

HOMEWORK 1

PID MMI (Man-Machine Interface)

จัดทำโดย

นางสาวจิรัชยา	สมุทรสินธุ์	63070501201
นายชัยพัชร	แก้ววรรณ	63070501203
นายพิรเดช	ตันติสุขารมย์	63070501212
นางสาวตรีรัตน์	คำเวช	63070501215
นายลัทธพล	ปิยะสุข	63070501216
นายศิริวิทย์	เดชประสพชัย	63070501220
นางสาวฉันทสิณี	เมืองหนู	63070501221

INC 354

ภาคการศึกษาที่ 1/2565

อาจารย์สมชัย ตรีรัตนจารุ

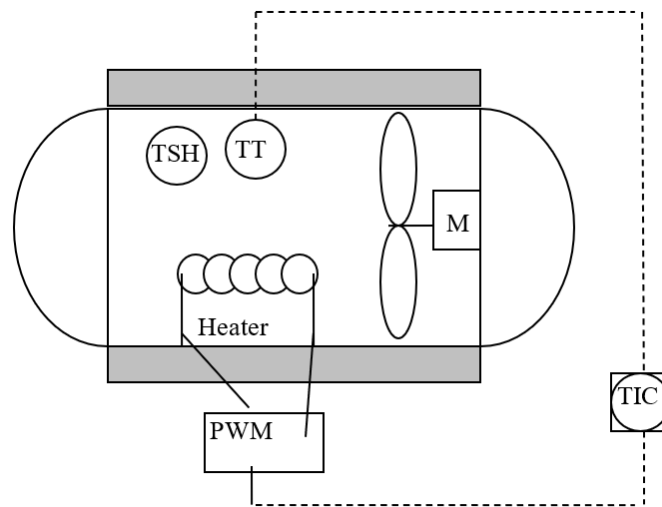
Homework 1

PID MMI (Man-Machine Interface)

Objective:

Understanding modern MMI (or sometime called HMI: Human machine interface) based on Web base technology.

Problem: Rework PID control on hot air process. Including software alarm high cut off heater at 75% of PV.



Make the user define Web page on the Simatic PLC, including these objects and on-line link to display web page as simple HMI.

PV as temperature. (Display only)

SP as setpoint temperature. (Display and can be changed in Auto mode)

MV as power output for heater. (Display and can be changed in Manual mode)

(All in percentages and in real formatted)

Switch or push-button object to switch between Auto and Manual mode.

P setting in real format, I setting in IEC time (DINT of ms) format.

Three lamps for AUT, MAN and ALM (high temperature alarm).

Set fan speed fix at color mark on the model (around 30%, not in HMI).

Report will include hard copy of control program and Web pages code, discussion about using Web page as HMI compare with dedicated HMI unit. (Siemens KTP 700)

➤ Symbol:

Default tag table							
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	System_Byte	Byte	%MB1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	FirstScan	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	DiagStatusUpdate	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Always TRUE	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Always FALSE	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Clock_Byte	Byte	%MB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Clock_10Hz	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Clock_5Hz	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Clock_2.5Hz	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Clock_2Hz	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Clock_1.25Hz	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Clock_1Hz	Bool	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Clock_0.625Hz	Bool	%M0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Clock_0.5Hz	Bool	%M0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	current_input	Word	%IW6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	current_dword	DWord	%MD50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	current_real	Real	%MD54	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	output	Real	%MD58	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	Tag_1	Bool	%M20.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	Tag_2	Word	%MW24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	AUT_MAN	Bool	%M150.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	SP	Real	%MD120	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
23	PV	Real	%MD124	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
24	Pgain	Real	%MD128	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
25	Igain	Time	%MD132	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
26	MV	Real	%MD136	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
27	new_current_input	Word	%IW8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
28	current_output	Int	%QW0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
29	MV_norm	Real	%MD160	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
30	MV_real	Real	%MD164	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
31	SW_mode	Bool	%IO.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
32	SP_IN	Real	%MD100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
33	MV_IN	Real	%MD104	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
34	Auto_lamp	Bool	%M150.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
35	Man_lamp	Bool	%M150.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
36	ALM_lamp	Bool	%M150.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
37	SW_HMI	Bool	%M150.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
38	Tag_3	Word	%IW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
39	Tag_4	Int	%QW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
40	Tag_5	Int	%MW300	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

ตารางที่ 1: สัญลักษณ์ที่ใช้ในการเขียนโปรแกรม PID MMI ทั้งหมด

➤ Program (LADDER):

