

HOMEWORK3

- LAB1: DISTANCE D
- LAB2: SUPER COUNTER
- LAB3: TRAFFIC LIGHT

จัดทำโดย

นายธนกฤต	จาตทอง	63070501208
นายลัทธิพล	ปิยะสุข	63070501216
นางสาวฉันท์สินี	เมืองหนู	63070501221

INC251

ภาคการศึกษาที่ 2/2564

อ.สมชัย ตีรรัตนจารุ

3.1 DISTANCE D

Name: Thanakrit Jadthong
Name: Latthaphon Piyasuk
Name: Chansinee Mueangnu

Student ID: 63070501208
Student ID: 63070501216
Student ID: 63070501221

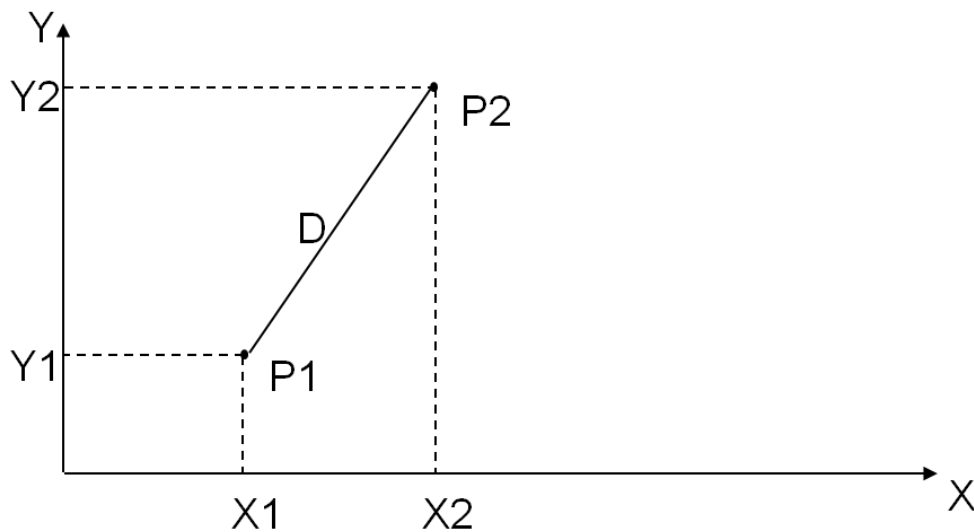
Group: INCA01
Group: INCA01
Group: INCA01

Homework 3

Write the program for howork3. then, achieve the project name as: HW3_group Axx and report. Please send to email: inc251.2a@gmail.com

3.1 Distance D

Write the IEC standard program according to the job descriptions as state below.



Calculating the distance D between two points and create HMI (WINCC) that can change parameters (X1,X2,Y1 and Y2) and show the Distance D

Remark : Using **m** for memory register and create symbol for all parameters.

Name: Thanakrit Jadthong
 Name: Latthaphon Piyasuk
 Name: Chansinee Mueangnu

Student ID: 63070501208
 Student ID: 63070501216
 Student ID: 63070501221

Group: INCA01
 Group: INCA01
 Group: INCA01

➤ Symbol:

Symbol /	Address	Data type
3.1.distanceD	FC 1	FC 1
3.2.super counter	FB 2	FB 2
3.3.traffic light	FC 2	FC 2
auto/manual button	I 124.0	BOOL
counter1	FB 3	FB 3
counter2	FB 4	FB 4
cvcount	MD 70	DINT
Distance	MD 60	REAL
GREEN1	Q 124.2	BOOL
GREEN2	Q 124.5	BOOL
i125.0	I 124.7	BOOL
mode	Q 125.0	BOOL
Power Start	I 124.2	BOOL
RED1	Q 124.0	BOOL
RED2	Q 124.3	BOOL
reset	I 125.1	BOOL
sigma_XY	MD 56	REAL
step button	I 124.1	BOOL
supercount	FB 1	FB 1
X	M 20.0	BOOL
X_dif	MD 28	REAL
X_sqr	MD 32	REAL
X1	MD 20	REAL
X2	MD 24	REAL
Xd	MW 22	INT
Y_dif	MD 48	REAL
Y_sqr	MD 52	REAL
Y1	MD 40	REAL
Y2	MD 44	REAL
YELLOW1	Q 124.1	BOOL
YELLOW2	Q 124.4	BOOL

ตารางที่ 1: สัญลักษณ์ที่ใช้ในการเขียนโปรแกรม S7-300 ทั้งหมด
 (สำหรับโปรแกรม Distance D เฉพาะที่ไฮไลต์)

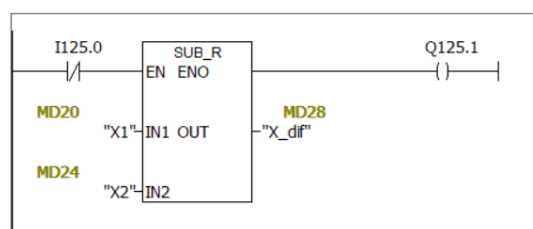
➤ Program:

โปรแกรม Distance D เป็นโปรแกรมที่สร้างขึ้นมาเพื่อคำนวณระยะทางระหว่างจุด 2 จุด (D) ซึ่งการที่เราจะสามารถหาระยะทางระหว่างจุด 2 จุดได้นั้น เราจะต้องรู้พิกัดของจุดทั้งสอง โดยกำหนดให้จุดแรกเป็นจุด P1(X1, Y1) และจุดที่สองเป็นจุด P2(X2, Y2) หลังจากนั้นก็คำนวณหาระยะทางซึ่งเป็นเส้นตรงระหว่างจุด P1 และ P2 โดยใช้สมการทางคณิตศาสตร์ ดังนี้

$$D = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

ในที่นี้เราได้ออกแบบโปรแกรมโดยแบ่งขั้นตอนการคำนวณเป็น 6 step ดังนี้

1. หาระยะทางตามแนวนอน (X_dif = X₁-X₂) โดยใช้คำสั่ง SUB_R ดังรูปที่ 1



รูปที่ 1: Network1 -> (X₁-X₂)

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

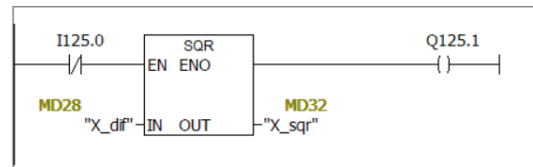
Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

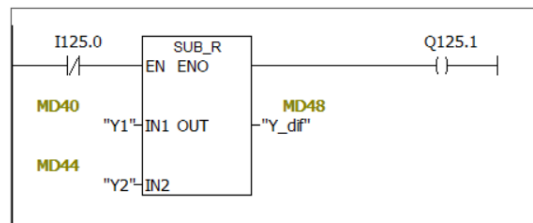
Group: INCA01

2. หาค่าของ $X_{sqr} = (X_1 - X_2)^2$ โดยใช้ค่าของ X_{dif} จากขั้นตอนที่ 1 มาคำนวณโดยใช้คำสั่ง SQR ดังรูปที่ 2



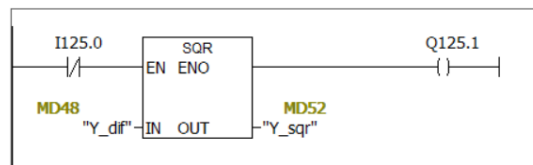
รูปที่ 2: Network2 -> $(X_1 - X_2)^2$

3. หาระยะทางตามแนวตั้ง ($Y_{dif} = Y_1 - Y_2$) โดยใช้คำสั่ง SUB_R ดังรูปที่ 3



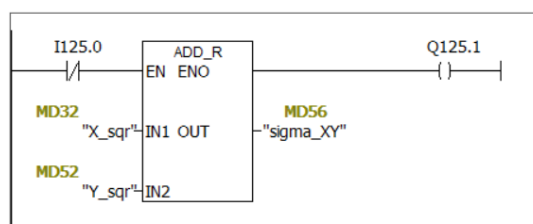
รูปที่ 3: Network3 -> $(Y_1 - Y_2)$

4. หาค่าของ $Y_{sqr} = (Y_1 - Y_2)^2$ โดยใช้ค่าของ Y_{dif} จากขั้นตอนที่ 3 มาคำนวณ โดยใช้คำสั่ง SQR ดังรูปที่ 4



รูปที่ 4: Network4

5. หาค่าของ $\sigma_{XY} = (X_1 - X_2)^2 + (Y_1 - Y_2)^2$ โดยใช้ค่า X_{sqr} จากขั้นตอนที่ 2 และค่า Y_{sqr} จากขั้นตอนที่ 4 มาคำนวณ โดยใช้คำสั่ง ADD_R ก็จะได้กำลังสองของความยาวเส้น-เฉียงที่เชื่อมระหว่างสองจุด ซึ่งเขียน LADDER ได้ดังรูปที่ 5



