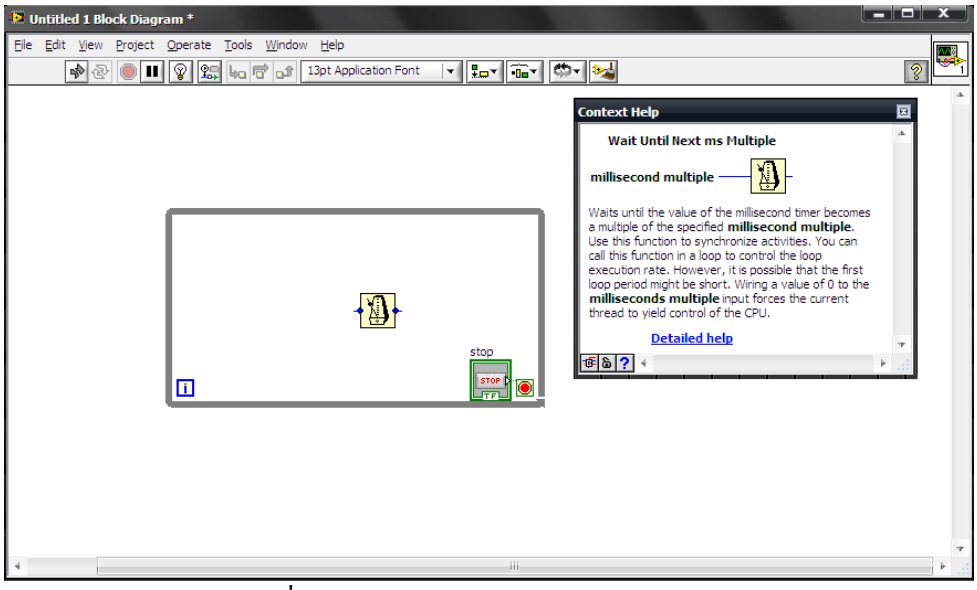
2) Help Utilities

เป็นเอกสารและตัวอย่างโปรแกรมให้ศึกษาทั้งที่ติดตั้งมาให้พร้อมกับ LabVIEW สามารถหาตัวอย่างโปรแกรมได้โดยเลือกที่เมนู Help >> Find Example จะปรากฏหน้าต่าง NI Example Finder ดังภาพที่ 2.29 ซึ่งจะแบ่งตัวอย่างโปรแกรมตามหมวดหมู่ในแท็บ Browse ส่วนในแท็บ Search จะเป็นการหาตัวอย่างตามคีย์เวิร์ดที่ใส่เข้าไป และถ้าต้องการได้ตัวอย่างเพิ่ม สามารถหาได้จาก www.ni.com



ภาพที่ 2.29 หน้าต่าง NI Example Finder

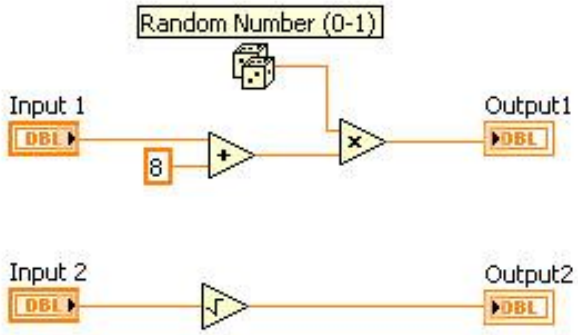
ในกรณีที่เปิดตัวอย่างแล้วไม่รู้จักโค๊ดบางตัว สามารถขอความช่วยเหลือได้จาก Context Help โดยเลือกที่เมนู Help >> Show Context Help หรือกด Ctrl+H ก็จะเห็นหน้าต่างเล็กๆ ทางมุมบน ดังภาพที่ 2.30 จากนั้นให้เอาเมาส์ไปวางบนสิ่งที่ต้องการคำอธิบาย หน้าต่าง Context Help จะแสดงคำอธิบายสั้นๆ ของโค๊ดตัวนั้นทันทีว่ามีหน้าที่อะไร มี อินพุต/เอาต์พุต อะไรบ้าง และถ้าต้องการเอกสารแบบเต็มรูปแบบก็ให้คลิ๊กที่ลิงก์ Detailed Help ซึ่งจะโยงไปหาเอกสารที่มีเอกสารแบบละเอียด



ภาพที่ 2.30 หน้าต่าง Context Help บนมุมขวา

3) Dataflow Programming Concept

หลักการทำงานของ LabView ซึ่งเป็นภาษาแบบกราฟฟิก จะมีข้อแตกต่างจากภาษาที่เป็นตัวหนังสือ อย่างเช่น โปรแกรมภาษาซี ที่มีการทำงาน(Execute) เป็นบรรทัดจากบนลงล่าง ทีละบรรทัดแต่ LabView จะมีหลักการทำงานแบบ Dataflow ซึ่งก็คือจะทำงานเป็นโหนด โดยโหนดใด ๆ จะทำงานได้ก็ต่อเมื่อโหนดนั้นมีอินพุตทุกตัวครบ ดังภาพที่ 2.65



ภาพที่ 2.31 การทำงานแบบ Dataflow

หลักการทำงานของโปรแกรมข้างบนคือ โปรแกรมประกอบด้วย โหนด 4 โหนด ได้แก่ การบวก, Random Number(การสุ่มตัวเลข), การคูณ, สแควร์รูท และถ้าพิจารณาจากหลักการข้างต้นโหนด ที่มีอินพุตครบและพร้อมที่จะทำงานได้เลยเมื่อรันโปรแกรมมีอยู่สามโหนด คือ การบวก, Random Number และสแควร์รูท ซึ่งสามตัวนี้จะทำงานพร้อมกัน ส่วนโหนดที่จะทำงานก่อนไม่ได้เลยก็คือ การคูณ เพราะจำเป็นต้องรอผลการบวกและ Random Number ก่อน

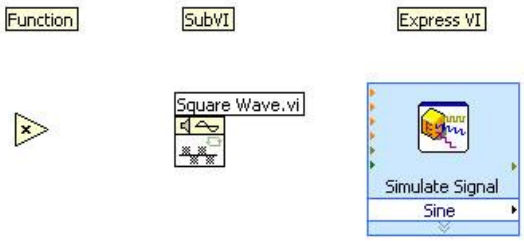
4) Block Diagram Nodes

คือไอคอนที่อยู่บน Block Diagram ซึ่งจะมีอินพุตหรือเอาต์พุต และจะทำงานตามหน้าที่เมื่อมีการรันโปรแกรม โดยแบ่งเป็นสามชนิดหลักๆ ได้แก่

ก) Functions คือ โหนดที่มีหน้าที่พื้นฐานของคอมพิวเตอร์ซึ่งไม่สามารถที่จะเจาะดูรายละเอียดภายในได้อีก เช่น การบวก, การคูณ, การเปิด-ปิดไฟล์ เป็นต้น

ข) SubVIs หรือในภาษาซอฟต์แวร์อาจเรียกว่า Subroutine หรือ Subprogram คือโปรแกรมย่อยที่ถูกเขียนขึ้นมาเพื่อถูกนำมาเรียกใช้ในอีกโปรแกรมหนึ่ง ดังภาพที่ 2.32 ซึ่งสามารถเปิดเข้าไปดู Front Panel และ Block Diagram ของมันได้เมื่อดับเบิ้ลคลิ๊กที่ตัวไอคอน SubVIs

ค) Express VIs เป็น SubVI ประเภทพิเศษ คือเมื่อเลือก Express VI มาวางบน Block Diagram มันจะปรากฏหน้าต่าง Configuration ขึ้นมาเพื่อให้ป้อนค่า Parameters ต่าง ๆ ตามต้องการและเมื่อป้อนค่าเสร็จ ก็จะสร้างโค๊ดไว้ภายในโดยอัตโนมัติตามที่ได้ตั้งค่าไว้ ซึ่งความสามารถของ Express VI นี้ทำให้แทบไม่ต้องต่อสาย Input เลยเพราะ Parameters ทั้งหมดถูกสร้างและเก็บอยู่ภายในเรียนร้อยแล้ว จึงทำให้การเขียน LabVIEW ง่ายและเร็วมากขึ้น สังเกตง่าย ๆ Express VI จะมีไอคอนใหญ่ที่พื้นหลังเป็นสีฟ้า ดังภาพที่ 2.32