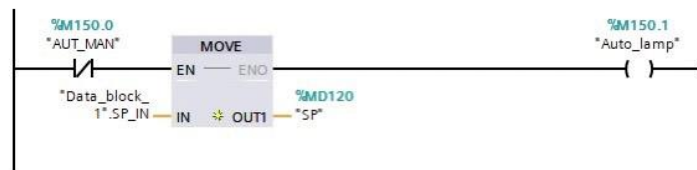
โปรแกรม PID MMI (Man-Machine Interface) เป็นโปรแกรมควบคุม PID ใน hot air process โดยที่จะมีการแจ้งเตือนและตัดการทำงานของโปรแกรมเมื่อ heater ทำงานถึง 75% ของ PV โดยการทำงานของโปรแกรมนี้อาศัยตัวแปรสำหรับควบคุม 3 ตัว คือ PV (Process Variable) เป็นค่าอุณหภูมิภายใน heater ที่เราต้องการควบคุม, MV (Manipulated Variable) เป็นค่าที่มาจาก output ของ PID-controller เพื่อส่งไปให้ heater ในเปลี่ยนแปลงค่า PV, และ SP (Set Point) เป็นค่าอุณหภูมิที่ต้องการให้ heater ทำงาน

การเขียนโปรแกรม PID MMI มีรายละเอียดดังต่อไปนี้

1. การทำงานของโปรแกรม แบ่งเป็น 2 โหมด คือ โหมด auto ซึ่งเราสามารถกำหนดค่า SP หรือ อุณหภูมิที่ต้องการให้ heater ทำงานได้ โดยจะส่งการผ่าน PID-controller ในการเปลี่ยนค่า MV ที่ไปควบคุม heater ให้ได้ค่าอุณหภูมิของ heater (ค่า PV) ใกล้เคียงกับค่า SP ที่ตั้งไว้ และโหมด manual ซึ่งสามารถเปลี่ยนค่า MV หรือค่าที่ไปควบคุม heater ได้ นั่นคือ ค่าของอุณหภูมิภายใน heater ก็จะทำงานตามค่า MV ที่เราตั้งค่าไว้ การทำงานของโปรแกรมส่วนนี้ เราได้เขียนโปรแกรมไว้ใน FC1 network 1-3 ดังรูปที่ 1-3



รูปที่ 1: network1-FC1 (switch to select mode)

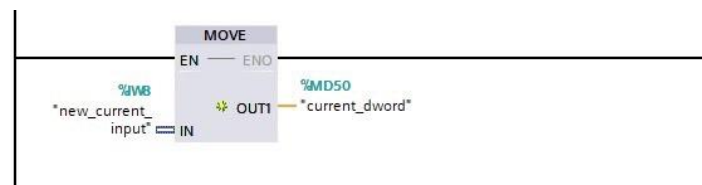


รูปที่ 2: network2-FC1 (change SP in auto mode)

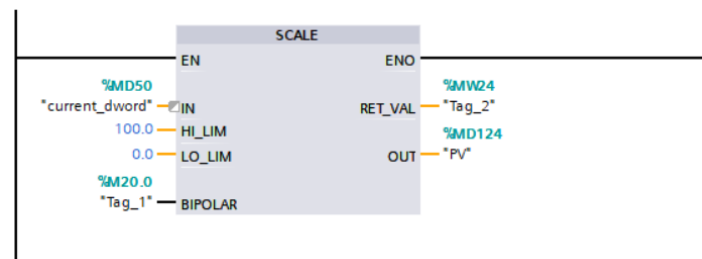


รูปที่ 3: network3-FC1 (change MV in manual mode)

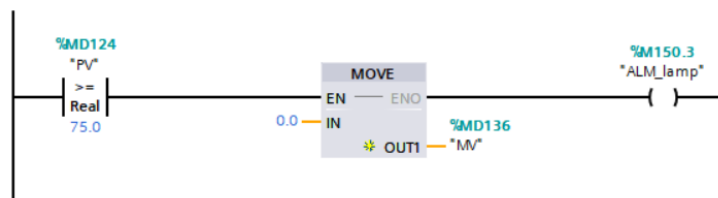
2. ส่วนของการรับค่าเพื่อใช้งานกับ PID controller เป็นส่วนที่ใช้รับค่าอุณหภูมิจาก hot air process หรือค่า PV ซึ่งเป็นสัญญาณ AI 4-20 mA เข้ามาที่ channel AI3 ของ analog card โดยก่อนที่จะนำค่า PV ไปใช้งานกับ PID controller นั้น จะต้องมีการแปลงค่า PV ที่ได้ ให้อยู่ในช่วง 0-100% ซึ่งในที่นี้เราได้เขียนโปรแกรมไว้ใน FC10 “scale” network 1-2 โดยมีรายละเอียดของโปรแกรม ดังรูปที่ 4-5 และจากโจทย์ หากค่า PV มากกว่า 75% จะมีการแจ้งเตือนเป็นไฟ alarm และสั่งการให้ค่า MV = 0 หรือตัดการทำงานของ heater ซึ่งการทำงานของโปรแกรมส่วนนี้ เราได้เขียนโปรแกรมไว้ใน FC1 network 4 ดังรูปที่ 6



