

2018156

by Jonathan Davies

FILE	16749_JONATHAN_DAVIES_2018156_179196_2021040104.PDF (774.03K)		
TIME SUBMITTED	24-MAY-2021 11:58PM (UTC+0100)	WORD COUNT	2044
SUBMISSION ID	154686350	CHARACTER COUNT	10495

Course: HNC Electrical and Electronic Engineering

Module: SGEE4004 Programmable Logic Controllers

Assignment: 2. Level and Flow Control System

Name: Jonathan Daffydd Lee Davies

University number: 2018156

Date submitted: 24/05/2021

Word count: 1772

1. Research

The International Electrotechnical Commission (IEC) is a worldwide standard organisation that was founded in 1906 and is recognised around the world for its standards within the controls industry. The standard is well established in Europe with rapid growth in popularity in Asia and North America as the standard programming standard for process control within industry. The IEC is the first vendor independent standardized programming language for industrial automation. (RTAutomation, 2021) Correctly cited

The standard IEC 61131 summarises PLC system requirements, which include the PLC hardware and programming systems. The standard has the inclusion of new programming methods as well as programming already in use. With the standard only a guideline for PLC programming as the enormous number of details defined means that only partial implementation is expected within programming systems. With the constant increase in complexity with PLC systems there is a rise in costs for training of programmers, creating larger programs and the implementation of more complex programming systems. By having a standard, the PLC programming systems costs can be reduced by standardisation, synergy and reusability as PLC programming systems gradually follow the mass software market trend of the PC systems. (Tiegelkamp, 2010) Correctly cited

When creating the programming side of the PLC there are textual languages and graphical languages used these are:

Textual languages

Structured text (ST)

Structured text is known as a high-level language as it does not use low level machine orientated operators but instead offers a large range of statements that describe the complex functionality of the program in a compressed format. (Tiegelkamp, 2010) Correctly cited

Structured text is like BASIC, C, or C11 that uses statements to define execution commands. It is best used for complex mathematical operations or specialized array/table loop processing. (Cox, 2014) Correctly cited

The statements within a ST program are separated by semicolons and can have its statements extended over several lines or several statements can be written on one line. Comments can be added alongside instructions by using brackets and asterisks ("* comment *"). ST is recommended to follow a layout and the comment feature as this allows easier readability although the language is a free format. (Tiegelkamp, 2010) Correctly cited

ST language can be difficult to monitor and trouble shoot without structure and should be left to special applications that would rarely require troubleshooting by the technician. (Cox, 2014) Correctly cited

Below is an example of the ST language commands.

Keyword	Description	Example	Explanation
:=	Assignment	d := 10;	Assignment of a calculated value on the right to the identifier on the left.
	Call of an FB ^a	FBName(Par1:=10, Par2:=20, Par3:=>Res);	Call of another POU of type FB including its parameters. “:=” for input parameters, “=>” for output parameters
RETURN	Return	RETURN;	Leave the current POU and return to the calling POU
IF	Selection	IF d < e THEN f:=1; ELSIF d=e THEN f:=2; ELSE f:= 3; END IF;	Selection of alternatives by means of Boolean expressions.
CASE	Multi-selection	CASE f OF 1: g:=11; 2: g:=12; ELSE g:=FunName(); END CASE;	Selection of a statement block, depending on the value of the expression "f".
FOR	Iteration (1)	FOR h:=1 TO 10 BY 2 DO f[h/2] := h; END FOR;	Multiple loop of a statement block with a start and end condition and an increment value.
WHILE	Iteration (2)	WHILE m > 1 DO n := n / 2; END WHILE;	Multiple loop of a statement block with end condition at the beginning.
REPEAT	Iteration (3)	REPEAT i := i*j; UNTIL i < 10000 END REPEAT;	Multiple loop of a statement block with end condition at the end.
EXIT	End of loop	EXIT;	Premature termination of an iteration statement
;	Dummy statement	;;	

(Tiegelkamp, 2010)