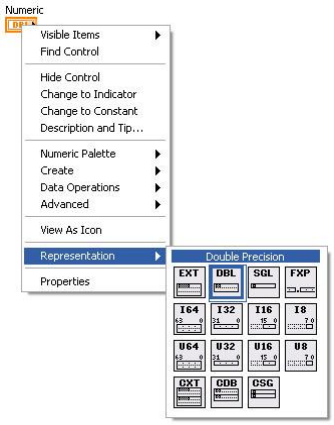


ภาพที่ 2.32 Block Diagram Nodes ในแบบต่างๆ

5) Data Types(ประเภทข้อมูล)

ในการเขียนโปรแกรมโดยทั่วๆ ไปจะต้องมีการประกาศตัวแปร ก่อนที่จะใช้ตัวแปรนั้น แต่สำหรับ LabVIEW นั้นจะจัดการให้เองหมดโดยผู้ใช้ไม่ต้องทำเอง เพียงแค่เลือกประเภทของข้อมูลมาวางบนโค๊ดเท่านั้น ประเภทของข้อมูลใน LabVIEW มีหลายอย่างที่ใช้กันเหมือนกับในภาษาอื่นๆ และยังมีอีกบางประการที่ใช้ใน LabVIEW เท่านั้น

ก) Numeric คือข้อมูลประเภทตัวเลข มีทั้งจำนวนเต็ม ซึ่งใน Block Diagram จะเห็นเป็นสีน้ำเงิน และจำนวนทศนิยมจะเห็นเป็นสีส้ม วิธีการเปลี่ยนประเภทของตัวเลข ให้คลิ๊กขวาที่ตัวเลขนั้นแล้วเลือก Representation และเลือกประเภทของตัวเลขได้เลย ซึ่งแถวบนจะเป็นตัวเลขประเภททศนิยม(จำนวนจริง) แถวสองคือ ตัวเลขจำนวนเต็ม(Integer) แถวสามคือจำนวนเต็มแบบไม่มีเครื่องหมายติดลบ(Unsigned Integer) และแถวล่างสุดคือตัวเลขเชิงซ้อน (Complex Number) ดังภาพที่ 2.33



ภาพที่ 2.33 การปรับเปลี่ยนค่าข้อมูลประเภทตัวเลข

ข) Boolean คือข้อมูลประเภทที่มีสองค่า คือ “จริง” และ “เท็จ” บน Block Diagram จะแสดงสีของข้อมูลนี้ด้วยสีเขียว และสำหรับบน Front Panel ตัว Boolean Control จะมีคุณสมบัติสวิตช์ ซึ่งมีหลายประเภท วิธีเปลี่ยนชนิดของสวิตช์ให้คลิ๊กขวาที่ Boolean Control บน Front Panel แล้วเลือก Mechanical Action ดังภาพที่ 2.34 จะมีด้วยกัน 6 รูปแบบดังนี้

Switch When Pressed คือ สวิตช์แบบกดติดกดดับ

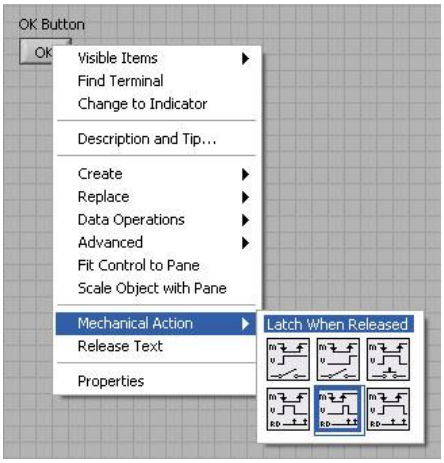
Switch When Released คือ สวิตช์แบบกดติดกดดับ แต่จะมีผลเมื่อปล่อยมือแล้วเท่านั้น

Switch Until Released คือ สวิตช์แบบกดติดปล่อยดับ

Latch When Pressed คือ สวิตช์จะเปลี่ยนค่าทันทีเมื่อทำการคลิ๊กแล้วจะเด้งกลับเป็นค่าเดิมเมื่อโปรแกรมรับรู้ ถึงแม้ว่าจะยังไม่ปล่อยมือก็ตาม

Latch When Released คือ จากกดสวิตช์แล้วจะเปลี่ยนค่าก็ต่อเมื่อปล่อยมือ แล้วเด้งกลับเป็นค่าเดิมอีกทีเมื่อโปรแกรมรับรู้

Latch Until Released คือ การกดติดปล่อยดับ แต่จะมีการรอให้โปรแกรมอ่านค่าตอนปล่อยมือก่อน แล้วเปลี่ยนกลับค่าเดิม



ภาพที่ 2.34 วิธีการเปลี่ยนรูปแบบของสวิตช์

ค) String คือข้อมูลที่เป็นตัวอักษร ไอคอนของ String จะเป็นสีชมพู สำหรับการแสดงผลของ String บน Front Panel จะมีอยู่ 4 รูปแบบ

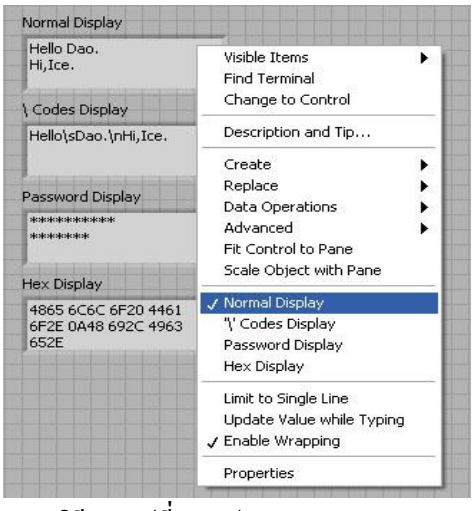
Normal Display คือ การแสดงผลปกติ

‘/’ Code Display คือ การแสดงผลแบบโค๊ด มีประโยชน์สำหรับการแสดงตัวอักษรที่ตาเปล่าไม่เห็น เช่น การเว้นวรรค(/s), แท็บ(/t), การขึ้นบรรทัดใหม่(/n) เป็นต้น

Password Display จะแทนตัวอักษรด้วยเครื่องหมาย *

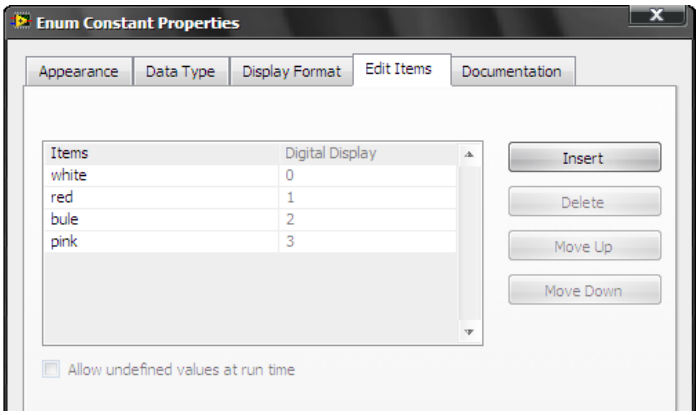
Hex Display แสดงผลเป็นรหัสเลขฐานสิบหก

ในการเปลี่ยนรูปแบบการแสดงผลสามารถคลิ๊กขวาที่ String บน Front Panel แล้วเลือกเปลี่ยนจากเมนูได้ทันที ดังภาพที่ 2.35



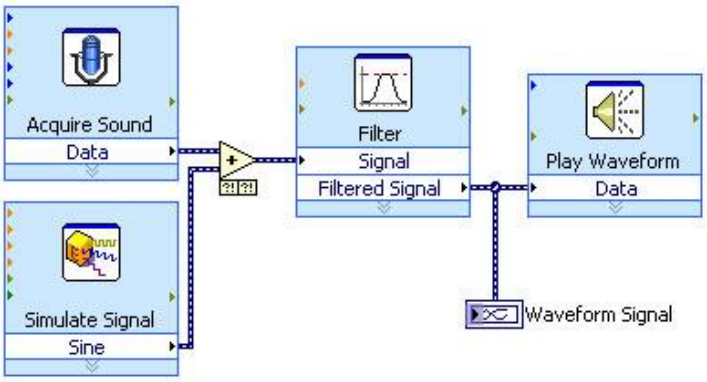
ภาพที่ 2.35 วิธีการเปลี่ยนรูปแบบการแสดงผลของ String

ง) Enum คือ ข้อมูลประเภทที่แสดงให้ผู้ใช้เห็นเป็นตัวหนังสือ แต่ค่าจริงของมันคือ ตัวเลขดังนั้นบน Block Diagram จึงมองเห็นข้อมูลประเภทนี้เป็นสีน้ำเงิน ซึ่งเหมือนกับจำนวนเต็มวิธีการสร้าง Enum Control ให้วาง Enum ลงไปบน Front Panel แล้วคลิ๊กขวาและเลือก Edit Item จากนั้นจะมีหน้าต่าง Enum Properties ก็พิมพ์ตัวหนังสือบนคอลัมน์ซ้าย ซึ่งจะไปจับคู่กับตัวเลขทางคอลัมน์ขวามือ ดังภาพที่ 2.36



