

# การใช้งาน TwidoSoft V 3.5



## Introduction to PLC

โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ หรือ PLC ถูกสร้างขึ้นมา โดยกลุ่มวิศวกร ของ บริษัท General Motor เนื่องจากมีความต้องการที่จะสร้างอุปกรณ์ควบคุมมาทดแทนวงจรรีเลย์ ในการควบคุม กระบวนการต่างๆ ซึ่งจะต้องสามารถรองรับการทำงานหรือกระบวนการผลิต ใหม่ๆ ได้และยังต้องมีคุณสมบัติดังนี้

1. ใช้หลักการเขียนโปรแกรมในการสร้างฟังก์ชันการทำงานแทนสายไฟฟ้า และจะต้องสามารถแก้ไขหรือเปลี่ยนแปลงฟังก์ชันการควบคุม ได้ตลอดเวลา
2. มีประสิทธิภาพในการควบคุม มีขนาดเล็ก เมื่อเทียบกับการใช้รีเลย์ในการควบคุม
3. การบำรุงรักษาทำได้ง่าย และมีค่าใช้จ่ายน้อย
4. ราคาถูกเมื่อเทียบกับการใช้รีเลย์

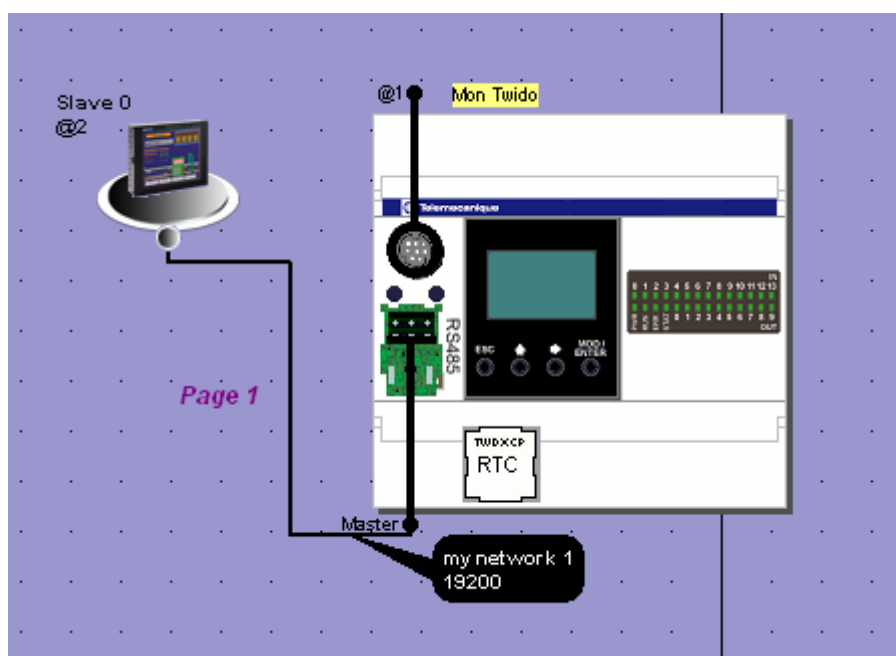
และจากการพัฒนาโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์อย่างต่อเนื่อง ทั้งทางด้าน ฮาร์ดแวร์ และ ซอฟต์แวร์ ทำให้ความสามารถในการทำงานเพิ่มสูงขึ้น เช่น มีหน่วยความจำสูงขึ้น สามารถติดต่อกับอินพุต เอาท์พุต แบบระยะไกล Remote input/output สามารถรับสัญญาณที่เป็นอนาล็อกอินพุต เอาท์พุต และการควบคุม แบบ PID และในปัจจุบันยังสามารถสื่อสารข้อมูล ระหว่าง PLC กับ PLC หรือ PLC กับอุปกรณ์ต่างๆ โดยมีรูปแบบการสื่อสารเป็น Modbus , ASCII , Remote Link , Ethernet , Field Bus , ASI BUS , etc. ทำให้เราสามารถรับรู้ข้อมูล จากกระบวนการผลิตหรือควบคุม และปัญหาหรือข้อบกพร่องต่างๆในกระบวนการควบคุม เป็นต้น

## การจำแนกขนาดของโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์

การจำแนกขนาดของโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ ตามโครงสร้างได้ 2 ลักษณะ คือ

### 1. แบบ *Compact*

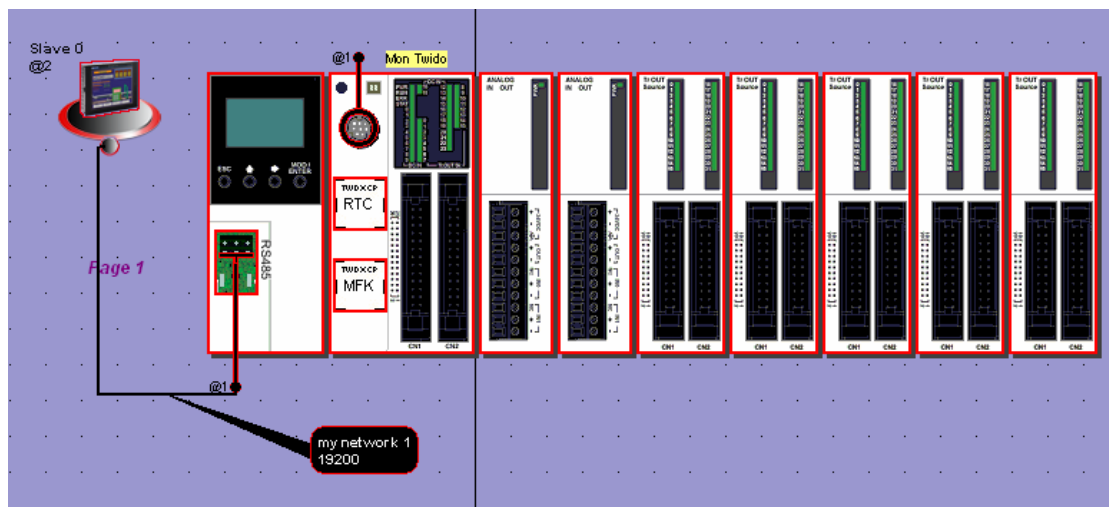
จะเป็นโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ที่มีขนาดเล็ก กะทัดรัด มีหน่วย อินพุต เอาท์พุต และพอร์ตสื่อสารข้อมูล ประกอบรวมกันอยู่ภายในโครงสร้างเดียวกันซึ่งเหมาะสำหรับการใช้งานที่มีอินพุต เอาท์พุตที่แน่นอนและมีจำนวนไม่มากนักเช่น ใช้ควบคุมเครื่องจักรขนาดเล็ก เป็นต้น



จากรูปจะเป็นพีแอลซี แบบ Compact ซึ่งแบ่งได้เป็น 10 , 16 , 24 , 40 I/O  
ในรุ่น 24 I/O จะสามารถต่อการ์ดเพิ่มได้ อีก 4 การ์ด และรุ่น 40 I/O จะสามารถต่อการ์ด  
เพิ่มได้ 7 การ์ด ส่วนรุ่น 10 และ 16 I/O จะไม่สามารถต่อการ์ดเพิ่มได้

## 2. แบบ *Modular*

จะเป็นโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ ที่มีลักษณะเป็นโมดูล เชื่อมต่อกันเป็นโมดูล สามารถที่จะทำการถอดหรือเสียบโมดูลที่ต้องการใช้งานได้ ในปัจจุบันนิยมที่จะมีโครงสร้างในลักษณะนี้ เนื่องจากมีความยืดหยุ่นในเรื่องของ จำนวน อินพุต เอาท์พุต อนาล็อก Temperature Sensor โดยจะสามารถต่อใช้งานได้ 7 การ์ด หรือจะสามารถใช้อินพุต เอาท์พุตได้ ประมาณ 300 I/O

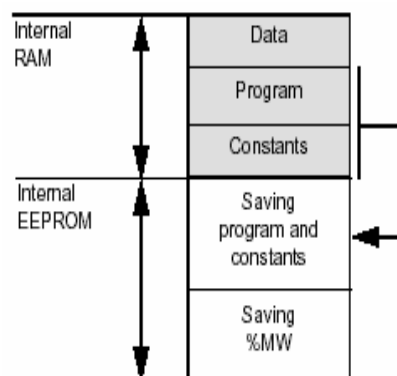


จากรูปจะเป็นพีแอลซี แบบ Modular ซึ่งสามารถเพิ่มการ์ดได้ 7 การ์ด และการอ้างอิงการ์ดในแต่ละ สล็อตจะเริ่มที่ สล็อตที่ 0 นับจากตัว PLC Base และจะเริ่มนับสล็อตที่ 1 ทางด้านขวามือเป็นต้นไปจนครบ 7 สล็อต ตัวอย่างการ์ดที่สามารถเพิ่มได้ เช่น Digital I/O , Analog I/O Temperature Card ส่วนทางด้านซ้ายมือจะไม่มีแอดเดรสอ้างอิงและสามารถต่อได้ 1 การ์ดเท่านั้น เช่น Serial Port ของพีแอลซี Twido Modular ฯ

## หน่วยความจำในโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์

การจำแนกหน่วยความจำภายในระบบของโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์  
จะแบ่งออกได้เป็น 2 ประเภทคือ

1. RAM จะเป็นหน่วยความจำที่ข้อมูล จะสูญหายเมื่อ ไม่มีการจ่ายกำลังไฟฟ้าให้กับหน่วยความจำ
2. EEPROM จะเป็นหน่วยความจำที่สามารถเก็บข้อมูลได้ แม้ในขณะที่ไม่มีการจ่ายกำลังไฟฟ้า ให้กับหน่วยความจำ



- ภายใน Twido จะมี RAM และ Internal EEPROM
- RAM เก็บ ข้อมูลได้ 30 วันโดย Battery Backup
- Internal EEPROM ใช้ Backup Program ได้
- External Memory Cartridge เก็บข้อมูลเหมือน Internal EEPROM

Memory type	Compact Controllers			Modular Controllers		
	10DRF	16DRF	24DRF	20DUK 20DTK	20DRT	40DUK 40DTK
Internal RAM	10KB	32KB	32KB	32KB	32KB	32KB
Available extended memory*					64KB	64KB
Maximum application size	10KB	32KB	32KB	32KB	32KB or 64KB*	32KB or 64KB*
Maximum external backup	32KB	32KB	32KB	64KB	32KB or 64KB	32KB or 64KB

รูปด้านบนแสดงพื้นที่ในการเก็บข้อมูลของ PLC ในแต่ละรุ่น

## หน่วยอินพุท เอาท์พุท

หน่วยอินพุท จะทำหน้าที่เชื่อมต่อระหว่าง หน่วยประมวลผลกลาง กับอุปกรณ์ภายนอก โดยจะรับสัญญาณ ที่มีสถานะ หรือปริมาณทางกายภาพ ต่างๆ แล้วแปลงสัญญาณส่งให้ หน่วยประมวลผลกลาง เพื่อทำการประมวลผลต่อไป

หน่วยเอาท์พุทนั้นจะทำหน้าที่เชื่อมต่อระหว่างหน่วยประมวลผลกลาง กับอุปกรณ์เอาท์พุทภายนอก โดยหน่วยประมวลผลกลางจะส่งค่าสถานะ หรือปริมาณการควบคุมต่างๆออกไปควบคุมอุปกรณ์ภายนอก

หน่วยอินพุท เอาท์พุทสามารถแบ่งได้เป็น

1. Digital Input/Output จะมีสัญญาณ 2 สถานะคือ OFF 0 VDC และ ON 24 VDC ซึ่งจะใช้งานในการรับแรงดัน ที่ป้อนให้กับอินพุทเพื่อเช็คสถานะการทำงาน หรือเพื่อให้เอาท์พุท เปิดหรือปิดอุปกรณ์ไฟฟ้าต่างๆ
2. Analog Input/Output จะเป็นสัญญาณที่มีความต่อเนื่องกันทางไฟฟ้าคือ จะมีแรงดัน ในช่วง 0 – 10 VDC , 4 – 20 mA เป็นต้น

\*\*\* โดยข้อแตกต่างกันระหว่าง Digital Input/Output และ Analog Input/Output คือ

- สัญญาณ ดิจิตอล จะสามารถรับรู้หรือสั่งเอาท์พุท ได้ 2 สถานะ คือ ON และ OFF
- สัญญาณ อนุาล็อก จะเป็นสัญญาณที่ต่อเนื่องคือ สามารถรับรู้แรงดันหรือกระแสได้ตั้งแต่ 0 – 10 VDC , 4 – 20 mA ซึ่งมีความละเอียดมากกว่าแบบ ON OFF จึงเหมาะสำหรับงานที่ต้องการความละเอียดในการควบคุม หรืองานควบคุมพิเศษ อย่างเช่น PID Control หรือ FUZZY Control เป็นต้น

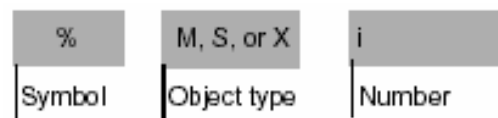
## ***Bit Object Format*** รูปแบบการใช้งานข้อมูลแบบบิต

การใช้งานข้อมูลในรูปแบบบิตนั้นจะใช้งานกับข้อมูลประเภท อินพุต เอาท์พุท รีเลย์ภายใน หรือ บิตในเวิร์ด และ System Bit จะแสดงและควบคุมสถานะแบบบิตของระบบ

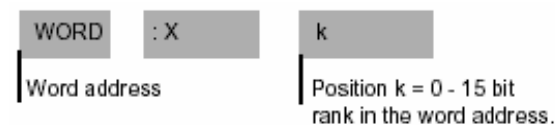
### ■ Input / Output Format



### ■ Bit Object Format



### ■ Bit Objects Extracted from Words



การใช้งานข้อมูลบิตจะมีรูปแบบการเขียนโปรแกรม %I0.0 ซึ่งหมายความว่าอินพุต ช่องที่ 0 Bit 0 โดยในการเขียนโปรแกรมจะต้องขึ้นต้นด้วย % เสมอเพราะว่าเป็นรูปแบบหรือ Syntax ของการเขียนโปรแกรม โดยจะสามารถจัดรูปแบบข้อมูลแบบบิตได้คือ

ข้อมูลอินพุต	%I0.0 – XXX	ขึ้นอยู่กับรุ่นของ PLC และ Expansion Card
ข้อมูลเอาท์พุต	%Q0.0 – XXX	ขึ้นอยู่กับรุ่นของ PLC และ Expansion Card
รีเลย์ภายใน	%M0 - 255	
System Bit	%S0 - 119	
บิตในเวิร์ด	%MW0:X0 - 15	

## Word Object Format รูปแบบการใช้งานข้อมูลแบบเวิร์ด

การใช้งานข้อมูลแบบเวิร์ดนั้นจะใช้งานเป็นตัวเก็บหรือพักข้อมูลขนาด 1 เวิร์ด หรือ 16 บิต และสามารถเลือกเก็บข้อมูลเป็นแบบ ดับเบิลเวิร์ด 32 บิต ก็ได้เช่นกัน โดยในการเก็บหรือพักข้อมูลนั้นจะรับค่าจากการประมวลผลทางคณิตศาสตร์ บวก ลบ คูณ หาร หรือรับค่าจากอนาล็อก อินพุตก็ได้ และสามารถรับค่าจากการสื่อสารต่างๆ เช่น Modbus , ASCII , Ethernet ฯลฯ



### ■ Word

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Bit position
0	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	Bit state
16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1		Bit value

การใช้งานข้อมูลแบบเวิร์ดจะมีรูปแบบการเขียนโปรแกรม %MW0 ซึ่งหมายความว่าป็นหน่วยความจำ เวิร์ดตัวที่ 0 โดยการเก็บข้อมูลแบบเวิร์ดนั้นจะมีรูปแบบดังต่อไปนี้

Word	%MW0 - 2999	ข้อมูลแบบเวิร์ด
DoubleWord	%MD0 - 2998	ข้อมูลแบบดับเบิลเวิร์ด
ConstantWord	%KW0 - 255	ค่าคงที่แบบเวิร์ด
SystemWORD	%SW0 - 122	ข้อมูลระบบ
Analog Word	%IW1.0, %QW1.0	ข้อมูลอนาล็อกแบบเวิร์ด

\* เราสามารถที่จะดู ค่าของ System Word และจะทำการแก้ไขค่าของ System Word ได้บางตัวสามารถดูได้จาก Help ของโปรแกรม



## ***Floating Point*** การใช้งานข้อมูลในรูปแบบทศนิยม

ในการใช้งานทศนิยมของ PLC TWIDO จะสามารถใช้งานได้ดังตาราง

Twido Controller	Floating Point Supported
TWDLMDA40DUK	Yes
TWDLMDA40DTK	Yes
TWDLMDA20DUK	No
TWDLMDA20DTK	No
TWDLMDA20DRT	Yes
TWDLMDA40DRF	Yes
TWDLCA24DRF	No
TWDLCA16DRF	No
TWDLCA10DRF	No

ในการใช้งานทศนิยมนั้นรูปแบบจะเป็นไปตาม IEEE STD 734-1985 โดยจะใช้พื้นที่ในการเก็บค่าทศนิยม 32 บิต หรือ 2 เวิร์ด และสามารถที่จะใช้งานได้ในช่วง



จากรูปในการใช้งานเราสามารถใช้งานเขียนแบบ Exponent ได้โดยตรงเช่น 1E-03 จะมีค่าเท่ากับ 0.001 และนอกจากนี้ยังมีฟังก์ชันพิเศษในการจัดการข้อมูลที่เป็นทศนิยมเพื่อให้่ายในการทำงานหรือ่ายในการเขียนโปรแกรม

การหาค่ารากที่สองของข้อมูล  $\text{SQRT}(x)$   
ฟังก์ชันการยกกำลังทศนิยม  $\text{expt}(x^y=\%mw\wedge\%mf)$   
ฟังก์ชัน  $\text{Log}(x)$   
ฟังก์ชัน  $\text{ln}(x)$   
ฟังก์ชัน  $\text{exp}(x)$

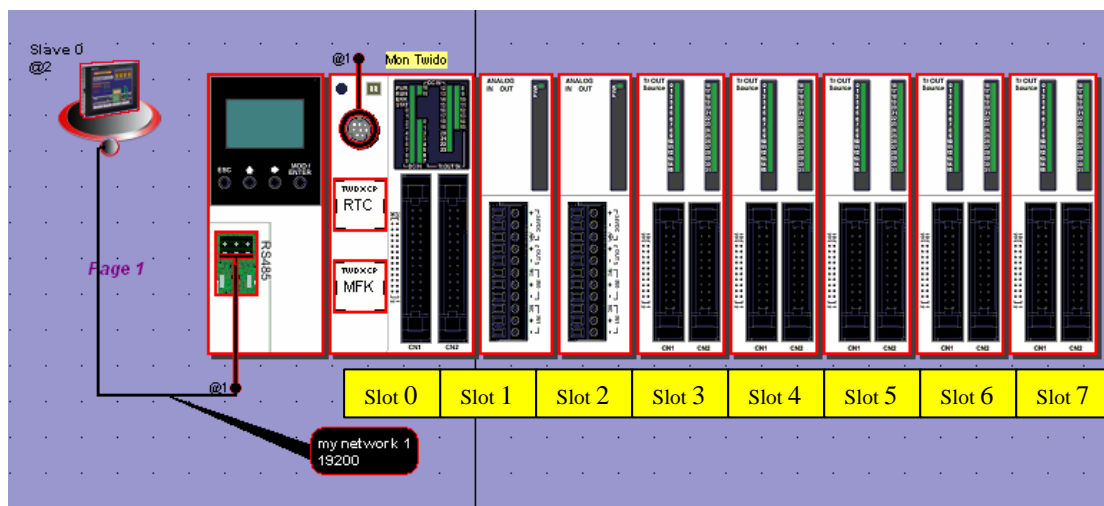
ในการใช้งานเมมโมรีเวิร์ดจัดการข้อมูลด้านเทคนิคสามารถใช้ได้ในช่วง %MF0 ถึง %MF1499 และสามารถใช้ค่าคงที่แบบเทคนิคได้ในช่วง %KF0 ถึง %KF127 และในการใช้งานต้องระวังการใช้งานพื้นที่ทับซ้อนกันเช่น

Floating Point	Internal Word
%MF0	%MW0
	%MW1
%KF0	%KW0
	%KW1

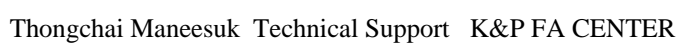
ถ้าเราใช้เมมโมรีเวิร์ด %MF0 แล้วเราจะไม่สามารถใช้งาน %MW0 และ %MW1 ได้ เนื่องจากใช้งานเมมโมรีเวิร์ด แบบเทคนิคจะใช้พื้นที่ในส่วนของเมมโมรีแบบเวิร์ด 2 เวิร์ด หรือที่เรียกว่า Union หรือการใช้งานพื้นที่เดียวกันหลายตัวแปร

## การอ้างแอดเดรส ของ PLC TWIDO

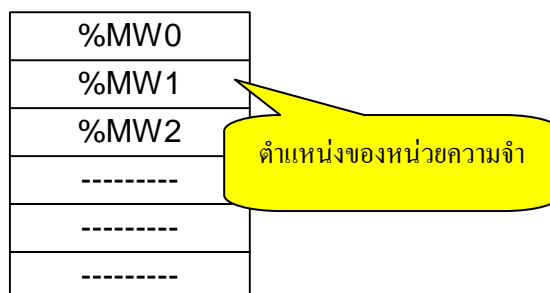
ในการใช้งานอุปกรณ์ อินพุต เอาท์พุต และหน่วยความจำ จะมีหมายเลขหรือแอดเดรสของอุปกรณ์ ให้ผู้ใช้งานสามารถระบุ เพื่อสะดวกในการนำไปใช้งาน และไม่เกิดความสับสนในการใช้งาน จะมีการจัดแบ่งแอดเดรสของอุปกรณ์ดังนี้



1. การอ้างแอดเดรสต่างๆ ของ PLC Twido จะขึ้นต้นด้วย %
2. ระบุชนิดของข้อมูลที่ต้องการอ้างเช่น  
Input                    %I  
Output                    %Q  
Memory Bit            %M  
Memory Word          %MW
3. ระบุหมายเลข Slot ที่ใช้งานโดย Slot 0 จะเป็นตัว PLC Base และ Slot1 to Slot 7 จะเป็นการ์ด Expansion ที่นำมาต่อเพิ่ม
4. จะใช้ทศนิยมคั่นระหว่าง slot กับ Bit หรือ Channel ที่ใช้งาน
5. ระบุ Bit หรือ Channel ที่ต้องการใช้งาน



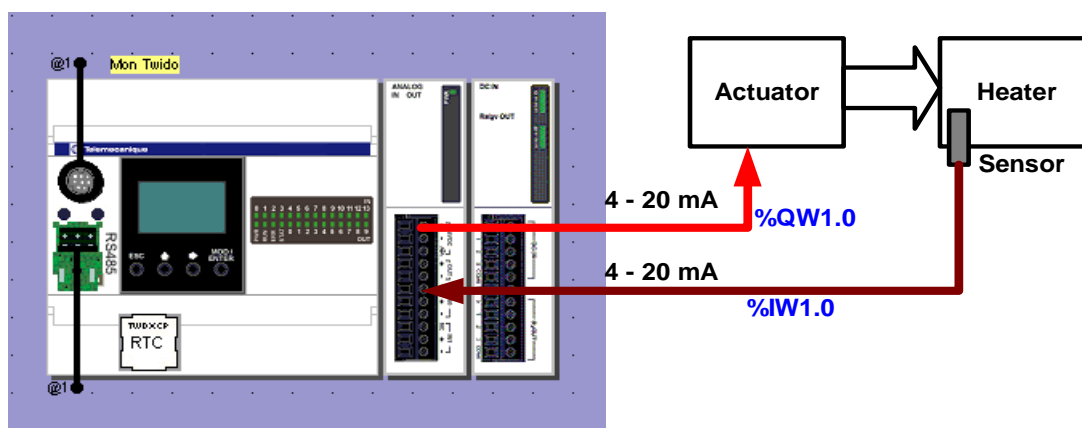
## การอ้างตำแหน่งแอดเดรสของหน่วยความจำของ PLC