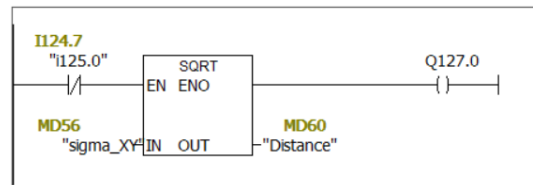
รูปที่ 5: Network5

Name: Thanakrit Jadthong
Name: Latthaphon Piyasuk
Name: Chansinee Mueangnu

Student ID: 63070501208
Student ID: 63070501216
Student ID: 63070501221

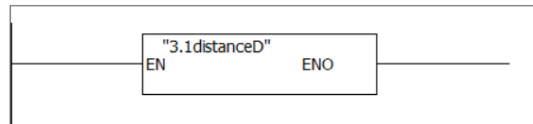
Group: INCA01
Group: INCA01
Group: INCA01

6. หารากที่สองของผลบวกที่ได้จากขั้นตอนที่ 5 ซึ่งผลลัพธ์ที่ได้ คือ ระยะทางระหว่างจุดสองจุด (ความยาวของเส้นตรงที่เชื่อมระหว่างสองจุด) โดยขั้นตอนนี้เราจะใช้คำสั่ง SQRT ดังรูปที่ 6



รูปที่ 6: Network6

โดยการทำงานทั้งหมดของโปรแกรมคำนวณหาระยะทางระหว่างจุดสองจุดนี้ เราได้เขียนโปรแกรมไว้ใน FC1 และเรียกใช้งานผ่าน OB1 ดังรูปที่ 7



รูปที่ 7: เรียกใช้ FC1 บน OB1

Name: Thanakrit Jadthong
 Name: Latthaphon Piyasuk
 Name: Chansinee Mueangnu

Student ID: 63070501208
 Student ID: 63070501216
 Student ID: 63070501221

Group: INCA01
 Group: INCA01
 Group: INCA01

➤ HMI (WinCC):

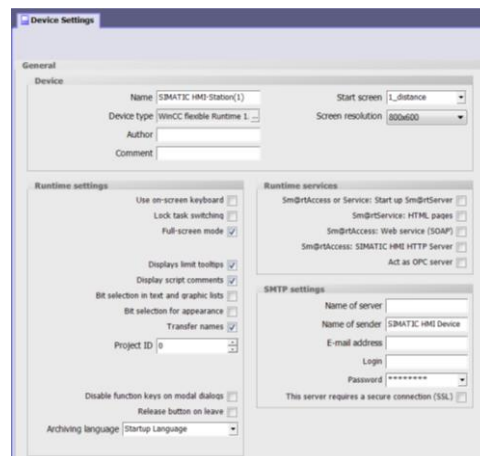
1. Tags:

Name	Display name	Connection	Data type	Symbol	Address	Array elements	Acquisition cycle
AUTO/MANUAL		Connection_1	Bool	<Undefined>	I 124.8	1	1 s
distance		Connection_1	Real	<Undefined>	MD 60	1	1 s
GREEN1		Connection_1	Bool	<Undefined>	Q 124.2	1	1 s
GREEN2		Connection_1	Bool	<Undefined>	Q 124.5	1	1 s
RED1		Connection_1	Bool	<Undefined>	Q 124.6	1	1 s
RED2		Connection_1	Bool	<Undefined>	Q 124.3	1	1 s
STEP BUTTON		Connection_1	Bool	<Undefined>	I 124.1	1	1 s
X1		Connection_1	Real	<Undefined>	MD 20	1	1 s
X2		Connection_1	Real	<Undefined>	MD 24	1	1 s
Y1		Connection_1	Real	<Undefined>	MD 40	1	1 s
Y2		Connection_1	Real	<Undefined>	MD 44	1	1 s
YELLOW1		Connection_1	Bool	<Undefined>	Q 124.1	1	1 s
YELLOW2		Connection_1	Bool	<Undefined>	Q 124.4	1	1 s

ตารางที่ 2: Tags ทั้งหมดที่ใช้ในการเขียน HMI (สำหรับโปรแกรม Distance D เฉพาะที่ไฮไลต์)

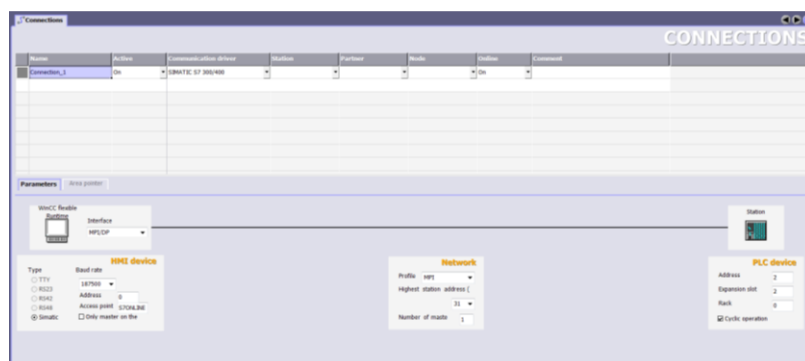
2. Setting:

- ตั้งค่า resolution ของหน้าจอ



รูปที่ 8: Device Settings

- เลือก protocol ในการสื่อสาร



รูปที่ 9: Communication → Connections

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

Group: INCA01

Name: Chansinee Mueangnu

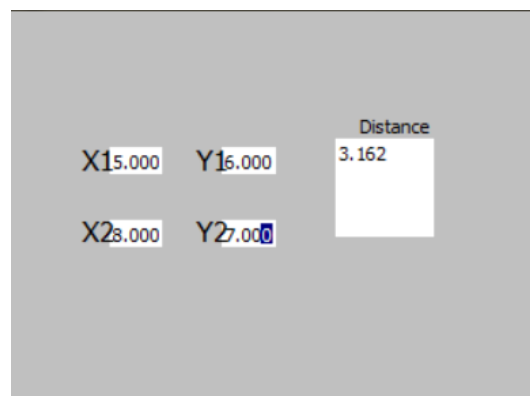
Student ID: 63070501221

Group: INCA01

3. HMI screens:

โปรแกรม Distance D ที่เราออกแบบ สามารถคำนวณระยะทางระหว่างจุด 2 จุด และแสดงค่าออกมาที่ช่อง Distance ดังรูปที่ 10 ได้ โดยการใช้กรอกพิกัดของจุดทั้งสอง นั่นคือ จุด $P1(X_1, Y_1)$ และจุด $P2(X_2, Y_2)$

จากตัวอย่าง HMI screens รูปที่ 10 ผู้ใช้ต้องการหาระยะทางระหว่างจุด $P1(5,6)$ และ $P2(8,7)$ ซึ่งโปรแกรมก็สามารถคำนวณออกมาได้ทันที ได้ระยะทางระหว่างจุดสองจุดเป็น 3.162 ซึ่งผลลัพธ์ที่ออกมา นั้นถูกต้องตามสูตร $D = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$



รูปที่ 10: ตัวอย่าง HMI screens

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

Group: INCA01

- **Questions:** What is the most suitable IEC object for solve this problem between FC and FB? Why?

ทางกลุ่มของเราคิดว่าสำหรับโจทย์ข้อนี้การใช้ FC จะเหมาะสมที่สุดเนื่องจากเราแสดงค่าระยะทางที่ผ่านการคำนวณทันที ไม่ได้มีการเก็บค่าไว้เพื่อนำไปใช้ต่อ

- **Summary:**

จากโปรแกรม Distance D เป็นโปรแกรมที่สร้างขึ้นมาเพื่อคำนวณระยะห่างระหว่างจุด 2 จุด (D) โดยในการคำนวณเราจำเป็นต้องรู้พิกัดของจุดทั้งสองผ่านตัวแปร x1, x2 และ y1, y2 โดยใช้สมการทางคณิตศาสตร์ $D = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$ โดยเราได้ทำการสร้างโปรแกรมตามขั้นตอนทางคณิตศาสตร์ซึ่งสามารถแบ่งออกได้เป็น 6 step โดยใช้การเขียนโปรแกรมผ่าน FC และเรียกใช้งานใน OB1 เนื่องจากเราต้องการใส่ค่าและแสดงผลการคำนวณออกมา ไม่ได้เก็บค่าไปใช้ต่อ จึงไม่มีความจำเป็นต้องใช้ FB จากนั้นเราจึงได้สร้าง HMI ผ่าน WINCC ซึ่งเมื่อทำการกรอกค่าลงไป โปรแกรมก็จะสามารถคำนวณผลลัพธ์ออกมาได้อย่างถูกต้อง

3.2 SUPER COUNTER

Name: Thanakrit Jadhong
Name: Latthaphon Piyasuk
Name: Chansinee Mueangnu

Student ID: 63070501208
Student ID: 63070501216
Student ID: 63070501221

Group: INCA01
Group: INCA01
Group: INCA01

3.2 Super Counter

Write the IEC standard program according to the job descriptions as state below.
Created control object to perform the counting task.