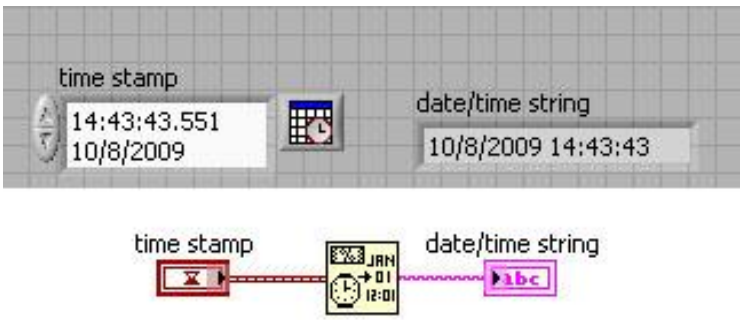
ภาพที่ 2.36 วิธีการสร้าง Enum Control

จ) Dynamic คือ ข้อมูลที่อยู่ในรูปของเว็ฟฟอร์มสัญญาณบน Block Diagram ถูกแสดงด้วยสีน้ำเงินเข้ม เส้นหนา ซึ่งภายในประกอบด้วยข้อมูลมากมาย เช่น Array ของเว็ฟฟอร์ม, Time Stamp, ชื่อของสัญญาณ ฯลฯ ข้อมูลประเภท Dynamic นี้ส่วนใหญ่ใช้ใน Express VI จำนวนการอ่าน, กำเนิด และวิเคราะห์สัญญาณ เป็นต้น แสดงการใช้งานแบบ Dynamic ดังภาพที่ 2.37



ภาพที่ 2.37 การใช้งานข้อมูลแบบ Dynamic

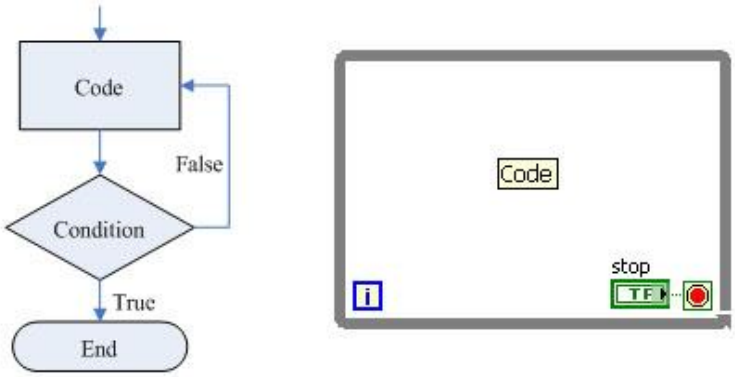
ฉ) Time Stamp คือข้อมูลที่ประกอบด้วยวันที่ และเวลาที่มีความละเอียดถึงมิลลิวินาที Time Stamp บน Block Diagram จะมีหน้าตาเป็นสีน้ำตาลเส้นหนา สามารถนำมาแปลงให้เป็นวันที่และเวลาแบบ String ได้ดังภาพที่ 2.38



ภาพที่ 2.38 การใช้งานข้อมูลแบบ Time Stamp

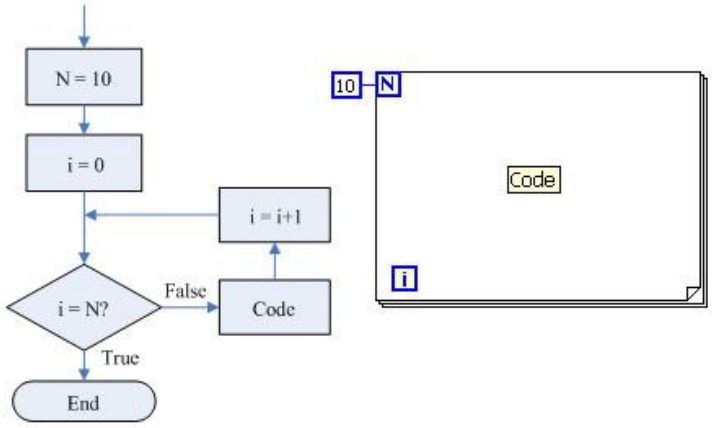
6) การเขียนโค๊ดให้มีการทำซ้ำหรือวนลูปเป็นสิ่งที่ใช้บ่อยมากสำหรับการเขียนโปนแกรม เนื่องจากการคำนวณหลายอย่างจำเป็นจะต้องทำในลักษณะซ้ำๆ การทำลูปจึงเป็นเทคนิคสำคัญที่จะต้องรู้ สำหรับ LabVIEW ลูปคือ กรอบสี่เหลี่ยมที่ล้อมรอบโค๊ดที่จะรันซ้ำเอาไว้มี 2 ประเภทตามลักษณะการใช้งาน

ก) While Loop คือกรอบสีเทา ประกอบด้วย Terminal Index ซึ่งจะส่งค่าตัวเลขออกมาบอกว่าตอนนี้ While Loop วิ่งมากี่ครั้งแล้วโดยเริ่มนับจากศูนย์ และ While Loop จะรันจนกว่า Terminal เงื่อนไข จะได้รับ Boolean ค่า True จึงจะหยุดรัน ซึ่งเงื่อนไขตัวกลมสีแดงนี้เรียกว่า Stop If True ก็คือ หยุดเมื่อได้รับค่าจริงนั่นเอง การรัน While Loop เป็นไปตามโฟล์วชาร์ต ดังภาพที่ 2.39



ภาพที่ 2.40 การใช้งาน While Loop

ข) For Loop ในส่วนนี้มีไว้สำหรับการรันลูปที่แน่นอน ว่าต้องการรันทั้งหมดกี่ครั้งโดยจำเป็นต้องระบุจำนวนครั้ง(N) ของ For Loop ไว้ก่อนมิฉะนั้นจะรันโค๊ดไม่ได้ ส่วนตัว Index จะเหมือนกับ While Loop คือให้ผลเป็นตัวเลขแสดงจำนวนลูปที่รันไปแล้วโดยเริ่มนับจากศูนย์ ซึ่งกลไกใน For Loop เป็นดังโฟล์วชาร์ต ดังภาพที่ 2.41

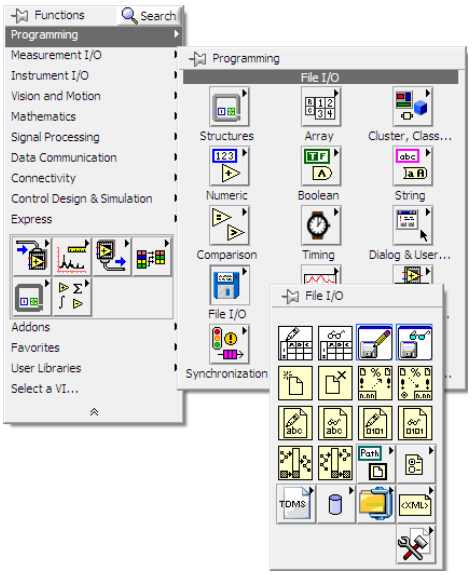


ภาพที่ 2.41 การใช้งาน For Loop

7) การบันทึกและการอ่านไฟล์

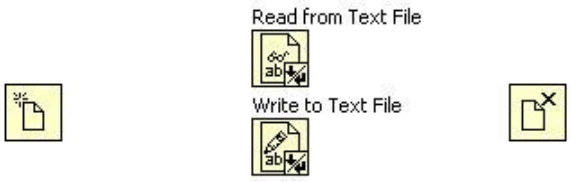
การบันทึกข้อมูลจาก LabVIEW ลงไฟล์ และการอ่านข้อมูลจากไฟล์แล้วนำไปใช้งานใน LabVIEW เป็นสิ่งสำคัญที่จะต้องนำมาใช้งานอยู่เสมอ ในหัวข้อนี้จะพูดถึงเรื่องของไฟล์ที่บันทึกเป็นตัวอักษรหรือ Text File ซึ่งเป็นโค๊ดในระบบ ASCII ที่สามารถบันทึกหรือนำมาเปิดด้วยโปรแกรมต่าง ๆ ได้เช่น Note Pad, MS Word, Ms Excel ฯลฯ และในการเขียนข้อมูลเข้า LabVIEW มี 2 แบบดังนี้