ในการอ้างถึงหน่วยความจำจะขึ้นต้นด้วย %MW+ตำแหน่งของหน่วยความจำที่ต้องการใช้งาน ในการใช้งานสามารถที่จะอ้างข้อมูลได้ตลอดเวลา

**หมายเหตุ** โดยในการระบุตำแหน่งหน่วยความจำทั้งแบบ บิต และ เวิร์ด ไม่ต้องมีทศนิยมและบิต หรือ Channel ที่จะใช้งาน

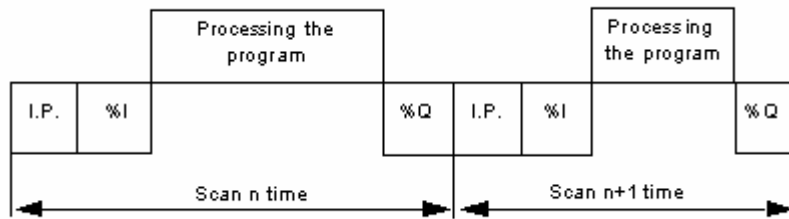
## การอ้างตำแหน่งแอดเดรสของ *Analog I/O* ของ PLC



ในการอ้างแอดเดรส ของ *Analog I/O* จะเป็นการอ้างถึงแบบเวิร์ด โดยอินพุต Ch 0 ของโมดูลตัวที่ 1 จะอ้างเป็น %IW1.0 ส่วนเอาต์พุต Ch 0 ของโมดูลตัวที่ 1 จะอ้างเป็น %QW1.0 ซึ่งการ์ด *Analog I/O* ตัวนี้จะมีทั้งอินพุตและเอาต์พุต ในตัวเดียวกัน

## Scanning Program

### Cyclic Scan



การสแกนหรือการทำงานใน 1 รอบการทำงานของ PLC TWIDO จะมีการทำงานดังรูป โดย

I.P. จะจัดการในส่วนของ System word, bit, update current timer, status light, detect run/stop switch, และกระบวนการต่างๆของ TWIDO SOFT เช่น Animation table เป็นต้น

%I,%IW จะทำการ Update Input Status แล้วส่งไปเก็บในอินพุตบัฟเฟอร์เพื่อให้ CPU ได้รับทราบการเปลี่ยนแปลงของอินพุต

Processing จะเป็นส่วนที่ทำงานตาม Application หรือ แลตเตอร์ที่ผู้ใช้เขียนขึ้นมาเพื่อควบคุมกระบวนการอย่างใดอย่างหนึ่ง

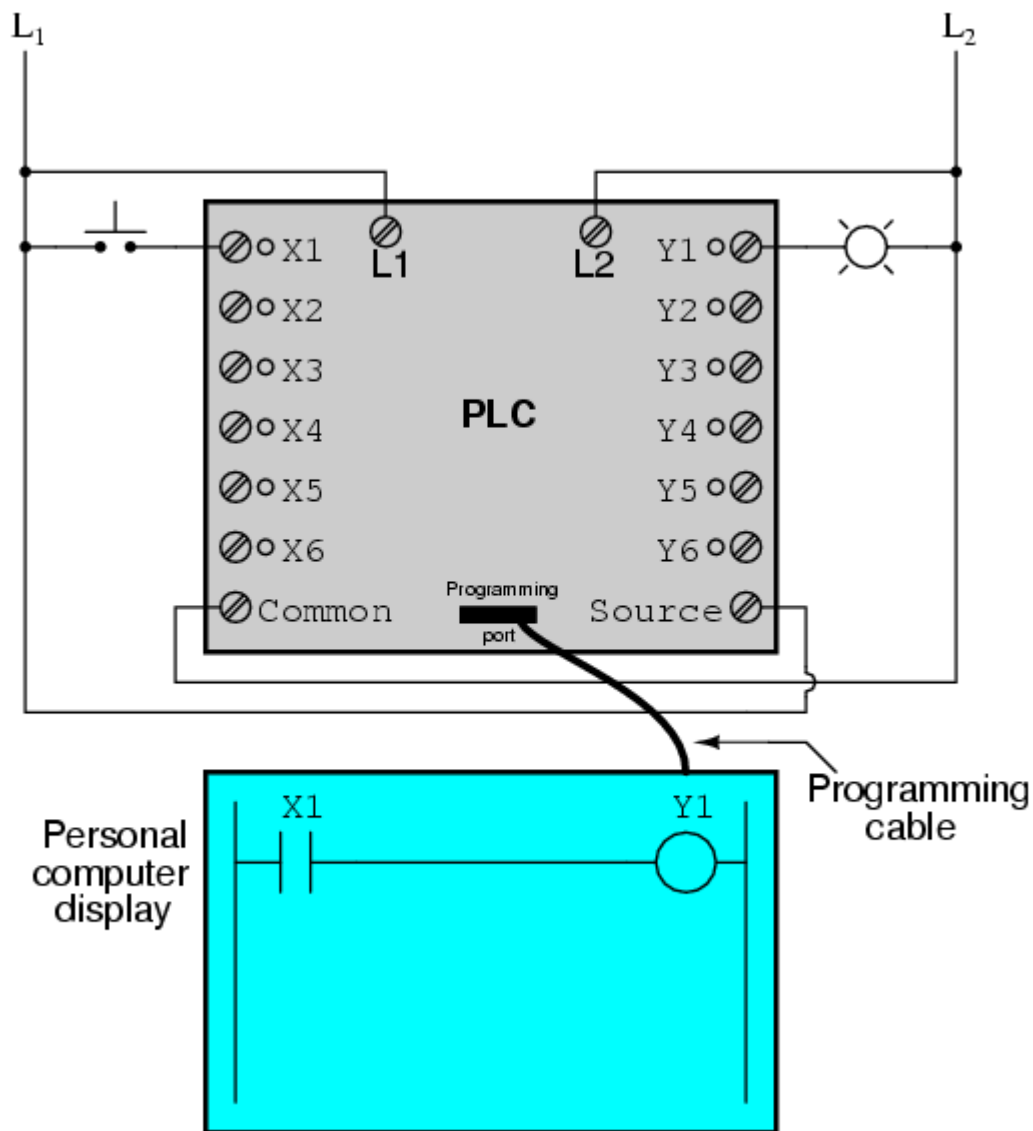
%Q,%QW จะเป็นส่วนที่ส่งสัญญาณไปสั่งให้อุปกรณ์เอาต์พุตทำงานโดยคำสั่งควบคุมจะมาจาก CPU หรือหน่วยประมวลผล

## ***Base Controller Models***

Part	Compact Base Controllers				Modular Base Controllers				
	10DRF	16DRF	24DRF	40DRF	20DUK	20DTK	20DRT	40DUK	40DTK
Timers	64	64	128	128	128	128	128	128	128
Constants	256	256	256	256	256	256	256	256	256
Counters	128	128	128	128	128	128	128	128	128
Fast counters	3	3	3	4	2	2	2	2	2
Very fast counter:	1	1	1	2	2	2	2	2	2
LIFO/FIFO registers	4	4	4	4	4	4	4	4	4
Schedule blocks	16	16	16	16	16	16	16	16	16
Drum controllers	4	4	8	8	8	8	8	8	8
PLS/PWM	0	0	0	2	2	2	2	2	2
Step counter	8	8	8	8	8	8	8	8	8
Shift bit register	8	8	8	8	8	8	8	8	8

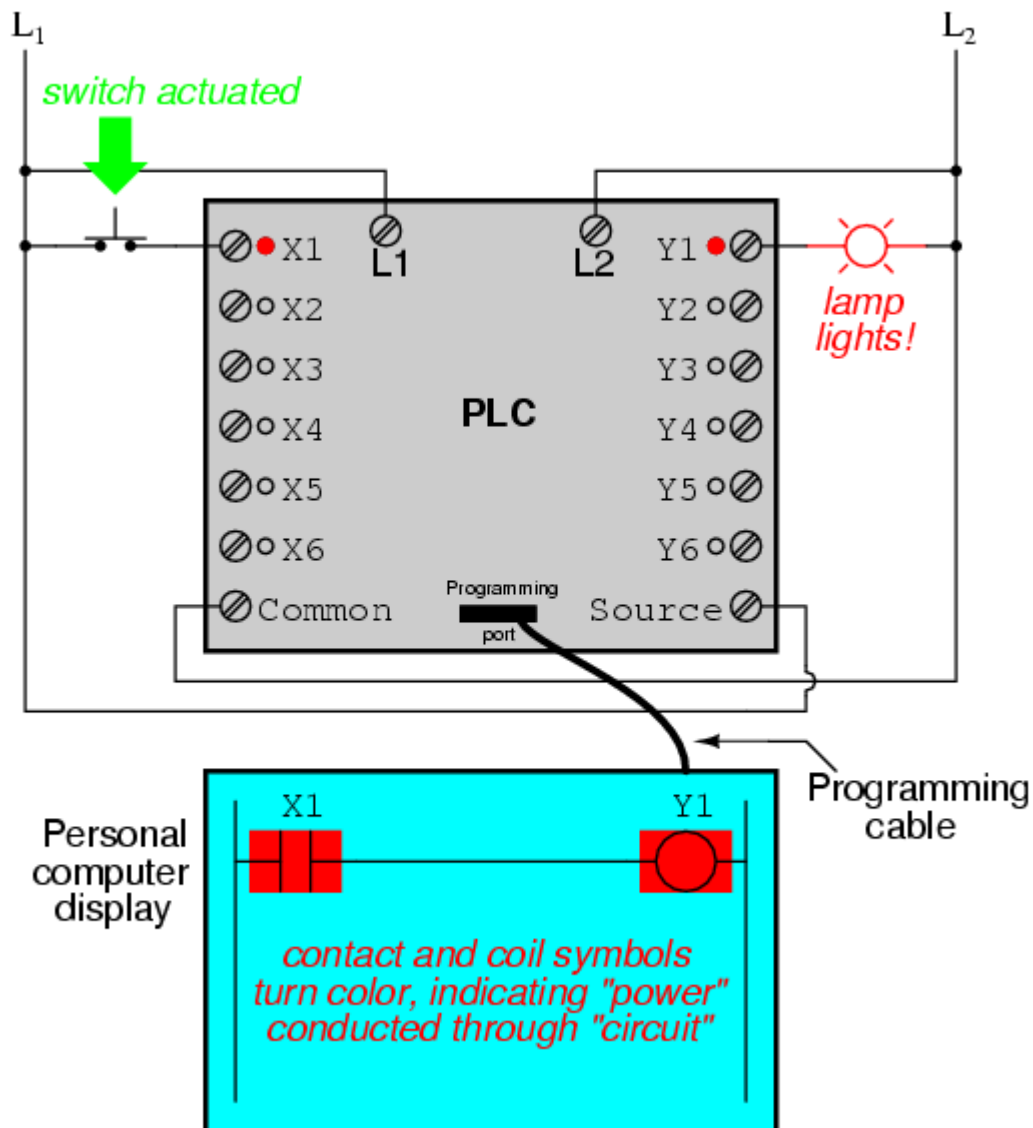
ส่วนนี้จะเป็นการแสดงฟังก์ชันต่างๆที่ของ PLC Twido จะมีและสามารถใช้งานได้ ซึ่งจะทำให้เราทราบว่าในการใช้งานนั้นเราควรเลือก PLC Twido รุ่นอะไรได้บ้างและ แต่ละรุ่นนั้นจะสามารถใช้งานอะไรได้บ้าง

การต่อใช้งาน PLC กับอุปกรณ์อินพุต เอาท์พุท

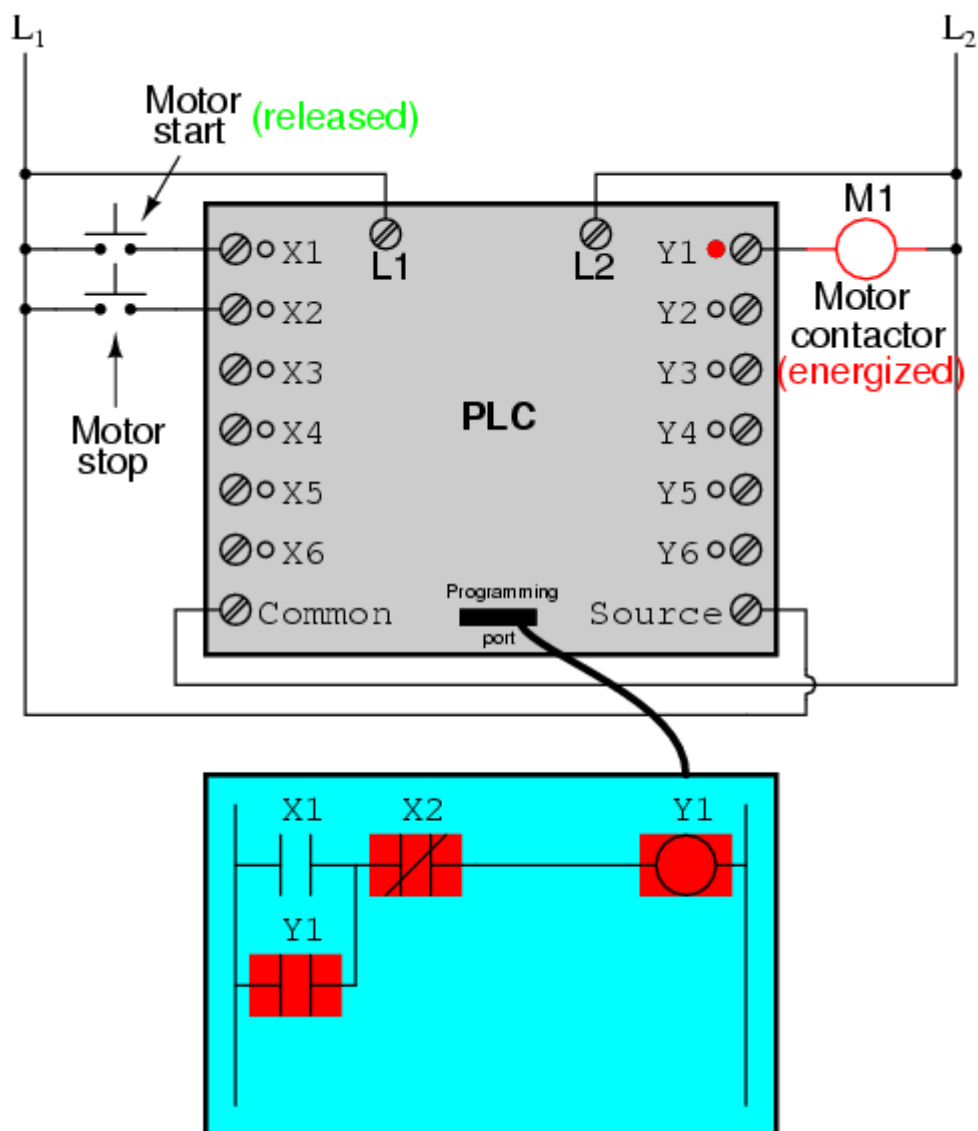


จากรูปแสดงการใช้งาน PLC กับ อุปกรณ์ อินพุท เอาท์พุท โดย L1 จะรับสัญญาณ 24 V และ L2 จะเป็น สายนิวตรอน X เป็นอินพุท Y เป็นเอาท์พุท โดยในภาพด้านล่างจะจะเป็นการ ดับเบิ้ลโปรแกรม หรือ การทำงานใน Animation Mode





เมื่อเกิดการกดสวิตช์  $X1$  ON จะทำให้เอาต์พุต  $Y1$  ON ถ้าเราปล่อยมือจะทำให้  $Y1$  Off หรือไม่ทำงาน และภาพด้านล่างจะแสดงการทำงานในรูปแบบ Animation



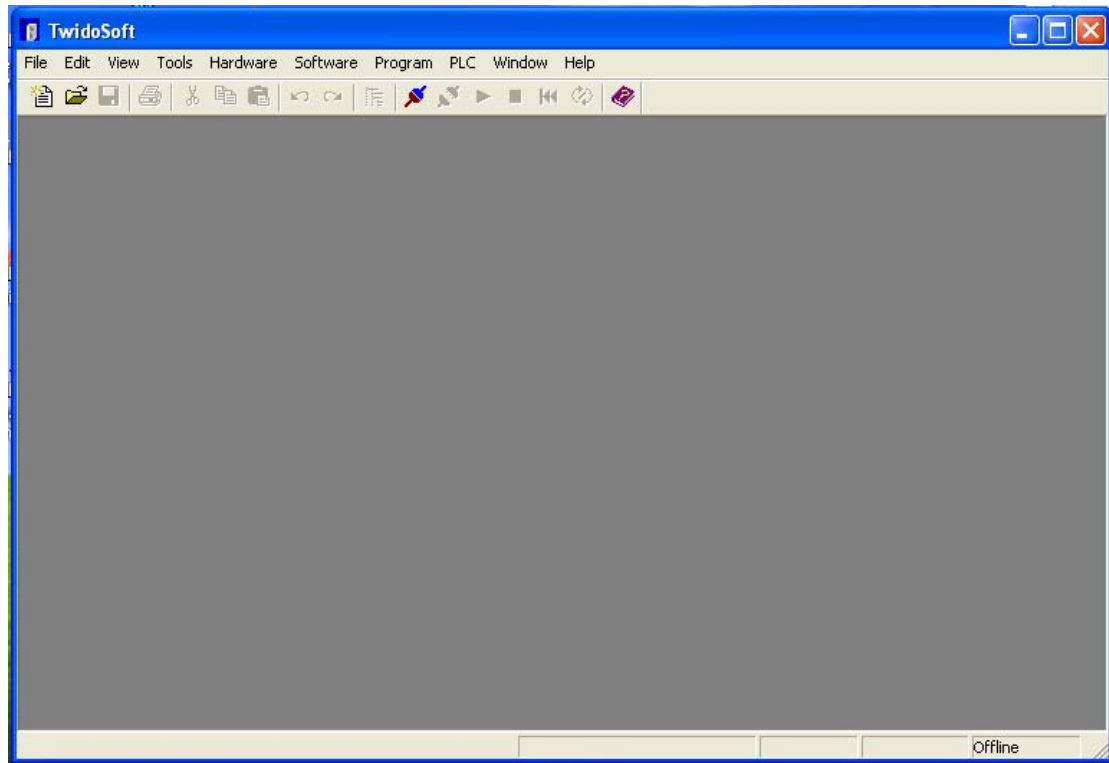
ตัวอย่างการใช้งาน ในการสตาร์ท สตอป มอเตอร์ไฟฟ้า โดยการ Holding Output


\*\*\* หมายเหตุ การใช้งานจริงต้องอ้างอิงจาก *Data Sheets* ของ PLC แต่ละรุ่นของ *Twido*

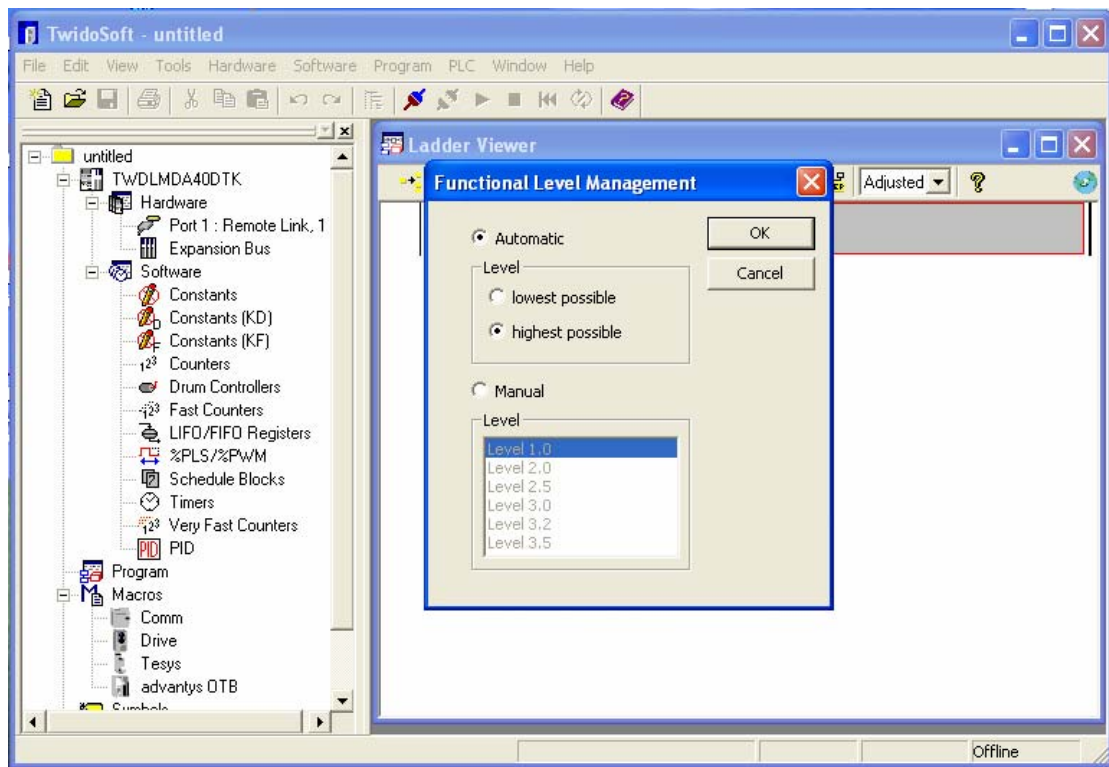
## รายละเอียดส่วนประกอบต่างๆ ในโปรแกรม *TWIDO SOFT*



Double Click Shortcut เพื่อเข้าสู่โปรแกรม Twido Soft

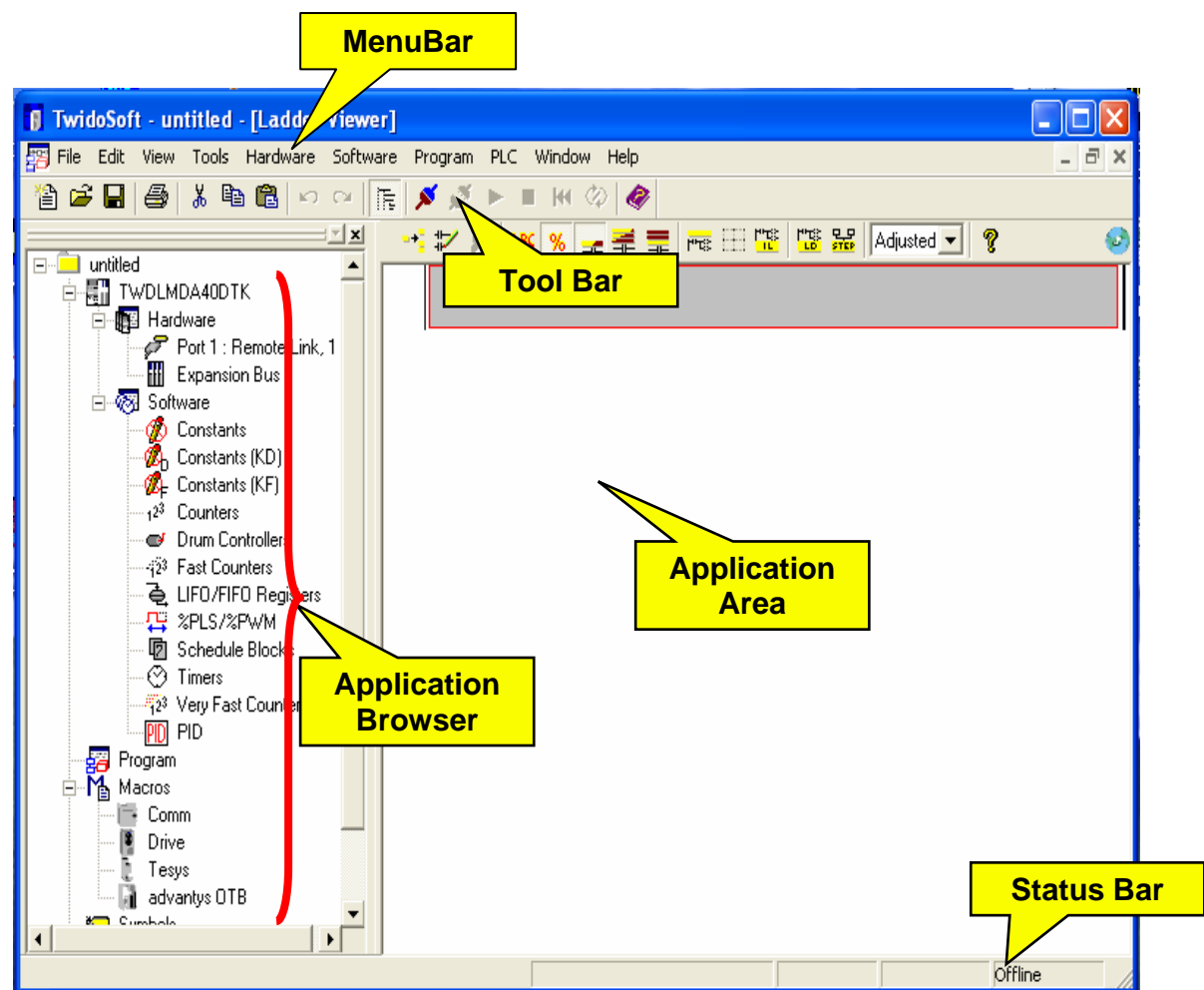


จะปรากฏหน้าจอ TwidoSoft ดังรูป ให้คลิก  เพื่อเข้าสู่โปรแกรมหลักหรือกดที่เมนู File -> New