รูปที่ 4: network1-FC10 (รับค่าจาก AI3 มาเก็บไว้ใน MD50)

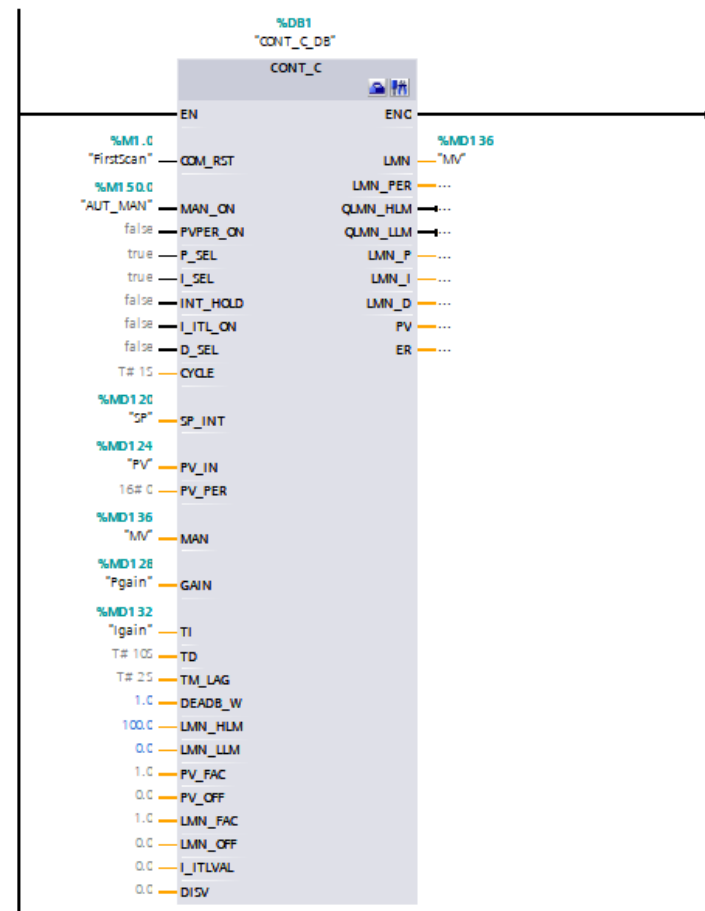


รูปที่ 5: network2-FC10 (scaling PV analog input)



รูปที่ 6: network4-FC1 (check PV to cut off heater)

3. ส่วนควบคุมการทำงานของ heater ด้วยค่า MV ส่วนนี้จะทำงานทั้งในโหมด auto และโหมด manual โดยในโหมด auto ค่า MV จะถูกควบคุมโดย PID-controller เพื่อให้ได้ค่า PV ตามค่า SP ที่เราได้ตั้งค่าไว้ ซึ่งค่า MV นี้ จะถูกใช้งานเป็น input ใน PID-controller และจะได้ MV output จาก PID controller เป็นค่าในช่วง 0-100% ซึ่งเราเขียนโปรแกรมไว้ใน OB 30 network 1 “Cyclic interrupt” ดังรูปที่ 7

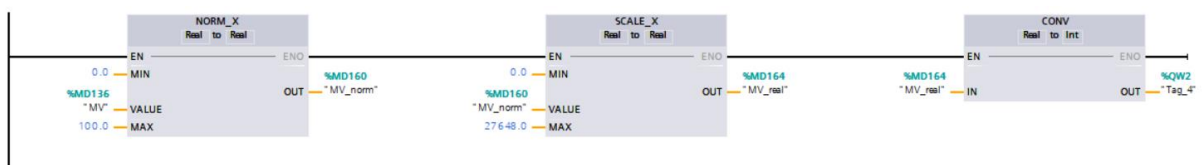


รูปที่ 7: network1-OB30 “Cyclic interrupt” (PID controller)