ก) High Level File I/O ในโปรแกรม LabVIEW นั้นมี VI สำเร็จรูปสำหรับการเขียนและอ่านไฟล์โดยที่ไม่ต้องสนใจเรื่องการเปิดไฟล์ปิดไฟล์ในระดับล่าง ซึ่งเรียกว่าแบบ High Level ซึ่ง VI เหล่านี้จะอยู่ใน Functions Palette : >> File I/O สี่ตัวแถวบน ดังภาพที่ 2.42



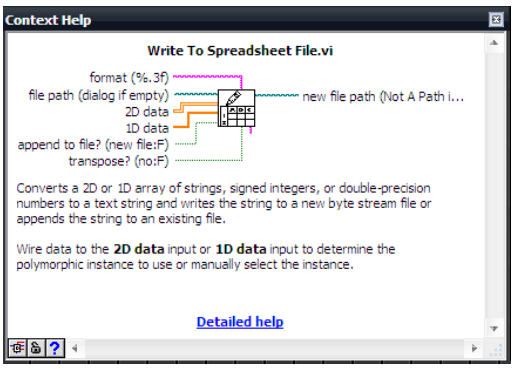
ภาพที่ 2.42 เมนูการบันทึกและการอ่านไฟล์

ข) Low Level File I/O มีรูปแบบการเขียนอ่านไฟล์ โดยเริ่มจากการเปิดไฟล์หรือสร้างไฟล์ใหม่ จากนั้นก็เริ่มการเขียนและการอ่านไปเรื่อยๆ ในลูป และเมื่อจบลูปก็ปิดไฟล์ ซึ่งจะเห็นว่าการเปิดไฟล์และปิดไฟล์ทำเพียงครั้งเดียวเพราะอยู่นอกลูป ถ้านำทุกอย่างเข้ามาในลูปแล้วก็จะทำให้การเขียนอ่านไฟล์ไม่มีประสิทธิภาพเพราะต้องเสียเวลาปิดไฟล์เปิดไฟล์ทุกครั้ง รูปแบบของ Low Level File I/O ดังภาพที่ 2.43

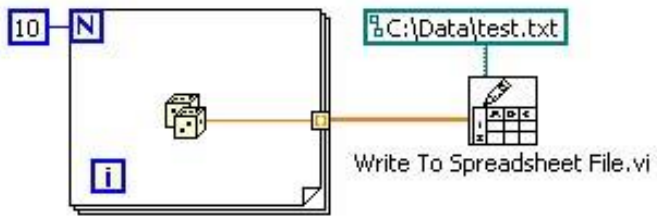


ภาพที่ 2.43 รูปแบบของ Low Level File I/O

ค) การเขียนข้อมูลเพื่อบันทึกไฟล์เป็นนามสกุล *.txt หรือ *.xls การบันทึกไฟล์นั้นทำได้โดยการใช้ฟังก์ชั่น Write To Spreadsheet File.Vi ดังภาพที่ 2.44 และการต่อข้อมูลหน้า Block Diagram ดังภาพที่ 2.78 เป็นตัวอย่างโปรแกรมการนำข้อมูลเก็บเป็นไฟล์ *.txt

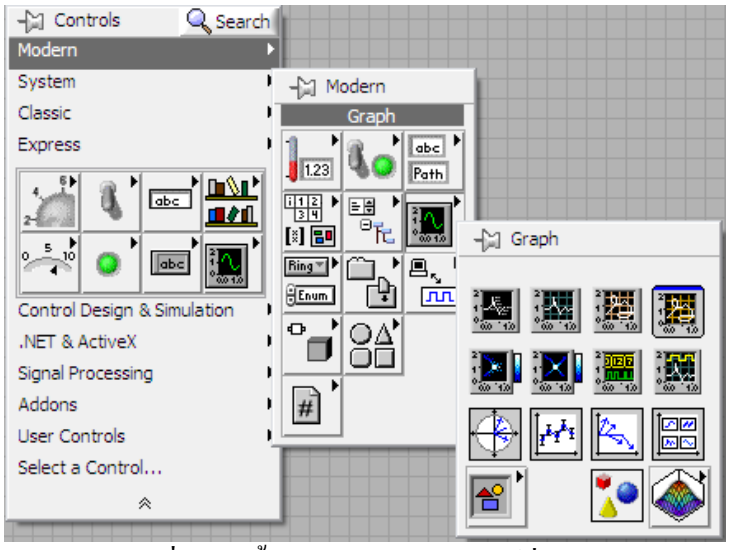


ภาพที่ 2.44 จุดต่อใช้งานของ Write to Spreadsheet File.vi



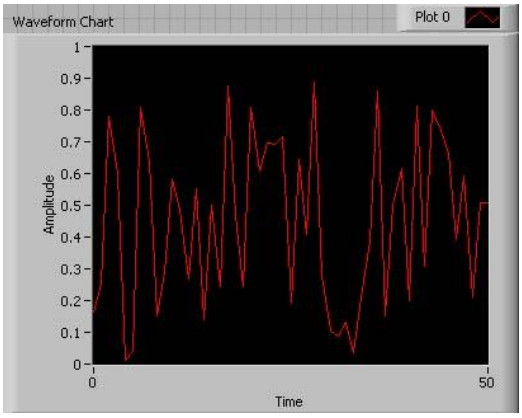
ภาพที่ 2.45 ตัวอย่างโปรแกรมการนำข้อมูลเก็บเป็นไฟล์ *.txt

8) การใช้งานกราฟฟังก์ชั่นกราฟของโปรแกรม LabVIEW 8.6 จะมีฟังก์ชั่นการใช้แสดงผลทางกราฟแบบสำเร็จรูปจึงง่ายต่อการใช้งาน โดยเริ่มต้นต้องทำการสร้างกราฟที่ต้องการจะให้แสดงผล โดยคลิ๊กขวาที่ Front Panel >> Controls>> Graph ดังภาพที่ 2.46



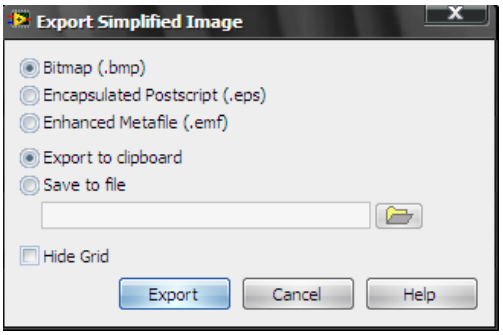
ภาพที่ 2.46 ขั้นตอนการเลือกใช้ฟังก์ชั่นกราฟ

ฟังก์ชั่นกราฟ นี้ยังมีกราฟอีกหลายแบบที่สามารถเลือกใช้ได้ตามต้องการ โดยในแต่ละแบบจะมีการใช้งานอย่างไรนั้นสามารถใช้ฟังก์ชั่น Help โดยเลือกที่เมนู Help >> Show Contact Help หรือกด Ctrl+H เมื่อนำเคอร์เซอร์ไปใกล้ๆ กับฟังก์ชั่นที่เลือกก็จะมีคำอธิบายปรากฏขึ้นบนมุมขวาของโปรแกรม ในการแสดงข้อมูลทางกราฟ ดังตัวอย่างกราฟชนิด Waveform Chart ดังภาพที่ 2.47

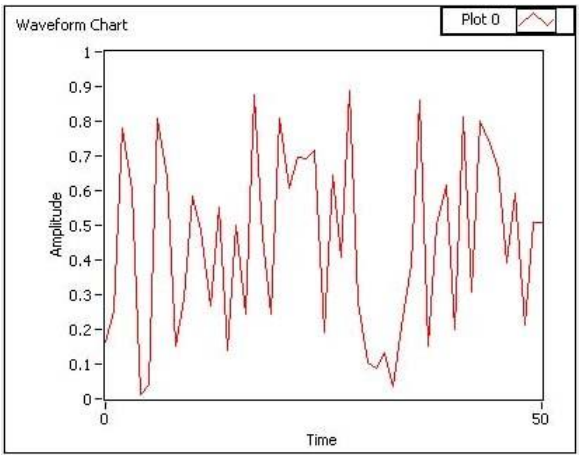


ภาพที่ 2.47 กราฟชนิด Waveform Chart

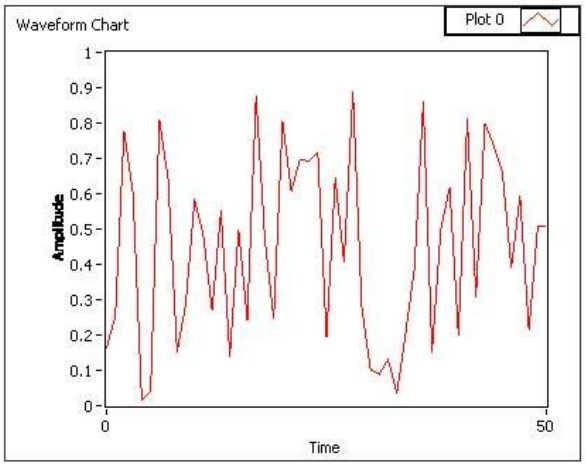
สามารถที่จะบันทึกเป็นไฟล์ภาพนามสกุล *.bmp และ *.emf ได้โดยการคลิ๊กขวาที่พื้นที่กราฟ จะมีข้อความขึ้นให้เลือกบันทึกเป็นไฟล์ภาพได้ 2 แบบคือ *.emf และ *.bmp ดังภาพที่ 2.48 และเมื่อทำการบันทึกแล้วผลที่ได้ ดังภาพที่ 2.49 - 2.50