เลือกการใช้งานแบบ Auto หรือ Manual ซึ่งจะเป็นการเลือกระดับของซอฟต์แวร์ที่ใช้งานให้  
เหมาะกับ ฟิแอลซี เช่น ใช้งานซอฟต์แวร์เวอร์ชัน 3.5 ก็เลือก Level 3.5 ในกรณีเลือกในโหมด  
Manual หรือในโหมด Auto ก็ให้เลือกเป็น Highest Possible

เมื่อเลือก **Functional Level Manager** แล้วจะปรากฏ หน้าตาของ Software TwidoSoft ดังรูป



**Menubar** เป็นส่วนในการจัดการ ทางด้าน ซอฟต์แวร์ ฮาร์ด การเชื่อมต่อกับคอมพิวเตอร์ และ ส่วนช่วยเหลือในการเขียนโปรแกรม Help

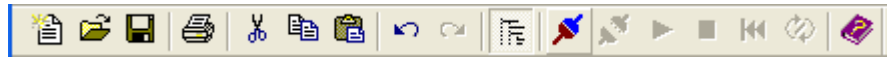
**Tool bar** จะคล้ายกับ Menubar และจะมี ไอคอน เป็นรูปภาพเพื่อความสะดวกในการใช้งาน

**Application Browser** จะเป็นส่วนที่ในการแสดงและจัดการ CPU ของพีแอลซี พอร์ตการ สื่อสารและการ เพิ่มการ์ดต่างๆ ตลอดจนการจัดการฟังก์ชันต่างๆใน PLC

**Application Area** จะเป็นพื้นที่ ในการเขียนโปรแกรม

**Status Bar** เป็นส่วนแสดงรายละเอียดต่างๆในการใช้งานโปรแกรม Online Offline

## การใช้งาน Tool Bar



**Toggle Application Browser** ใช้เปิดปิด Application Browser



**Connection** เชื่อมต่อกับ PLC



**Disconnect** หยุดการเชื่อมต่อกับ PLC



**Run** สั่งให้พีแอลซีทำงาน PLC RUN



**Stop** สั่งให้พีแอลซีหยุดทำงาน PLC Stop




**Animation** ตรวจสอบการทำงานของโปรแกรมและใช้ทดสอบการทำงานของโปรแกรม

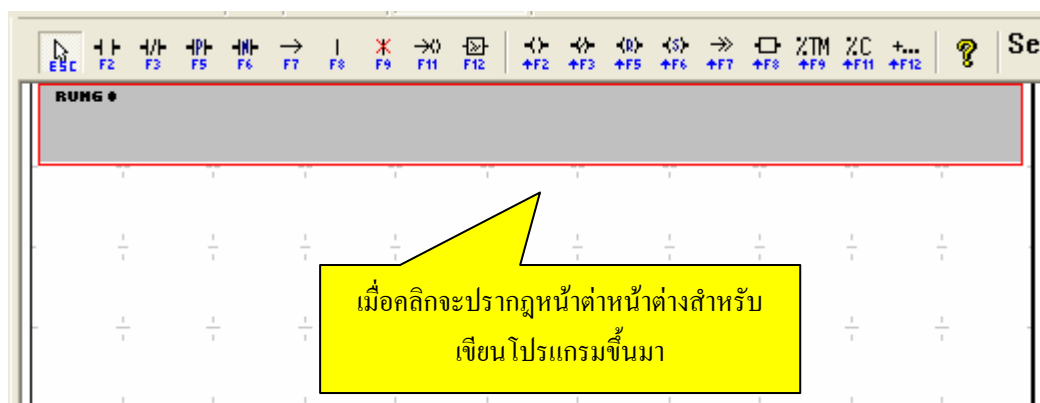


**Help** ส่วนช่วยเหลือสำหรับการเขียนโปรแกรม

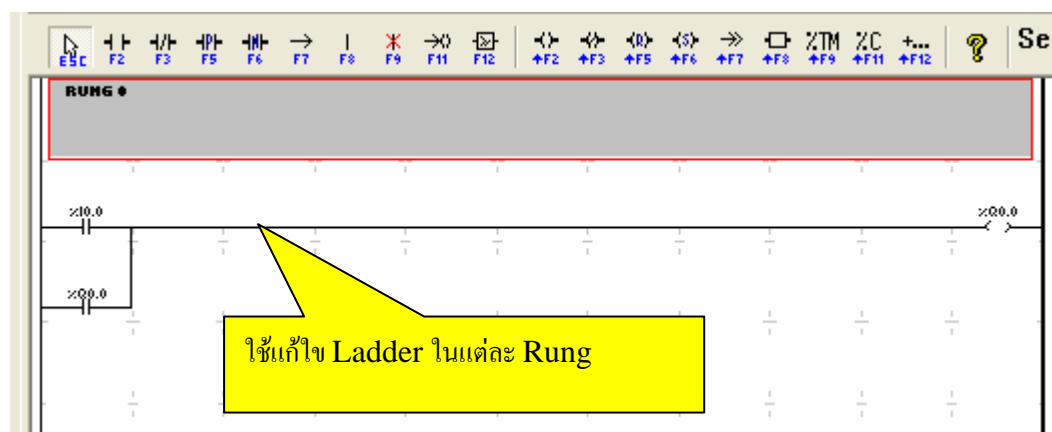
## การใช้งาน Tool Bar สำหรับเขียนโปรแกรม



 **Insert Rung** ใช้ในการเพิ่ม รังในการเขียนโปรแกรมแบบ แลadder



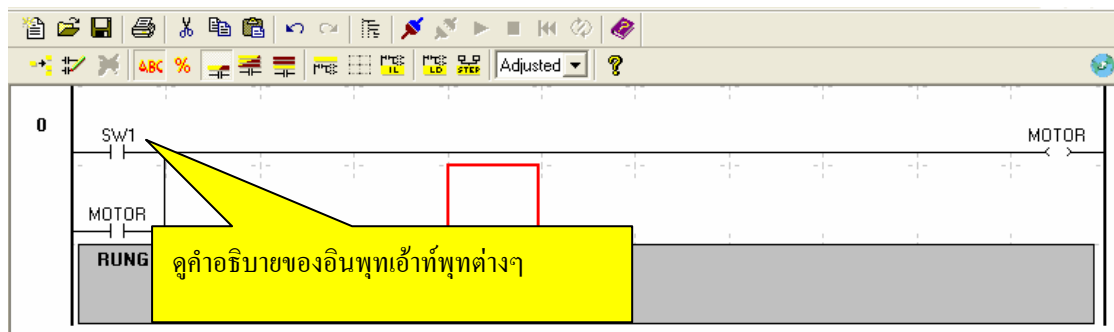
 **Edit Rung** ใช้แก้ไข เปลี่ยนแปลง ฟังก์ชันต่างๆ ในแลadderโปรแกรม



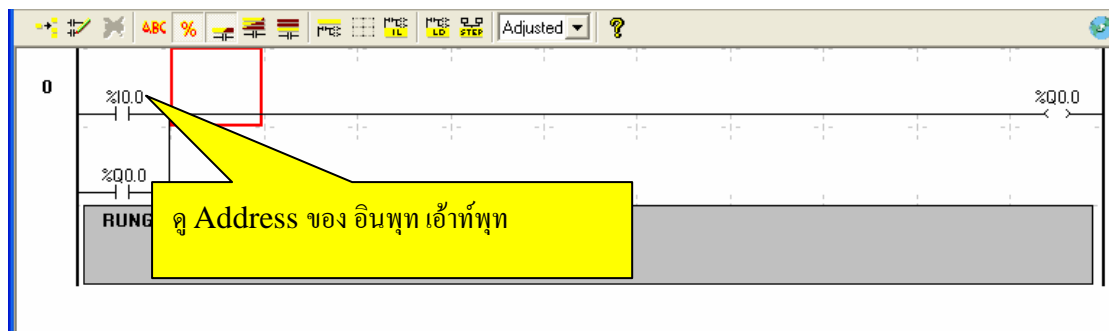




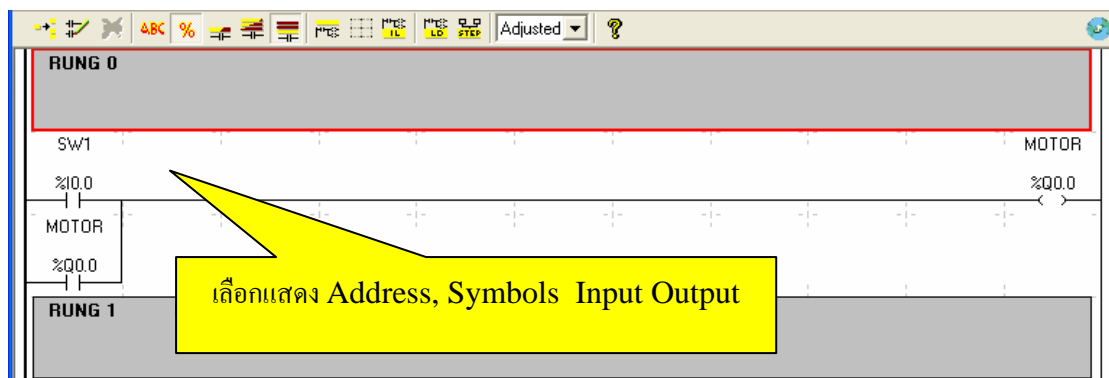
**View Symbols** เป็นตัวแสดงหรือ อธิบายความหมายของ อินพุต เอาท์พุท



**Show Addresses** เป็นตัวแสดงตำแหน่งแอดเดรสของ อินพุต เอาท์พุท ที่ใช้งานอยู่

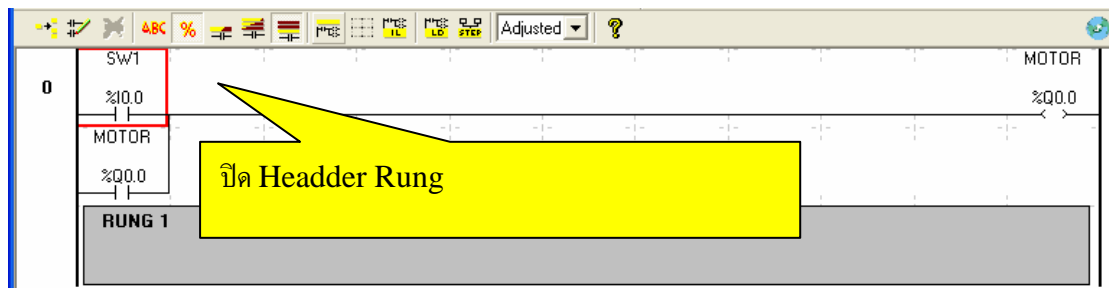
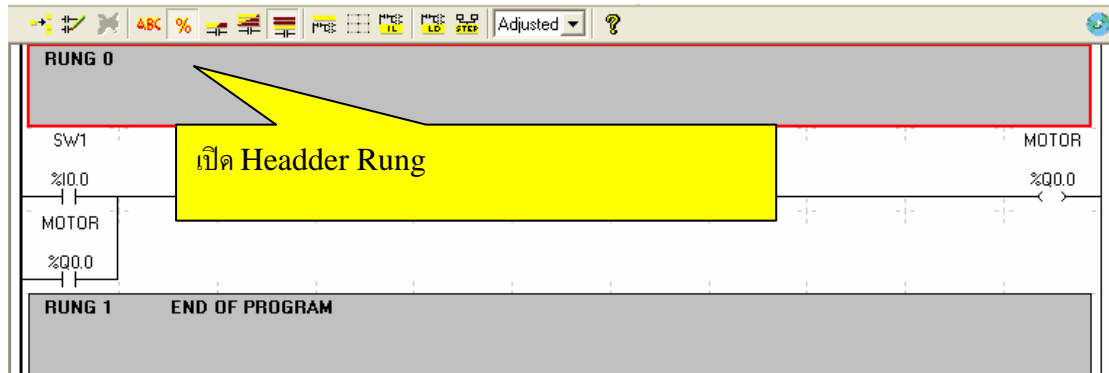


**Show Address & Symbols** ใช้แสดง แอดเดรสและสัญลักษณ์ของ อินพุต เอาท์พุท ในแลตเตอร์โปรแกรม

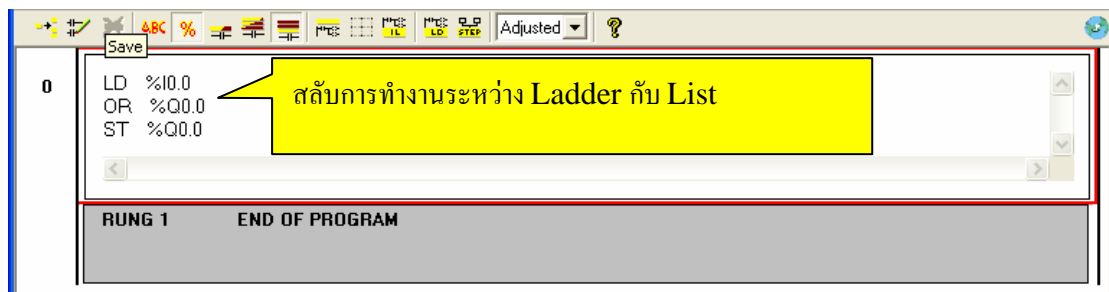




## Toggle Header ใช้สลับการทำงานของ Header Rung

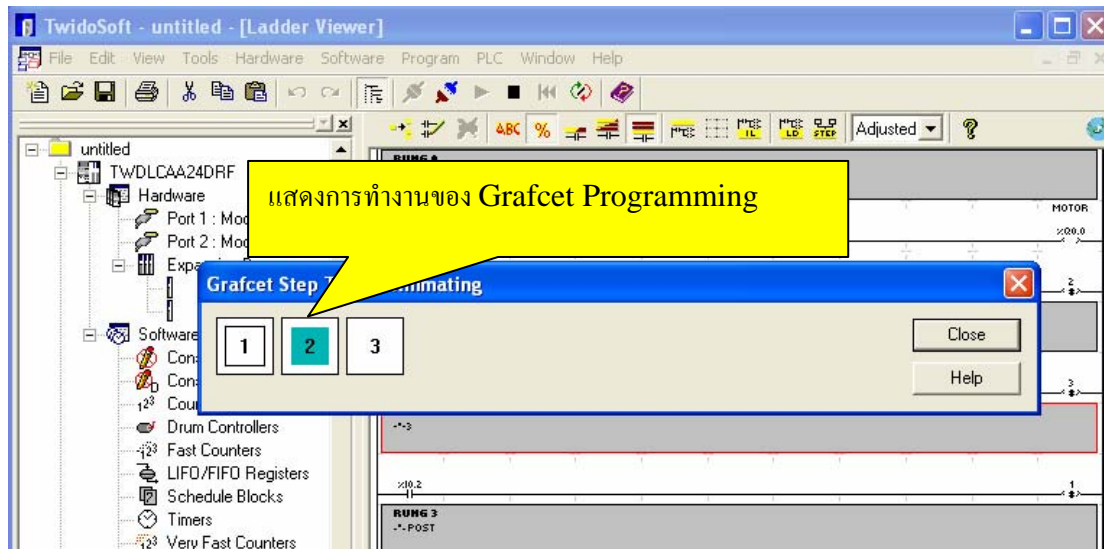


## Toggle Ladder/List ใช้สลับการทำงานระหว่าง Ladder และ Instruction List

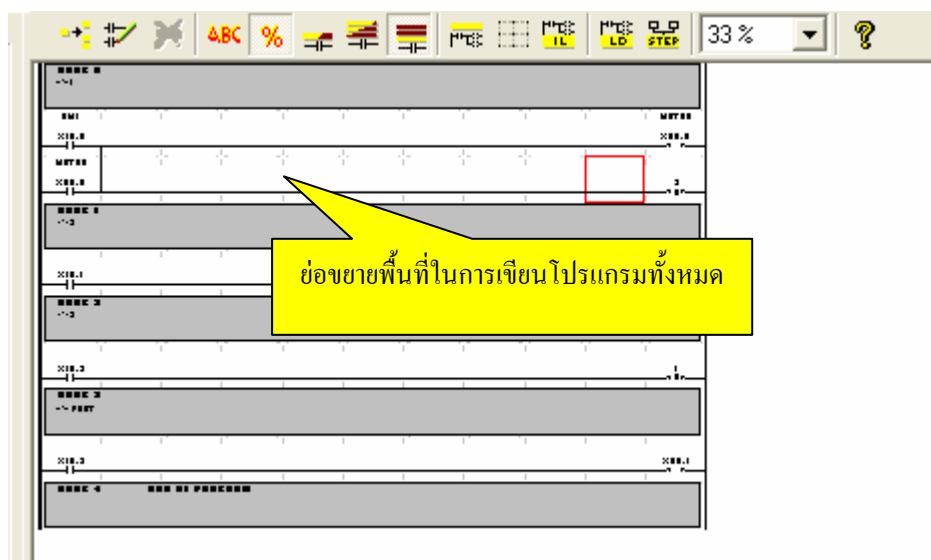




**Show Grafcet Step** แสดงขั้นตอนการทำงานของ Grafcet Programming

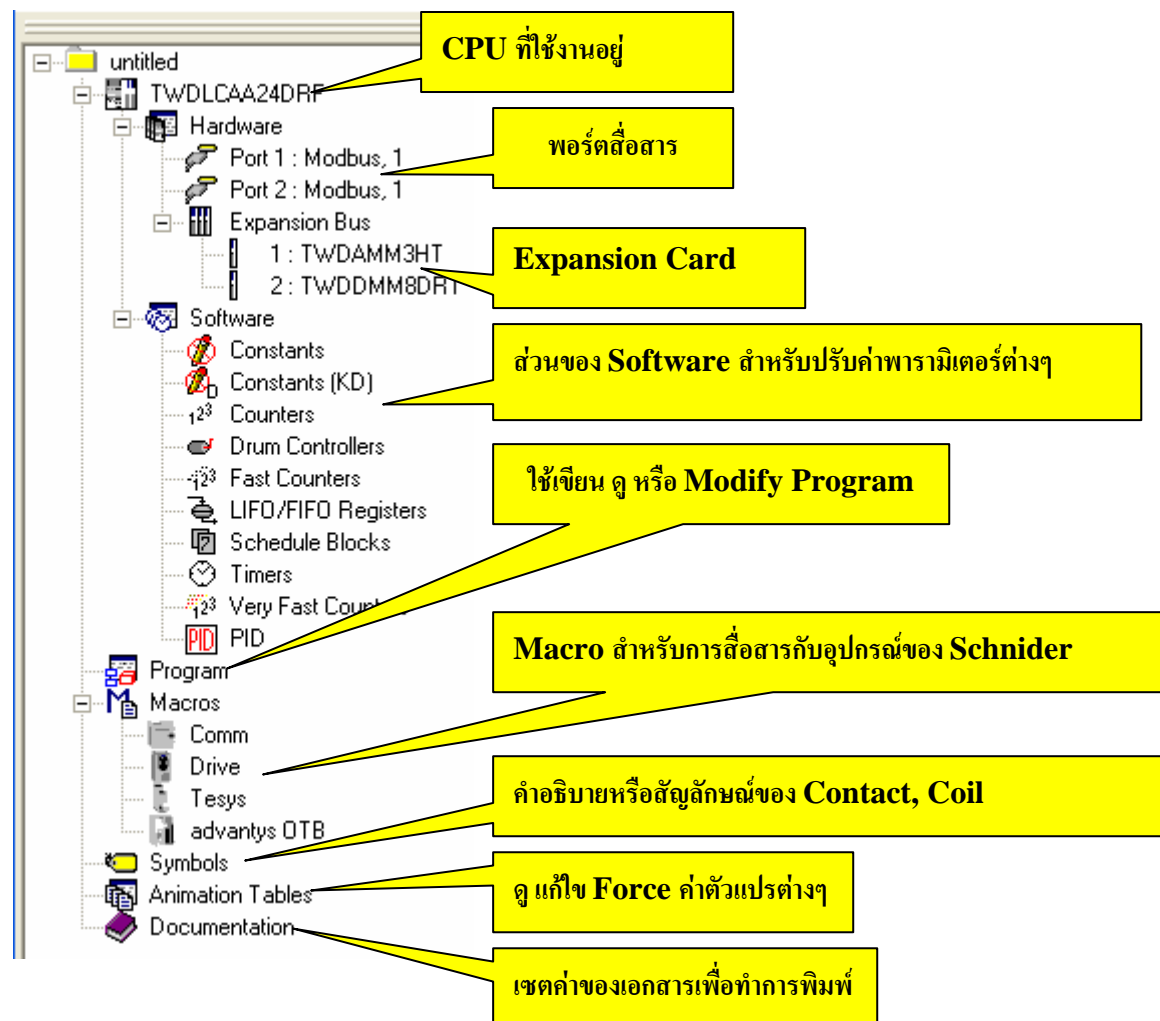


**Zoom Factor** ให้อยู่ขยายหน้าจอ เพื่อความสะดวกในการเขียนโปรแกรม



## Application Browser

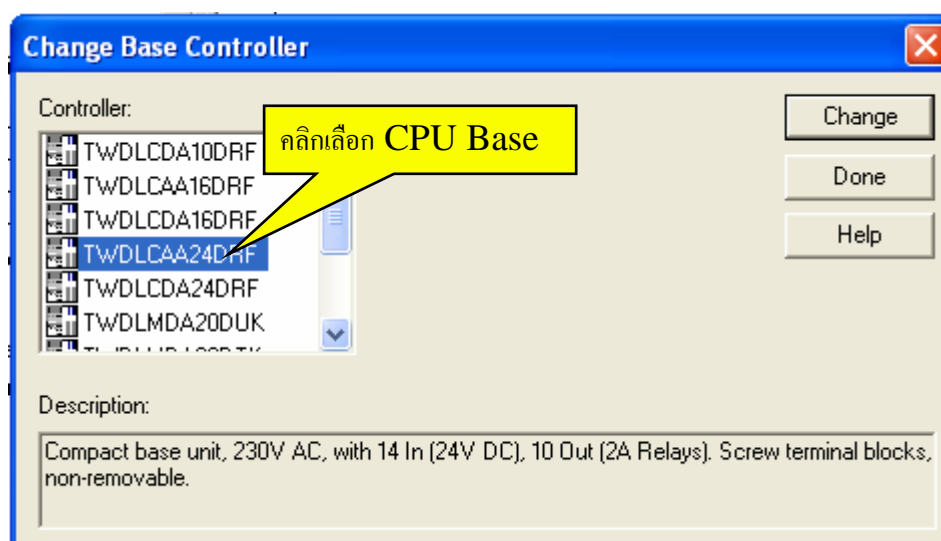
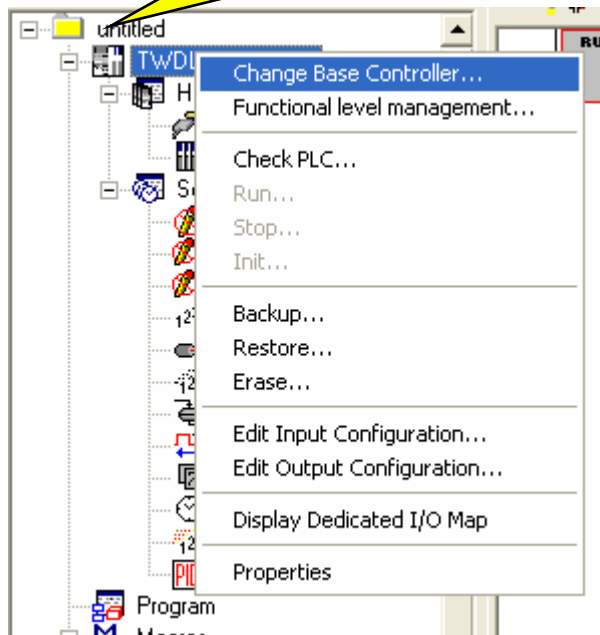
จะเป็นตัวช่วยในการเขียนโปรแกรมให้สะดวกขึ้น โดยจะแสดง Hardware และ Software ที่ใช้งาน และสามารถ Configure Hardware โดยผ่าน Software ได้ทันที



