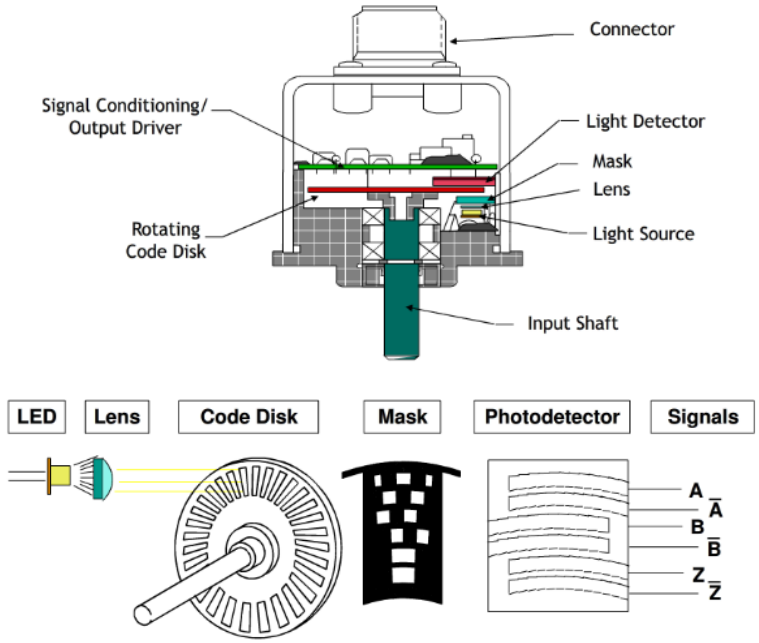
บทที่ 11 Digital Encoder

เป็นอุปกรณ์บอกองศาการหมุน (ตำแหน่งเชิงมุม) โครงสร้างภายในประกอบด้วย 1) เพลา (Shaft) เป็นแกนหมุนต่อเข้ากับอุปกรณ์ 2) แผ่นดิสก์ (Disc หรือ Slit) เป็นแผ่นที่เจาะเป็นแถบเล็กๆให้แสงผ่านได้ 3) Photo detector ใช้รับแสงสำหรับแปลงเป็นรหัสข้อมูล 4) LED ใช้เป็นแหล่งกำเนิดแสง ส่องผ่านเลนส์ (Convex lens) ให้ลำแสงส่องผ่านแถบเล็กๆ บน Disc



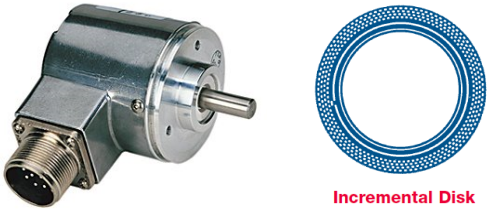
(รูปจาก Rockwell Automation)

Digital Encorder โดยทั่วไปแบ่งได้ 2 ชนิดคือ Incremental Encoder และ Absolute Encoder

Incremental Encoder

บอกองศาการหมุน (ตำแหน่งเชิงมุม) ในรูปของสัญญาณพัลส์สี่เหลี่ยม ความละเอียด(Resolution) ของ Encoder ขึ้นอยู่กับจำนวนพัลส์ต่อหนึ่งรอบการหมุน เรียกว่า Pulse per revolution (PPR) Incremental Encoder มีค่า PPR มาก ยิ่งมีความละเอียดสูง เราสามารถทราบตำแหน่งเชิงมุมได้จากการนับจำนวนพัลส์ โดยใช้การ์ด Counter เทียบกับ PPR

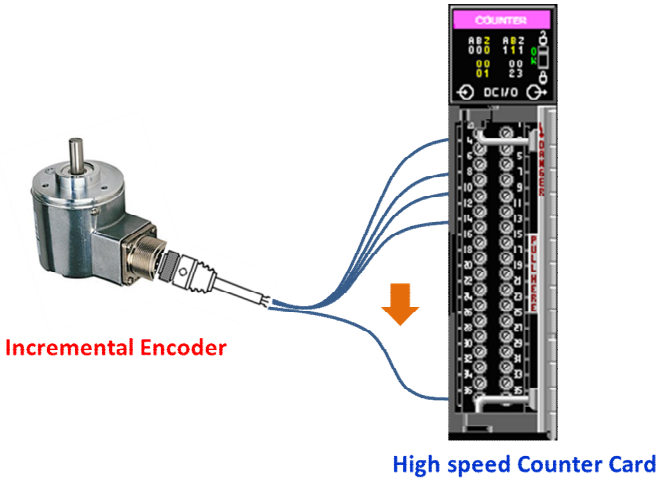
โดยทั่วไป Encoder ชนิดนี้มีสัญญาณ Output อยู่ 2 สัญญาณ คือ A และ B ซึ่งมีเฟสต่างกัน 90 องศา แต่บางรุ่นได้เพิ่มสัญญาณตอนหมุนครบรอบ เรียกว่า “Marker” หรือ Channel Z เข้ามาด้วย



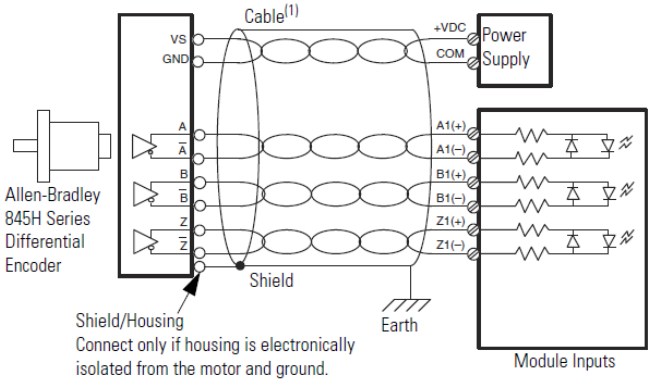
(รูปจาก Rockwell Automation)

ตัวอย่าง

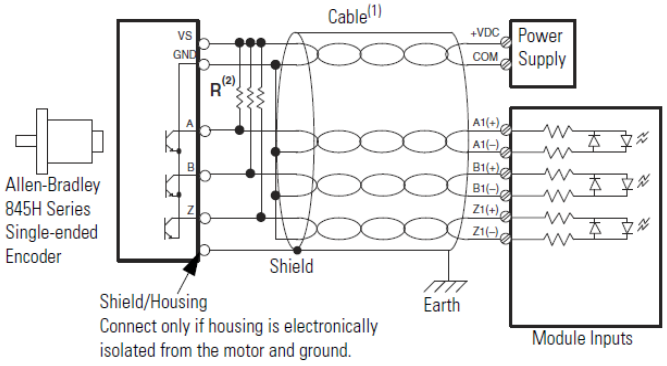
การอ่านค่าองศาจาก Incremental Encoder ด้วยการ์ด High Speed Counter (1756-HSC)