The input parameters should be **CU** for counting up, **CD** for counting down, **ST** for setting counter to specific value assign by input **CS** and the last input **CR** for resetting counter to zero. All input parameters will be activated at positive edge except CR is activated as long as the level is high.

And the output parameters should be **CO** for counter output, turn on for non-zero counter value in which is the second output parameter **CV**. Counting rang shall be -1,000,000 to +1,000,000 or better. Making call this new created object at least twice in main program.

Remark : Using **m** for memory register and create symbol for all parameters and you must create VAT table for easy testing.

➤ Symbol:

Symbol /	Address	Data type
3.1.distanceD	FC 1	FC 1
3.2.super counter	FB 2	FB 2
3.3.traffic light	FC 2	FC 2
auto/manual button	I 124.0	BOOL
counter1	FB 3	FB 3
counter2	FB 4	FB 4
cvcount	MD 70	DINT
Distance	MD 60	REAL
GREEN1	Q 124.2	BOOL
GREEN2	Q 124.5	BOOL
I125.0	I 124.7	BOOL
mode	Q 125.0	BOOL
Power Start	I 124.2	BOOL
RED1	Q 124.0	BOOL
RED2	Q 124.3	BOOL
reset	I 125.1	BOOL
sigma_XY	MD 56	REAL
step button	I 124.1	BOOL
supercount	FB 1	FB 1
X	M 20.0	BOOL
X_dif	MD 28	REAL
X_sqr	MD 32	REAL
X1	MD 20	REAL
X2	MD 24	REAL
Xd	MW 22	INT
Y_dif	MD 48	REAL
Y_sqr	MD 52	REAL
Y1	MD 40	REAL
Y2	MD 44	REAL
YELLOW1	Q 124.1	BOOL
YELLOW2	Q 124.4	BOOL

ตารางที่ 3: สัญลักษณ์ที่ใช้ในการเขียนโปรแกรม S7-300 ทั้งหมด
(สำหรับโปรแกรม Super Counter เฉพาะที่ไฮไลต์)

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

Group: INCA01

➤ Program:

จากโจทย์ต้องการให้สร้างโปรแกรม Super Counter (S_CUD) ที่สามารถตั้งค่า Value for Presetting Counter (CS) ในช่วง -1,000,000 ถึง 1,000,000 ได้ และแม้ว่าในโปรแกรม S7-300 จะมีคำสั่ง counter UD สำเร็จรูปให้ แต่เราจำเป็นต้องสร้าง counter UD ขึ้นมาเอง โดยการเขียนโปรแกรม LADDER ลงใน Function Block (FB) เนื่องจากคำสั่ง counter สำเร็จรูปนั้นสามารถกำหนดค่าตั้งต้นในการเริ่มนับที่ค่าต่ำสุด 0 ถึงค่าสูงสุดเพียง 999 ซึ่งไม่เพียงพอตามความต้องการ แต่หากเป็น Function Block ที่เราสร้างขึ้นมาเองจะสามารถนับค่าได้มากกว่านั้น ขึ้นอยู่กับ data type ของค่าต่าง ๆ ที่เราเขียนไว้ในโปรแกรม

เราได้เขียนโปรแกรม สร้าง counter UD ไว้ใน FB1 และมีการใช้ FB คู่กับ data block (DB) เสมอ เพื่อใช้ DB เป็นตัวเก็บข้อมูล พารามิเตอร์ต่าง ๆ ของ FB ซึ่งมีวิธีการเขียนโปรแกรม ดังต่อไปนี้

1. กำหนดตัวแปรที่ต้องการใช้เพื่อสร้าง Counter UD โดยมีรายละเอียดต่าง ๆ ดังนี้

	Name	Data Type	Address	Initial Value
CU		Bool	0.0	FALSE
CD		Bool	0.1	FALSE
ST		Bool	0.2	FALSE
CR		Bool	0.3	FALSE
CS		DInt	2.0	L#0

ตารางที่ 4: Interface INPUT

	Name	Data Type	Address	Initial Value
CO		Bool	6.0	FALSE
CV		DInt	8.0	L#0

ตารางที่ 5: Interface OUTPUT

	Name	Data Type	Address	Initial Value
CVmem		DInt	12.0	L#0
CUmем		Bool	16.0	FALSE
CDmem		Bool	16.1	FALSE
STmem		Bool	16.2	FALSE

ตารางที่ 6: Interface STAT

Name: Thanakrit Jadhong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

Group: INCA01

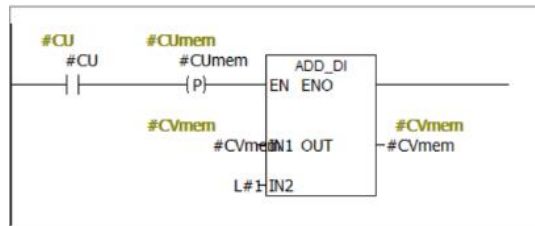
Name: Chansinee Mueangnu

Student ID: 63070501221

Group: INCA01

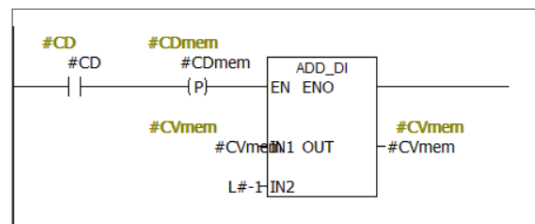
2. เขียนโปรแกรมบน FB1 เพื่อสร้างฟังก์ชันต่าง ๆ ในตัว Counter UD

- สร้างฟังก์ชัน Count up (CU) : มีการใช้คำสั่ง rising up เพื่อใช้นับขอบขาขึ้น และใช้ฟังก์ชัน ADD_DI เพื่อทำการบวกค่า CVmem (ค่าที่ counter นับได้) ไปเรื่อย ๆ โดย $CVmem = CVmem + 1$ ซึ่งเขียนโปรแกรมได้ดังรูปที่ 11



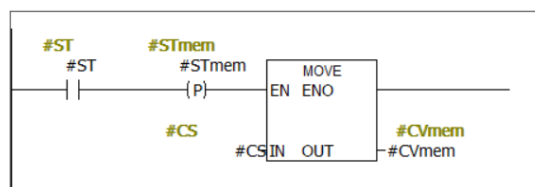
รูปที่ 11: Network1

- สร้างฟังก์ชัน Count down (CD) : มีการใช้คำสั่ง rising up เพื่อใช้นับขอบขาขึ้น และใช้ฟังก์ชัน ADD_DI เพื่อทำการลดค่า CVmem (ค่าที่ counter นับได้) ไปเรื่อย ๆ โดย $CVmem = CVmem - 1$ ซึ่งเขียนโปรแกรมได้ดังรูปที่ 12



รูปที่ 12: Network2

- สร้างฟังก์ชัน Set (ST) : โดยหน้าที่หลัก ๆ ของฟังก์ชัน Set คือ จะทำให้ค่า CV แสดงค่าตาม CS ที่เราได้ตั้งค่าไว้ ซึ่งเราจะสร้างโดยใช้คำสั่งขอบขาขึ้น และใช้คำสั่ง move เพื่อย้ายค่าจาก CS ไปยัง CVmem ดังรูปที่ 13



รูปที่ 13: Network3

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

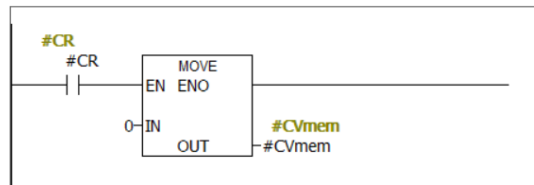
Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

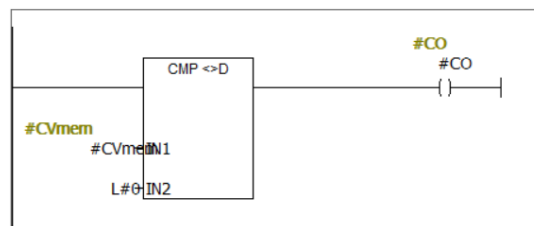
Group: INCA01

- สร้างฟังก์ชัน Reset (CR) : เราจำเป็นต้องสร้างฟังก์ชัน Reset อยู่ท้ายสุดของโปรแกรม เพื่อให้ฟังก์ชัน Reset มี priority ที่สูงกว่าฟังก์ชันอื่น ๆ นั่นคือ ถ้าหากกด Reset จะทำให้ค่า counter ที่นับได้เป็น 0 นั่นคือ ฟังก์ชันนี้ เป็นการย้ายค่า 0 ไปใส่ไว้ใน CVmem ดังรูปที่ 14



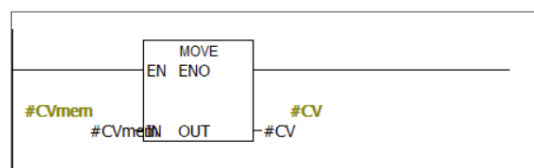
รูปที่ 14: Network4

- การนำค่ามาแสดงที่ฝั่ง output : ค่าของ CO จะแสดงเมื่อ CVmem ไม่เท่ากับ 0 ซึ่งเขียนโปรแกรมได้ ดังรูปที่ 15