ภาพที่ 2.48 การเลือกบันทึกข้อมูลให้เป็นไฟล์ภาพ



ภาพที่ 2.49 ไฟล์ภาพนามสกุล *.bmp

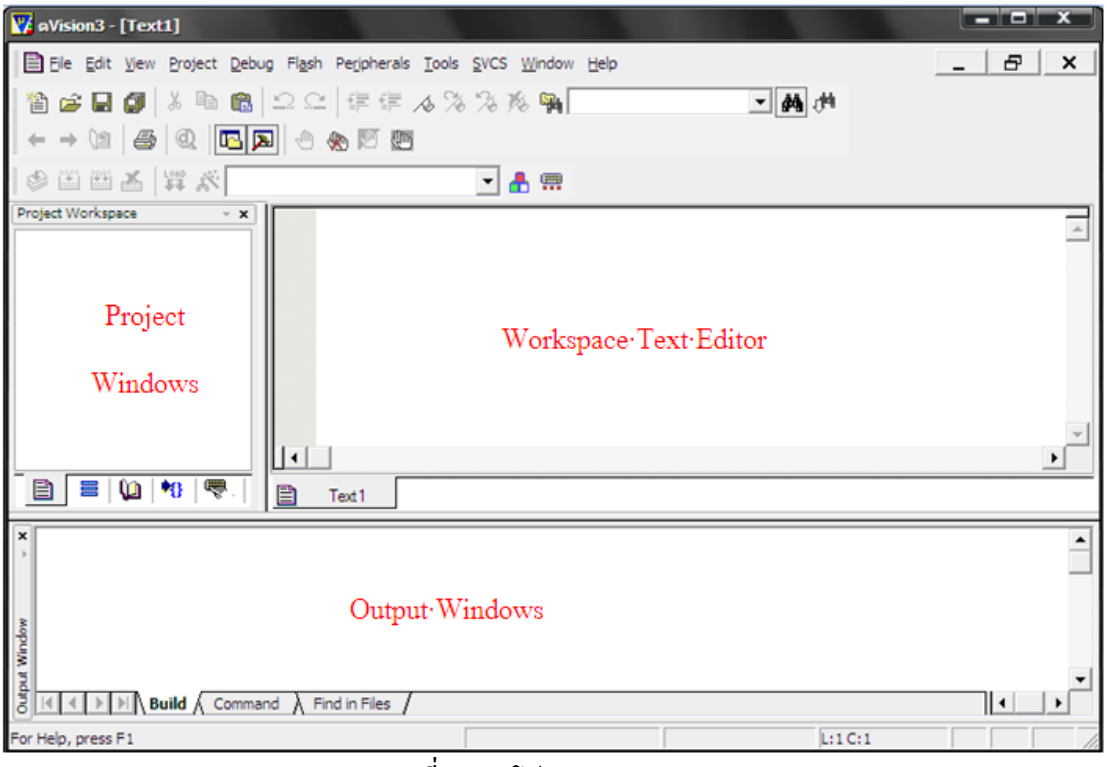


ภาพที่ 2.50 ไฟล์ภาพนามสกุล *.emf

2.8.2 โปรแกรม Keil C51

โปรแกรม Keil C51 นี้เป็นส่วนประกอบหลักๆ ในการใช้เขียนโปรแกรมภาษาซี เพื่อควบคุมการใช้งานของเครื่อง เมื่อต้องการเข้าสู่โปรแกรม Keil C51 ให้ดับเบิ้ลคลิ๊กที่ไอคอน Keil C51 ดังภาพที่ 2.51 และในหน้าต่างหลักของโปรแกรม Keil C51 ดังภาพที่ 2.52

ภาพที่ 2.51 โปรแกรม Keil C51



ภาพที่ 2.52 โปรแกรม Keil C51

ซึ่งประกอบไปด้วย

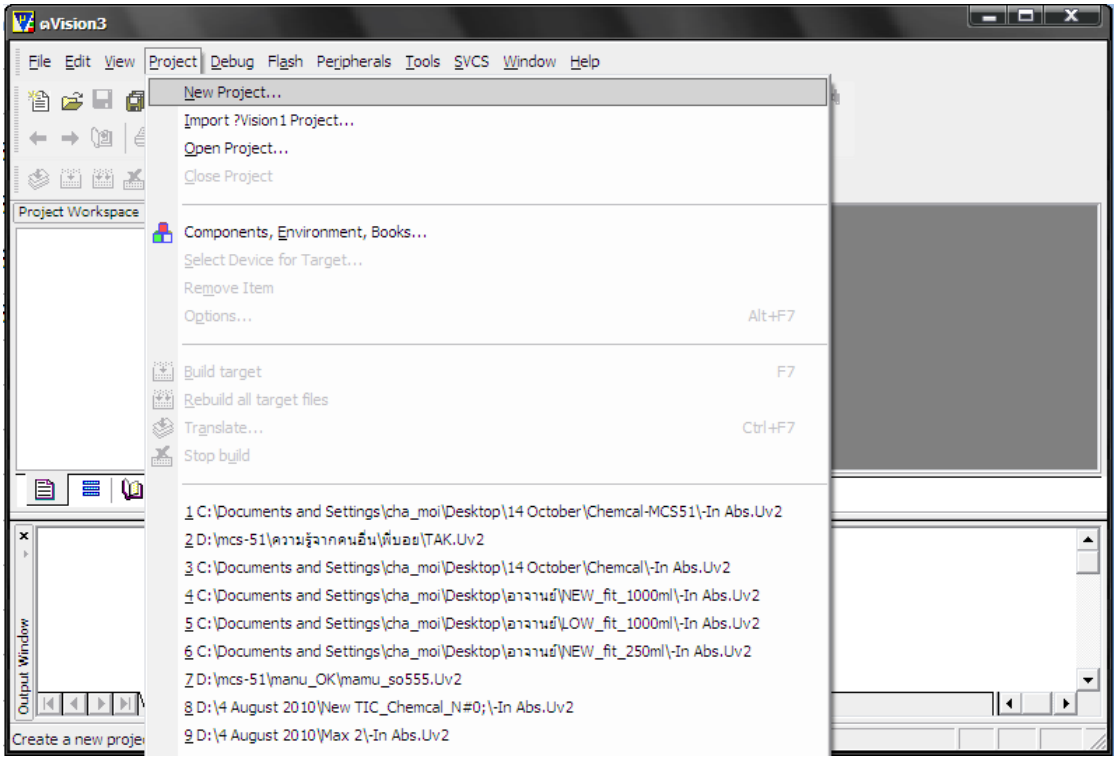
1) Project Windows ส่วนนี้ใช้สำหรับสร้างโปรแกรม และแปลคำสั่งในโปรแกรมให้อยู่ในรูป *.hex ไฟล์

2) Workspace Text Editor ส่วนนี้ใช้สำหรับเขียนโปรแกรมภาษา ซึ่งเวลาแก้ ไขต้องใช้หน้าต่างในส่วนนี้

3) Output Windows ส่วนนี้ใช้สำหรับตรวจสอบความผิดพลาดทางไวยากรณ์ของโปรแกรมที่เขียนขึ้น