สายสัญญาณของ Incremental Encoder แบบ Differential จะต่อเข้ากับอินพุต A(+),A(-),B(+),B(-) และ Z(+),Z(-) ของการ์ด High speed counter แต่ถ้าเป็น encoder แบบ Single End สัญญาณ A(-),B(-) และ Z(-) จะรวมกันรวมเข้ากับ Com (-) แทน ดังรูปด้านล่าง



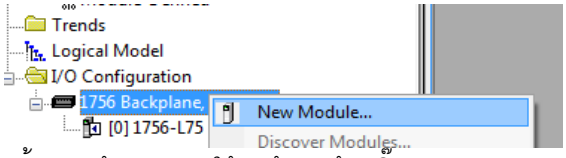
(ต่อแบบ Differential รูปจาก Rockwell Automation)



(ต่อแบบ Single End รูปจาก Rockwell Automation)

Add High Speed Counter Module

1) คลิ๊กขวาที่ I/O Configuration ->New Module -> Add การ์ด High Speed Counter 1756-HSC



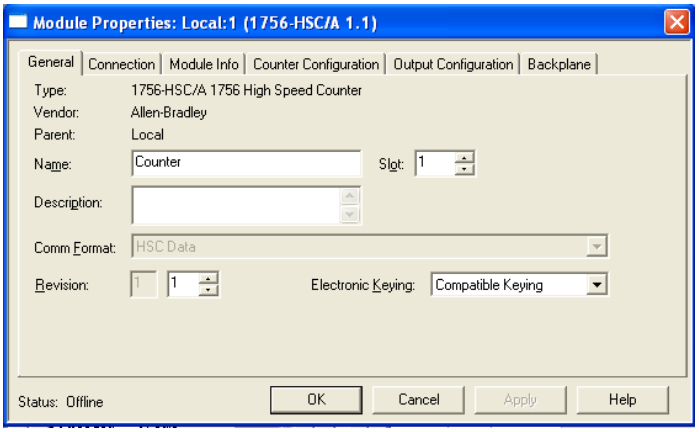
2) คลิ๊กแท็บ General จากนั้นกรอกข้อมูลต่างๆให้ถูกต้อง แล้วคลิ๊ก OK

Name – ชื่อการ์ด

Slot – หมายเลข Slot ที่การ์ดติดตั้งอยู่

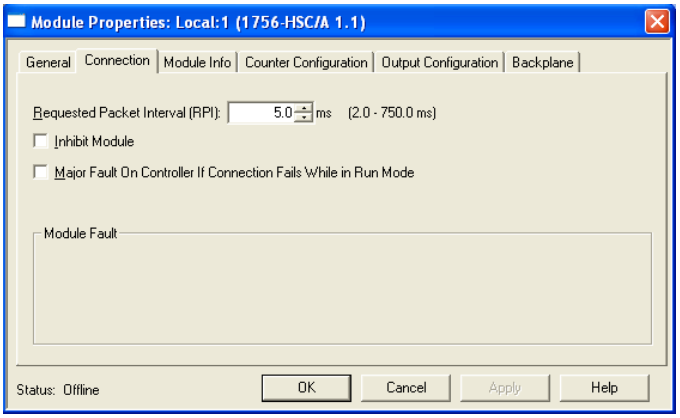
Description – ป้อนคำอธิบายของการ์ด

Comm Format – เลือกชนิดการเชื่อมต่อกับการ์ดให้เลือกเป็น HSC Data



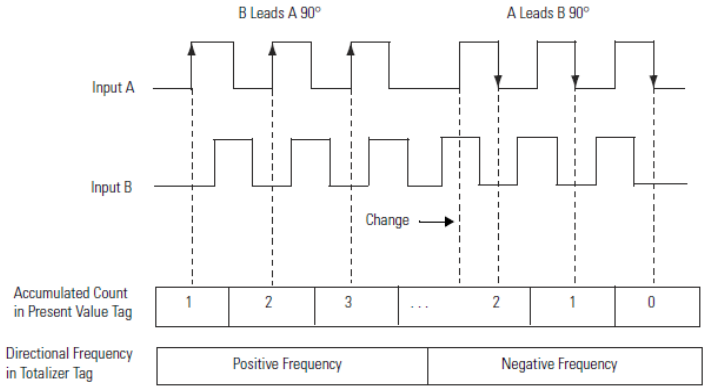
3) กำหนดค่า RPI (Requested Packet Interval) ในแท็บ Connection แล้วคลิ๊ก OK

Requested package interval (RPI) – เวลาส่งข้อมูล (ค่าพัลส์ที่นับได้) จากหน่วยความจำของการ์ด ออกไปยัง Controller



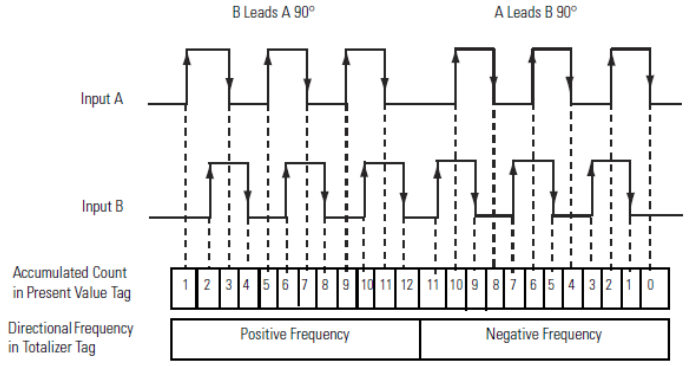
4) คลิ๊กแท็บ Counter Configuration จากนั้นกรอกข้อมูลต่างๆให้ถูกต้อง แล้วคลิ๊ก OK

Operational Mode – กำหนดวิธีการนับพัลส์ที่ส่งมาจาก Encoder เช่นโหมด X1 จะนับขึ้นเมื่อสัญญาณจาก B นำหน้าสัญญาณจาก A อยู่ 90 องศา และนับลงเมื่อสัญญาณจาก A นำหน้าสัญญาณจาก B อยู่ 90 องศา



(รูปจาก Rockwell Automation)