รูปที่ 15: Network5

- การนำค่าที่ counter นับได้มาแสดงที่ CV : ใช้คำสั่ง move ในการย้ายค่าที่ counter นับได้ ซึ่งเก็บไว้ใน CVmem ไปแสดงที่ CV ซึ่งเป็นฝั่ง output เขียนโปรแกรมได้ ดังรูปที่ 16



รูปที่ 16: Network6

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

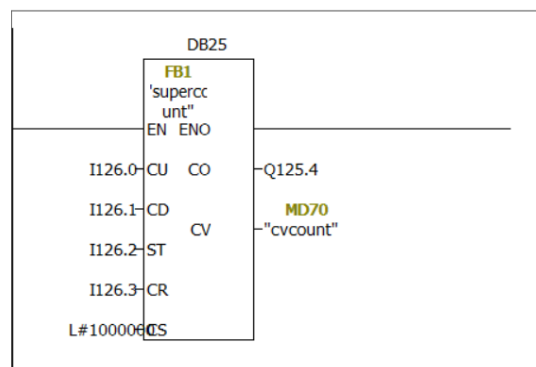
Group: INCA01

Name: Chansinee Mueangnu

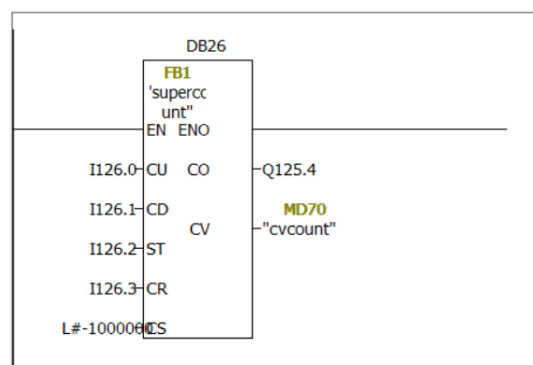
Student ID: 63070501221

Group: INCA01

3. เมื่อสร้าง Function Block ของ counter ใน FB1 เสร็จแล้ว เราจะนำ Function Block ที่สร้างไว้มาเรียกใช้ใน FB2 ซึ่งในที่นี้จะเรียกใช้ counter ขึ้นมาสองตัว ดังรูปที่ 17 ตั้งค่า CS เป็น 1,000,000 และรูปที่ 18 ตั้งค่า CS เป็น -1,000,000 โดยเมื่อทดลองรันโปรแกรม โดยการกด Set (ST) ค่าที่แสดงที่ CV ก็จะเป็นไปตามค่าที่เราได้ทำการ Set ไว้ และนอกจากนี้โปรแกรมก็สามารถที่จะนับค่า count up และ count down ได้อย่างสมบูรณ์

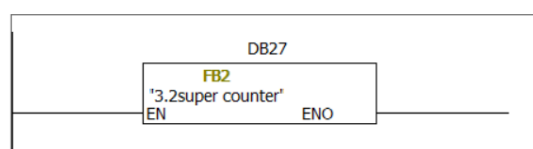


รูปที่ 17: DB25



รูปที่ 18: DB26

โดยการทำงานทั้งหมดของโปรแกรม Super Counter นี้ เราได้เขียนโปรแกรมไว้ใน FB1 โดยเราสามารถที่จะเรียกใช้ FB1 ที่เป็นฟังก์ชัน Counter UD นี้ได้ เช่น ตัวอย่างข้างต้นที่เราได้เรียกใช้งานฟังก์ชัน Counter UD ที่เราสร้างขึ้นเองบน FB1 มาใช้ใน FB2 และเรียกใช้งาน FB2 ผ่าน OB1 ดังรูปที่ 19



รูปที่ 19: รัน FB2 ใน OB1

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

Group: INCA01

- **Questions:** What is the most suitable IEC object for solve this problem between FC and FB? Why?

ทางกลุ่มของเราคิดว่าสำหรับโจทย์ข้อนี้ควรใช้ FB จึงจะเหมาะสมที่สุดเนื่องจากหลังการเปลี่ยนแปลงค่าแล้ว ในการทำงานครั้งต่อไปหากมีการเพิ่มหรือลดค่า เราจะนำค่าเดิมมาใช้ทำให้หากใช้ FC ที่ไม่มีการเก็บข้อมูล จะทำให้เป็นการเปลี่ยนแปลงค่าเริ่มต้นที่เรากำหนดไว้แทน

- **Summary:**

เราได้สร้าง function block ที่เป็น counter UD ขึ้นมา โดยที่สามารถตั้งค่า Value for Presetting Counter (CS) ในช่วง -1,000,000 ถึง 1,000,000 ได้ โดยเราได้เขียนโปรแกรม LADDER ลงใน FB1 และมีการใช้ FB คู่กับ data block (DB) เสมอ เพื่อใช้ DB เป็นตัวเก็บข้อมูล พารามิเตอร์ต่าง ๆ ของ FB

หลังจากนั้นเราได้นำ Function Block ที่สร้างไว้มาเรียกใช้ใน FB2 ซึ่งในที่นี้จะเรียกใช้ counter ขึ้นมาสองตัว โดยตัวแรกตั้งค่า CS เป็น 1,000,000 และอีกตัวตั้งค่า CS เป็น -1,000,000 เมื่อทดลองรันโปรแกรม โดยการกด Set (ST) ค่าที่แสดงที่ CV ก็จะเป็นไปตามค่าที่เราได้ทำการ Set ไว้ และนอกจากนี้โปรแกรมก็สามารถที่จะนับค่า count up และ count down ได้อย่างสมบูรณ์

3.3 TRAFFIC LIGHT

Name: Thanakrit Jadhong
Name: Latthaphon Piyasuk
Name: Chansinee Mueangnu

Student ID: 63070501208
Student ID: 63070501216
Student ID: 63070501221

Group: INCA01
Group: INCA01
Group: INCA01

3.3 Traffic light

Write a stage diagram for traffic light. Then, write the Program (S7-300) and HMI (WinCC) for this application. (You must create “AUTO/Manual” button and “STEP” button on WINCC screen.)

1.A traffic light system is designed to control traffic flowing into an intersection of two roads.

2.The system should perform the following actions.

- When the switch, AUTO/MANUAL is 'on', the traffic lights follow the sequence shown in Figure 1. The activation of the pushbutton STEP has no effect on the system.
- When the switch, AUTO/MANUAL is turned off, the current operation state of the traffic light system is held until the pushbutton STEP is pressed. Then the system enters the next operation state and so on.
- If there is a power failure and recovery (CPU stop and switch to run), only YELLOW1 and YELLOW2 are activated for ten second (the other lights are off) before the system restarts from the first state of the operation sequence.

You should name your inputs and outputs according to their functions.

Try to minimize the number of timers used in your program.

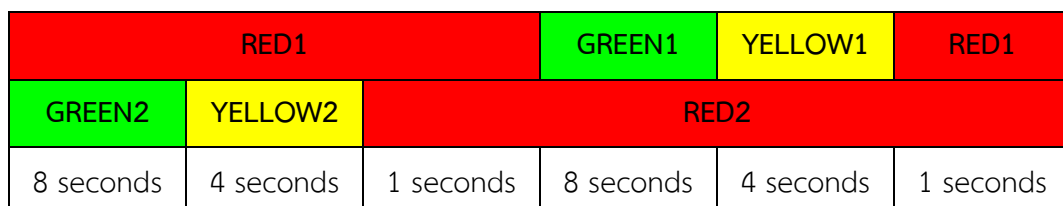


Figure 1: Traffic light operation sequence

Inputs	Outputs
AUTO/MANUAL: I124.0	RED1: Q 124.0
STEP: I124.1	YELLOW1: Q 124.1
	GREEN1: Q 124.2
	RED2: Q 124.3
	YELLOW2: Q 124.4
	GREEN2: Q 124.5

Name: Thanakrit Jadthong
 Name: Latthaphon Piyasuk
 Name: Chansinee Mueangnu

Student ID: 63070501208
 Student ID: 63070501216
 Student ID: 63070501221

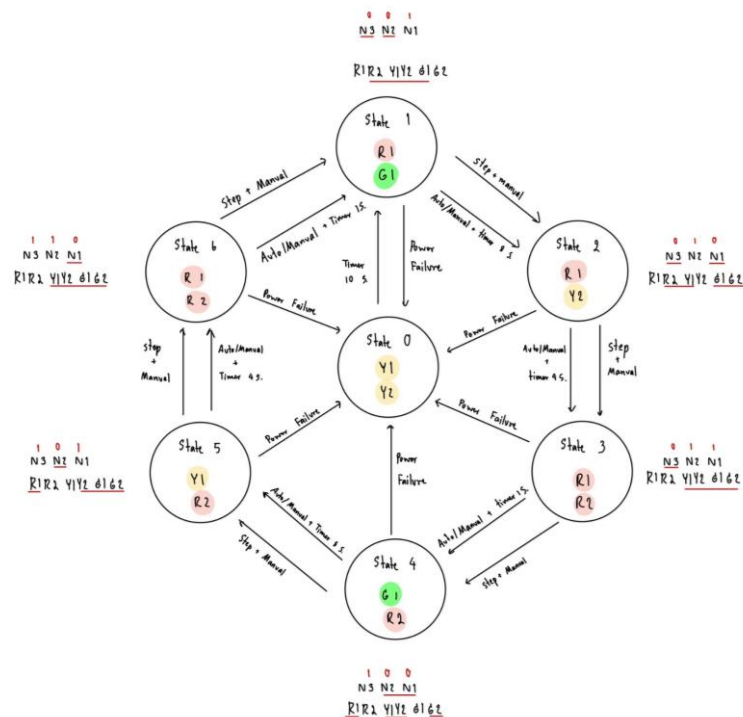
Group: INCA01
 Group: INCA01
 Group: INCA01