## การ Configure CPU

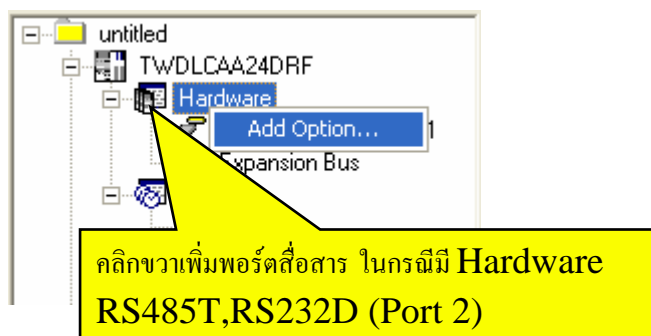
โดยในการใช้งานเราต้องเลือก CPU ให้ตรงกับ พีแอลซี ที่เราใช้งานอยู่

คลิกขวาแล้วเลือก Change BaseController

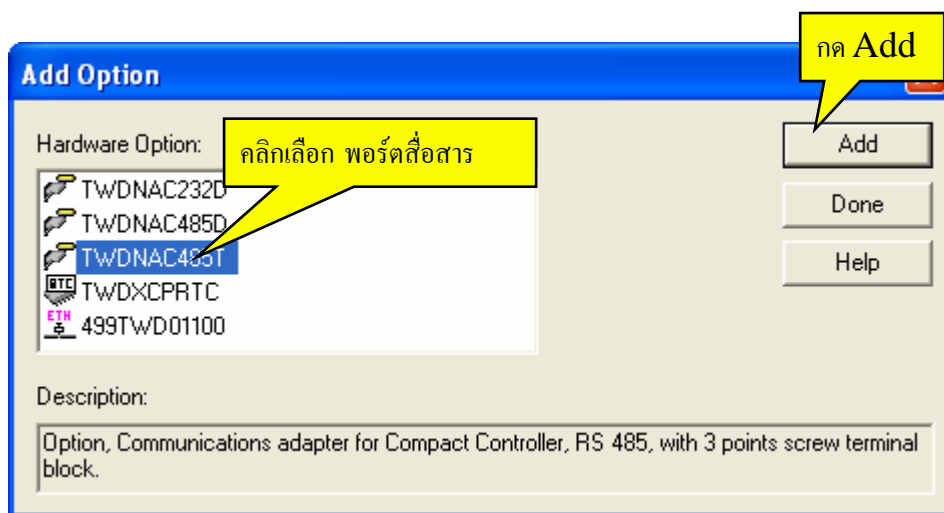


## ***Configure Communication Port***

ในการใช้งานนั้นเราจะต้อง Configure Port ที่ใช้ในการสื่อสาร สำหรับสื่อสารกับคอมพิวเตอร์ และการสื่อสารกับอุปกรณ์อื่นๆ โดยทั่วไป

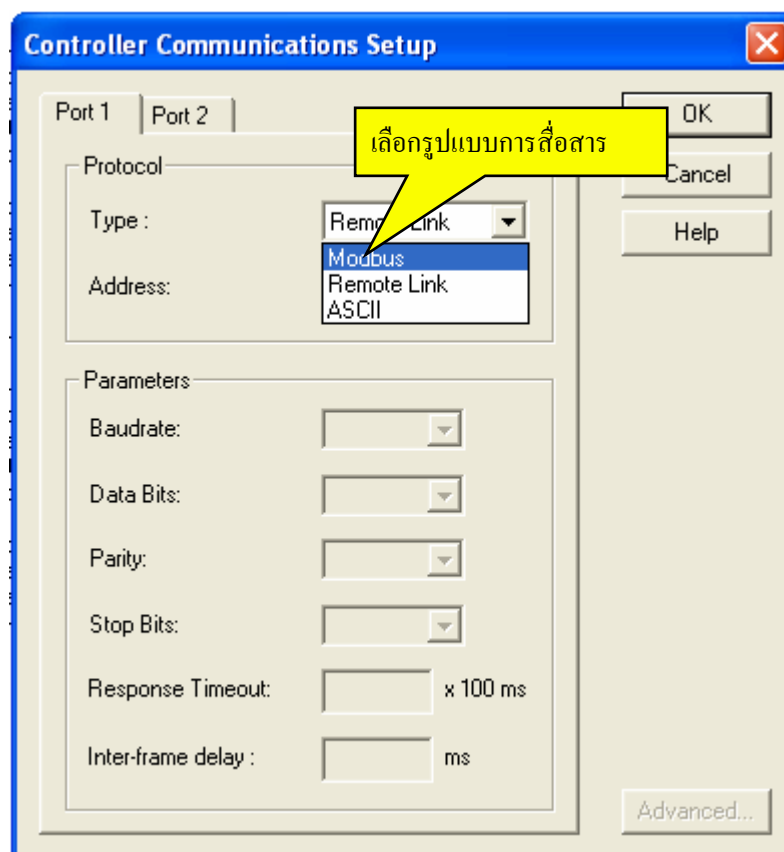
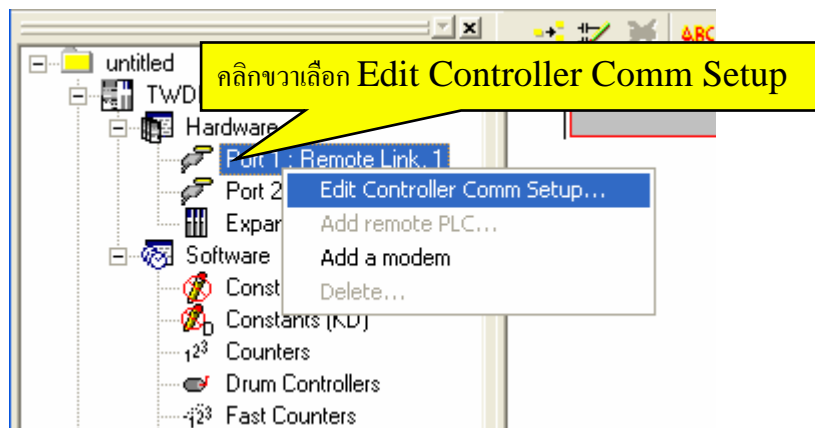


***เลือกพอร์ตสื่อสารให้ตรงกับ Hardware ที่ใช้งานแล้วกด Add***



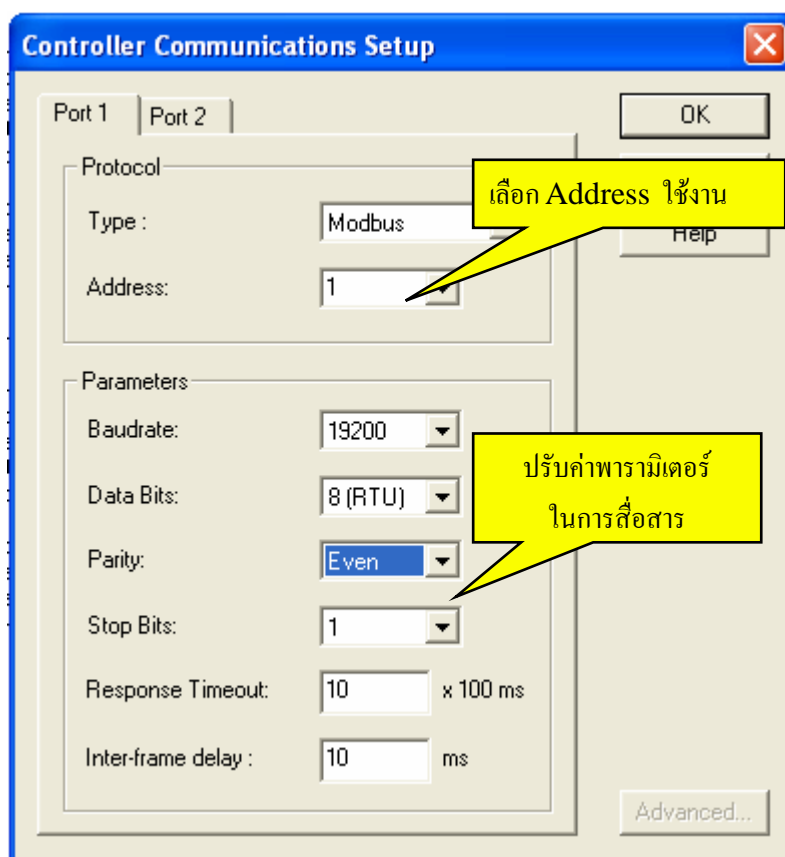
## การปรับตั้งพอร์ตสื่อสาร

การปรับตั้งพอร์ตสื่อสาร เพื่อสื่อสารกับคอมพิวเตอร์ หรือสื่อสารกับพีแอลซี ด้วยตัวเอง หรืออุปกรณ์อื่นๆ โดยผ่าน โปรโตคอล Modbus , ASCII , Remote Link , etc.



## ปรับ Address และพารามิเตอร์ในการสื่อสารต่างๆ

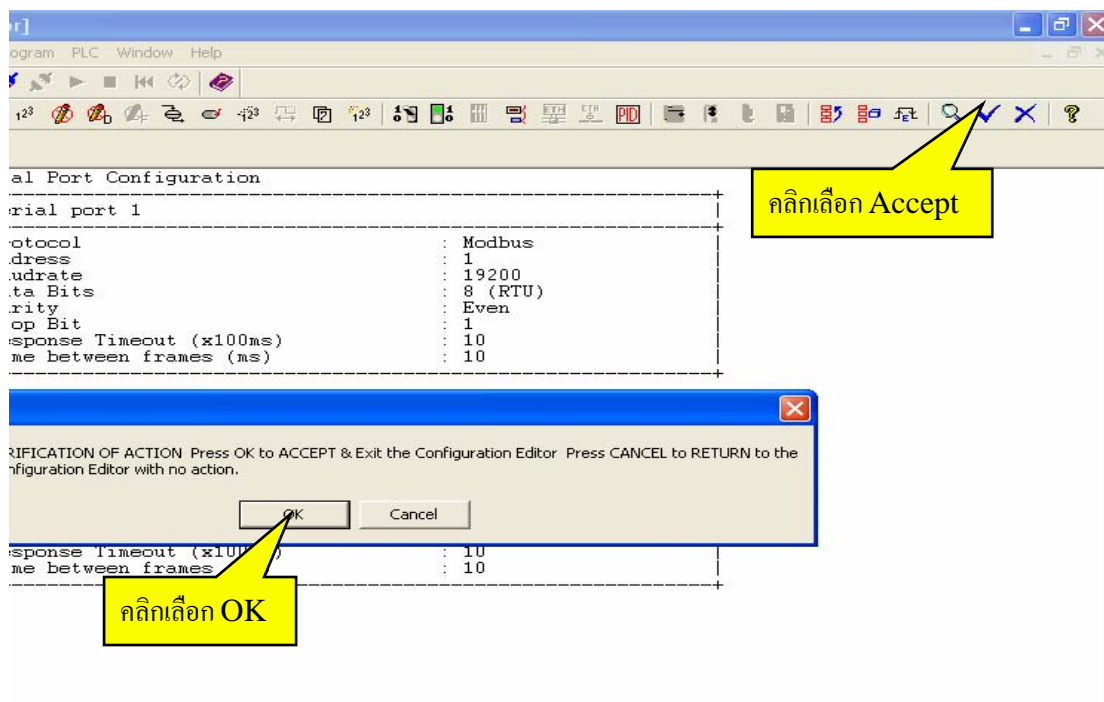
การปรับค่า แอดเดรส ก็เพื่อเป็นการระบุชื่อ หรือหมายเลขของอุปกรณ์แต่ละตัวส่วนพารามิเตอร์ จะเป็นการปรับความเร็วในการรับส่งข้อมูล จำนวนข้อมูลในการรับส่ง 8 บิต หรือ 7 บิต รูปแบบการสื่อสาร Even , Odd ฯ เพื่อความสะดวกในการติดต่อสื่อสาร และเป็นข้อบังคับของโปรโตคอลต่างเหล่านั้น เช่น Modbus , Ethernet , ASCII เป็นต้น



ถ้าเลือกการสื่อสารเป็น Modbus จะสามารถปรับค่า Address และค่าพารามิเตอร์ในการสื่อสารต่างได้



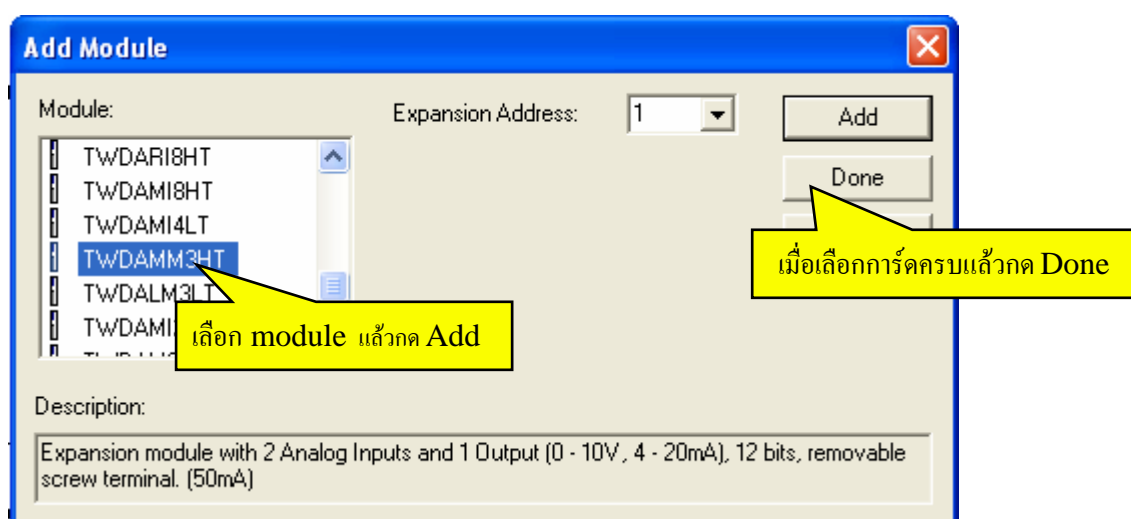
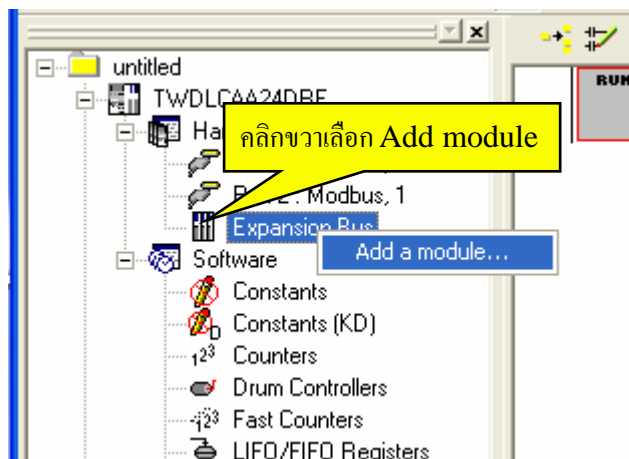
เมื่อเราทำการ *Configure* หรือปรับค่าต่างๆ แล้ว จะปรากฏรายละเอียดต่างๆ ที่เราได้  
ปรับแต่งให้เรายอมรับโดยการ *คลิกที่เครื่องหมายถูก* หรือ *Accept* และคลิก *OK*



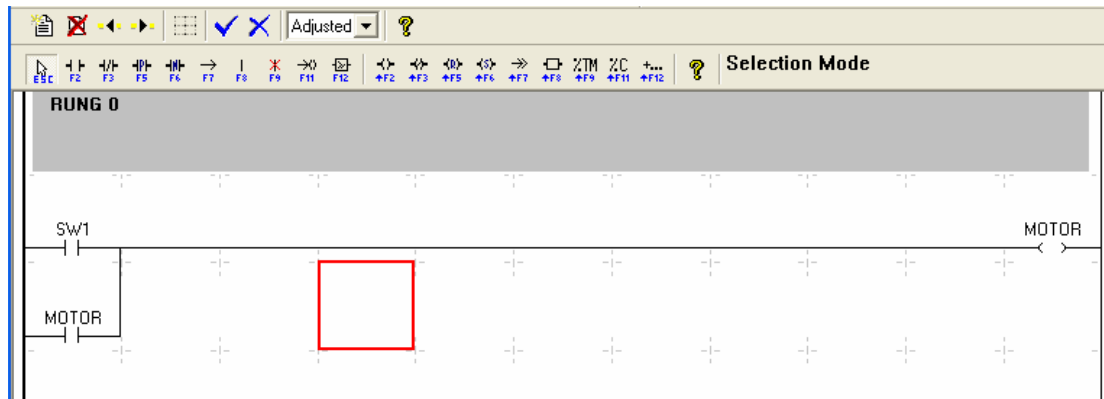
ในทุกครั้งที่มีการ *Configure* หรือปรับค่าพารามิเตอร์ต่างๆ จะปรากฏหน้าต่าง  
รายละเอียดหรือข้อมูล ของอุปกรณ์ที่เราทำการปรับแต่ง ขึ้นมาให้เราได้ตรวจสอบว่าถูกต้องหรือไม่  
ถ้าถูกต้องก็ให้เราเลือก *Accept* หรือเครื่องหมายถูก แต่ถ้าข้อมูลไม่ถูกต้องก็ให้เราเลือก *Cancel*  
หรือเครื่องหมาย ผิด เพื่อไปทำการแก้ไขข้อผิดพลาดก่อนแล้วจึงกดเครื่องหมายถูก

## Configure Expansion Card

การที่จะเพิ่มการ์ดต่างๆ ให้กับ PLC นั้นจะต้องมีการประกาศหรือบอกให้ PLC ได้รับรู้  
หรือที่เรียกว่า การ Configure Module



## การใช้งาน Tool bar สำหรับเขียน โปรแกรม 1

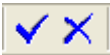


 **New Rung** เพิ่มรังในการเขียนโปรแกรม

 **Delete Rung** ลบรังในการเขียนโปรแกรม

 **Previous Rung And Next Rung** ย้อนกลับไป 1 รัง และ ไปยังรังถัดไป

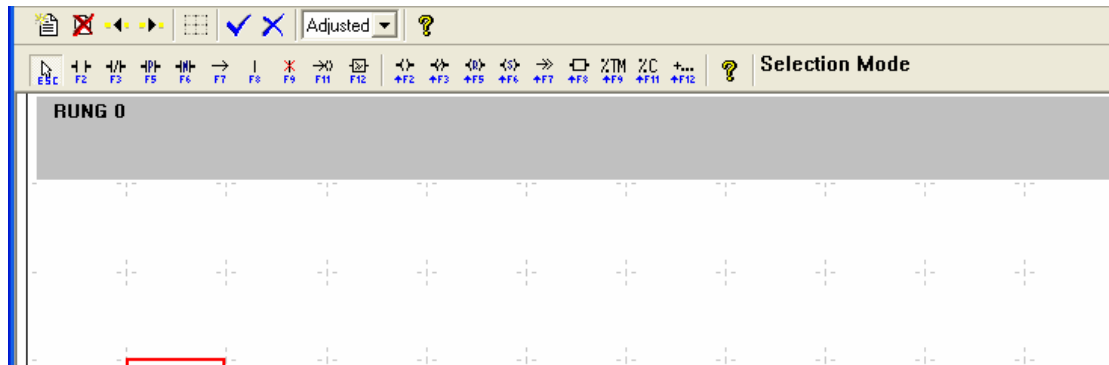
 **On/Off Grid** เปิดปิด กริด

 **Accept Cancel** ยอมรับ หรือ ไม่ยอมรับ โปรแกรมที่เขียนหรือแก้ไข

 **Zoom Factor** การย่อขยายพื้นที่การทำงาน

## การใช้งาน Tool bar สำหรับเขียน โปรแกรม 2

ส่วนนี้จะเป็นส่วนที่ใช้ในการเขียนโปรแกรม Ladder หรือ Instruction List โดยจะมี อินพุต เอาท์พุต และฟังก์ชันบล็อก ต่างๆให้ใช้งาน