เนื่องจากค่า MV output จาก PID-controller มีค่าอยู่ในช่วง 0-100% นั่นคือ เราจะต้องแปลงค่า MV output ให้อยู่ในช่วง 0-27648 ซึ่งเทียบเท่ากับสัญญาณ AO 4-20 mA เพื่อที่จะสามารถนำไปสั่งการ heater ให้ทำงานจนได้ค่า PV ตามค่า SP ที่เราได้ตั้งค่าไว้กรณีโหมด auto

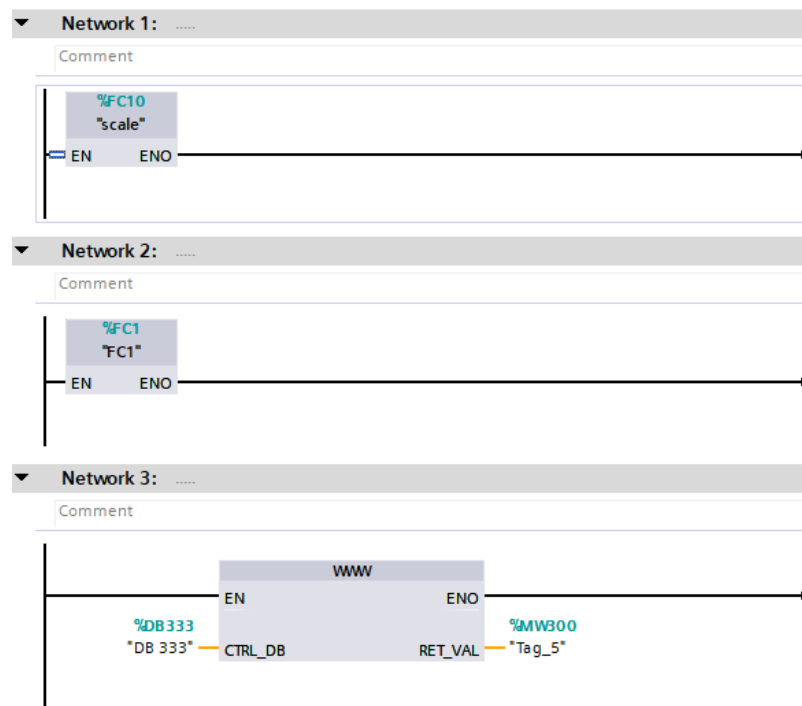
และสำหรับโหมด manual เราสามารถตั้งค่า MV (0-100%) ไปสั่งการ heater ให้ทำงานได้โดยตรง ไม่ต้องผ่านการควบคุมจาก PID-controller เพื่อให้ได้ค่า PV ตามที่เราต้องการ แต่ยังคงต้องมีการแปลงค่าในช่วง 0-100% ให้อยู่ในช่วง 0-27648 เหมือนโหมด auto เช่นกัน ถึงจะสามารถสั่งการ heater ให้ทำงานตามที่ต้องการได้

โดยที่ส่วน scale ค่า MV จาก 0-100% เป็นค่าที่เทียบเท่าสัญญาณ analog output (AO) 4-20 mA เพื่อนำไปสั่งการให้ heater ทำงานได้ตามต้องการนี้ เราได้เขียนโปรแกรมไว้ใน FC10 network 3 “scale” ดังรูปที่ 8



รูปที่ 8: network3-FC10 “scale” (scaling MV analog output)

4. ในโหมด auto เราสามารถตั้งค่า SP ได้ โดยค่า SP จะเป็นการกำหนดขึ้นมาเพื่อให้ PID controller เปลี่ยนแปลงค่า MV และสั่งการไปที่ heater เพื่อให้ได้ค่า PV ตามค่า SP ที่เราได้ตั้งค่าไว้ ซึ่งค่า SP นี้ จะถูกใช้งานเป็น input ใน PID controller ซึ่งเราเขียนโปรแกรมไว้ใน OB 30 network 1 “Cyclic interrupt” ดังแสดงในรูปที่ 7 ข้างต้น
5. การทำงานทั้งหมดของโปรแกรม PID MMI นี้ เราได้เขียนโปรแกรมไว้ใน FC1, FC10 และ OB30 โดยเราได้เรียกใช้งาน FC1, FC10, และ WWW ผ่าน OB1 ดังรูปที่ 9 โดยที่ WWW เป็น System Function ที่ช่วยให้ CPU และ web page สามารถโต้ตอบกันได้ผ่านการเขียน html



รูปที่ 9: OB1 “Main Program”