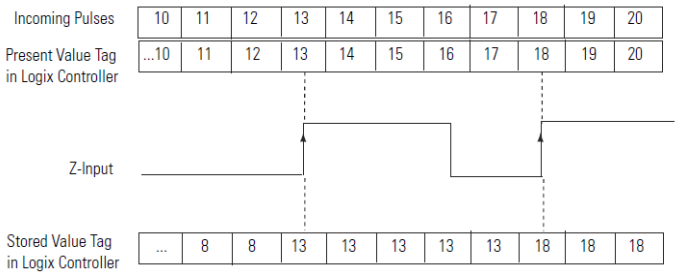
ส่วนโหมด X4 การทำงานเหมือนกับโหมด X1 ยกเว้นนับพัลส์ที่ขอบขาขึ้นและขาลงของสัญญาณ A และ B ทำให้ได้ความละเอียดมากกว่าโหมด X1 ถึง 4 เท่า



(รูปจาก Rockwell Automation)

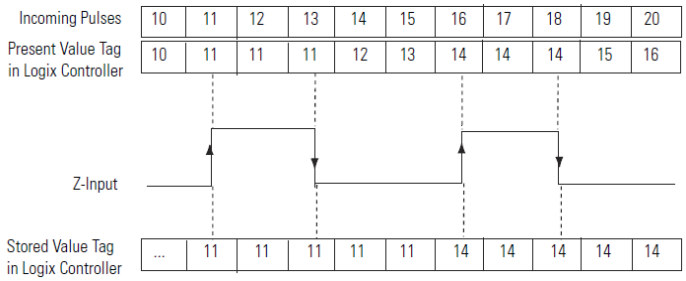
Storage Mode – เลือกวิธีในการเก็บค่าพัลส์ที่นับได้ลงใน Present Value Tag ของ Controller โดยอาศัยสัญญาณ Z (Gate/Reset) เป็นตัว trigger แบ่งย่อยได้ 4 แบบ คือ

- Store and Continue Mode – ค่า Present Value Tag เท่ากับ Incoming Pulses วิธีนี้ใช้กับ Incremental Encoder



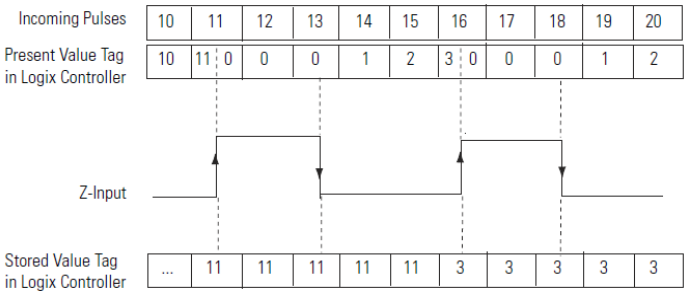
(รูปจาก Rockwell Automation)

- Store, Wait, and Resume – ค่า Present Value Tag จะคงค่าเท่ากับค่า Stored Tag ในช่วงที่มีอินพุตของสัญญาณ Z (Wait) หลังจากนั้น ค่า Present Value Tag กลับมาอัพเดจตามเดิม (Resume) วิธีนี้ใช้เมื่อต้องการให้นับในช่วงที่กำหนดเท่านั้น



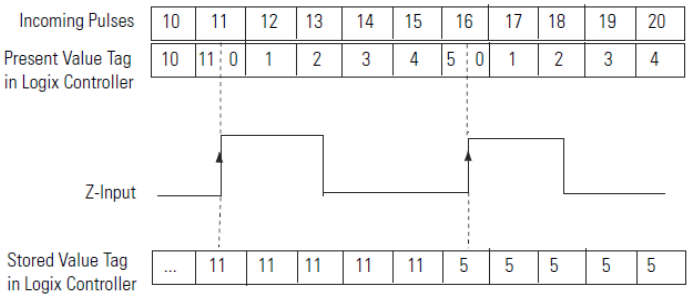
(รูปจาก Rockwell Automation)

- Store and Reset, Wait, and Start– ค่า Present Value Tag จะถูกรีเซ็ตเป็นศูนย์เมื่อมีอินพุตของสัญญาณ Z (Reset) รอจนสิ้นสุดสัญญาณ Z (Wait) จึงกลับมาเริ่มนับใหม่ (Start) วิธีนี้ใช้เมื่อต้องการให้นับเป็นรอบๆ ตามที่กำหนด

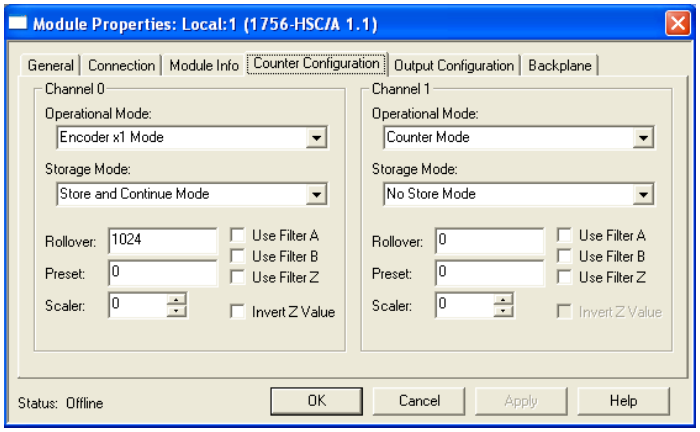


(รูปจาก Rockwell Automation)

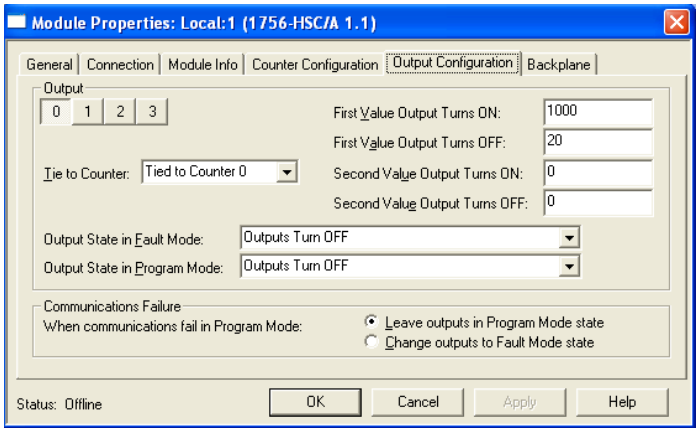
- Store and Reset, and Start– ค่า Present Value Tag จะถูกรีเซ็ตเป็นศูนย์เมื่อมีอินพุตของสัญญาณ Z (Reset) แล้วกลับมาเริ่มนับใหม่ (Start) ทันที



Rollover – กำหนดจำนวนพัลส์สูงสุดเลือกวิธีในการเก็บค่าพัลส์ ถ้าค่าที่นับได้เกินค่านี้ Counter จะถูกรีเซ็ตเป็นศูนย์แล้วเริ่มนับใหม่อีกครั้ง