ออกจากโหมด



อินพุตแบบปกติเปิด Contact NO



อินพุตแบบปกติปิด Contact NC



อินพุตแบบจับขอบขาขึ้น ↗



อินพุตแบบจับขอบขาลง ↘



Horizontal Connector ใช้เชื่อมต่อสายในแนวนอน



Download Connector ใช้เชื่อมต่อสายในแนวตั้ง



**Erase Down Connector** ใช้ลบสายในแนวตั้ง



**Horizontal Connector Fill** ใช้ต่อสายแนวนอนไปจนถึง คอลย์ เอาท์พุท



**Compare Box** ใช้สำหรับเปรียบเทียบค่าใน Memory หรือค่าคงที่ ต่างๆ



**เอาท์พุทแบบปกติเปิด Coil NO**



**เอาท์พุทแบบปกติปิด Coil NC**



**เอาท์พุทแบบรีเซ็ต Reset Coil**



**เอาท์พุทแบบเซต Set Coil**



**Jump or Subroutine Call**



**Operate Box** ใช้สำหรับเก็บค่าจากการประมวลผล การแปลงค่าต่าง ๆ



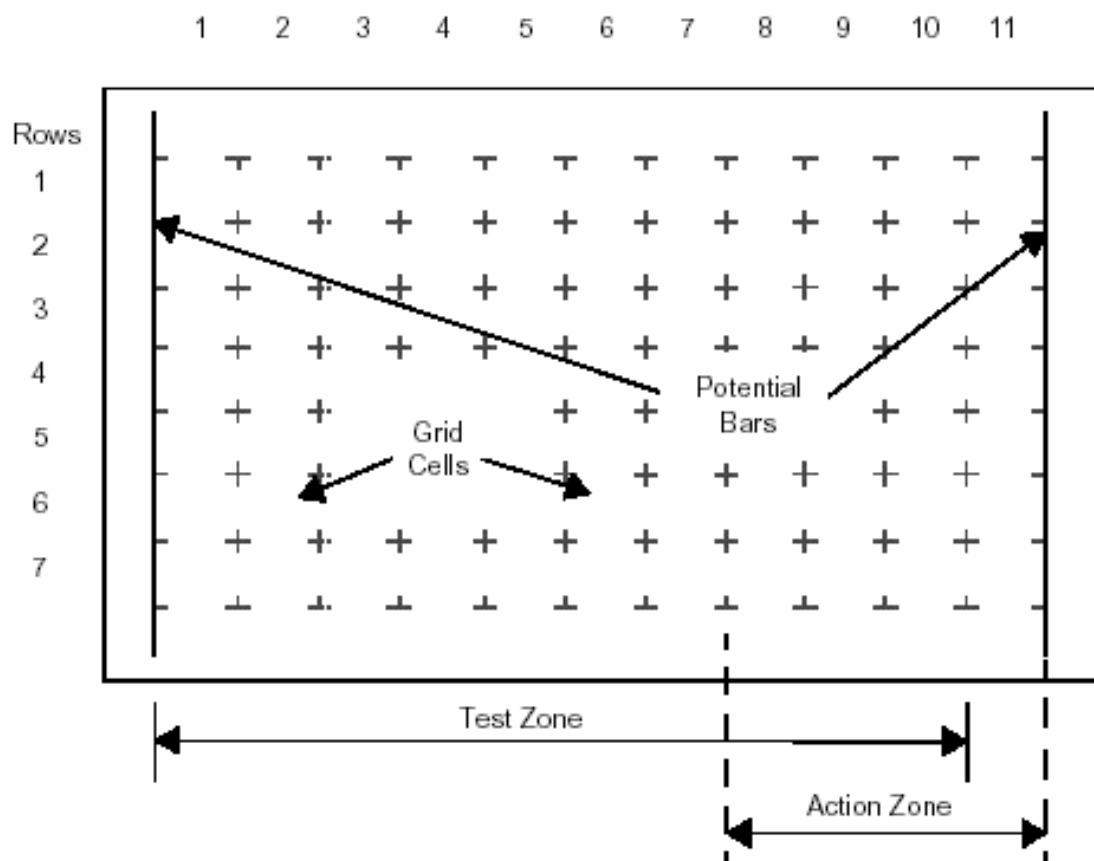
**Timer** ใช้หน่วยเวลาในการทำงาน



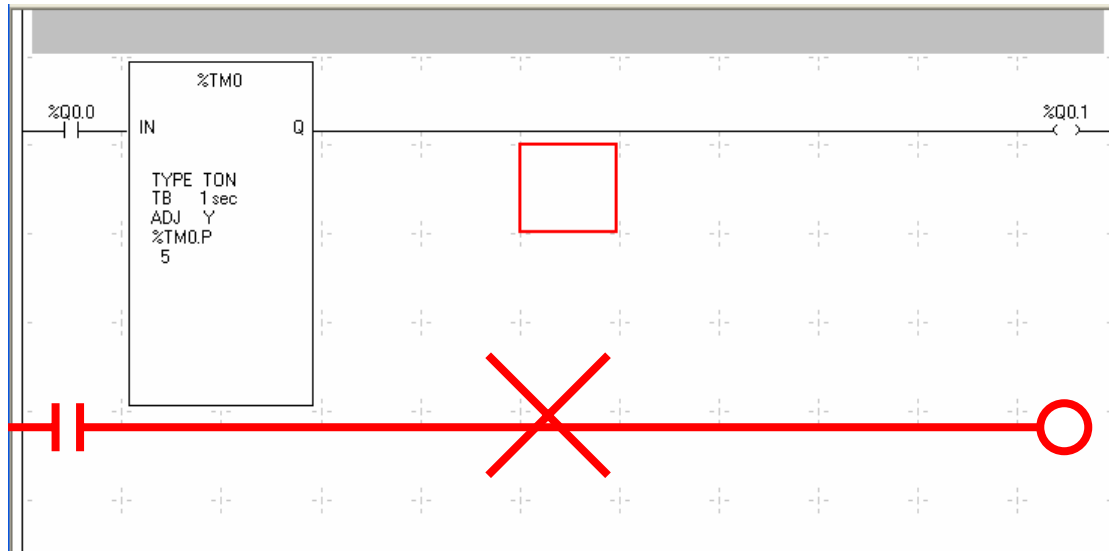
**Counter** ใช้นับเอาท์พุทจากอุปกรณ์ต่างๆ

## การใช้งานพื้นที่สำหรับเขียนโปรแกรม *Application Area*

การใช้งานพื้นที่สำหรับเขียนโปรแกรมนั้นจะต้องเข้าสู่โหมด Insert หรือ Edit โดยสามารถที่จะวาง Contact, Coil หรือ ฟังก์ชันบล็อกต่างลงไปได้ โดยใน 1 รางจะสามารถวาง Contact Input, Coil Function Block ได้ ไม่เกิน 77 ตัว ขึ้นอยู่กับขนาดของพาร์ตอุปกรณ์แต่ละตัว

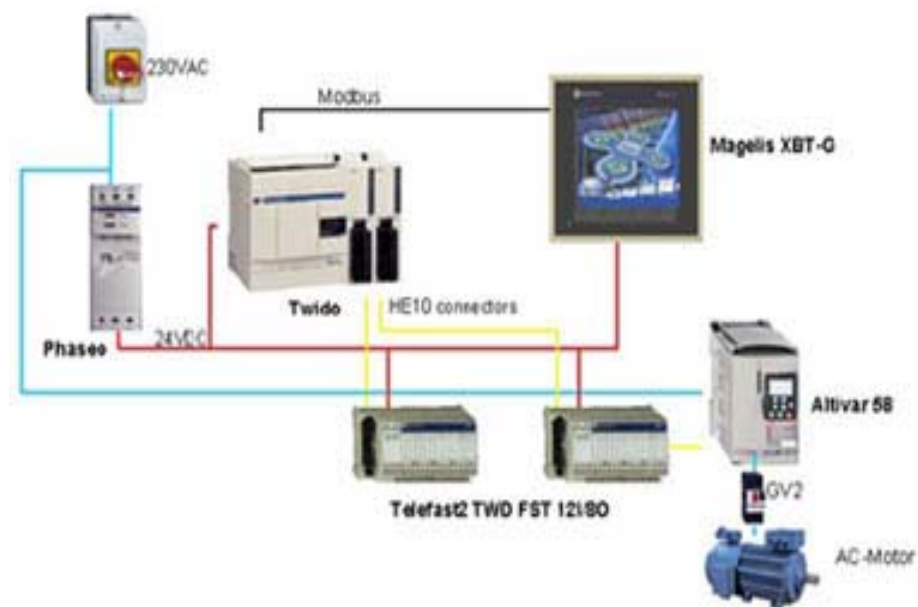


## ข้อควรระวังในการเขียนโปรแกรม



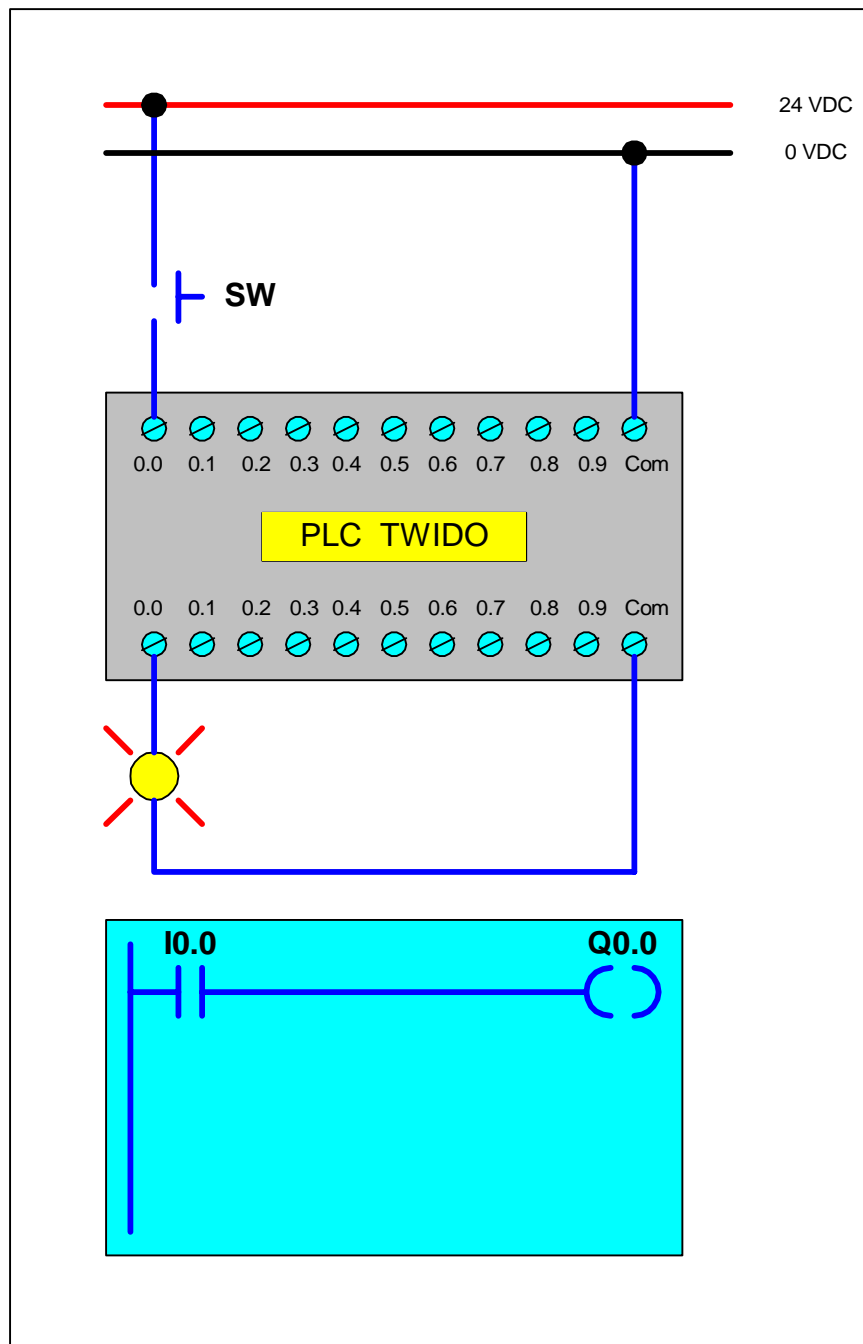
1. ก่อนการเขียนโปรแกรมทุกครั้งให้ทำการ Configure CPU ,  
Communication Port, Expansion ก่อนทุกครั้ง
2. ไม่สามารถเขียน Ladder ข้ามฟังก์ชันบล็อกใดๆได้
3. ไม่สามารถวางบล็อกเดียวกันหรือต่างกันมากกว่า 1 ตัวได้ใน 1 รังได้ เช่น Timer,  
Counter ยกเว้น Compare Box และ Operate Box
4. ต้องเขียนโปรแกรมให้อยู่ในขอบเขตที่กำหนด 7 แถว 11 หลัก ถ้าไม่พอให้ใช้  
ฟังก์ชัน AND OR มาช่วยในการจัดการ
5. ทุกครั้งที่ไฟฟ้าดับหรือไฟที่เลี้ยงวงจรขาดหายไป PLC จะจำสถานะไว้ เมื่อไฟเลี้ยงวงจรมา  
ก็จะทำงานต่อไปได้เลย แต่ถ้าต้องการให้เริ่มการทำงานใหม่ก็ให้ใช้ %S0 หรือ Cold Start  
มาช่วยในการจัดการ

# ***TWIDO Software Operation***




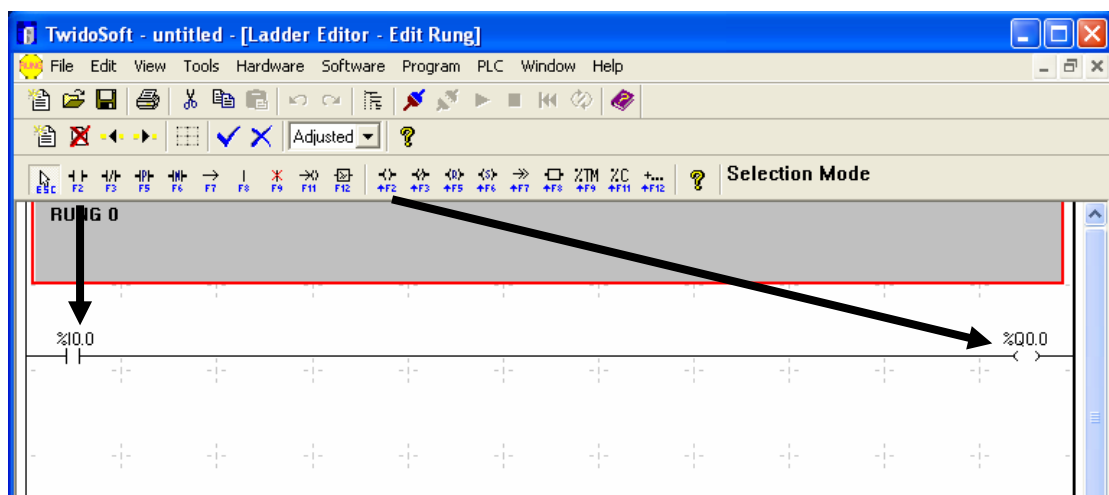


## การใช้งานอินพุท เอาท์พุท




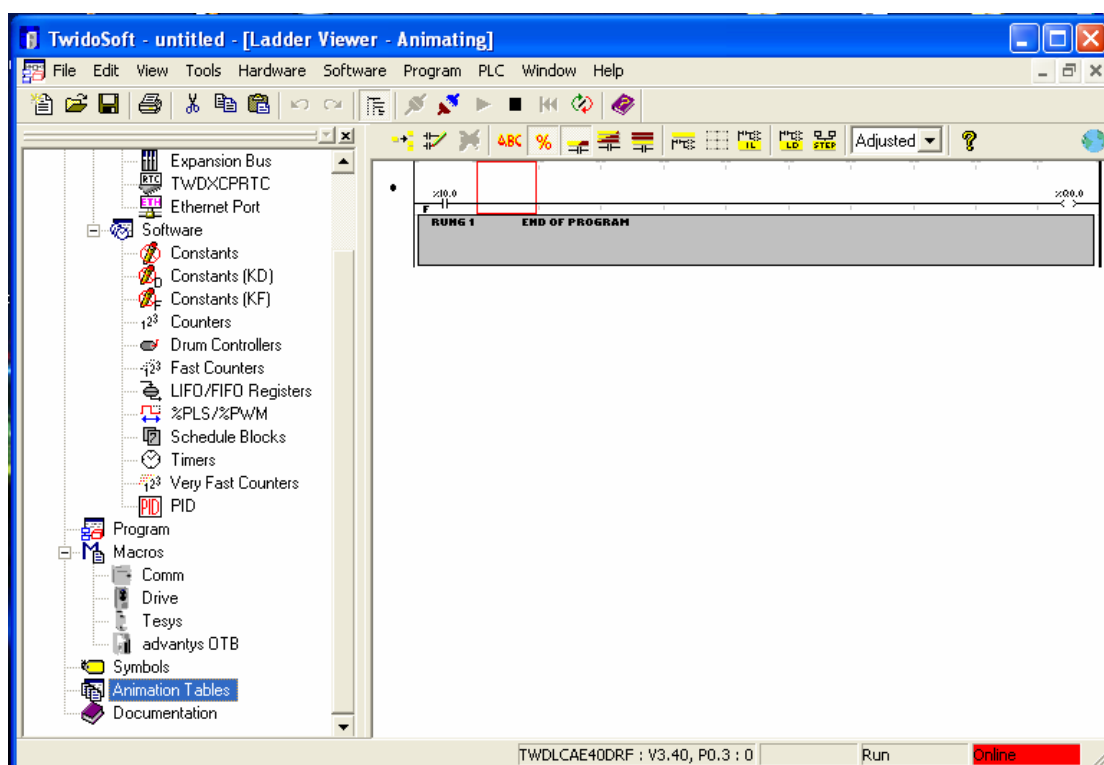
ตัวอย่างการใช้งานอินพุท เอาท์พุท ของพีแอลซี และการมอนิเตอร์โปรแกรม โดยอินพุท  
ที่ใช้เป็น I0.0 เอาท์พุท เป็น Q0.0 เป็นต้น

จากตัวอย่างการใช้งานอินพุต เอาท์พุทของ พีแอลซี ในหน้าที่ผ่านมา เราสามารถที่จะเขียนโปรแกรมได้โดยเข้าไปที่  **Menu Insert Rung** จะปรากฏหน้าต่างดังรูป

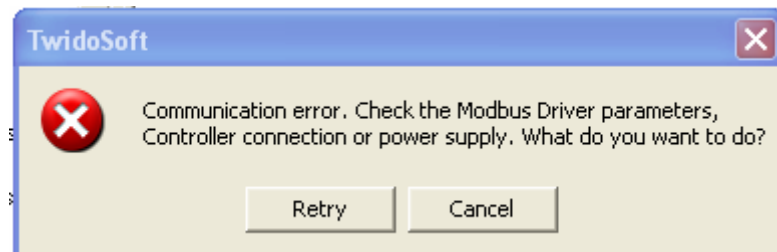


ภายในจะมี คอนแทค คอยล์ ฟังก์ชันบล็อกต่างๆ ให้เราได้เลือกใช้งานโดยการเลือกแล้วนำมาวาง หรือกด F2, F3 ที่ขึ้นอยู่กับพาร์ทแต่ละตัว เมื่อวางแล้วให้เราทำการ ดับเบิลคลิก ที่ตัวคอนแทค หรือ คอยล์ เพื่อใส่ แอดเดรสให้กับอุปกรณ์เหล่านั้น แล้วกด ยอมรับโดยการคลิกที่เครื่องหมายถูก

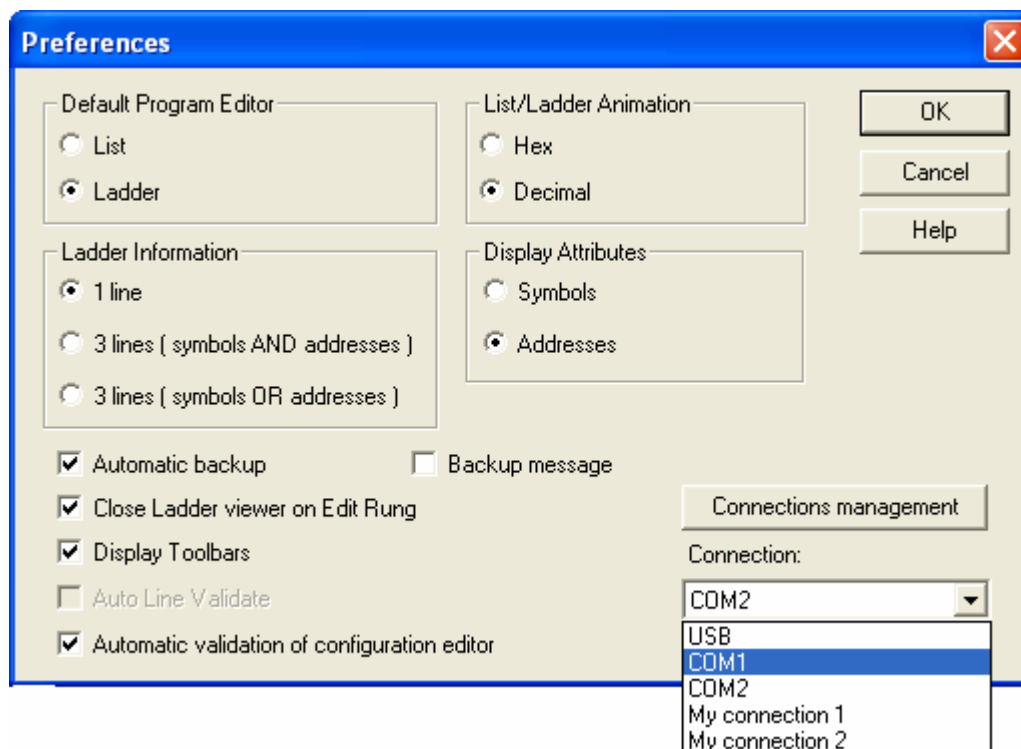
เชื่อมต่อพีแอลซีกับคอมพิวเตอร์ โดยการคลิกที่  แล้วจะปรากฏหน้าจอการทำงานดังรูป โดยให้สังเกตคำว่า *Online* ที่ด้านล่างมุมขวา



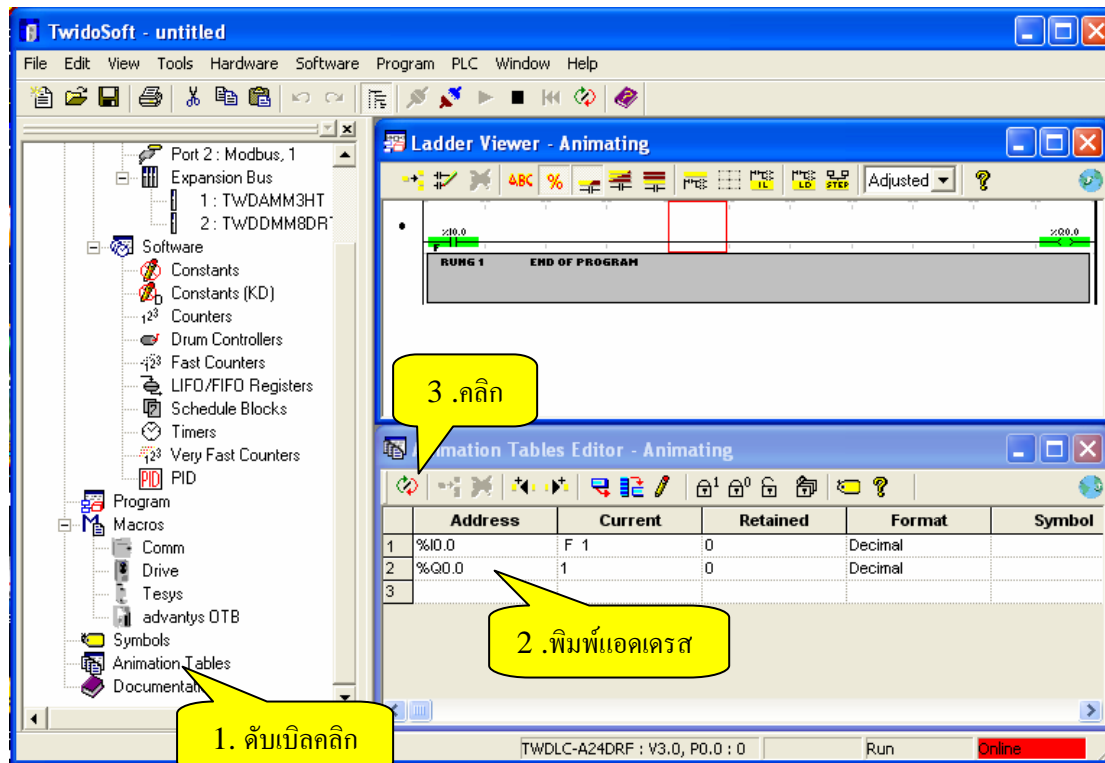
ถ้าไม่สามารถเชื่อมต่อกันได้โดยปรากฏ **Pop up** ดังรูป ให้ทำการแก้ไขโดยเช็คว่  
เสียบสายลิงค์ว่าเสียบอยู่หรือป่าว



ถ้ายังเชื่อมต่อไม่ได้ให้ไปที่ **File → Preferences** แล้วปรับพอร์ตสื่อสาร  
ให้ตรงกับที่ใช้งาน

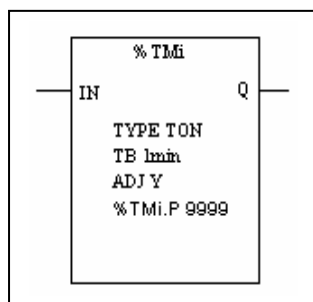


## การใช้งาน Animation Table



การใช้งาน Animation Table เพื่อดูสถานะการทำงานของอินพุท เอาท์พุท และ หน่วยความจำที่เก็บข้อมูล โดยสามารถทำการ Force อินพุท เอาท์พุท หรือใส่ค่า ใน หน่วยความจำเพื่อเป็นการจำลองการทำงาน หรือดีบั๊กการทำงานของโปรแกรม PLC