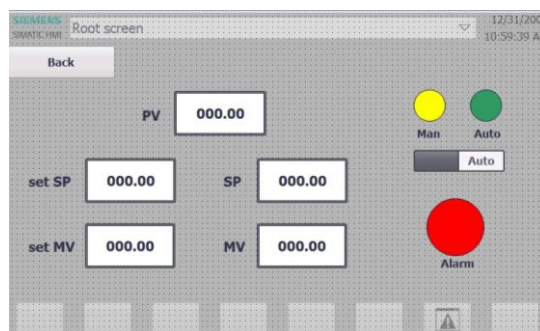
➤ SIMATIC HMI (KTP700):

1. Tags:

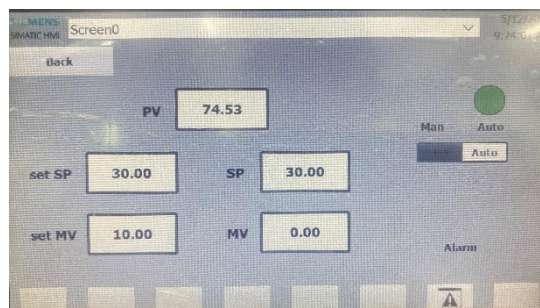
Name	Data type	Connection	PLC name	PLC tag
ALM_lamp	Bool	HMI_Connectio...	PLC_1	ALM_lamp
Auto_lamp	Bool	HMI_Connectio...	PLC_1	Auto_lamp
Man_lamp	Bool	HMI_Connectio...	PLC_1	Man_lamp
MV	Real	HMI_Connectio...	PLC_1	MV
MV_IN	Real	HMI_Connectio...	PLC_1	Data_block_1.MV_IN
PV	Real	HMI_Connectio...	PLC_1	PV
SP	Real	HMI_Connectio...	PLC_1	SP
SP_IN	Real	HMI_Connectio...	PLC_1	Data_block_1.SP_IN
SW_HMI	Bool	HMI_Connectio...	PLC_1	SW_HMI
Tag_ScreenNumber	UInt	<Internal tag>		<Undefined>

ตารางที่ 2: Tags ทั้งหมดที่ใช้ในการเขียนโปรแกรม (SIMATIC HMI) PID MMI ทั้งหมด

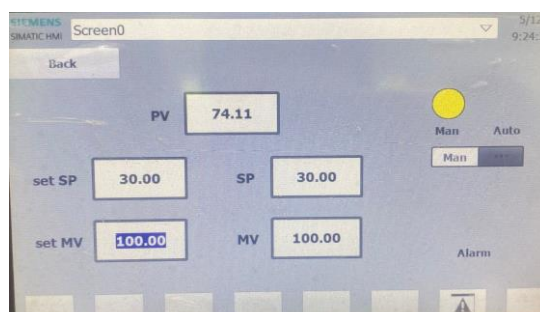
2. HMI Screen:



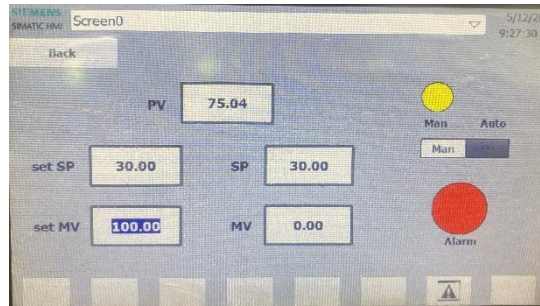
รูปที่ 10: SIMATIC HMI Screen



รูปที่ 11: Auto mode



รูปที่ 12: Manual mode



รูปที่ 13: กรณี PV มีค่าถึง 75 %

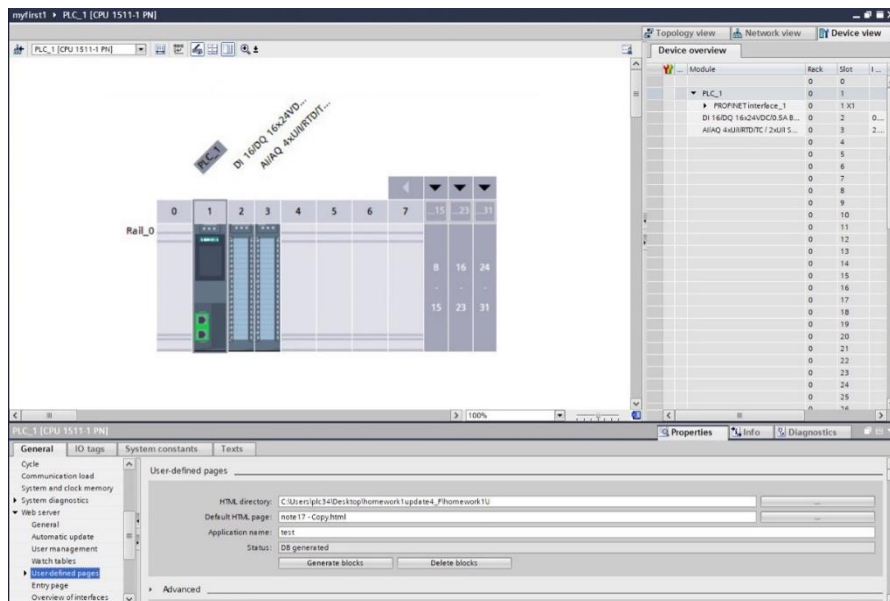
➤ HMI (Web Page):