5) คลิ๊กแท็บ Output Configuration จากนั้นเลือก Output ของการ์ดที่ต้องการส่งสัญญาณไปให้อุปกรณ์ภายนอกเมื่อค่าของCounter ตรงตามที่กำหนด (จะใช้งานหรือไม่ก็ได้) เสร็จแล้วคลิ๊ก OK



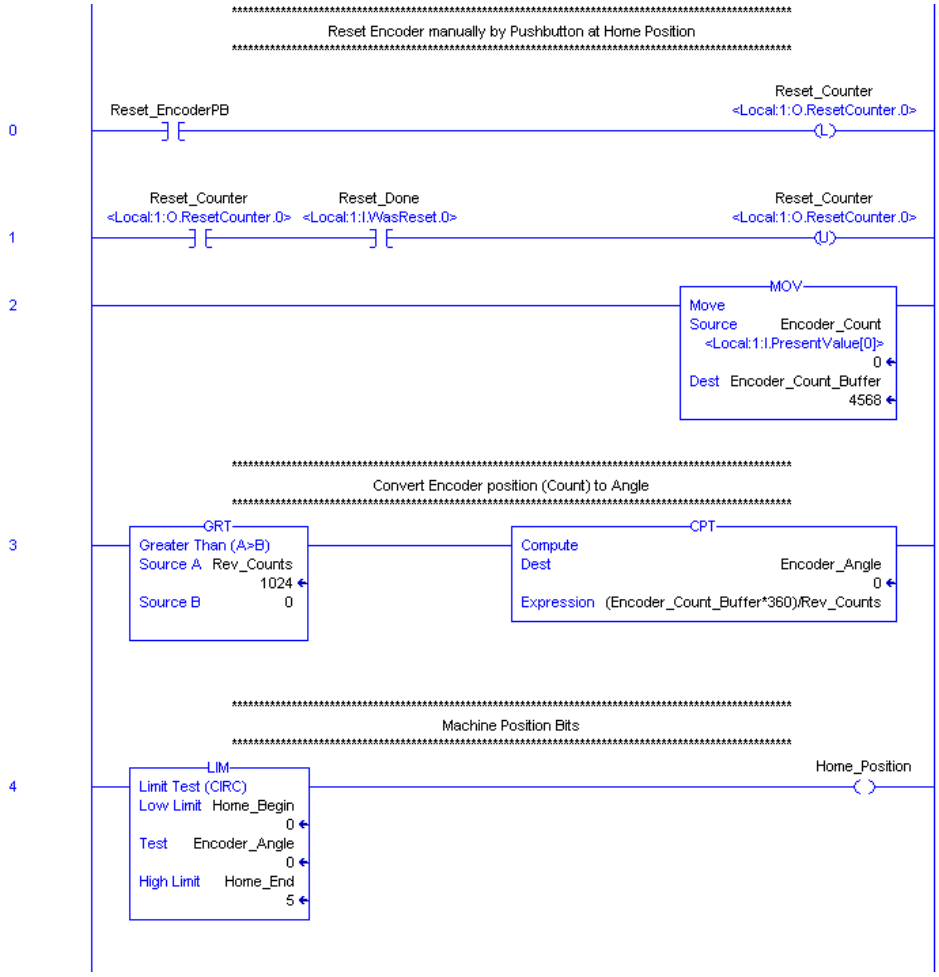
6) หลังจากคอนฟิก High Speed Counter การ์ดเสร็จแล้ว เราสามารถอ่านพารามิเตอร์และค่าที่นับได้ผ่าน Output Tag ของการ์ดได้โดยตรง ค่าพัลส์จาก Encoder จะเก็บอยู่ใน Tag Present Value

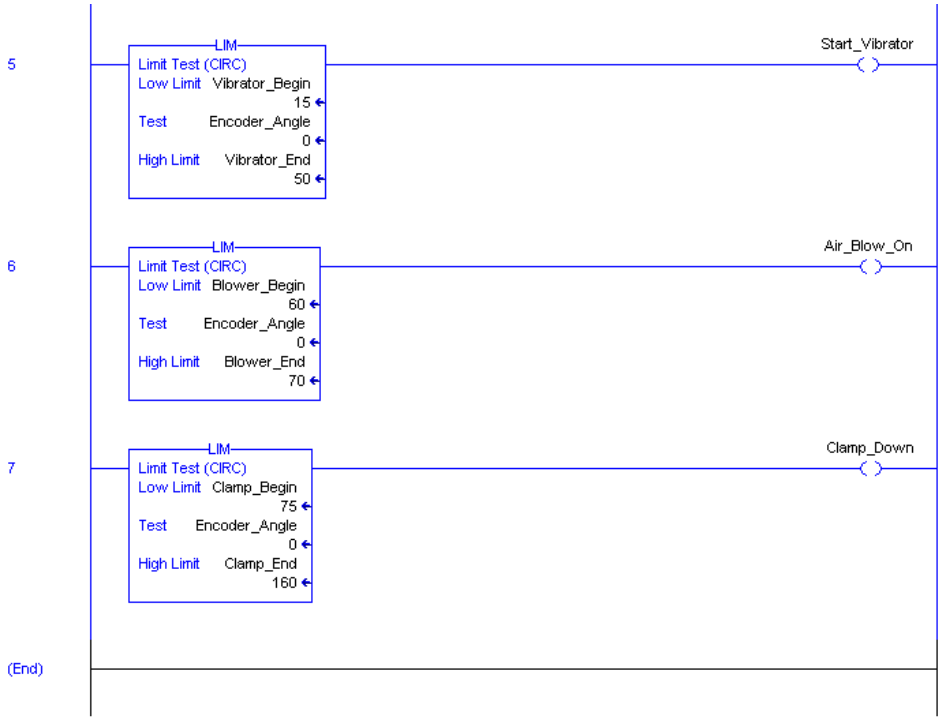
1756-HSC Module Output Tags

Name	Type	Style	Definition	Change During Operation
O.ResetCounter.*x*	BOOL	Decimal	Resets counter and begins counting. The reset occurs only on a zero to one transition. 0 = Do not reset. 1 = Reset.	Yes
O.OutputControl[*y*]	SINT	Decimal	Overrides current state of output. 0 = Normal operation. 1 = Override value to Off. 2 = Override value to On.	Yes
O.Output.[y].OnValue ⊕	DINT	Decimal	Designates the value at which an output turns On. Values are 0…16,777,214.	Yes
O.Output.[y].OffValue ⊕	DINT	Decimal	Designates the value at which an output turns Off. Values are 0…16,777,214.	Yes
O.Output.[y].ToThisCounter ⊕	SINT	Decimal	Designates counter to which this output is tied. 0 = Not tied to counter. 1 = Tied to counter (0). 2 = Tied to counter (1).	Yes
I.PresentValue[*x*]	DINT	Decimal	Displays the current count in the Counter and Encoder modes. Displays counts per sample in Frequency, Period Rate, or Continuous Rate modes. Values range from 0…16,777,214.	
I.WasReset.*x*	BOOL	Decimal	Displays whether the counter was reset. 0 = Counter was not reset. 1 = Counter was reset.	