## การใช้งาน *Timer*



*%TMI* เป็นหมายเลขของ Timer ที่ใช้งาน  
โดยจะสามารถใช้งาน Timer ได้ 128 ตัว

*IN* เป็นสัญญาณ Enable Timer

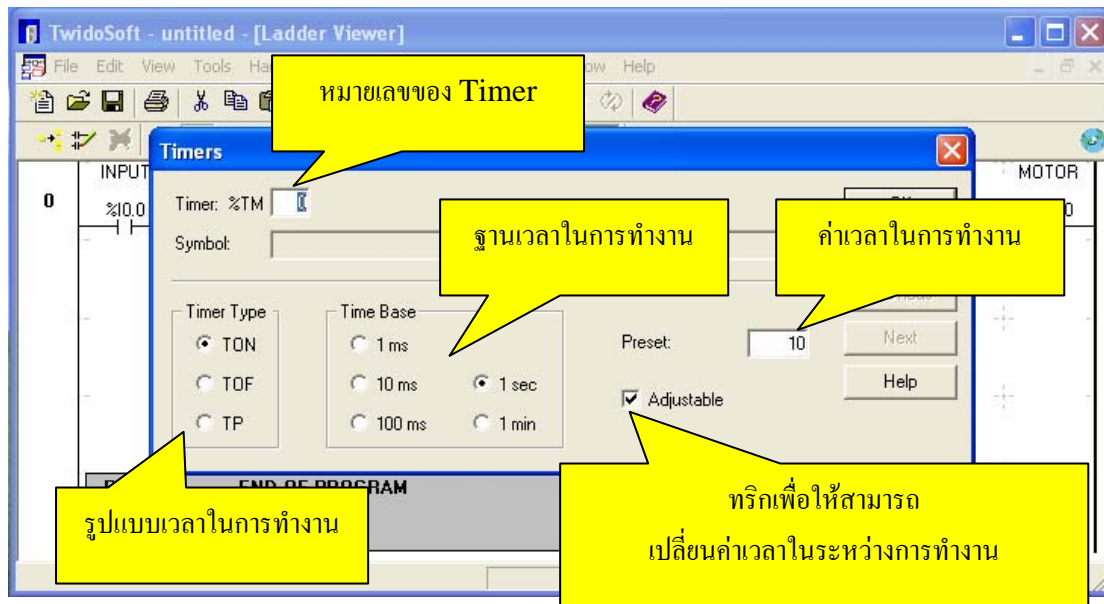
*Type XXX* เป็นชนิดของไทม์เมอร์โดยสามารถ  
เลือกการทำงานได้ 3 โหมด คือ

1. Timer ON
2. Timer OFF
3. Timer Pulse

*TB Time Base* เป็นฐานเวลาในการทำงานของ Timer จะมีให้ใช้งาน  
แบบ คือ 1 ms, 10 ms, 100 ms, 1 Sec, 1 Min

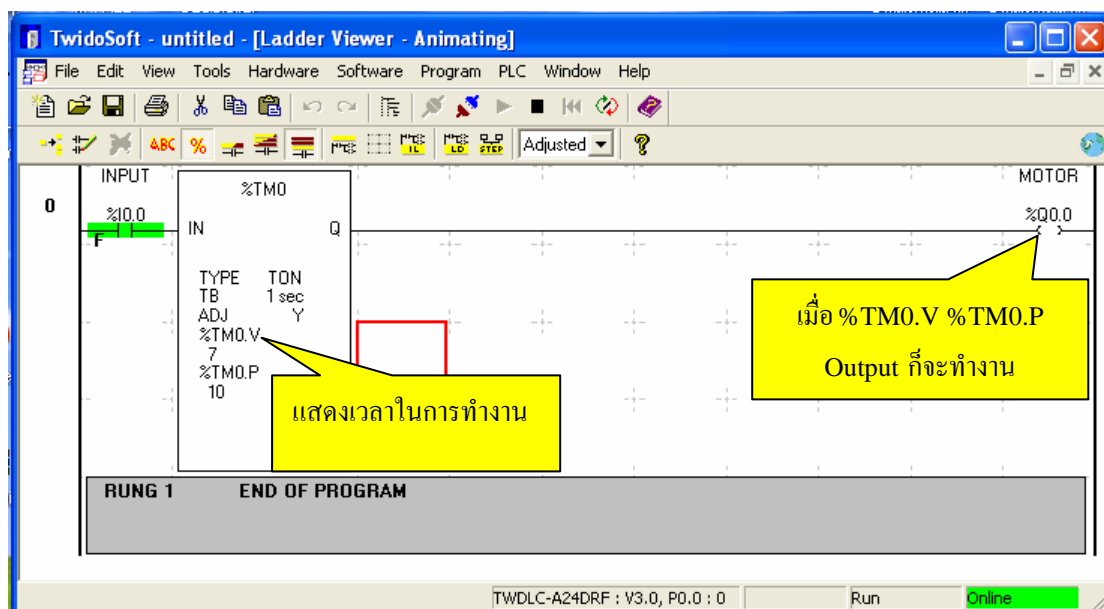
*ADJ Y* เป็นตัวกำหนดว่าจะสามารถแก้ไขเวลาในระหว่างการทำงานได้  
หรือไม่

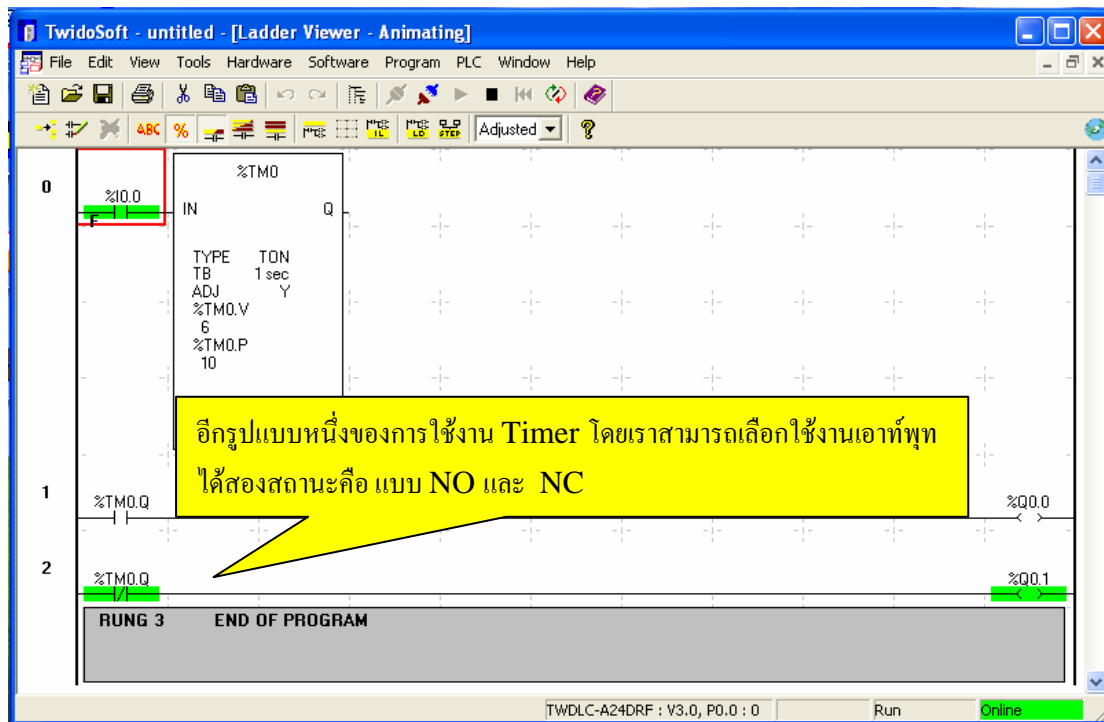
*%TMI.PXXXX* ค่าเวลาในการทำงานจะมีค่า 0 – 9999



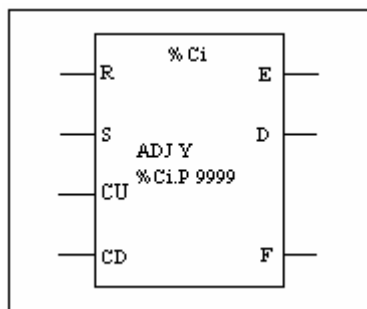
## ตัวอย่างการใช้งาน *Timer*

Timer การใช้เพื่อหน่วงเวลาในการทำงานในช่วงใดช่วงหนึ่ง เช่น หน่วงเวลาในการกดสตาร์ทของอินพุต ประมาณ 10 วินาที แล้ว เอาท์พุทหรือมอเตอร์จึงจะทำงาน





## การใช้งาน Counter



**%Ci** เป็นหมายเลขของ Counter สามารถใช้งานได้ 256 ตัว

**R** เป็นตัวรีเซ็ต Counter โดยจะรีเซ็ตเมื่อได้รับสัญญาณขอบขาขึ้น

**S** เซต เมื่อได้รับสัญญาณ On จะทำให้ Output On

**CU** เมื่อได้รับสัญญาณขอบขาขึ้นจะนับขึ้น 1

**CD** เมื่อได้รับสัญญาณขอบขาขึ้นจะทำการนับลง 1

**E** จะทำงานก็เมื่อค่าการนับน้อยกว่า 0 คือ จะมีค่า เท่ากับ 9999

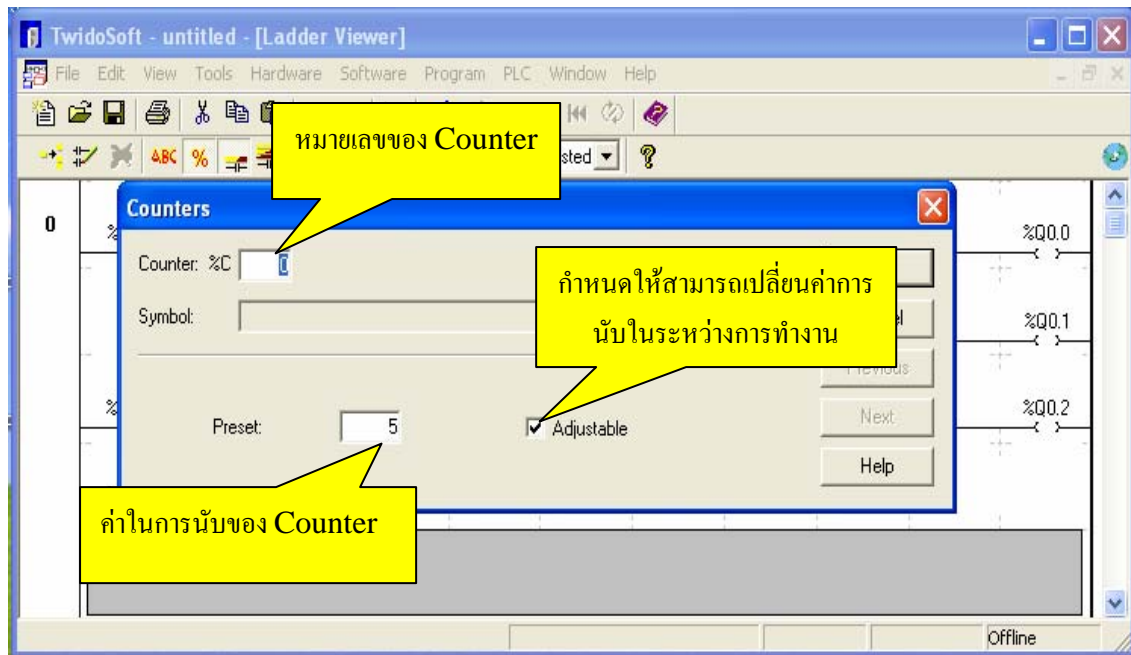
**F** จะทำงานก็ต่อเมื่อค่าการนับมากกว่า 9999 แล้วค่า วนกลับเป็น 0

**D** เป็นเอาต์พุตในกรณีค่าการนับเท่ากับค่าที่ตั้งไว้

**ADJ Y** เป็นตัวกำหนดว่าจะสามารถแก้ไขเวลาในระหว่างการทำงานได้หรือไม่

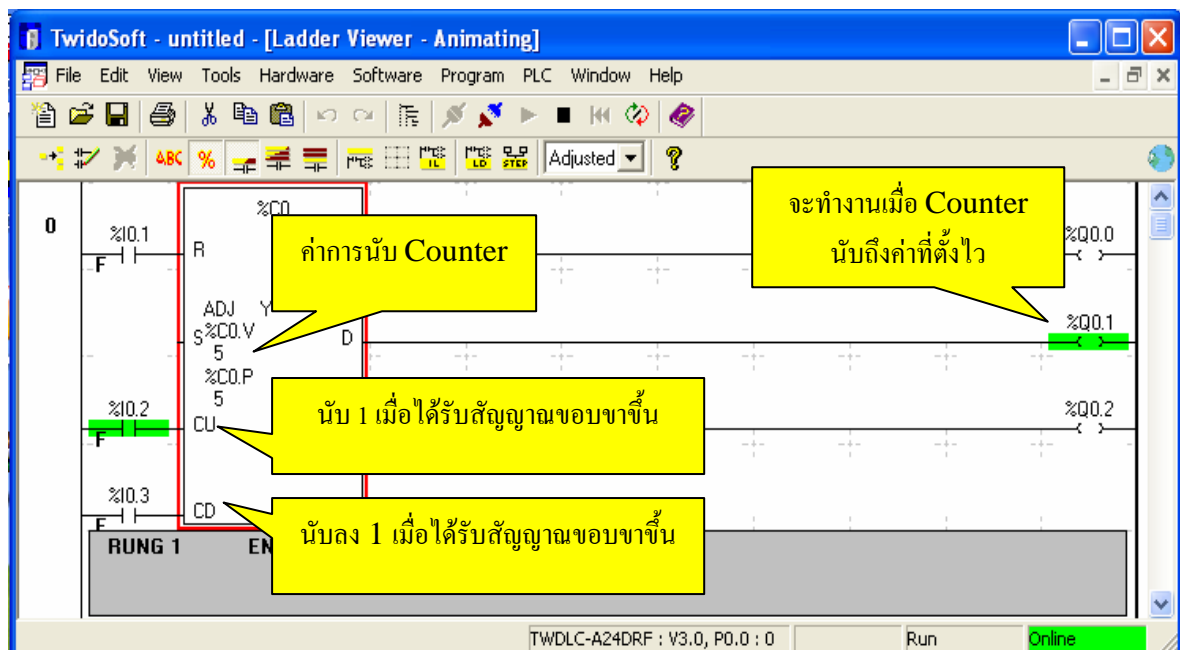
**%Ci.PXXXX** ค่าการนับของ Counter



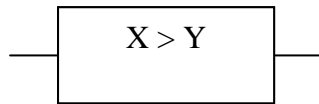


## ตัวอย่างการใช้งาน Counter

การใช้งาน เคอน์เตอร์ของพีแอลซี เพื่อนับสัญญาณอินพุตที่ป้อนให้กับพีแอลซี ตัวอย่าง เช่น การนับชิ้นงานหรือวัตถุดิบต่างๆ ในการผลิต การนับชิ้นงานเพื่อบรรจุลงกล่อง เป็นต้น



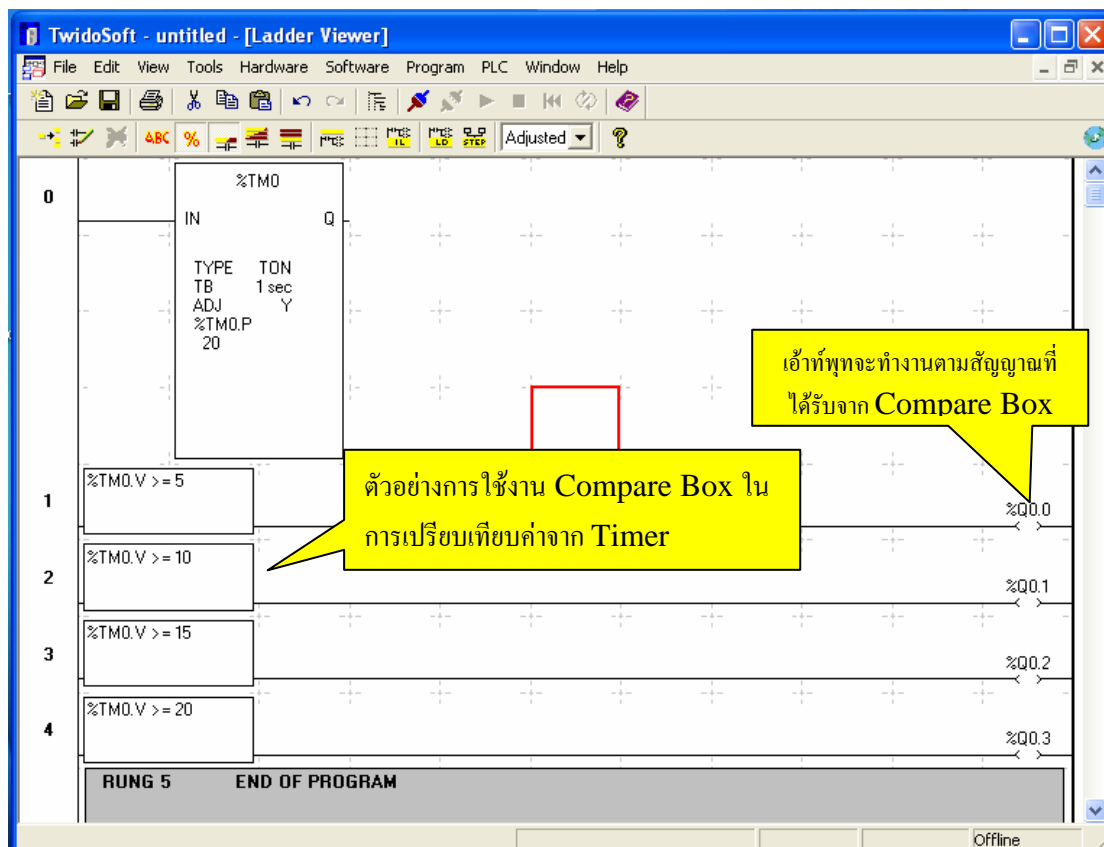
## การใช้งาน **Compare Box**



การใช้งาน Compare Box จะใช้ในการเปรียบเทียบตัวแปรหรือค่าคงที่ 2 ตัวแล้วส่งสัญญาณออกไปควบคุมเอาต์พุต โดยเราสามารถใส่เครื่องหมายทาง ตรรกศาสตร์ ลงไปโดยตรงได้เลยดังนี้

- > มากกว่า
- < น้อยกว่า
- <= น้อยกว่าหรือเท่ากับ
- >= มากกว่าหรือเท่ากับ
- <> ไม่เท่ากับ

## ตัวอย่างการใช้งาน **Compare Box**



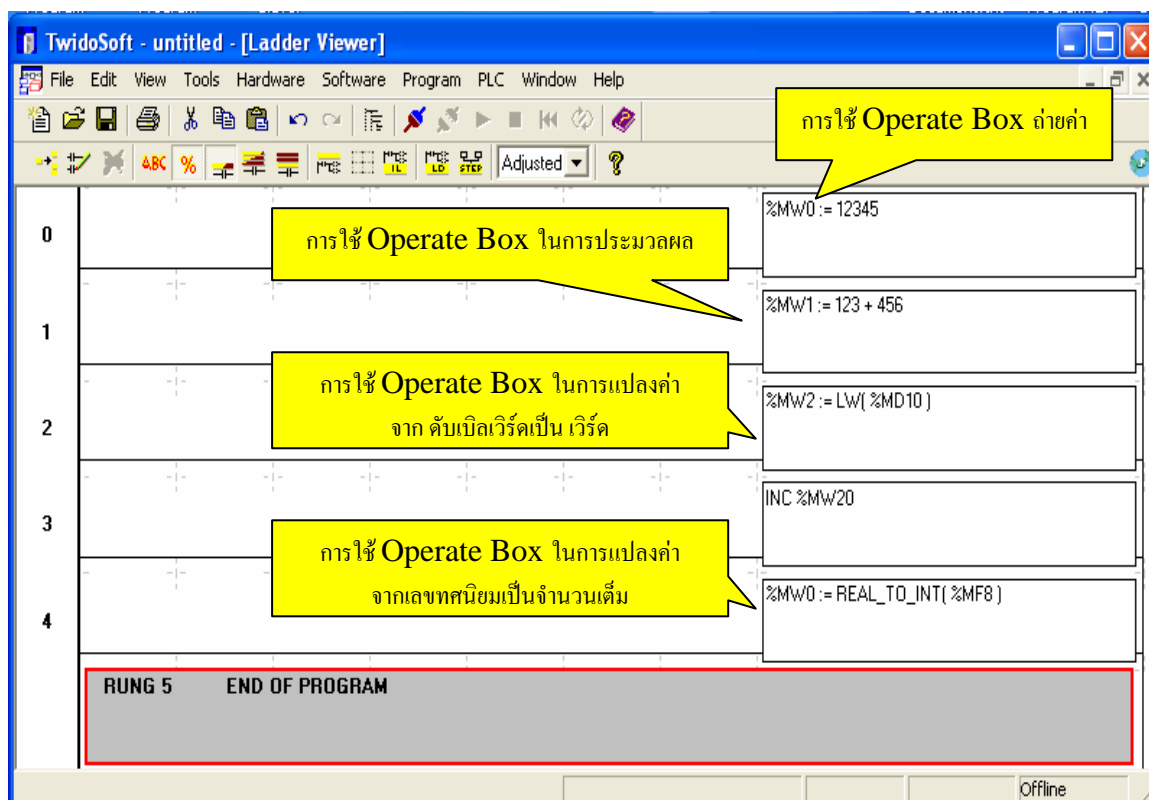
จากตัวอย่างเป็นการประยุกต์ใช้งาน Compare Box ให้ On Output ในช่วงเวลาต่างๆ นอกจากนี้เรายังสามารถนำ Compare Box ไปใช้กับค่าต่างๆ ที่อยู่ใน Memory Word เช่น Analog I/O , High speed Input และอื่นๆ อีกมากขึ้นอยู่กับการประยุกต์ใช้งานในด้านต่างๆ การเปรียบเทียบ สัญญาณอนาล็อก ที่รับเข้ามา

## การใช้งาน Operate Box

%mwi := 1234

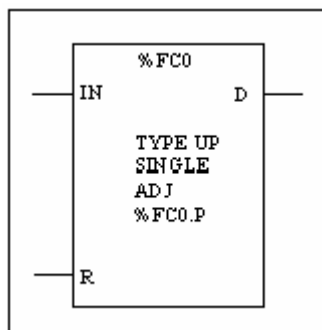
ในการเขียนโปรแกรมของ PLC TWIDO จะต้องมีการถ่ายค่า การเก็บค่าจากการประมวลผล หรือการแปลงข้อมูลต่าง เราจะใช้ Operate Box มาเป็นตัวจัดการกับข้อมูลดังกล่าว ดังตัวอย่างต่อไปนี้

## ตัวอย่างการใช้งาน Operate Box



## การใช้งาน *Fast Counter*

การใช้งาน Fast Counter จะใช้งานกับ งานที่ต้องรับ Input ที่มีความเร็วไม่เกิน 5 kHz โดยในการใช้งานจะใช้ %I0.2 เป็นตัวรับ Input โดยสามารถเลือกได้ว่าจะ นับขึ้นหรือนับลง