1. Tags:

Data_block_1								
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static							
2	SP_IN	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	MV_IN	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

รูปที่ 14: data block ที่เก็บค่าตัวแปรสำหรับการรับ-ส่งค่าระหว่าง PLC และ web page

2. Configuration:



รูปที่ 15: Generate blocks สำหรับเขียน html ลงบน web page

3. Coding:

การเขียน HMI ลงบน Web Page จะใช้ภาษา html และ JavaScript ในการเขียนโปรแกรม ซึ่งเขียนโปรแกรมได้ ดังนี้

- ประกาศตัวแปรที่รับค่ามาจาก PLC และเขียนโปรแกรมเพื่อแสดงข้อความในส่วนหัวของ web page รวมถึงการเขียนโปรแกรมเพื่อประกาศเรียกใช้ภาษา JavaScript แสดงได้ดังรูปที่ 16

```
3 <!-- AWP_In_Variable Name="SW_HMI" -->
4 <!-- AWP_In_Variable Name="Data_block_1.SP_IN" -->
5 <!-- AWP_In_Variable Name="Data_block_1.MV_IN" -->
6 <!-- AWP_In_Variable Name="PV" -->
7 <!-- AWP_In_Variable Name="SP" -->
8 <!-- AWP_In_Variable Name="MV" -->
9
10 <!DOCTYPE html>
11
12 <html lang="en">
13
14   <head>
15     <meta charset="utf-8">
16     <title>PLC INC354</title>
17     <script src="jquery-2.0.2.min.js"></script>
18   </head>
```

รูปที่ 16: coding part 1

- การเขียนโปรแกรมส่วนของการตั้งค่ารูปแบบของไฟแสดงผล สำหรับโหมดต่าง ๆ ของโปรแกรม PID MMI (Auto mode, Manual mode, และ Alarm) แสดงได้ดังรูปที่ 17

```
22 <style>
23
24   .alm_lamp {
25     width: 100px;
26     height: 100px;
27     border: solid 5px red;
28     border-radius: 100%;
29     background-color: #B8B8B8;
30     transform: translate(100%, -180%);
31   }
32
33   .man_lamp {
34     width: 100px;
35     height: 100px;
36     border: solid 5px green;
37     border-radius: 100%;
38     background-color: #B8B8B8;
39     transform: translate(100%, 30%);
40   }
41
42   .auto_lamp {
43     width: 100px;
44     height: 100px;
45     border: solid 5px blue;
46     border-radius: 100%;
47     background-color: #B8B8B8;
48     transform: translate(100%, 240%);
49   }
50
51 </style>
```

รูปที่ 17: coding part 2

- ส่วนของการแสดงผลค่า PV, การรับค่า SP สำหรับ Auto mode, และการรับค่า MV สำหรับ Manual mode สามารถเขียนโปรแกรมได้ดังรูปที่ 18

```

54 <body>
55 <h1>Hot air process</h1>
56 </p>
57
58 <form method="post" action="">
59 <span class="home-p-v">PV:</span>
60 &nbsp;
61 := "PV":
62 &nbsp;
63 &nbsp;
64 </br>
65
66 <tr>
67 <form method="post" action="" >
68 <td>Set Point</td> <td>:= "SP":</td>
69 <td><input type="text" name="Data_block_1.SP_IN" maxlength="8" ></td>
70 <td><input type="submit" value="Write"></td>
71 </form>
72 </tr>
73
74 <tr>
75 <form method="post" action="" >
76 <td>Manipulated Value</td> <td>:= "MV":</td>
77 <td><input type="text" name="Data_block_1.MV_IN" maxlength="8" ></td>
78 <td><input type="submit" value="Write"></td>
79 </form>
80 </tr>
81
82 </p>