(รูปจาก Rockwell Automation)

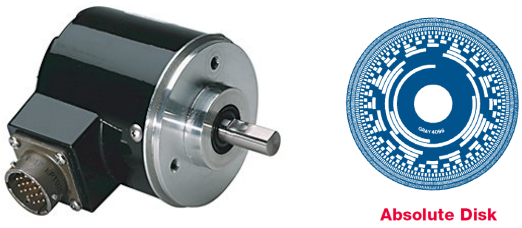
7) เขียนโปรแกรมดังรูปด้านล่าง





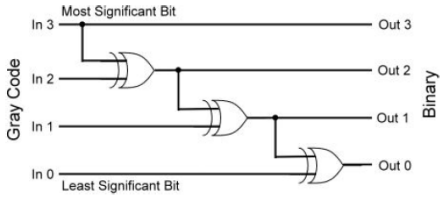
Absolute Encoder

บอกองศาการหมุน (ตำแหน่งเชิงมุม) ในรูปของรหัสแบบต่างๆ เช่น Binary,BCD และ Gray code แต่ละองศาให้ค่าสมบูรณ์ไม่ซ้ำกัน ข้อดีของ Absolute Encoder คือ ไม่ต้องใช้ Counter นับพัลส์เพื่อคำนวณหาค่าองศาและในกรณีแหล่งจ่ายไฟเกิดขัดข้อง Absolute Encoder ยังคงให้ค่าตำแหน่งเดิมได้หลังจากแหล่งจ่ายไฟกลับมาเป็นปกติ แตกต่างกับ Incremental Encoder ที่ต้องเคลื่อนไปยังตำแหน่ง Home ก่อนเสมอ ความละเอียด(Resolution) ของ Encoder ขึ้นอยู่กับจำนวนบิตของ Output เช่น 8 bits, 10 bits หรือ 24 bits เป็นต้น

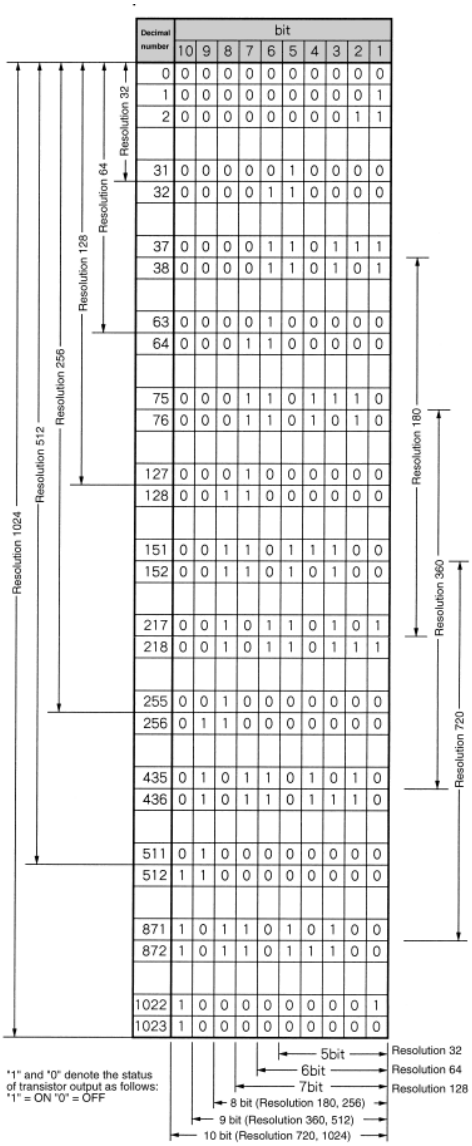


(รูปจาก Rockwell Automation)

Absolute Encoder ส่วนมากนิยมใช้แบบ Gray code โดยสามารถต่อ Encoder เข้ากับ Digital Input Card ของ ControlLogix PLC ได้เลย แล้วเขียนโปรแกรมเพื่อแปลงค่าองศาจาก Gray code เป็น Binary code โดยอาศัยหลักการของ XOR ดังรูป



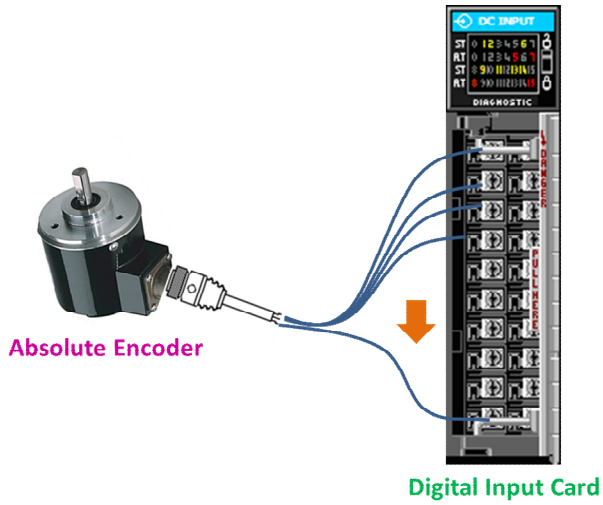
ถ้า Resolution ของ Encoder เท่ากับ 2 ยกกำลังจำนวนบิตเอาท์พุต ค่า 0 องศาจะเริ่มต้นที่ค่า 0 ในหน่วย Binary เสมอ แต่ถ้า Resolution ของ Encoder ไม่เท่ากับ 2 ยกกำลังจำนวนบิตเอาท์พุต ค่า 0 องศาจะเริ่มต้นที่ค่าอื่นดังตารางด้านล่าง



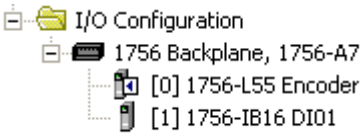
(รูปจาก Automationdirect)