➤ Symbol:

Symbol /	Address	Data type
3.1.distanceD	FC 1	FC 1
3.2.super counter	FB 2	FB 2
3.3.traffic light	FC 2	FC 2
auto/manual button	I 124.0	BOOL
counter1	FB 3	FB 3
counter2	FB 4	FB 4
cvcount	MD 70	DINT
Distance	MD 60	REAL
GREEN1	Q 124.2	BOOL
GREEN2	Q 124.5	BOOL
i125.0	I 124.7	BOOL
mode	Q 125.0	BOOL
Power Start	I 124.2	BOOL
RED1	Q 124.0	BOOL
RED2	Q 124.3	BOOL
reset	I 125.1	BOOL
sigma_XY	MD 56	REAL
step button	I 124.1	BOOL
supercount	FB 1	FB 1
X	M 20.0	BOOL
X_dif	MD 28	REAL
X_sqr	MD 32	REAL
X1	MD 20	REAL
X2	MD 24	REAL
Xd	MW 22	INT
Y_dif	MD 48	REAL
Y_sqr	MD 52	REAL
Y1	MD 40	REAL
Y2	MD 44	REAL
YELLOW1	Q 124.1	BOOL
YELLOW2	Q 124.4	BOOL

ตารางที่ 7: สัญลักษณ์ที่ใช้ในการเขียนโปรแกรม S7-300 ทั้งหมด
 (สำหรับโปรแกรม Traffic Light เฉพาะที่ไฮไลต์)

➤ State diagram:



Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

Group: INCA01

Name: Chansinee Mueangnu

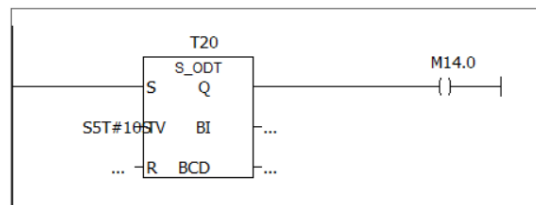
Student ID: 63070501221

Group: INCA01

➤ Program:

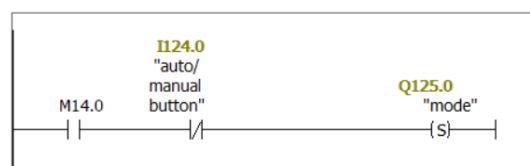
โปรแกรม Traffic Light นี้ เราได้ออกแบบโปรแกรมโดยสร้างให้มีปุ่มสำหรับเลือกโหมดระหว่าง AUTO/MANUAL ซึ่งถ้าหากเลือกโหมด AUTO โปรแกรมก็จะทำการรัน state ต่าง ๆ ตามที่เราได้ออกแบบไว้ โดยอัตโนมัติ แต่ถ้าหากเลือกโหมด MANUAL เราจะต้องทำการกดปุ่ม STEP เพื่อทำการรันในแต่ละ state ซึ่งเราได้เขียนโปรแกรมโดยมีขั้นตอน ดังนี้

- 1) ถ้าหากไฟตก หรือมีเหตุขัดข้องที่ทำให้ CPU stop ลง เมื่อเรา reset CPU ขึ้นมาใหม่ มันจะกลับเข้าสู่โหมดไฟสีเหลือง 10 วินาที ซึ่งเราเขียนโปรแกรมได้ตามรูปที่ 20 คือ เมื่อ M14.0 ทำงาน มันจะทำให้ไฟสีเหลืองทั้งสองดวงติด และเมื่อครบ 10 วินาที M14.0 จะไม่ทำงาน ซึ่งในทุก ๆ Network ในโปรแกรม จะมี M14.0 คั่นไว้ หมายความว่า จะต้องผ่านสถานะที่ไฟสีเหลืองทุกดวงติด 10 วินาทีนี้ไปก่อน ส่วนอื่น ๆ ถึงค่อยเริ่มทำงาน

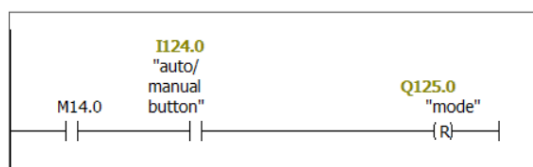


รูปที่ 20: Network1

- 2) รูปที่ 21 และรูปที่ 22 เป็นการเขียนโปรแกรมเกี่ยวกับการกดปุ่มเพื่อเปลี่ยนโหมดระหว่าง AUTO/MANUAL โดยเรามีการใช้คำสั่ง set เพื่อสั่งให้ Q125.0 มี logic เป็น 1 เมื่อกดปุ่ม AUTO/MANUAL และใช้คำสั่ง reset เพื่อสั่งให้ Q125.0 มี logic เป็น 0 ซึ่งการเขียนโปรแกรมเป็นไปตามรูปที่ 21 และรูปที่ 22



รูปที่ 21: Network2



รูปที่ 22: Network3

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

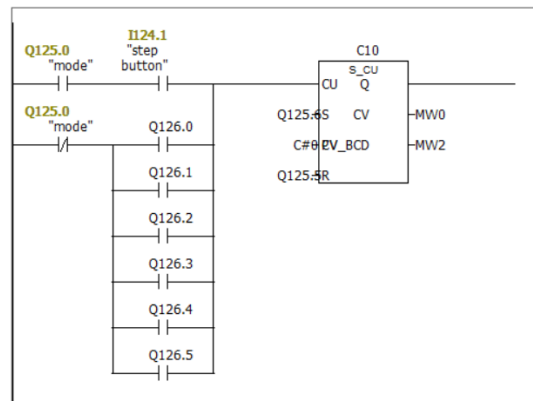
Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

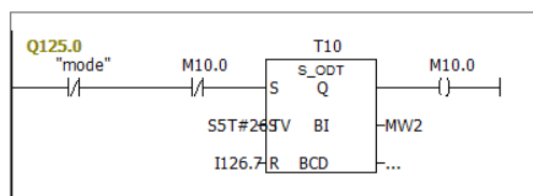
Group: INCA01

- 3) เราได้ออกแบบโปรแกรมโดยการสร้าง main counter ขึ้นมา ซึ่ง main counter นี้ จะเป็น ตัวนับค่า เพื่อสั่งให้โปรแกรมนั้น state ทั้งหมด โดยเอาหน้า contact Q125.0 (mode) มาคั่น ถ้า mode ทำงานก็จะเข้าไปเงื่อนไข AUTO และถ้าหาก mode ไม่ทำงานก็จะเข้าสู่โหมด MANUAL



รูปที่ 23: Network4

- โหมด MANUAL จะเพิ่มค่า counter ได้ด้วยการกดปุ่ม step
- โหมด AUTO จะมีการใช้เวลาเข้ามาเกี่ยวข้อง เริ่มจากมี Network5 เป็น main clock โดย Network5 นี้จะมีการใช้หน้า contact M10.0 มาคั่นไว้ที่ขา set ของ main clock เพื่อให้เกิดการกะพริบไปกะพริบมาของ timer ซึ่งถ้าหากเอาหน้า contact M10.0 ออก มันจะทำให้ไฟเข้าขา set ค้าง และเมื่อนับเวลาครบมันก็จะไม่เริ่มนับใหม่จนกว่าจะมีการตัดไฟแล้วจ่ายใหม่ เราเลยเอาหน้า contact M10.0 มาคั่นไว้ ดังนั้นเมื่อนับเวลาครบ 26 วินาที จะเริ่มจ่ายไฟไปที่ M10.0 แล้วเราก็เอาหน้า contact ของ M10.0 ไปคั่นไว้ที่ขา set ขา set ก็จะกลายเป็น open ทันทีทำให้ไฟไม่เข้า timer แล้วไฟก็จะไม่จ่ายไปที่หน้า contact M10.0 และเราก็จะเอาเวลาที่ MW2 มาใช้ในการควบคุมการเปรียบเทียบ