```

รูปที่ 18: coding part 3

- การโปรแกรมสำหรับเปลี่ยนโหมดระหว่าง auto และ manual แสดงได้ดังรูปที่ 19 โดยหากกดปุ่มเพื่อเลือกโหมด auto จะได้ค่า SW_HMI = 0 และถ้าหากเลือกโหมด manual ก็จะได้ค่า SW_HMI = 1 ซึ่งค่า value จาก SW_HMI นี้จะนำไปใช้สั่งการในส่วนการแสดงสีของหลอดไฟกรณีที่มีการเปลี่ยนโหมด

```

82 </p>
83
84 <form method="post" action="" >
85 <td><input type="submit" value="Auto"></td>
86 <td><input type="hidden" name="SW_HMI" value="0"></td>
87 </form>
88
89 </p>
90
91 <form method="post" action="" >
92 <td><input type="submit" value="Manual"></td>
93 <td><input type="hidden" name="SW_HMI" value="1"></td>
94 </form>
95
96 </p>

```

รูปที่ 19: coding part 4

- รูปที่ 20 เป็นการเขียนโปรแกรมเพื่ออ้างถึงตัวหลอดไฟที่ได้ออกแบบไว้ เพื่อให้สามารถเรียกใช้หลอดไฟตามโหมดที่ต้องการให้แสดงผลได้

```

96     </p>
97
98     <div id = "auto_lamp" class="auto_lamp"></div>
99     <div id = "man_lamp" class="man_lamp"></div>
100    <div id = "alm_lamp" class="alm_lamp"></div>
101
102    </p>
103
104    <p id=mode></p>
105    <p id=almtext></p>
106
107    <script>
108
109        setTimeout(function(){window.location.reload(1);},3000);
110
111        let auto = document.getElementById("auto_lamp");
112        let man = document.getElementById("man_lamp");
113        let alm = document.getElementById("alm_lamp");
114

```

รูปที่ 20: coding part 5

- การแสดงผลหลอดไฟ จะต้องมีการรับค่าจาก value ของ SW_HMI ดังที่กล่าวไปก่อนหน้านี้ โดยหากกดปุ่มเพื่อเลือกโหมด auto หลอดไฟสีน้ำเงินจะติด พร้อมกับแสดงข้อความว่า Auto mode ส่วนหากเลือกโหมด manual หลอดไฟสีเขียวจะติด พร้อมแสดงข้อความว่า Manual mode ซึ่งการเขียนโปรแกรม เป็นไปดังรูปที่ 21

```

117    mode = ':"SW_HMI:"';
118
119    if(mode == 0) {
120        document.getElementById("mode").innerHTML = "Auto mode";
121
122        auto.style.backgroundColor = "#0000FF";
123        man.style.backgroundColor = "#B8B8B8";
124    }
125    if(mode == 1) {
126        document.getElementById("mode").innerHTML = "Manual mode";
127
128        auto.style.backgroundColor = "#B8B8B8";
129        man.style.backgroundColor = "#00FF00";
130    }

```

รูปที่ 21: coding part 6

- และกรณีที่ค่า PV มีค่าตั้งแต่ 75 % ขึ้นไป จะมีการแจ้งเตือนเป็นไฟ alarm สีแดง พร้อมกับแสดงข้อความแจ้งเตือนว่า Alarm !! เงื่อนไขนี้สามารถเขียนโปรแกรมได้ดังรูปที่ 22 โดยที่หากค่า PV กลับมามีค่าลดลงต่ำกว่า 75 % การแจ้งเตือน alarm จะดับลง

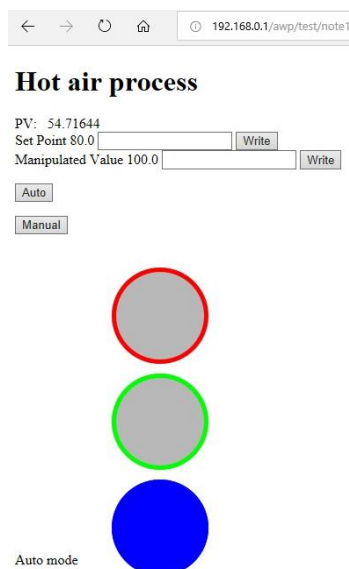
```

132     pv = ':=PV: ';
133     if(pv >= 75) {
134         alm.style.backgroundColor = "#FF0000";
135         document.getElementById("almtext").innerHTML = "Alarm!!";
136     }
137     else {
138         alm.style.backgroundColor = "#B8B8B8";
139     }
140
141     </script>
142     </body>
143 </html>
144

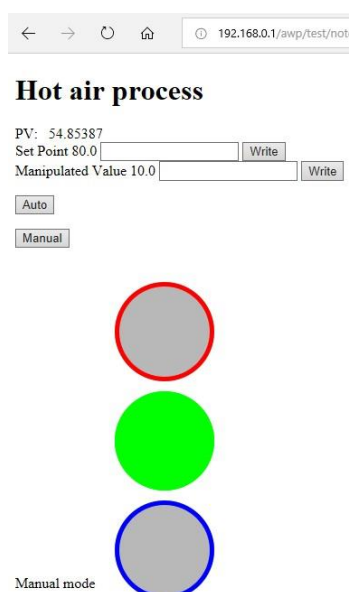
```