ตัวอย่าง

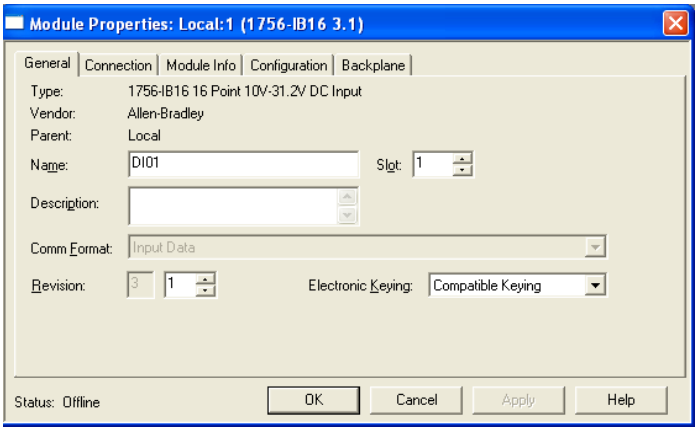
การอ่านค่าองศาจาก Absolute Encoder ไปใช้ควบคุมการทำงานของเครื่องจักร

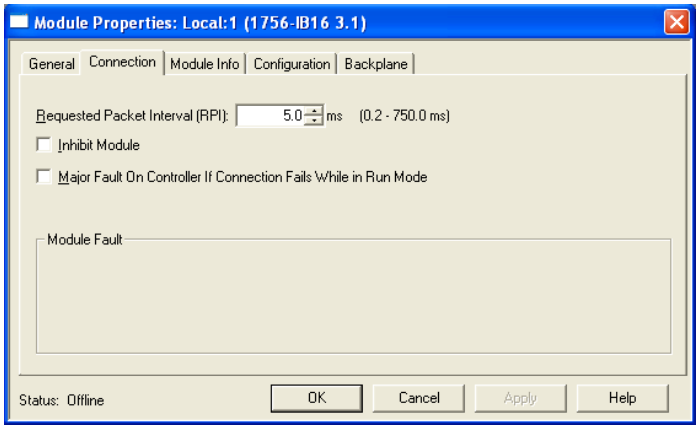


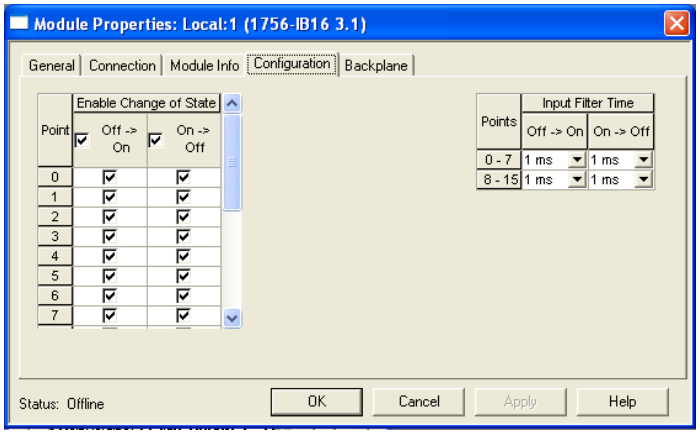
1) Add I/O การ์ดใน I/O Configuration ของโปรเจคไฟล์



2) คอนฟิกการ์ด Digital Input ให้ถูกต้อง ดังรูปด้านล่าง





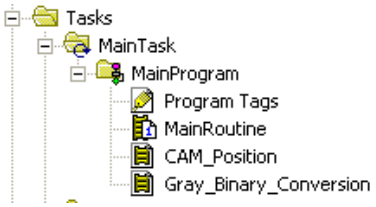


3) สร้าง Tag สำหรับใช้งานตามที่กำหนด

Scope: Encoder Show... Show All

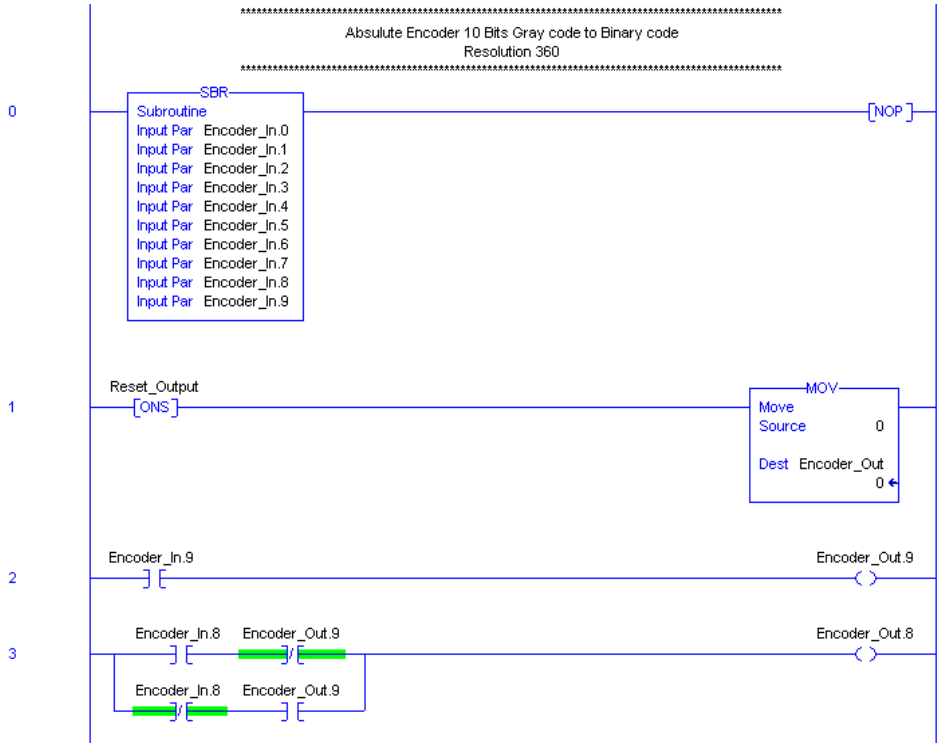
Name	Alias For	Base Tag	Data Type	Style
Encoder_In			DINT	Decimal
Encoder_Out			DINT	Decimal
Local:1:C			AB:1756_DI:C:0	
Local:1:I			AB:1756_DI:I:0	
Reset_Output			BOOL	Decimal
Encoder_Bit0	Local:1:I.Data.0	Local:1:I.Data.0	BOOL	Decimal
Encoder_Bit1	Local:1:I.Data.1	Local:1:I.Data.1	BOOL	Decimal
Encoder_Bit2	Local:1:I.Data.2	Local:1:I.Data.2	BOOL	Decimal
CAM_Position			DINT	Decimal
Encoder_Bit3	Local:1:I.Data.3	Local:1:I.Data.3	BOOL	Decimal
Encoder_Bit4	Local:1:I.Data.4	Local:1:I.Data.4	BOOL	Decimal
Encoder_Bit5	Local:1:I.Data.5	Local:1:I.Data.5	BOOL	Decimal
Encoder_Bit6	Local:1:I.Data.6	Local:1:I.Data.6	BOOL	Decimal
Encoder_Bit7	Local:1:I.Data.7	Local:1:I.Data.7	BOOL	Decimal
Encoder_Bit8	Local:1:I.Data.8	Local:1:I.Data.8	BOOL	Decimal
Encoder_Bit9	Local:1:I.Data.9	Local:1:I.Data.9	BOOL	Decimal
Machine_Position			DINT	Decimal