*%FCi* เป็นหมายเลขของ Fast Counter

*IN* Enable Fast Counter

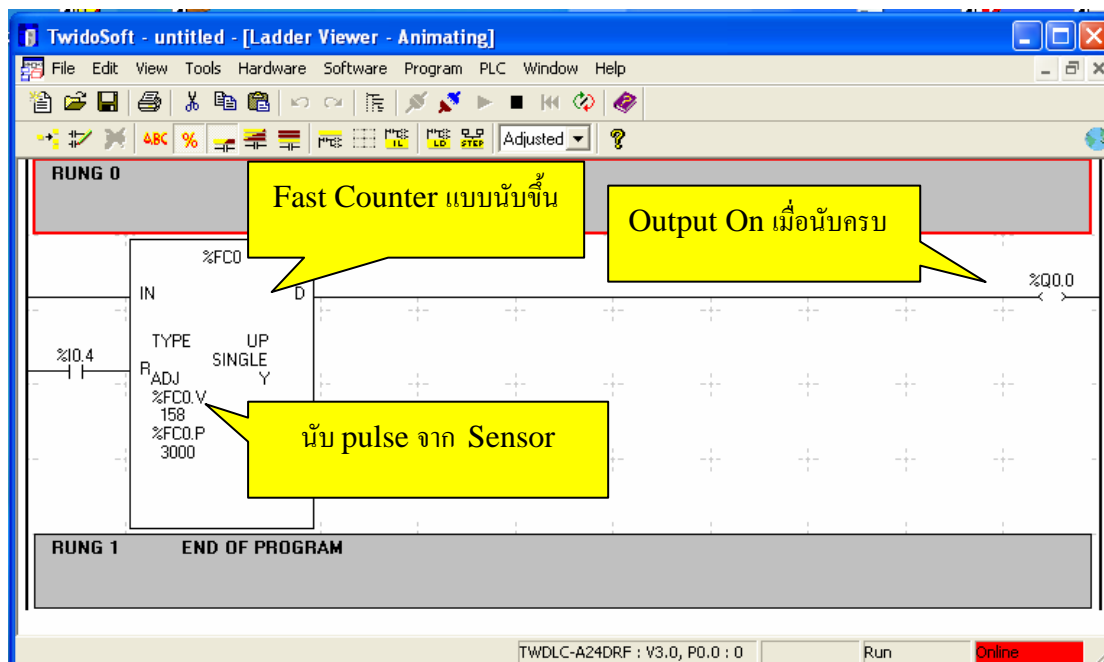
*R* Reset Faster

*D* Output เมื่อนับครบตามที่ตั้งไว้

*%FCi.P* ตั้งค่าการนับของ Fast Counter สามารถนับได้ 65536 และ 42949672956 Double Word

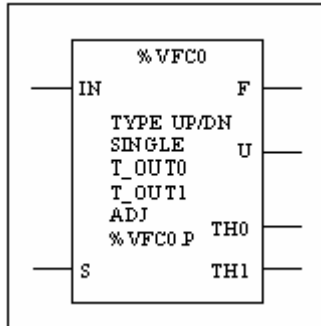
**ข้อควรระวัง** ในการใช้งาน Fast Counter 0 จะไม่สามารถใช้ Input %I0.2 ได้ หรือใน กรณี ใช้งาน Input %I0.2 ในแลตเตอร์โปรแกรมแล้วเราจะไม่สามารถใช้งาน Fast Counter 0 ได้

## ตัวอย่างการใช้งาน *Fast Counter*



จากตัวอย่างเป็นการใช้งาน Fast Counter นับสัญญาณอินพุต จาก Encoder LED Detector และอุปกรณ์อื่นๆ ที่ส่ง เอาท์พุท ความถี่ไม่เกิน 5 kHz

# การใช้งาน Very Fast Counter



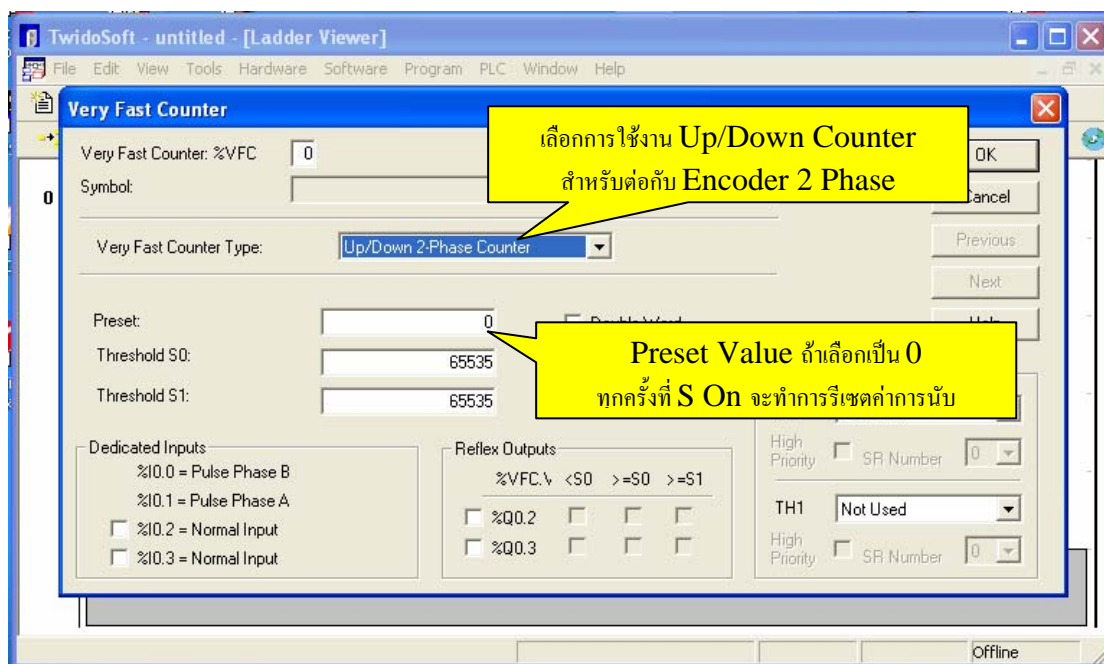
การใช้งาน Very Faster Counter จะสามารถรับอินพุตได้ไม่เกิน 20 kHz และมีฟังก์ชันในการใช้งาน 5 ฟังก์ชันดังต่อไปนี้

1. Up/Down Counter
2. Up/Down 2 - Phase Counter
3. Single Up Counter
4. Single Down Counter
5. Frequency Meter

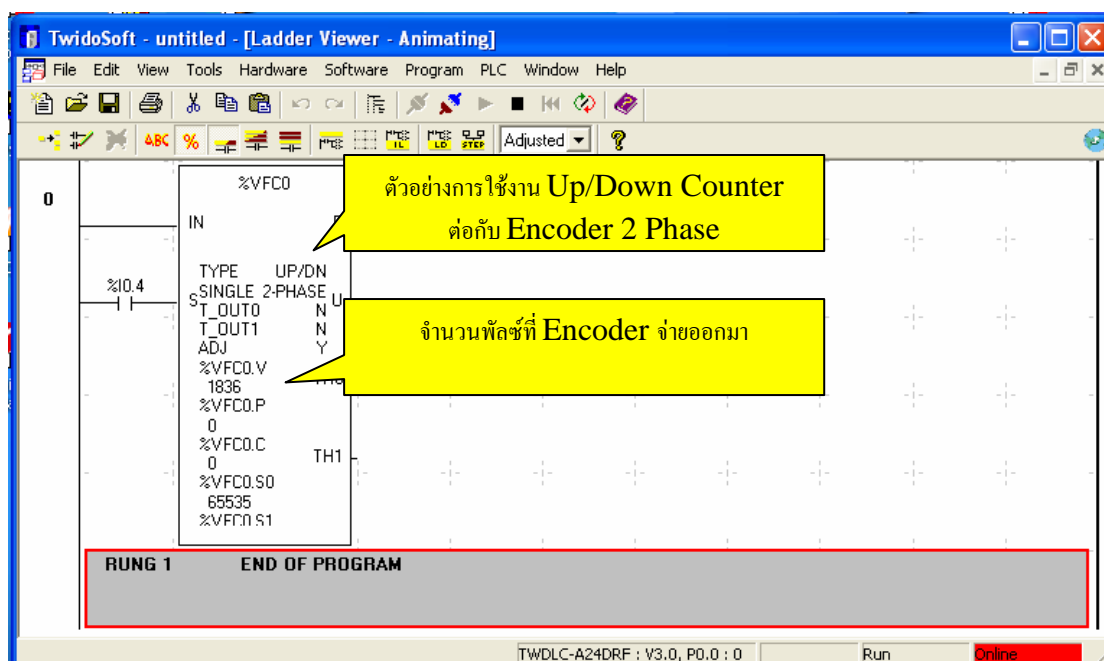
		Main inputs		Auxiliary inputs	
%VFC0	Selected Use	IA input	IB input	IPres	Ica
	Up/down counter	%I0.0.1	%I0.0.0 (UP=0/DO=1)	%I0.0.2 (1)	%I0.0.3 (1)
	Up/Down 2-Phase Counter	%I0.0.1	%I0.0.0 (Pulse)	%I0.0.2 (1)	%I0.0.3 (1)
	Single Up Counter	%I0.0.1	(2)	%I0.0.2 (1)	%I0.0.3 (1)
	Single Down Counter	%I0.0.1	(2)	%I0.0.2 (1)	%I0.0.3 (1)
	Frequency Meter	%I0.0.1	(2)	(2)	(2)
%VFC1	Selected Use	IA input	Input IB)	IPres	Ica
	Up/down counter	%I0.0.7	%I0.0.6 (UP = 0/DO = 1)	%I0.0.5 (1)	%I0.0.4 (1)
	Up/Down 2-Phase Counter	%I0.0.7	%I0.0.6 (Pulse)	%I0.0.5 (1)	%I0.0.4 (1)
	Single Up Counter	%I0.0.7	(2)	%I0.0.5 (1)	%I0.0.4 (1)
	Single Down Counter	%I0.0.7	(2)	%I0.0.5 (1)	%I0.0.4 (1)
	Frequency Meter	%I0.0.7	(2)	(2)	(2)

จากตารางด้านบนจะเป็นรายละเอียดการใช้งาน อินพุต ที่เชื่อมต่อกับอุปกรณ์  
เข้าพุทต่างๆ ซึ่งในการใช้งาน เราจะไม่สามารถใช้งานอินพุต ต่างๆที่ Very fast  
Counter ใช้งานได้

## ตัวอย่างการใช้งาน *Very Fast Counter*



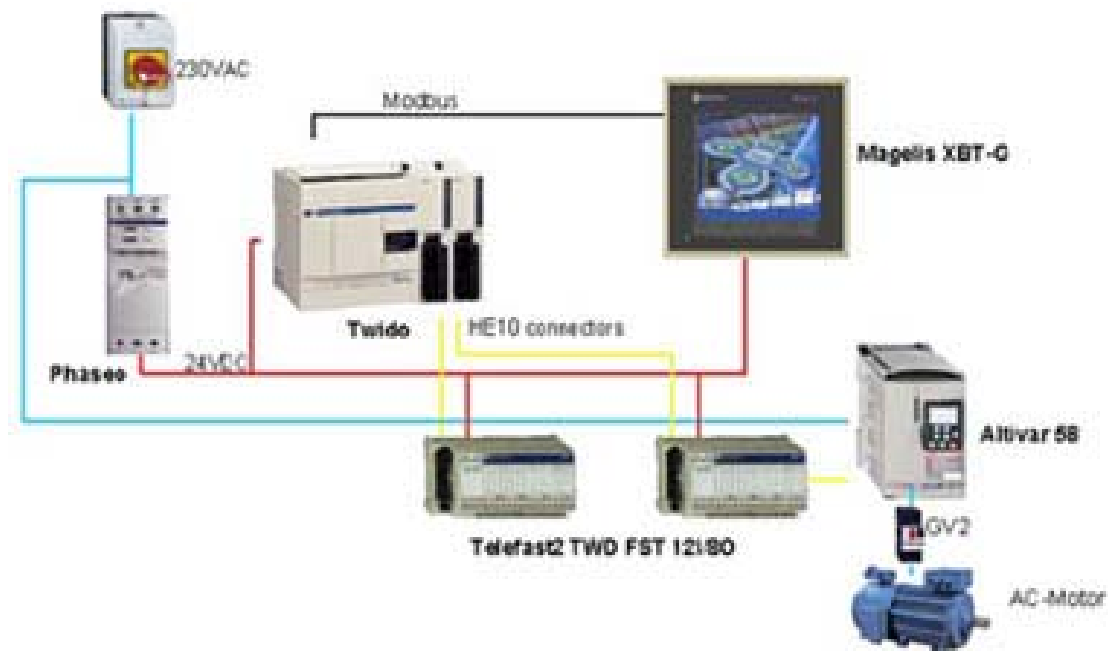
ตัวอย่างการเซตค่า Very Fast Counter สำหรับต่อใช้งานกับ Encoder



จากรูปเป็นค่าพัลส์ที่ Encoder จ่ายออกมาให้กับ PLC เราสามารถนำค่าไปประมวลผลหรือทำงานในรูปแบบอื่นๆได้

## การใช้งาน Modbus RTU

Modbus RTU จะเป็นโปรโตคอลที่ใช้ในการสื่อสารข้อมูล ระหว่าง PLC TWIDO กับ PLC TWIDO หรือใช้ในการสื่อสารกับอุปกรณ์ที่สนับสนุนการสื่อสารแบบ Modbus RTU เป็นต้น โดยสามารถรับส่งข้อมูลได้สูงสุด 38400 Bit/sec และสามารถต่อร่วมกับอุปกรณ์ที่เป็น Modbus RTU ได้ 32 ตัว โดยจะมีหนึ่งตัวเป็นหัวควบคุมหลักหรือที่เราเรียกว่า Master Address 1 และตั้งแต่ Address 2 เป็นต้นไป จะเป็น Slave



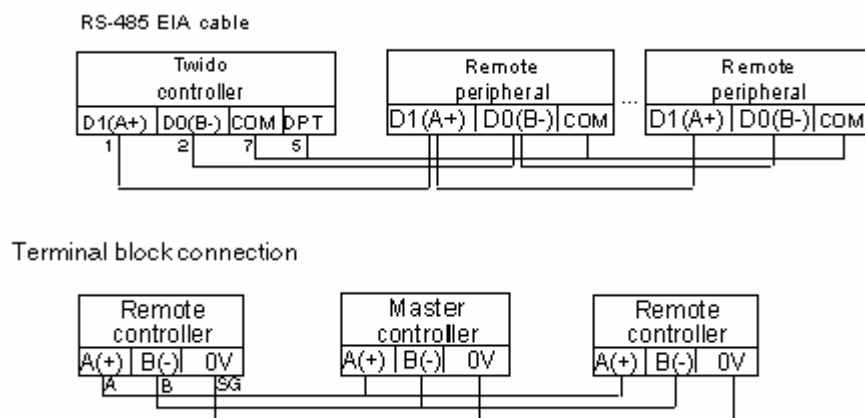
จากรูปเราสามารถสื่อสารกับอุปกรณ์ที่สนับสนุนการสื่อสารแบบ Modbus RTU ต่างๆได้ เช่น จอทัชสกรีน, PLC, Inverter, Temperature Controller, etc. สามารถ Read /Write ข้อมูล หรือสั่งให้อุปกรณ์ทำงาน หรือหยุดทำงานได้ โดยควบคุมจาก Master เพียงตัวเดียว จึงทำให้สะดวก รวดเร็ว และง่ายต่อการควบคุม อุปกรณ์หรือระบบ

โดยการเขียนโปรแกรม Modbus จะเป็นพื้นฐานไปสู่ การเขียนโปรแกรม ASCII , Remote Link, Ethernet, Can Open (field BUS), As Interface (ASI BUS) , etc.

# Configure Modbus RTU

ในการสื่อสารแบบ Modbus RTU จำเป็นต้องเข้าใจรูปแบบการเขียนโปรแกรมโดยรูปแบบการเขียนโปรแกรมจะแบ่งเป็น 3 ส่วนสำคัญ คือ

1. รูปแบบสายที่เชื่อมต่อ
2. Configure Port
3. Software



จากรูปด้านบนเป็นตัวอย่างการต่อสายสำหรับ Modbus RTU ด้านบนจะต่อกับพอร์ต Minidin 8 Pin ส่วนด้านล่างจะเป็นการต่อกับ Option 485T แบบสกรูเทอมินอล

	Most significant byte	Least significant byte
Control table	Command	Length (Transmission/Reception)
	Reception offset	Transmission offset
Transmission table	Transmitted Byte 1	Transmitted Byte 2
	...	...
	...	Transmitted Byte n
	Transmitted Byte n+1	
Reception table	Received Byte 1	Received Byte 2
	...	...
	...	Received Byte p
	Received Byte p+1	

โดยสามารถที่จะดูรายละเอียดได้จาก Help -> Index -> Modbus -> standard request จะเป็นรายละเอียดการ Read / Write Modbus แบบ บิทและเวิร์ด ซึ่งจะเป็นส่วนอธิบายในส่วนคอมมาน หรือเขียนคำสั่ง Read/Write Modbus



Input/Output	Definition	Description
R	Reset input	Set to 1: re-initializes communication or resets block (%MSGx.E = 0 and %MSGx.D = 1).
%MSGx.D	Communication complete	0: request in progress. 1: communication done if end of transmission, end character received, error, or reset of block.
%MSGx.E	Error	0: message length OK and link OK. 1: if bad command, table incorrectly configured, incorrect character received (speed, parity, and so on.), or reception table full.

ใช้ควบคุมการสื่อสาร  
Modbus RTU

คลิกขวาเพื่อทำการปรับ  
ค่าพารามิเตอร์ในการสื่อสาร

เลือกรูปแบบการสื่อสารเป็น  
Modbus และเลือก Address

ค่าพารามิเตอร์ต้องปรับให้ตรงกันทั้ง  
Master และ Slave

## ตัวอย่างการเขียนโปรแกรม

	Table Index	Most significant byte	Least significant byte
Control table	0	01 (Transmission/reception)	06 (Transmission length) (*)
	1	00 (Reception offset)	00 (Transmission offset)
Trans	2	Slave@(1..247)	06 (Request code)
	3	Address of the word to write	
	4	Word value to write	
Reception table (after response)	5	Slave@(1..247)	06 (Response code)
	6	Address of the word written	
	7	Value written	

คำสั่งในการ write 1 word

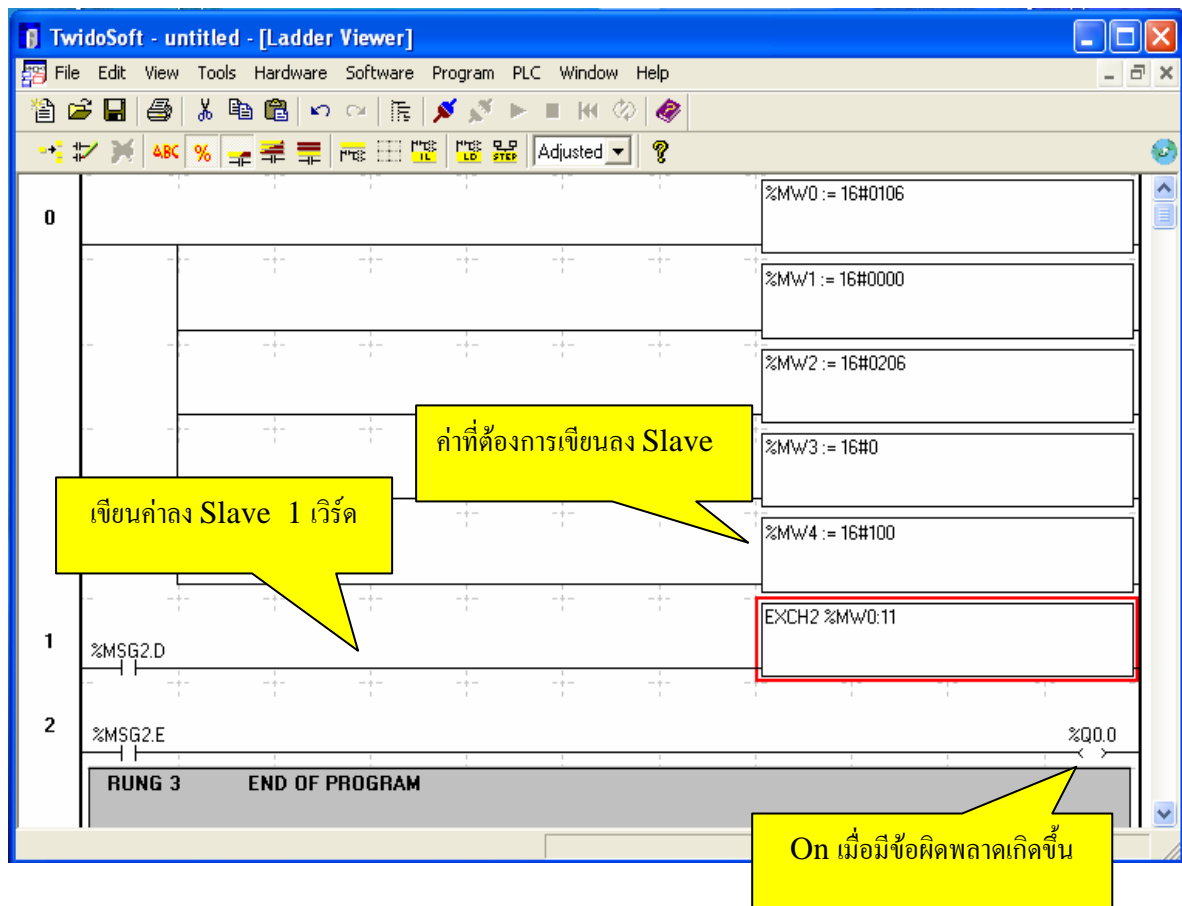
Address Word  
ของ Slave

Address ของ Slave

ค่าที่ต้องการ Write to Slave

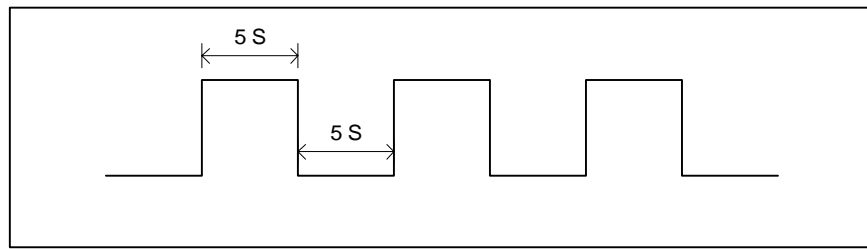
จากตัวอย่างเป็นการเขียนค่า 1 เวิร์ด ไปยัง Slave โดยในการใช้งานนั้นเราจะใช้  
 2 เวิร์ดแรก เป็นคำสั่งในการจัดการสื่อสาร เวิร์ดที่ 2 จะเป็น Slave Address กำหนดได้ 1... 247  
 เวิร์ดที่ 3 จะเป็น Address Word ที่จะเขียนข้อมูลลงไป และ เวิร์ดที่ 4 จะเป็นค่าที่ต้องการเขียน  
 ส่วนเวิร์ดที่ 5, 6, 7 เป็นส่วนที่ใช้ในการตรวจสอบ

## ตัวอย่างการเขียนโปรแกรม

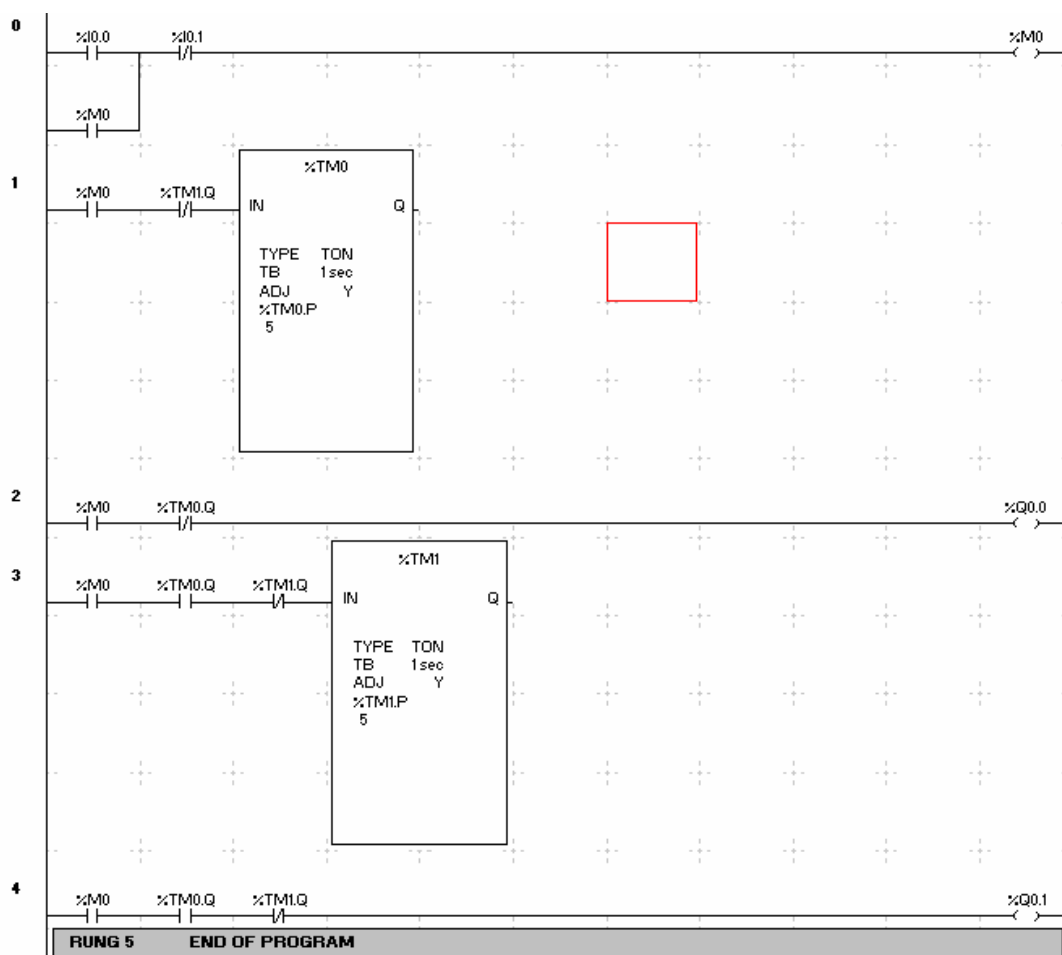


## การทดลองที่ 1

### การสร้างสัญญาณพัลส์



การสร้างสัญญาณพัลส์ โดยใช้ Timer เป็นตัวสร้างสัญญาณพัลส์ เป็นตัวอย่างง่ายๆ  
ในการประยุกต์ใช้งาน Timer และการใช้งาน อินพุต เอาท์พุตเบื้องต้น



## สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

## คำถามหลังการทดลอง

1. ในการทดลองสามารถปรับคาบเวลาของสัญญาณพัลส์ได้หรือไม่

.....