រូបទី 22: coding part 7

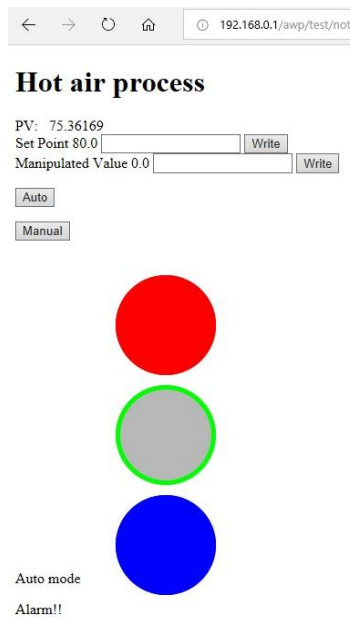
4. HMI Screen:



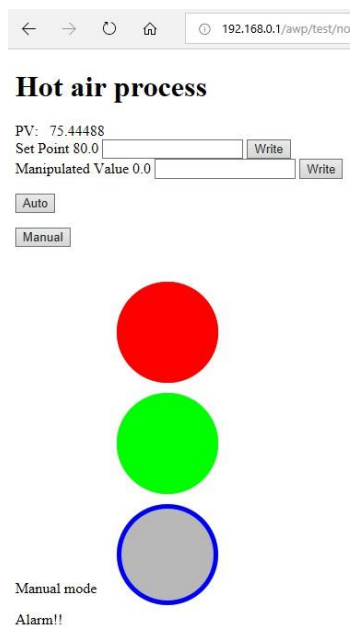
រូបទី 23: Auto mode



រូបទី 24: Manual mode



รูปที่ 25: Auto mode กรณี PV > 75% (มีแจ้งเตือน Alarm)



รูปที่ 26: Manual mode กรณี PV > 75% (มีแจ้งเตือน Alarm)

➤ Summary:

การเขียนโปรแกรมควบคุม Hot Air Process ด้วย PID MMI นี้ เราได้ออกแบบโปรแกรมโดยสร้างให้มีปุ่มสำหรับเลือกโหมดระหว่าง Auto/Manual พร้อมกับมีไฟแสดงสถานะการทำงานของแต่ละโหมด ซึ่งถ้าหากเลือกโหมด Auto จะสามารถกำหนดค่า SP ได้ แล้ว PID-controller ก็ทำการเปลี่ยนค่า MV ที่ไปควบคุม heater เพื่อให้ได้ค่า PV ใกล้เคียงกับ SP ที่เราตั้งไว้โดยอัตโนมัติ แต่หากเลือกโหมด Manual จะสามารถกำหนดค่า MV ได้แทน นั่นคือ การควบคุม heater ในโหมด Manual นี้ ไม่ต้องอาศัย PID-controller ในการสั่งการ แต่สามารถใช้ค่า MV ที่กำหนด ไปควบคุมอุณหภูมิภายในของ heater ได้โดยตรง นอกจากนี้โปรแกรมยังมีส่วนป้องกันความปลอดภัยอีกด้วย กรณีที่ PV มีค่าตั้งแต่ 75 % เป็นต้นไป ระบบจะตัดการทำงานของ heater โดยอัตโนมัติ พร้อมทั้งมีการแจ้งเตือน Alarm

อีกทั้งเราได้เขียนโปรแกรม HMI (KTP700) และเขียนโปรแกรม html เพื่อสร้าง HMI ลงบน web page สำหรับการแสดงผลค่าต่าง ๆ รวมถึงสามารถควบคุมการทำงานของระบบได้โดยตรงผ่าน HMI ทั้งสองแบบ ซึ่งการเขียนโปรแกรม html เพื่อสร้าง HMI ลงบน web page นั้น มีข้อดี คือ สามารถสั่งการ PLC แบบ remote ได้ โดยจำเป็นต้องรับ-ส่งค่าผ่าน data block ซึ่งต่างจากหน้าจอ HMI ที่มักใช้สั่งการ PLC จากหน้าจอ โดยมีการรับ-ส่งค่าผ่าน default tag ซึ่งไม่จำเป็นต้องใช้ data block ในการรับ-ส่งค่าเหมือน web page