รูปที่ 24: Network5

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

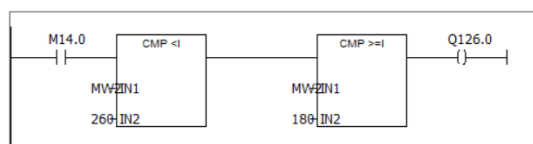
Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

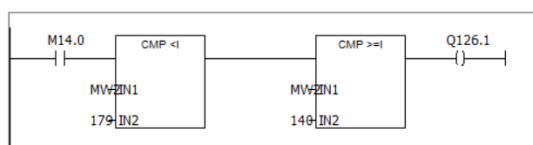
Group: INCA01

- 4) พิจารณา Network6 comparator ตัวแรกเอาไว้ check ว่าเวลาที่ timer นับมา (ค่าที่เก็บไว้ที่ MW2) มันน้อยกว่า 26 วินาที ใช่หรือไม่ ถ้าหากใช่ก็ check ต่อว่าเวลาที่นับมายังมากกว่า 18 วินาทีใช่หรือไม่ (timer นับเวลาถอยหลัง) นั่นคือ ช่วงเวลา 8 วินาทีแรกที่ timer นับ จะสั่งให้ Q126.0 ทำงาน ซึ่ง Q126.0 นี้ก็จะไปทำให้ count เพิ่มขึ้นเป็น 1



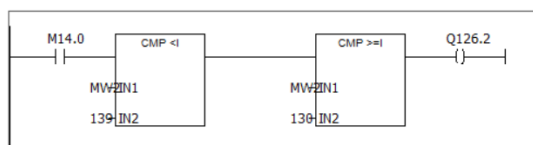
รูปที่ 25: Network6

- 5) เมื่อเวลาลดไปเรื่อย ๆ เกิน 8 วินาทีไปแล้ว โปรแกรมก็จะมา check ที่ Network7 โดยเราจะใช้ comparator เพื่อ check ว่าเวลาที่นับไปอยู่ในช่วงใด ซึ่งในโปรแกรมเราได้ใช้ MW2 เปรียบเทียบกับค่า 179 เนื่องจากเราไม่สามารถเขียนเวลาให้ซ้อนทับกันได้ ถ้าหากเวลาซ้อนทับกันมันจะเหมือนกับว่าเราสั่งให้โปรแกรมทำสองอย่างพร้อมกัน ดังนั้นโปรแกรมที่เราเขียนจึงสั่งให้มันหยุดทำ Network6 ก่อนเสียวินาทีหนึ่ง แล้วค่อยมาทำ Network7 ต่อไป โดย Network7 นี้จะเป็นช่วงหลังจาก 8 วินาทีแรก แล้วนับไปอีก 12 วินาที



รูปที่ 26: Network7

- 6) เมื่อเวลาลดลงไปเรื่อย ๆ จนเกิน 4 วินาทีแล้ว โปรแกรมจะมา check ที่ Network 8 โดยนำค่าจาก MW2 มาใช้เทียบกับ 139 ว่าน้อยกว่า 139 และมากกว่า 130 หรือไม่ ถ้าเป็นจริง ก็จะ ทำให้หน้า contact Q126.2 ทำงาน จากนั้นนับไปอีก 1 วินาที ในช่วงนี้คือช่วงไฟแดงทั้ง 2 ดวง ติดพร้อมกัน 1 วินาที



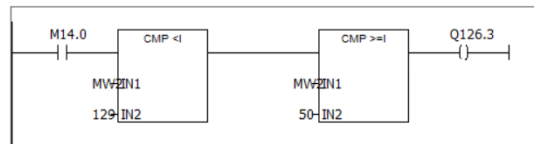
รูปที่ 27: Network8

Name: Thanakrit Jadthong
Name: Latthaphon Piyasuk
Name: Chansinee Mueangnu

Student ID: 63070501208
Student ID: 63070501216
Student ID: 63070501221

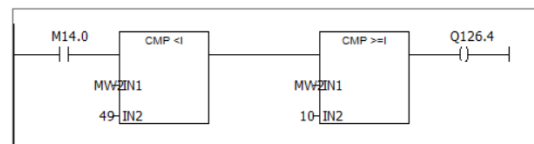
Group: INCA01
Group: INCA01
Group: INCA01

- 7) ใน Network 9 คือ comparator นำ MW2 มาเปรียบเทียบและสั่งให้หน้า contact Q126.2 ทำงาน โดยจะทำงานอยู่ 8 วินาที ซึ่งในช่วงนี้คือช่วงที่ไฟเขียวดวงที่ 1 และไฟแดงดวงที่ 2 ติดพร้อมกัน 8 วินาที



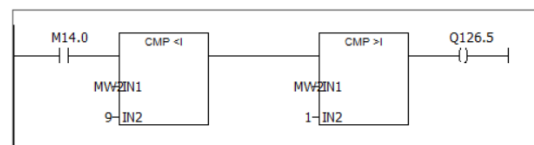
รูปที่ 28: Network9

- 8) หลังจาก Network 9 ทำงานมา 8 วินาทีแล้ว จะมาทำงานใน Network 10 ต่อคือนำ MW2 มาเทียบว่า อยู่ในช่วง 49-10 หรือไม่แล้วจึงสั่งให้ Q126.2 ทำงาน หากเป็นจริง ซึ่งจะทำงานอยู่ 4 วินาที ซึ่งคือช่วงที่ไฟเหลืองดวงที่ 1 และไฟแดงดวงที่ 2 ติดพร้อมกัน



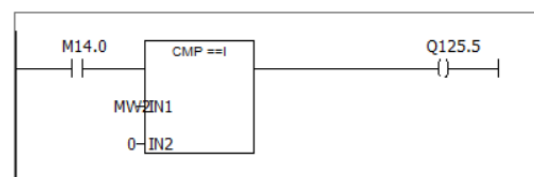
รูปที่ 29: Network10

- 9) ใน Network 10 จะเป็นการทำงานโดยนำค่าจาก MW2 มาเทียบว่าอยู่ในช่วง 0-9 หรือไม่ หากเป็นจริงจึงสั่งให้ Q126.2 ทำงาน ซึ่งก็คือช่วงที่ ไฟแดงทั้ง 2 ดวงติดพร้อมกัน 1 วินาที



รูปที่ 30: Network11

- 10) พิจารณาที่ Network 12 เมื่อเวลาเป็น 0 (นับครบ 26 วินาทีแล้ว) เราถึงจะจ่ายไฟเข้าที่ Q125.5 ซึ่งเป็นการจ่ายไฟเข้าที่ขา reset ของตัว counter นั่นคือสั่งให้ counter เป็น 0 จากนั้นการทำงานก็จะทำงานวนลูปไปเรื่อย ๆ



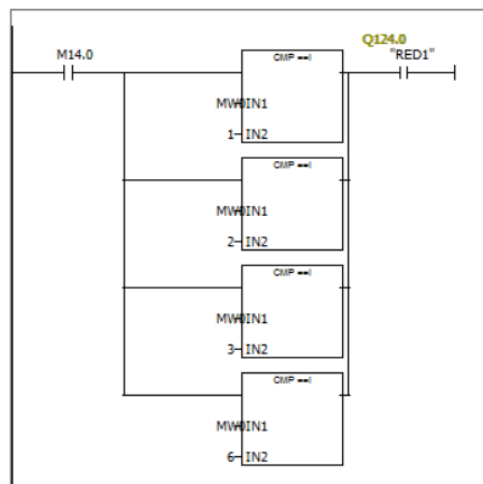
รูปที่ 31: Network12

Name: Thanakrit Jadhong
Name: Latthaphon Piyasuk
Name: Chansinee Mueangnu

Student ID: 63070501208
Student ID: 63070501216
Student ID: 63070501221

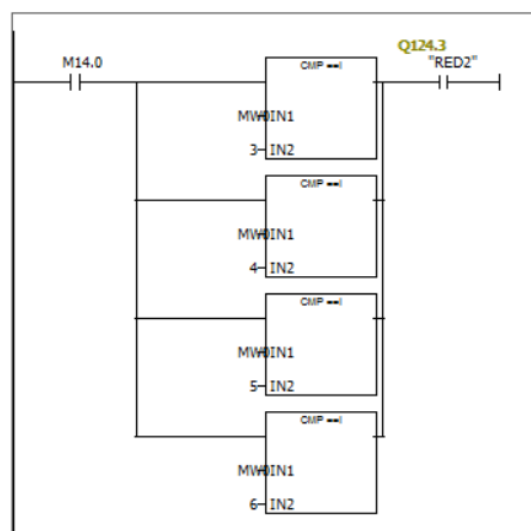
Group: INCA01
Group: INCA01
Group: INCA01