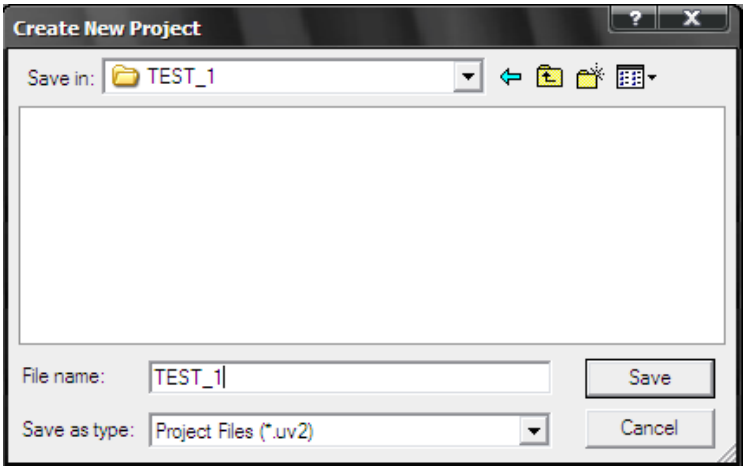
ขั้นตอนในการสร้างโปรเจ็ค

1) เลือกที่ Project/New Project เพื่อสร้างโปรเจ็คใหม่ขึ้นมา ดังภาพที่ 2.53



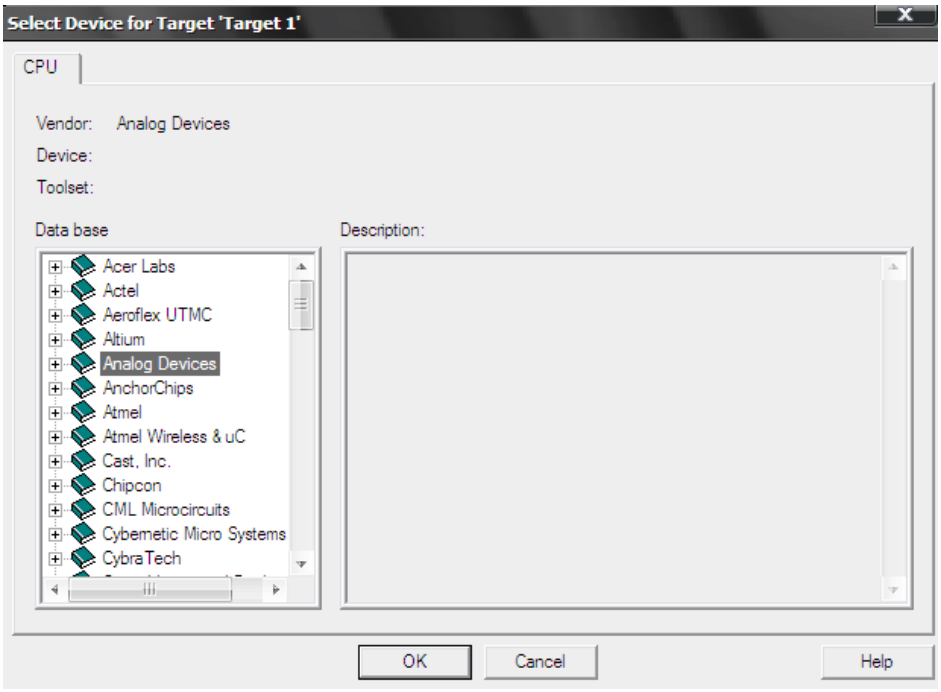
ภาพที่ 2.53 การสร้างโปรเจ็คใหม่

2) เมื่อเลือก New Project ขึ้นมาแล้วจะขึ้นหน้าต่างสำหรับใส่ชื่อโปรเจ็คและบันทึกงาน ดังภาพที่ 2.54 ซึ่งควรจะสร้างโฟลเดอร์ขึ้นมาใหม่ แล้วเข้าไป Save ในโฟลเดอร์นั้น ข้อสำคัญในการตั้งชื่อคือ ควรจะต้องตั้งให้สอดคล้องกับงานที่ทำ



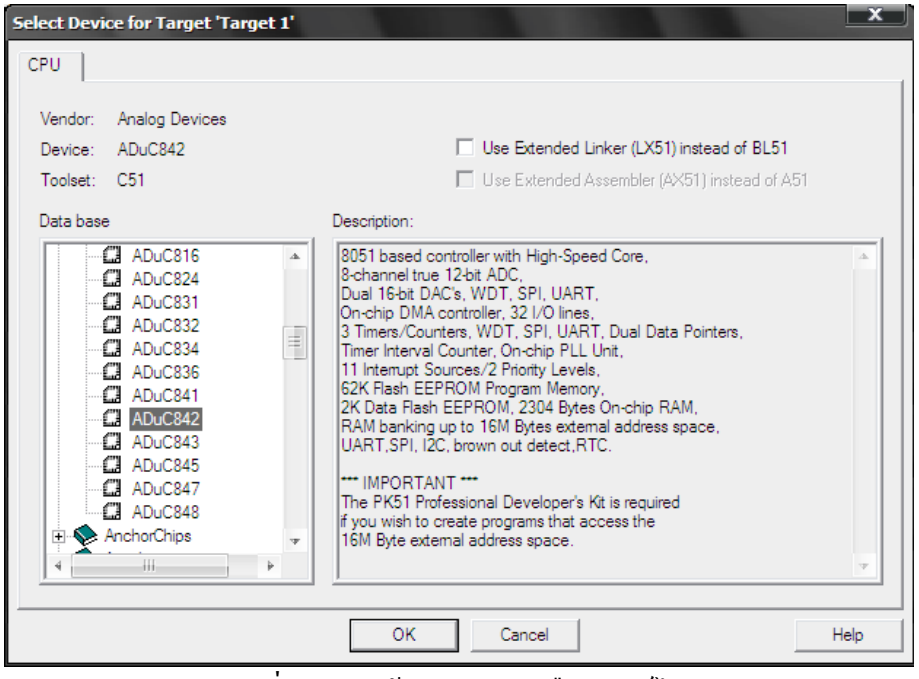
ภาพที่ 2.54 หน้าต่างสำหรับใส่ชื่อโปรเจ็คและบันทึกงาน

3) เมื่อคลิ๊ก Save จะปรากฏหน้าต่าง ดังภาพที่ 2.55 เพื่อให้ทำการเลือกไอซีที่จะทำการสร้างโปรเจ็ค แล้วจึงทำการเลือกยี่ห้อ ซึ่งในที่นี้จะใช้ยี่ห้อ Analog Devices