.....

.....

2. ในการทดลองสามารถปรับคาบเวลาในการ ON และ OFF ให้ต่างกันได้หรือไม่

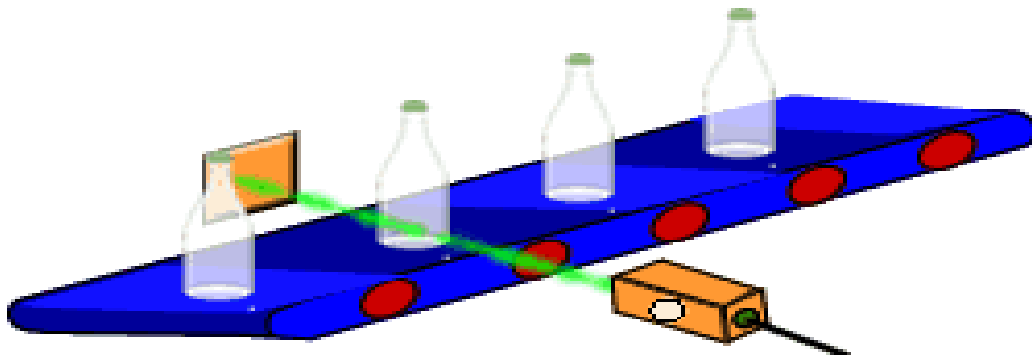
.....

.....

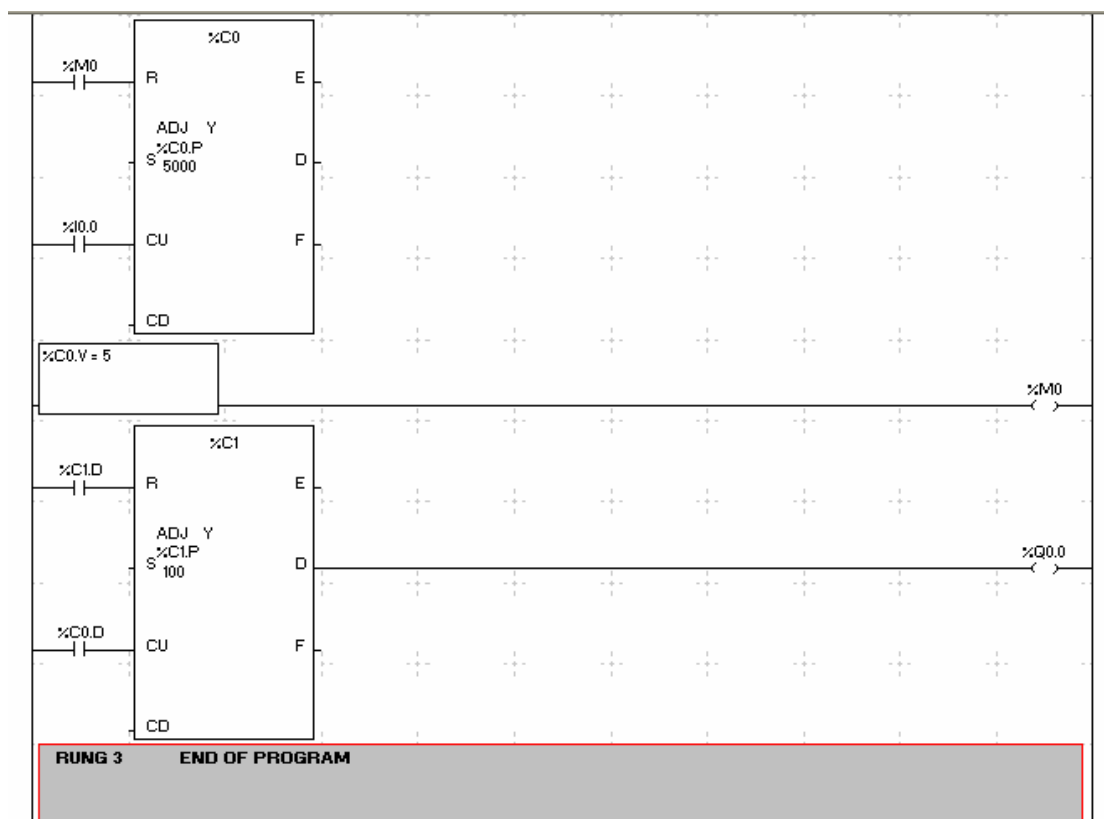
.....

## การทดลองที่ 2

## การใช้งานเคาน์เตอร์นับชิ้นงาน



ตัวอย่างเป็นการใช้งานเคาน์เตอร์นับชิ้นงานที่ส่งผ่านมาทางสายพานลำเลียงจำนวน 500,000 ชิ้น เพื่อนำไปเข้าสู่กระบวนการอื่นๆ ต่อไป



### สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

### คำถามหลังการทดลอง

1. ในการทดลองให้ทำการปรับเกนเตอร์เพื่อให้สามารถนับขึ้นและนับลงได้

.....

.....

.....

2. ในการทดลองหากสัญญาณอินพุตที่ส่งให้กับ พีแอลซี มีความถี่สูงมากแต่ไม่เกิน 5 kHz ควรใช้ฟังก์ชันอะไรมานับสัญญาณดังกล่าว

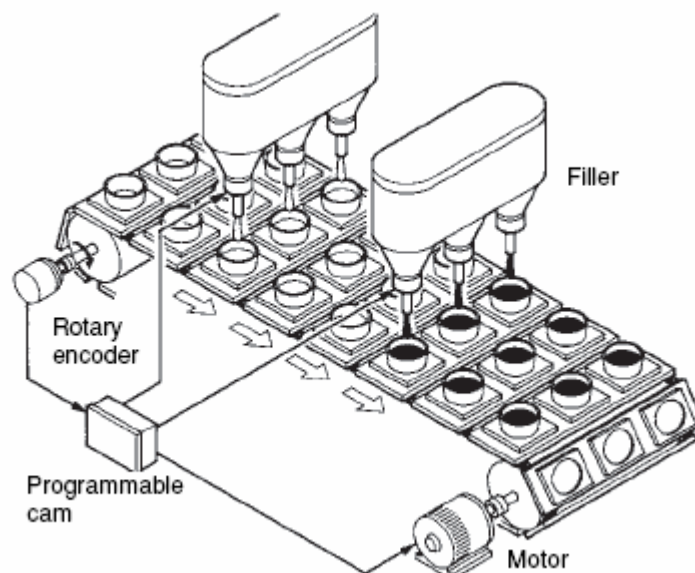
.....

.....

.....

### การทดลองที่ 3

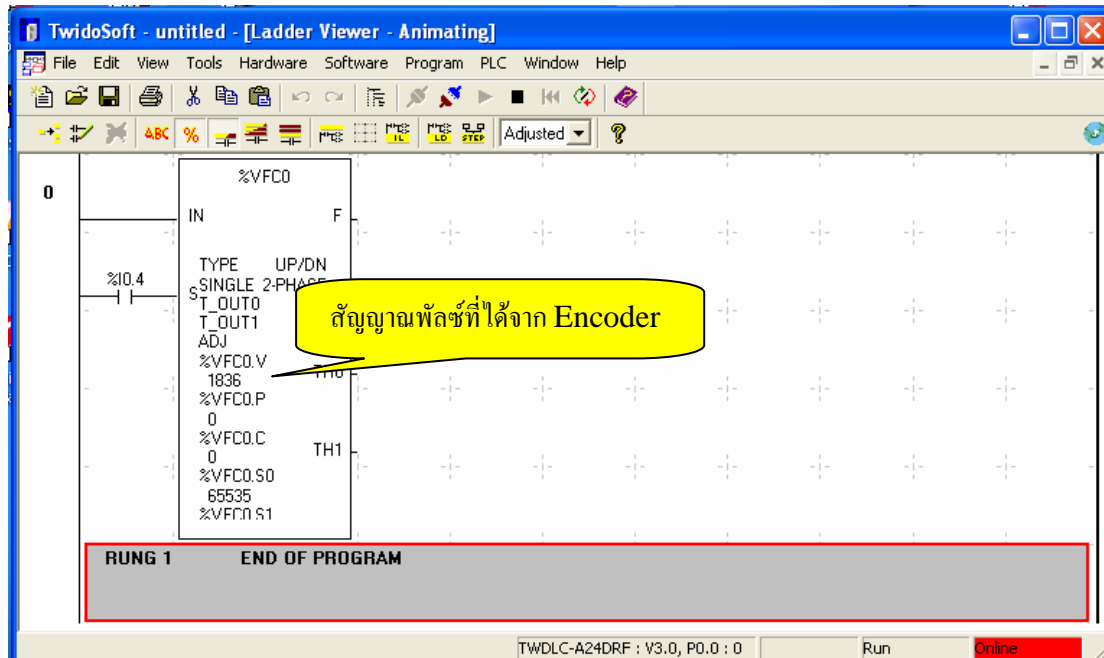
### การรับสัญญาณจาก Encoder



ตัวอย่างการควบคุมการบรรจุของเหลว ต้องมีการควบคุมตำแหน่งการเคลื่อนที่ของสายพานให้อุปกรณ์หรือภาชนะพอดีกับ ตัวหยอดของเหลว โดยคอนโทรลเลอร์จะสั่งให้มอเตอร์หมุน และรับรู้การเคลื่อนที่ ของภาชนะบรรจุได้จาก พัลส์เอาต์พุต ของ Encoder ตัวอย่างเช่นถ้า Encoder หมุนครบหนึ่งรอบจ่ายพัลส์ออกมา 1000 พัลส์ และได้ระยะทางในการเคลื่อนที่ 30 เซนติเมตร ถ้าจ่ายพัลส์ออกมา 2000 ก็จะได้ระยะทาง 60 เซนติเมตร เป็นต้น



## ตัวอย่างโปรแกรมในการรับสัญญาณพัลส์จาก Encoder



การรับสัญญาณจาก Encoder จะใช้ฟังก์ชัน Very fast Counter เป็นตัวนับนับสัญญาณ โดยเลือกใช้งานในโหมด UP/Down 2 Phase Counter โดยค่าจากการนับ จะแสดงที่ %VFC0.V และสามารถที่จะนำค่าดังกล่าวไปทำการประมวลผลหรือ ทำการเปรียบเทียบได้

### สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

### คำถามหลังการทดลอง

1. ในการทดลองสัญญาณพัลส์ที่ได้จากการหมุนของ Encoder มีค่าสูงสุดเป็นเท่าใด

.....

.....

.....

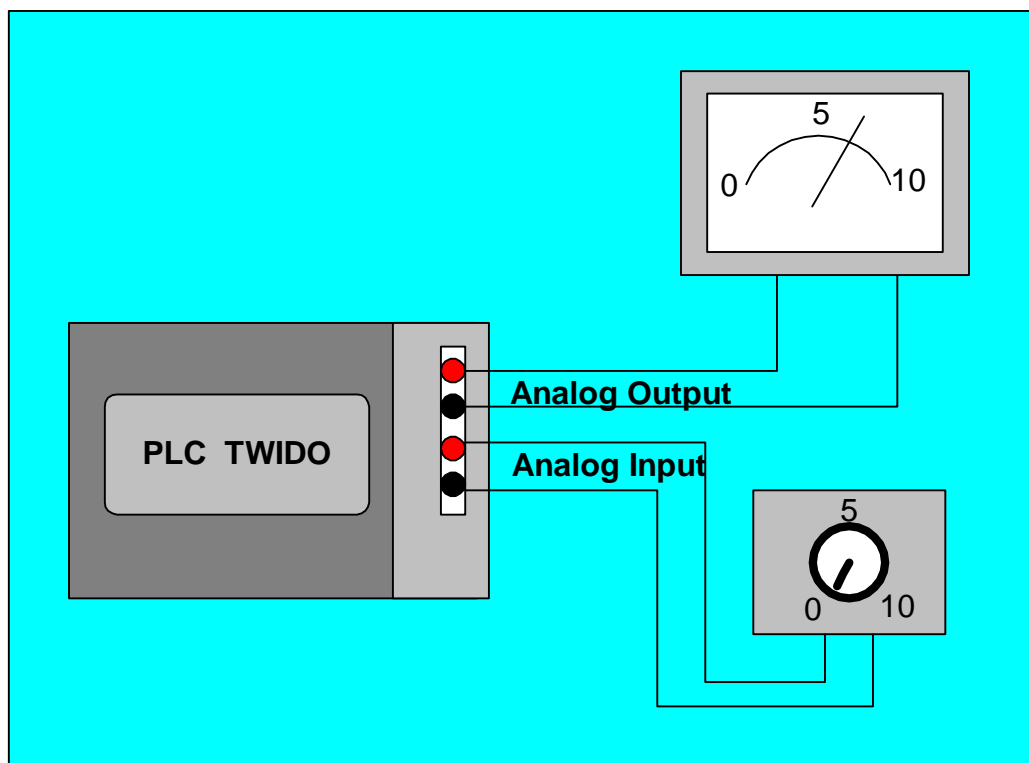
2. ในการทดลองถ้ามอเตอร์หมุน ทั้งซ้ายและขวา จะได้เอาต์พุตออกมาเป็นอย่างไร

.....

.....

.....

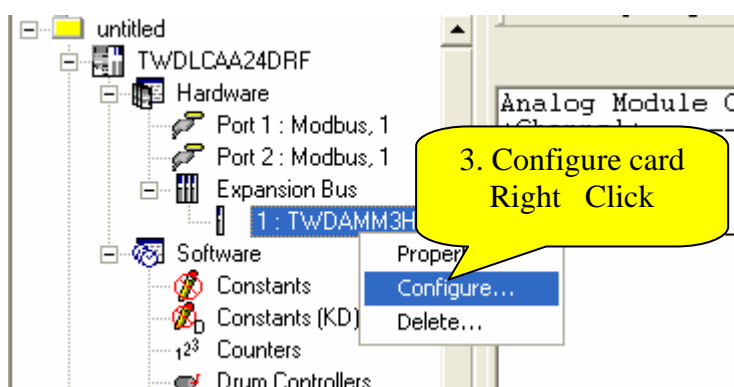
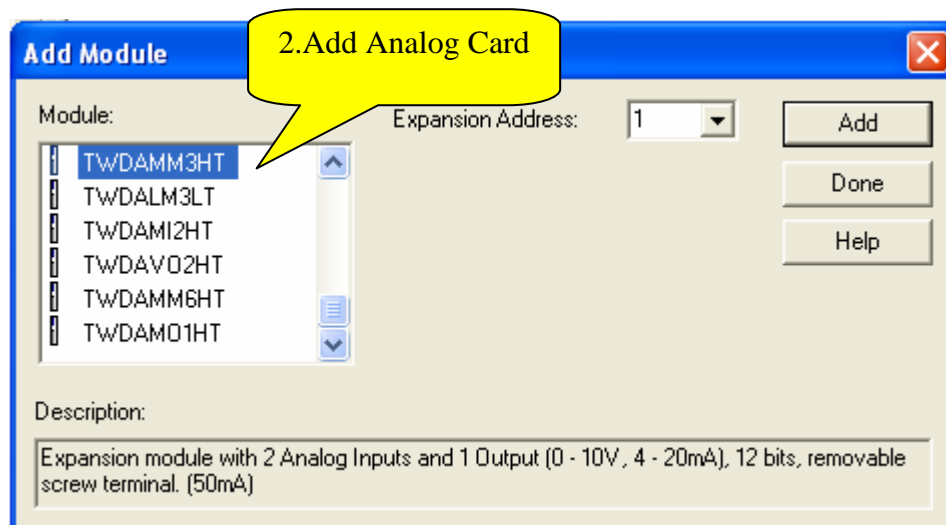
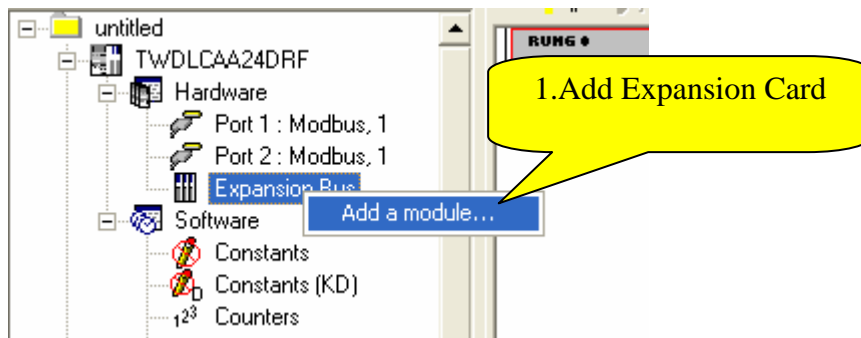
## การทดลองที่ 4 Analog I/O



การทดลองนี้แสดงตัวอย่างการใช้งาน การ์ด อนาล็อก อินพุต เอาท์พุท โดยใช้ การ์ด TWDAMM3HT โดยจะมี Input 2 Ch Output 1 Ch และสามารถเลือกการใช้งาน อินพุต กระแส 4 – 20 mA หรือ แรงดัน 0 – 10 V

ในการทดลอง จะป้อนแรงดัน 0 – 10 โวลต์ ให้กับ Input Analog Card แล้วนำค่าดังกล่าวไปเก็บไว้ยังหน่วยความจำ แล้วนำค่าดังกล่าวป้อนออกจาก Output Analog Card ซึ่งต่ออยู่กับ โวลท์มิเตอร์สำหรับแสดงผล

## Configure Analog Card



**Configure Module - TWDAMM3HT [Position 1]**

Description:  
Expansion module with 2 Analog Inputs and 1 Output (0 - 10V, 4 - 20mA (50mA))

	Symbol		
%MV1.0		Not Used	Normal
%MV1.1		Not Used	Normal
%QW1.0		0 - 10 V	Normal
		4 - 20 mA	

4. Select Volt /Current

5. เขียนโปรแกรม R/W

0

RUNG 1      END OF PROGRAM

%QW0.1.0 := %IW0.1.0  
1214/1214

## ผลการทดลอง

.....

.....

.....

.....

.....

.....

## คำถามหลังการทดลอง

1. จากการทดลองค่าที่ได้จากการทดลองมีค่าเป็นอย่างไร

.....

.....

.....

.....

2. ถ้าเราต้องการค่า 0 ที่แรงดัน 0 โวลต์ และ 10000 ที่แรงดัน 10 โวลต์ สามารถทำได้หรือไม่อย่างไร

.....

.....

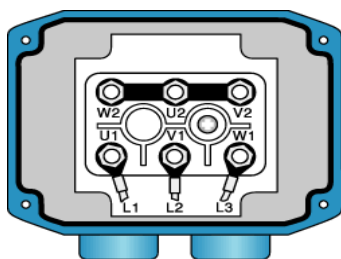
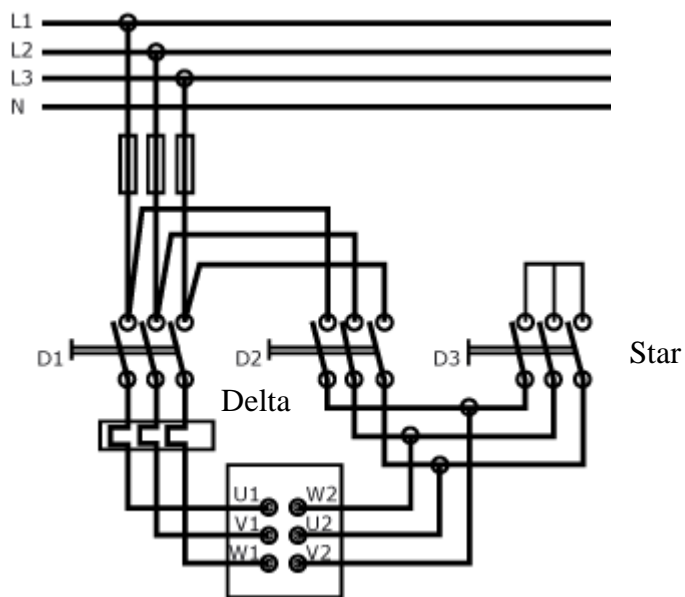
.....

.....

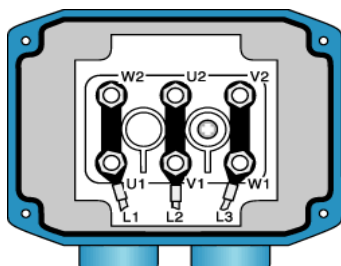
.....

## การทดลองที่ 5 การทดลองการสตาร์ทมอเตอร์ star delta

การสตาร์ทแบบ star/delta จะใช้ก็ต่อเมื่อโหลดของคุณต้องการ starting torque ไม่มากนักเช่นพัดลม โบลเวอร์ หรือปั๊ม เนื่องจากการสตาร์ทมอเตอร์แบบ star/delta เป็นวิธีสตาร์ทโดยการลดแรงดันขณะสตาร์ทลง แต่ตอนรันใช้แรงดันปกติ เพื่อเป็นการลดกระแสขณะสตาร์ท ส่วนทอร์ก ขณะสตาร์ทก็จะลดลงด้วยเหลือ 1 ใน 3 ของทอร์ก พิกัด (เนื่องจากทอร์ก แปรผันตามแรงดันยกกำลัง)



การต่อใช้งานแบบ Star



การต่อใช้งานแบบ Delta

## Ladder Diagram



## ผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....