- 11) ตั้งแต่ Network13 เป็นต้นไปจนถึง Network 18 เป็นการแสดงผลของไฟจราจร ซึ่งเราก็จะแยกเป็น red1 red2 yellow1 yellow2 green1 และ green2 เนื่องจากเราต้องการให้มี contact out เพียงตัวเดียว โดยเรามีการใช้ comparator โดยนำเอา mw0 มาเปรียบเทียบกับค่า counter เป็นเท่าไร เพื่อเปรียบเทียบกับเลข state เช่น ถ้า state เป็น 1, 2, 3 และ 6 ไฟสีแดงดวงที่ 1 ติด ซึ่งมีหน้าคอนแทค NO M14.0 มาคั่นคือให้ทำงานหลังจากที่เปิดเครื่องและไฟเหลืองติดไปแล้ว 10 วินาที



รูปที่ 32: Network13

- 12) ใน Network 14 คือเงื่อนไขที่ทำให้หลอดไฟสีแดงดวงที่ 2 คือติดเมื่อค่าจาก counter หรือ state นั้นมีค่าเป็น 3, 4, 5 และ 6



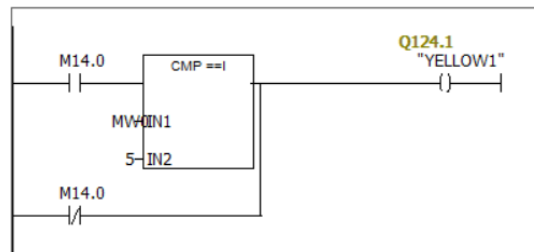
รูปที่ 33: Network14

Name: Thanakrit Jadhong
Name: Latthaphon Piyasuk
Name: Chansinee Mueangnu

Student ID: 63070501208
Student ID: 63070501216
Student ID: 63070501221

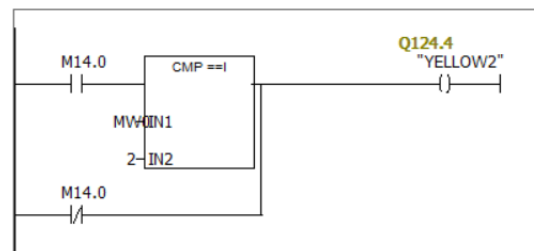
Group: INCA01
Group: INCA01
Group: INCA01

- 13) ใน Network 15 คือเงื่อนไขที่ทำให้หลอดไฟสีเหลืองดวงที่ 1 ติด คือติดเมื่อค่าจาก counter หรือ state นั้นมีค่าเป็น 5 และกรณีที่ PLC นั้นถูกเปิดในช่วง 10 วินาทีแรก



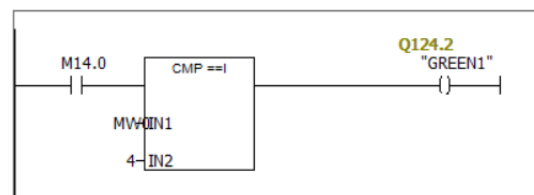
รูปที่ 34: Network15

- 14) ใน Network 16 คือเงื่อนไขที่ทำให้หลอดไฟสีเหลืองดวงที่ 2 ติด คือติดเมื่อค่าจาก counter หรือ state นั้นมีค่าเป็น 2 และกรณีที่ PLC นั้นถูกเปิดในช่วง 10 วินาทีแรก



รูปที่ 35: Network16

- 15) ใน Network 17 คือเงื่อนไขที่ทำให้หลอดไฟสีเขียวดวงที่ 1 ติด คือติดเมื่อค่าจาก counter หรือ state นั้นมีค่าเป็น 4



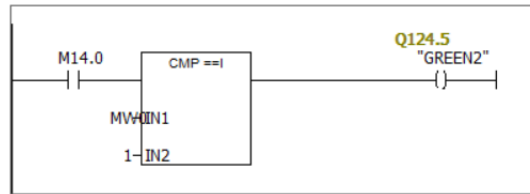
รูปที่ 36: Network17

- 16) ใน Network 18 คือเงื่อนไขที่ทำให้หลอดไฟสีเขียวดวงที่ 2 ติด ซึ่งจะติดเมื่อค่าจาก counter หรือ state นั้นมีค่าเป็น 1

Name: Thanakrit Jadthong
 Name: Latthaphon Piyasuk
 Name: Chansinee Mueangnu

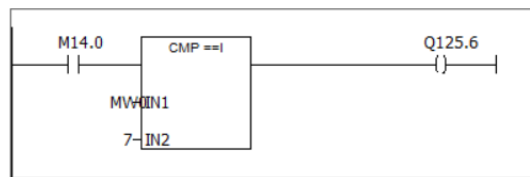
Student ID: 63070501208
 Student ID: 63070501216
 Student ID: 63070501221

Group: INCA01
 Group: INCA01
 Group: INCA01



รูปที่ 37: Network18

17) Network19 เมื่อค่าเป็น 7 ให้จ่ายไฟเข้า q125.6 คือหา set -counter ก็จะนับใหม่เป็น 1



รูปที่ 38: Network19

➤ HMI (WinCC):

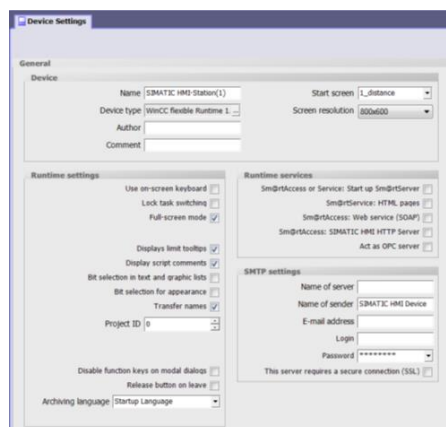
1. Tags:

Name	Display name	Connection	Data type	Symbol	Address	Array elements	Acquisition cycle
AUTO/MANUAL		Connection_1	Bool	<Undefined>	I 124.0	1	1 s
distance		Connection_1	Real	<Undefined>	MD 60	1	1 s
GREEN1		Connection_1	Bool	<Undefined>	Q 124.2	1	1 s
GREEN2		Connection_1	Bool	<Undefined>	Q 124.5	1	1 s
RED1		Connection_1	Bool	<Undefined>	Q 124.0	1	1 s
RED2		Connection_1	Bool	<Undefined>	Q 124.3	1	1 s
STEP BUTTON		Connection_1	Bool	<Undefined>	I 124.1	1	1 s
X1		Connection_1	Real	<Undefined>	MD 20	1	1 s
X2		Connection_1	Real	<Undefined>	MD 24	1	1 s
Y1		Connection_1	Real	<Undefined>	MD 40	1	1 s
Y2		Connection_1	Real	<Undefined>	MD 44	1	1 s
YELLOW1		Connection_1	Bool	<Undefined>	Q 124.1	1	1 s
YELLOW2		Connection_1	Bool	<Undefined>	Q 124.4	1	1 s

ตารางที่ 4: Tags ทั้งหมดที่ใช้ในการเขียน HMI (สำหรับโปรแกรม Traffic Light เฉพาะที่ไฮไลต์)

2. Setting:

- ตั้งค่า resolution ของหน้าจอ



รูปที่ 39: Device Settings

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

Student ID: 63070501216

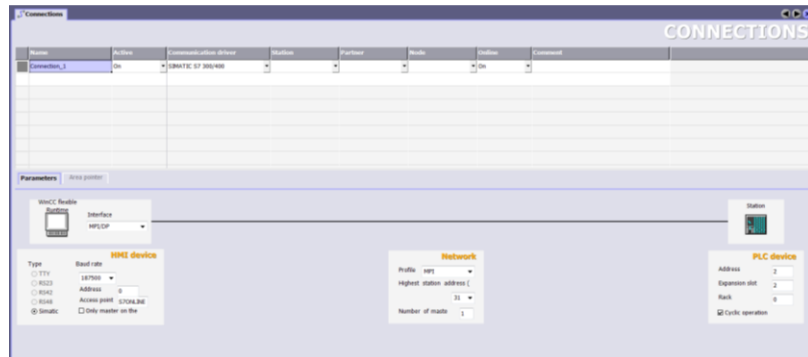
Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

Group: INCA01

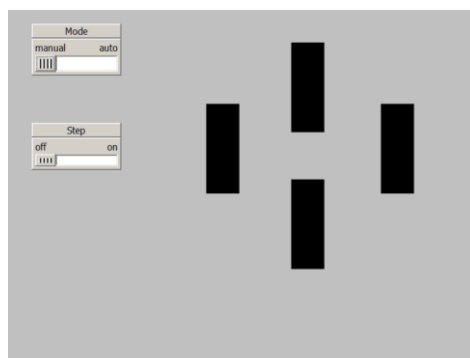
- เลือก protocol ในการสื่อสาร



รูปที่ 40: Communication → Connections

3. HMI screens:

- กรณีที่เลือกโหมด MANUAL จะต้องทำการกดปุ่ม step เพื่อให้โปรแกรมรันต่อที่ละ state
- กรณีที่มีเหตุขัดข้องที่ทำให้ CPU stop ลง เมื่อเรา reset CPU ขึ้นมาใหม่ มันจะกลับเข้าสู่โหมดไฟสีเหลือง 10 วินาทีก่อนทุกครั้งที่จะเริ่มทำงาน State อื่น ๆ ต่อไป
- กรณีที่เลือกโหมด AUTO โปรแกรมจะรัน state โดยอัตโนมัติตามเวลาที่ตั้งค่าไว้ ซึ่ง state ในการรันของโปรแกรมทั้งโหมด AUTO และ MANUAL จะเหมือนกัน ต่างกันเพียงแค่ถ้าเป็นโหมด AUTO โปรแกรมจะรันอัตโนมัติตามเวลาที่ตั้งค่าไว้ ดังนี้
 - State1 : RED1 และ GREEN2 ทำงานพร้อมกัน 8 วินาที
 - State2 : RED1 และ YELLOW2 ทำงานพร้อมกัน 4 วินาที
 - State3 : RED1 และ RED2 ทำงานพร้อมกัน 1 วินาที
 - State4 : GREEN1 และ RED2 ทำงานพร้อมกัน 8 วินาที
 - State5 : YELLOW1 และ RED2 ทำงานพร้อมกัน 4 วินาที
 - State6 : RED1 และ RED2 ทำงานพร้อมกัน 1 วินาที