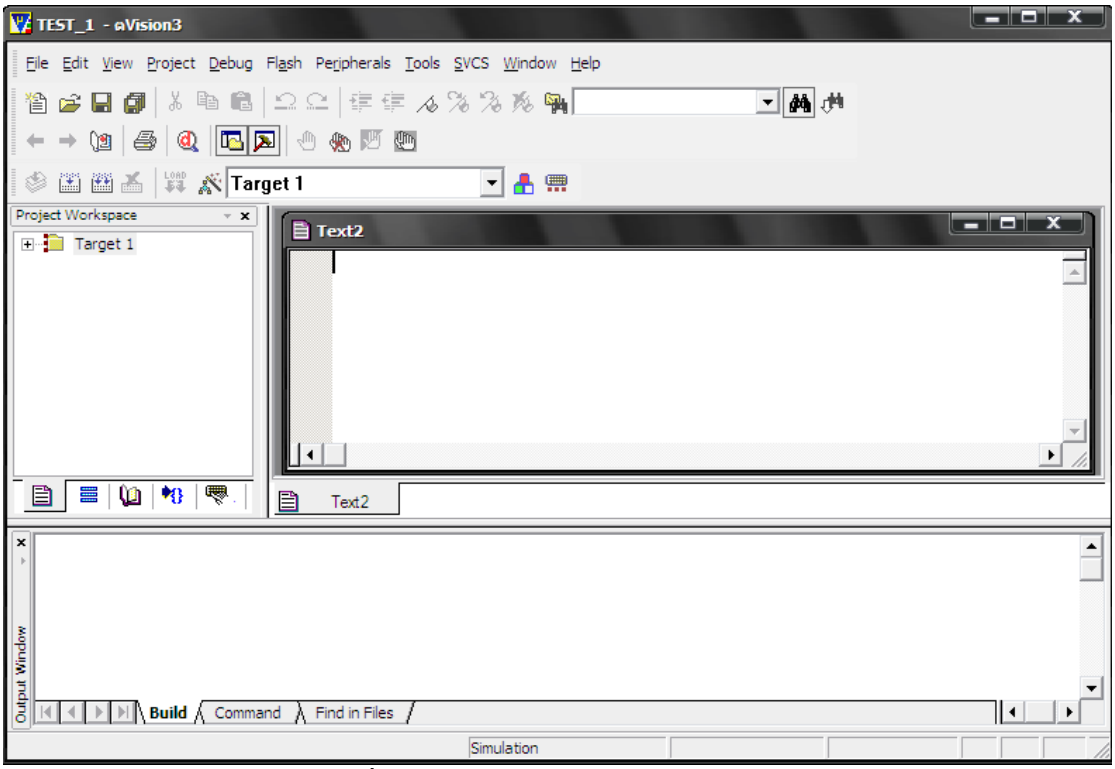
ภาพที่ 2.55 หน้าต่างสำหรับเลือกไอซี

4) เลือกเบอร์ไอซีที่ต้องการ แล้วคลิ๊ก OK ดังภาพที่ 2.56



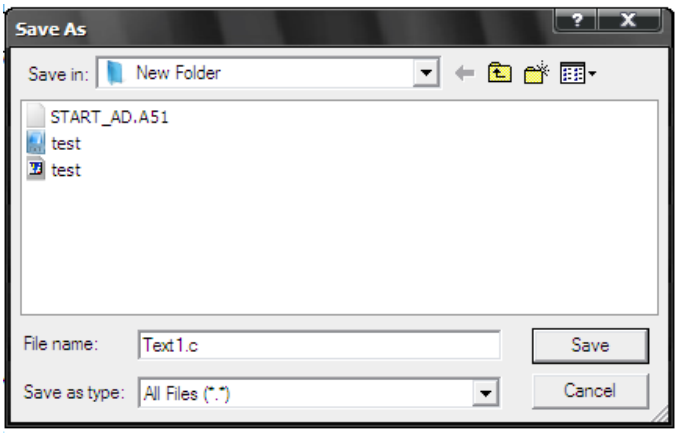
ภาพที่ 2.56 หน้าต่างสำหรับเลือกเบอร์ไอซี

5) เลือกที่ New จะขึ้นหน้าต่าง Text ขึ้นมาให้ทำการเขียนโปรแกรมที่ต้องการ ดังภาพที่ 2.57



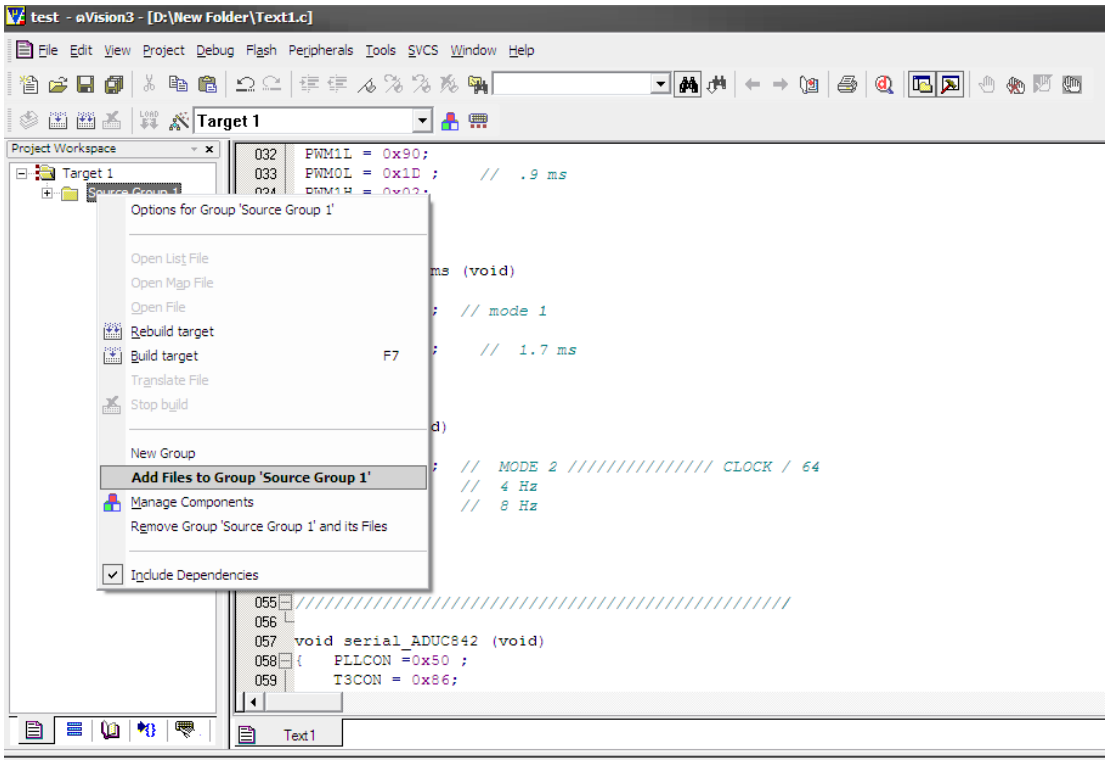
ภาพที่ 2.57 หน้าต่างสำหรับเขียนโปรแกรม

6) เมื่อเขียนโปรแกรมเสร็จแล้วให้ทำการบันทึกโปรแกรมโดยเลือกที่ Save ดังภาพที่ 2.91 ซึ่งในการ Save นั้นจำเป็นจะต้องใส่นามสกุล *.c เช่น test.c, robo51.c



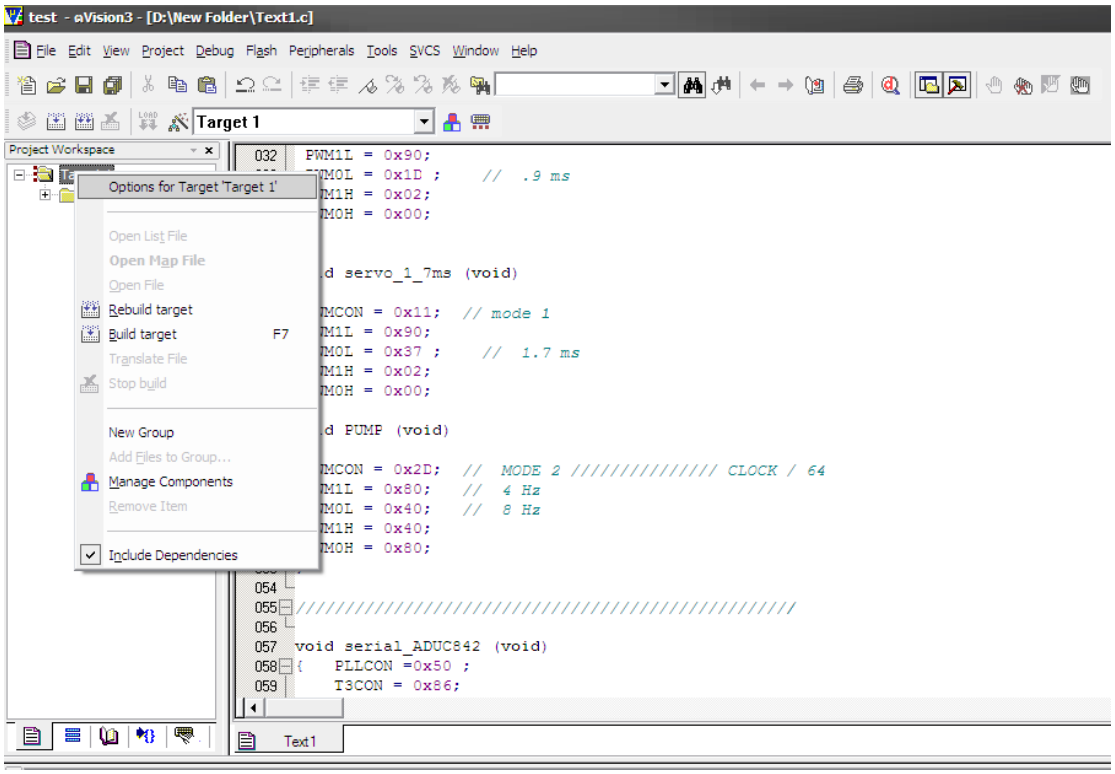
ภาพที่ 2.58 หน้าต่างสำหรับบันทึกโปรแกรม

7) ทำการคลิ๊กขวาที่ Source Group แล้วเลือกที่ Add File to Group ทำการเลือก File ที่ได้ ทำการ Save ไว้มาเข้ากลุ่มงานโปรเจ็คที่สร้าง ดังภาพที่ 2.59