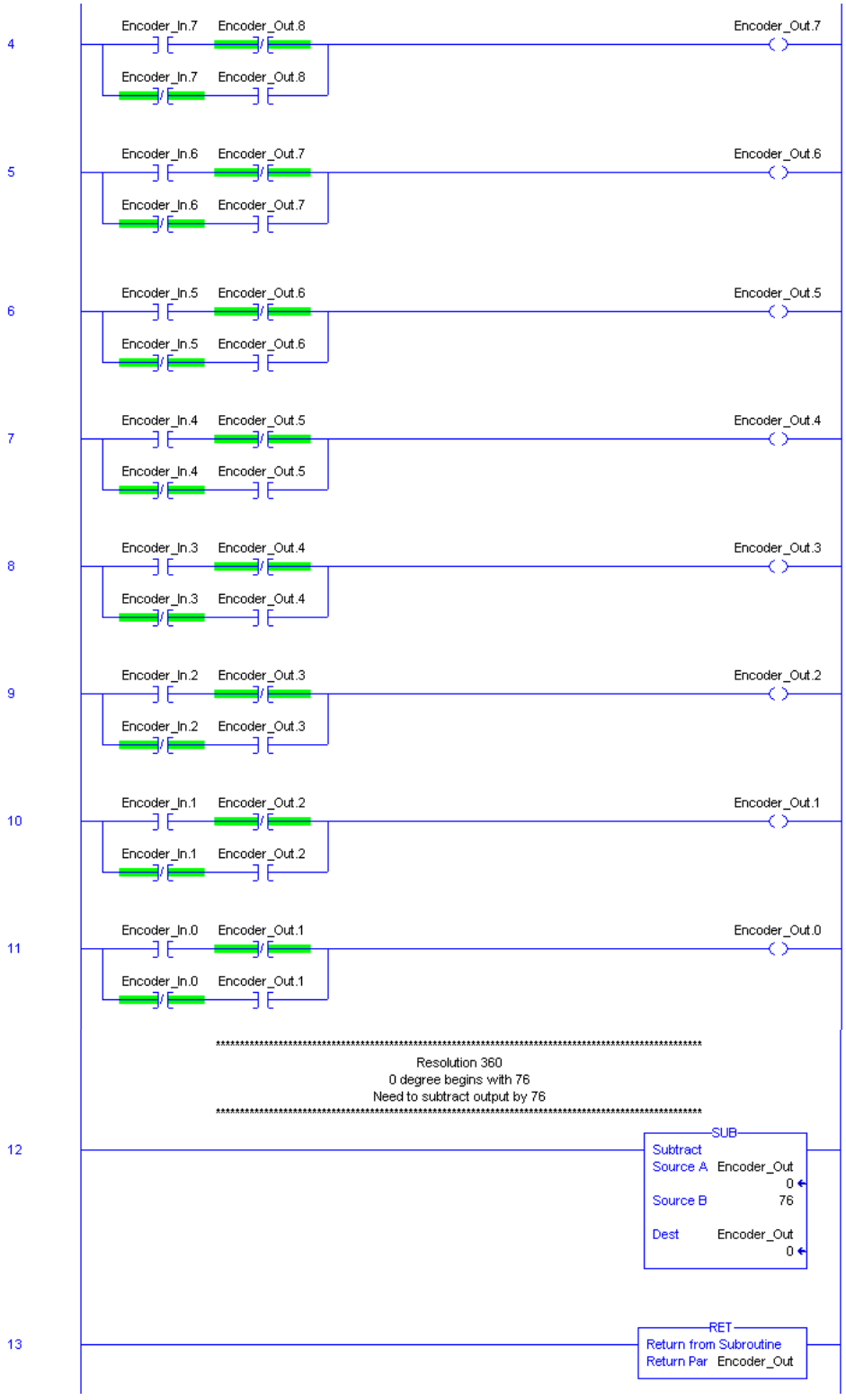
4) สร้าง New Routine ชื่อ CAM_Position และ Gray_Binary_Conversion โดยรูทีน Gray_Binary_Conversion ใช้สำหรับแปลงค่าองศาจาก Gray code เป็น Binary code ส่วน CAM_Position ทำหน้าที่กำหนด Position Bit ของ Master Shaft ที่องศาต่างๆกันแล้วนำไปเขียนโปรแกรมสั่งงานอุปกรณ์