รูปที่ 41: POWER FAILURE

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

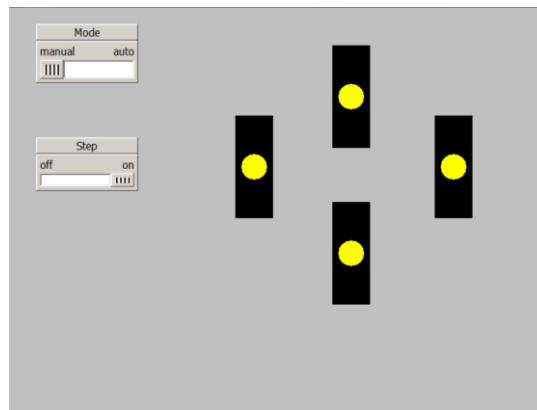
Student ID: 63070501216

Group: INCA01

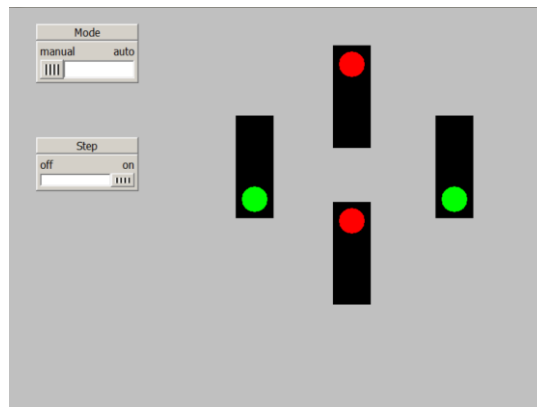
Name: Chansinee Mueangnu

Student ID: 63070501221

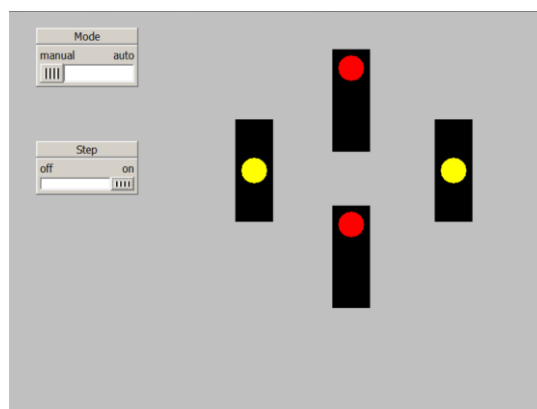
Group: INCA01



รูปที่ 42: สถานะการทำงานของไฟจราจรเมื่อกดปุ่ม reset หลัง power failure



รูปที่ 43: State1



รูปที่ 44: State2

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

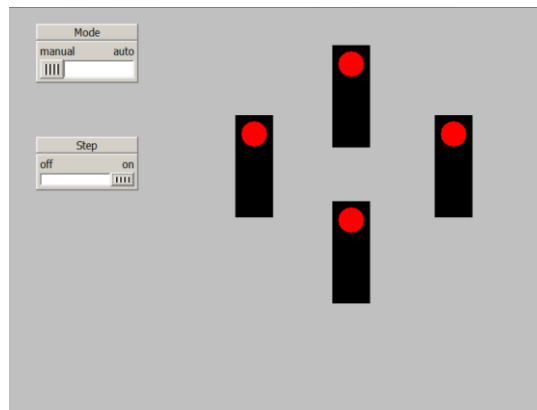
Student ID: 63070501216

Group: INCA01

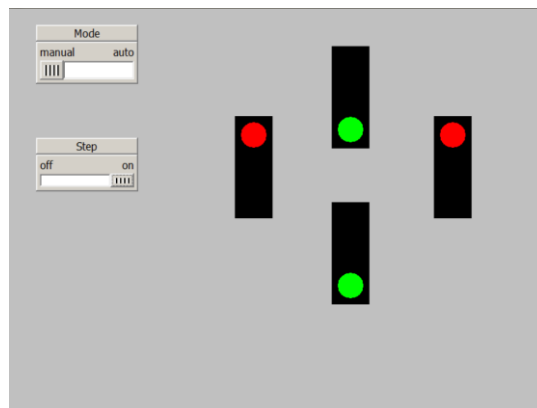
Name: Chansinee Mueangnu

Student ID: 63070501221

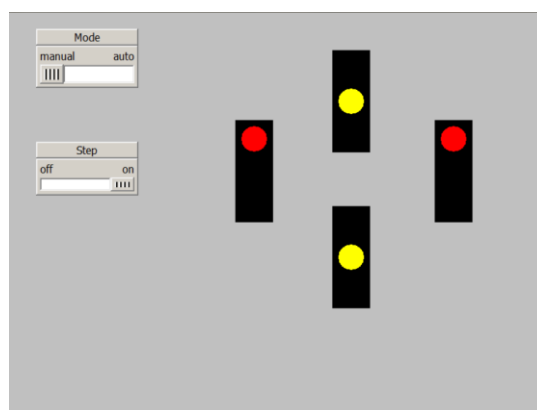
Group: INCA01



รูปที่ 45: State3



รูปที่ 46: State4



รูปที่ 47: State5

Name: Thanakrit Jadthong

Student ID: 63070501208

Group: INCA01

Name: Latthaphon Piyasuk

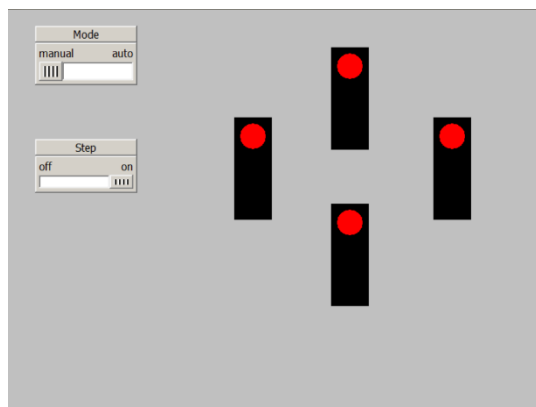
Student ID: 63070501216

Group: INCA01

Name: Chansinee Mueangnu

Student ID: 63070501221

Group: INCA01



รูปที่ 48: State6

➤ Summary:

จากการทำ Traffic light ซึ่งเป็นการควบคุมไฟจราจรโดยแบ่งออกเป็น 2 ชุด ซึ่งมีช่วงเวลาการทำงานที่เหมือนกันแต่แตกต่างกันตรงสัญญาณไฟที่แสดงพร้อมกัน โดยประกอบไปด้วยไฟ 3 สี 1.สีเขียว 2.สีแดง และ 3.สีเหลือง โดยได้ออกแบบให้มี Mode Auto และ Manual โดยใน Mode Manual ไฟจะทำงานเป็นลำดับก็ต่อเมื่อมีการกดปุ่ม Step และใน Mode Auto ไฟจะทำงานตามลำดับโดยอัตโนมัติ โดยเราจะสร้าง Main counter ขึ้นมาเพื่อทำการรัน state โดยแต่ละ state จะใช้ Comparator รับค่าเวลามาจาก Timer ทำให้เราสามารถกำหนดเวลาในแต่ละช่วง state ได้ จากนั้นเราก็ให้ตัว Main counter รัน state ตามลำดับ ไฟก็จะติดและดับตามแต่ละเงื่อนไขของแต่ละ state นั้นๆ และหากเกิด Power Failure หลังจากที่เราทำการ Reset PLC แล้วไฟจราจรทั้ง 2 ชุด จะต้องติดเป็นไฟสีเหลือง เป็นเวลา 10 วินาที จากนั้นจึงกลับไปทำงานตามปกติ ต่อมาทำการสร้าง HMI ผ่าน WinCC ซึ่งจากการทดสอบโปรแกรมพบว่าไฟจราจรสามารถทำงานได้ตามปกติทั้งใน Mode Manual , Mode Auto และกรณีที่เกิด Power failure (CPU Stop)