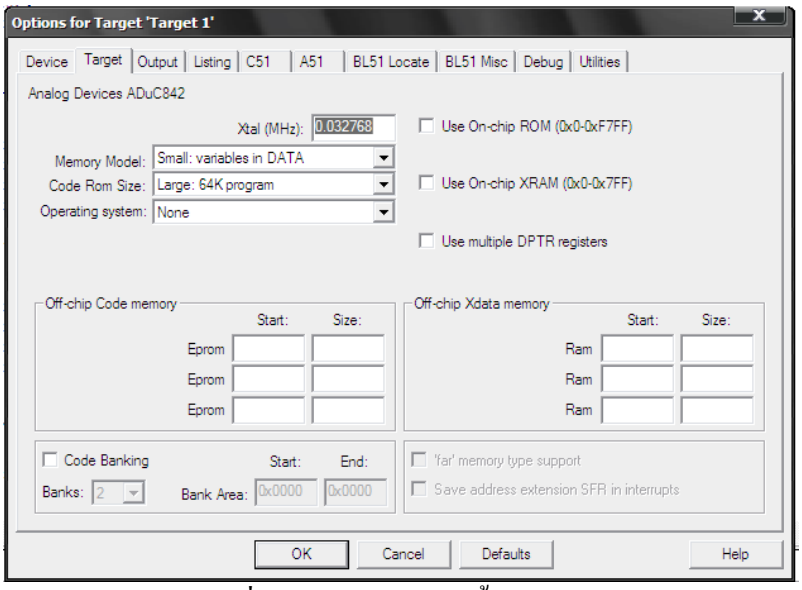
ภาพที่ 2.59 หน้าต่างการ Add File to Group

8) เมื่อเลือกงานเข้ามาในโปรเจ็คที่สร้างแล้ว ต่อมาจะต้องทำการ SET ค่าเพื่อที่จะทำ Output โดยทำการเลือก Option for Target ดังภาพที่ 2.60



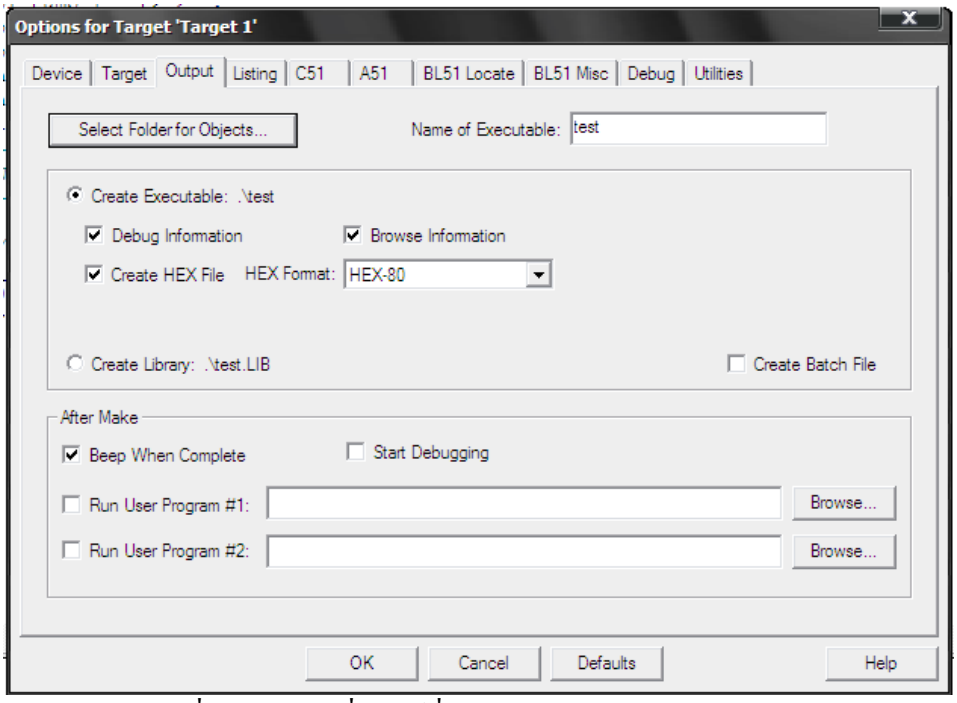
ภาพที่ 2.60 หน้าต่างการเลือก Option for Target

9) จากนั้นจะปรากฏหน้าต่าง ดังภาพที่ 2.61 ขึ้นมาทำการใส่ค่า Option ต่างๆ



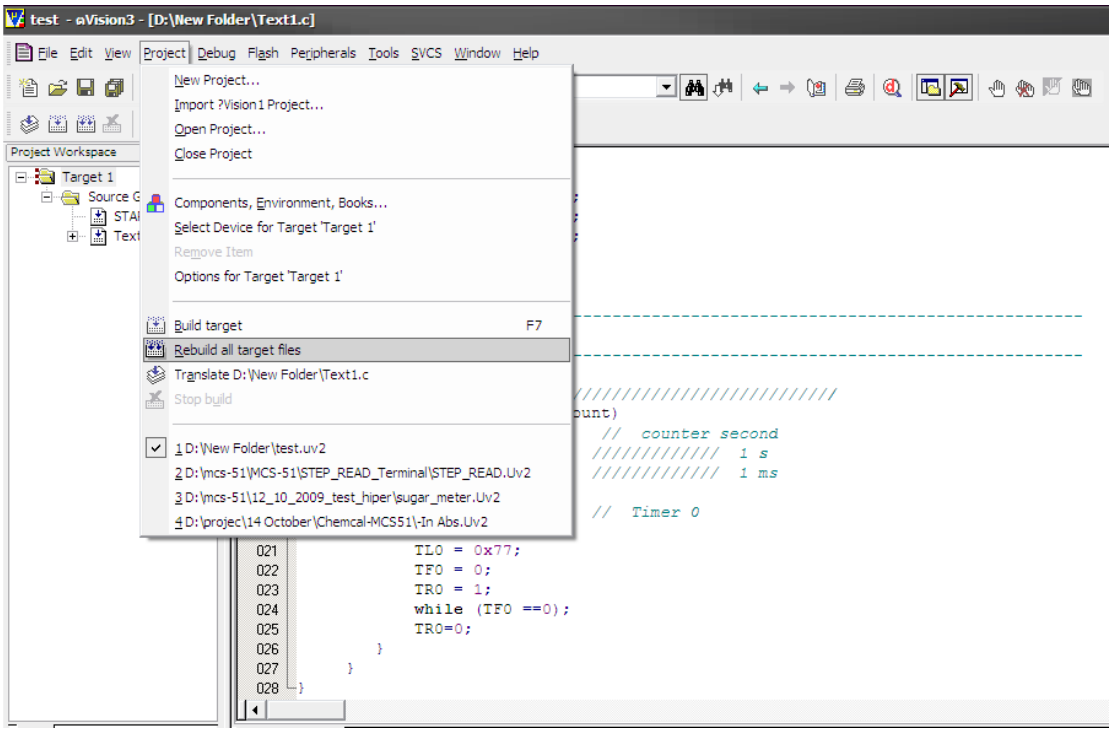
ภาพที่ 2.61 หน้าต่างการตั้งค่า Option

10) เลือกที่ Tap Output แล้วทำการเลือกชื่อไฟล์ที่จะ Complier แล้วคลิ๊กที่ Create HEX File ดังภาพที่ 2.62



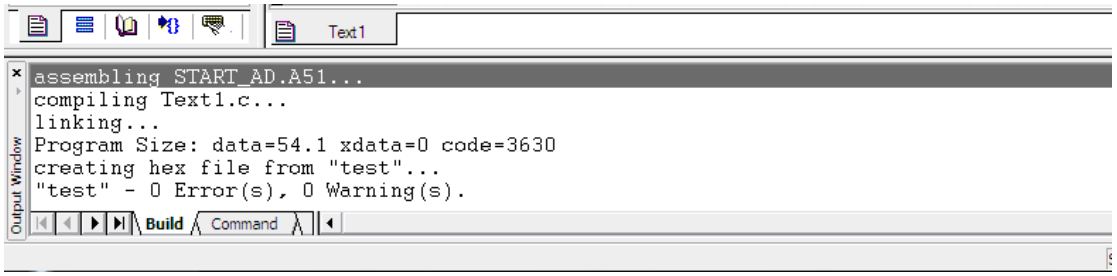
ภาพที่ 2.62 เลือกชื่อไฟล์ที่ใช้ Complier และสร้าง *.hex File

11) จากนั้นทำการสร้าง Output โดยเข้าที่ Project Build Project หรือกด F7 ดังภาพที่ 2.96



ภาพที่ 2.63 การสร้างเอาต์พุต

12) เมื่อทำการกดแล้วถ้าขึ้น Error แสดงว่าโปรแกรมที่เขียนมีจุดที่เขียนผิด ให้ทำการดับเบิ้ลคลิ๊กตรง Error ก็จะปรากฏบรรทัดที่ Error ของโปรแกรมแต่ถ้าทำการแก้ไขเรียบร้อยแล้วจนไม่มี Error แล้วโปรแกรมจะสร้าง ไฟล์ *.hex ให้ในโฟลเดอร์โปรเจ็คที่สร้าง ดังภาพที่ 2.63



ภาพที่ 2.63 โปรแกรมที่สามารถใช้งานได้