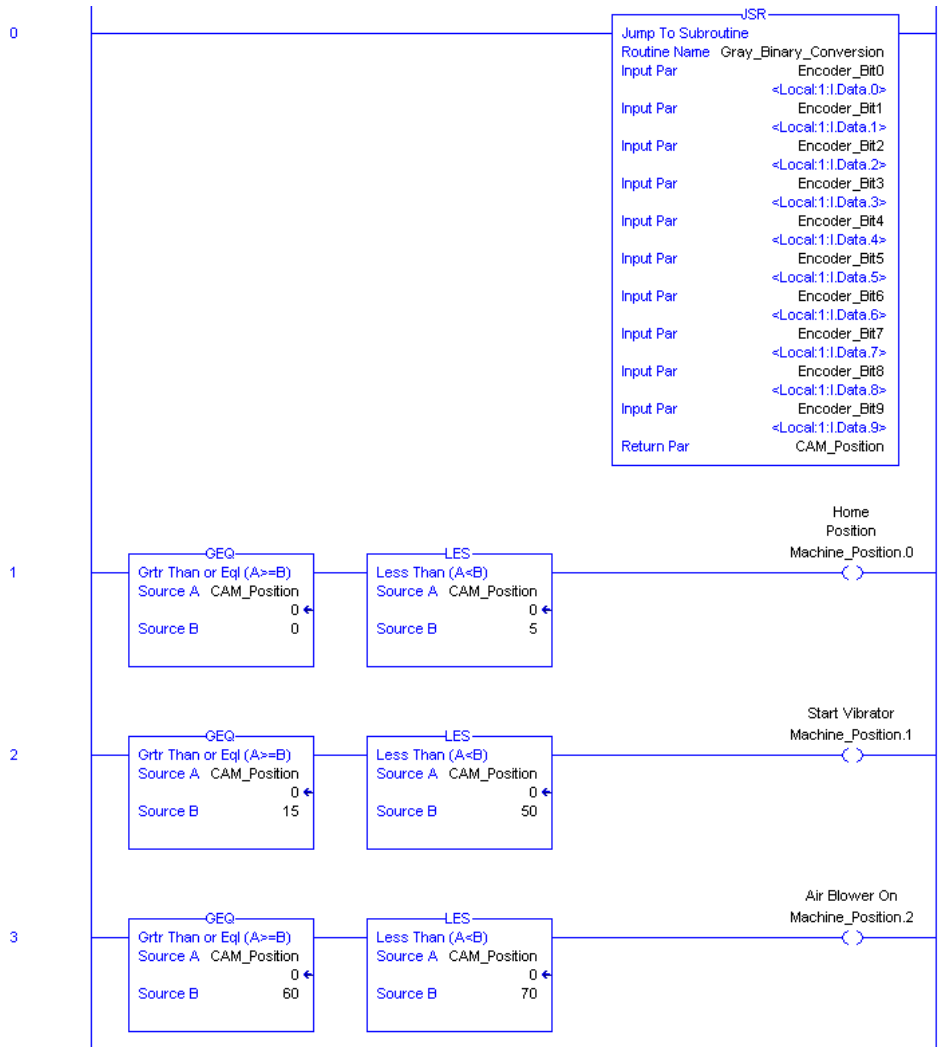


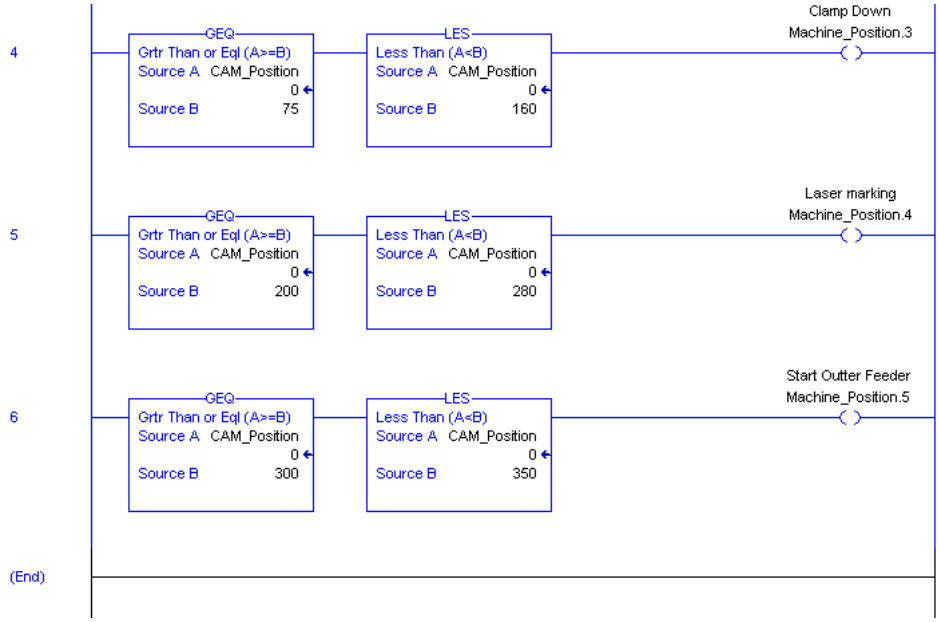
5) เริ่มเขียนโปรแกรมในรูทีน Gray_Binary_Conversion และ CAM_Position ดังรูปด้านล่าง

Gray_Binary_Conversion Routine




4
Encoder_In.7
Encoder_Out.8
Encoder_Out.7
Encoder_In.7
Encoder_Out.8
5
Encoder_In.6
Encoder_Out.7
Encoder_Out.6
Encoder_In.6
Encoder_Out.7
6
Encoder_In.5
Encoder_Out.6
Encoder_Out.5
Encoder_In.5
Encoder_Out.6
7
Encoder_In.4
Encoder_Out.5
Encoder_Out.4
Encoder_In.4
Encoder_Out.5
8
Encoder_In.3
Encoder_Out.4
Encoder_Out.3
Encoder_In.3
Encoder_Out.4
9
Encoder_In.2
Encoder_Out.3
Encoder_Out.2
Encoder_In.2
Encoder_Out.3
10
Encoder_In.1
Encoder_Out.2
Encoder_Out.1
Encoder_In.1
Encoder_Out.2
11
Encoder_In.0
Encoder_Out.1
Encoder_Out.0
Encoder_In.0
Encoder_Out.1
Resolution 360
0 degree begins with 76
Need to subtract output by 76
12
SUB
Subtract
Source A Encoder_Out
0
Source B 76
Dest Encoder_Out
0
13
RET
Return from Subroutine
Return Par Encoder_Out


0
JSR
Jump To Subroutine
Routine Name Gray_Binary_Conversion
Input Par Encoder_Bit0
<Local:1:I.Data.0>
Input Par Encoder_Bit1
<Local:1:I.Data.1>
Input Par Encoder_Bit2
<Local:1:I.Data.2>
Input Par Encoder_Bit3
<Local:1:I.Data.3>
Input Par Encoder_Bit4
<Local:1:I.Data.4>
Input Par Encoder_Bit5
<Local:1:I.Data.5>
Input Par Encoder_Bit6
<Local:1:I.Data.6>
Input Par Encoder_Bit7
<Local:1:I.Data.7>
Input Par Encoder_Bit8
<Local:1:I.Data.8>
Input Par Encoder_Bit9
<Local:1:I.Data.9>
Return Par CAM_Position
1
GEQ
Grtr Than or Eql (A>=B)
Source A CAM_Position
0
Source B 0
LES
Less Than (A<B)
Source A CAM_Position
0
Source B 5
Home
Position
Machine_Position.0
2
GEQ
Grtr Than or Eql (A>=B)
Source A CAM_Position
0
Source B 15
LES
Less Than (A<B)
Source A CAM_Position
0
Source B 50
Start Vibrator
Machine_Position.1
3
GEQ
Grtr Than or Eql (A>=B)
Source A CAM_Position
0
Source B 60
LES
Less Than (A<B)
Source A CAM_Position
0
Source B 70
Air Blower On
Machine_Position.2


4
GEQ
Grtr Than or Eql (A>=B)
Source A CAM_Position
0
Source B 75
LES
Less Than (A<B)
Source A CAM_Position
0
Source B 160
Clamp Down
Machine_Position.3
5
GEQ
Grtr Than or Eql (A>=B)
Source A CAM_Position
0
Source B 200
LES
Less Than (A<B)
Source A CAM_Position
0
Source B 280
Laser marking
Machine_Position.4
6
GEQ
Grtr Than or Eql (A>=B)
Source A CAM_Position
0
Source B 300
LES
Less Than (A<B)
Source A CAM_Position
0
Source B 350
Start Outter Feeder
Machine_Position.5
(End)