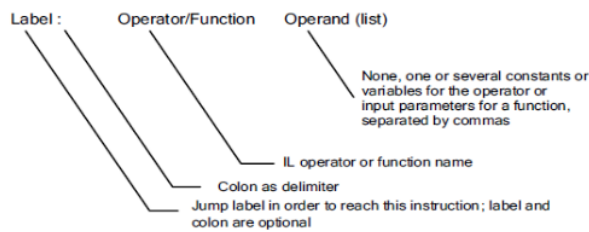
Correctly cited

Instruction list (IL)

The IL language is a programming language based on assembler type language and is known as an intermediate language. The IL follows line-oriented structure with instructions described in one line with empty instructions in the form of blank lines. (Tiegelkamp, 2010)

The IL language is like the ST language however as the name suggests the instructions are listed one under the other.

Below is an example of IL language commands.



(Tiegelkamp, 2010)

Graphical languages

Ladder diagram (LD)

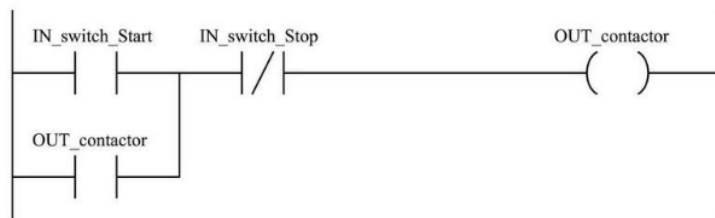
The LD language is graphical language that displays the logical connections between the inputs and outputs as if they were coils and contacts hardwired in a relay circuit. The purpose of the language created was to make the development of PLC programming feel more natural for electricians and technicians who are familiar with relay-based logic and control circuits. (Control Automation, 2021)

Correctly cited

LD language uses networks that are processed in a sequence of top to bottom and from left to right. The power rails from the left are powered by logic state 1 and the power reaches the connected elements. The dependence of the elements logic state or element type will either allow the power to pass through or interrupt the flow like a physical relay circuit. (Tiegelkamp, 2010)

Correctly cited

The image below shows an example of the LD language.



(Control Automation,

2021)

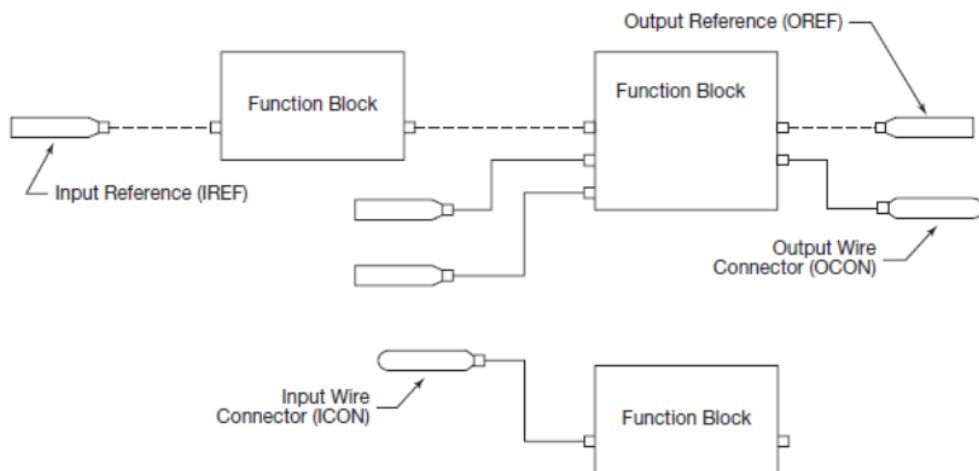
Function block diagram (FBD)

The FBD language originally comes from signal processing industry where the floating-point values and the integer values are important. Over the years the FBD language has become established as a universally useable language within the PLC industry. (Tiegelkamp, 2010)

FBD uses function blocks to make calculations and to make decisions and can be found within process control applications. Process control applications require larger data handling and calculations compared to more discrete machine control applications. (Cox, 2014)

The function blocks work by taking one or more input, depending on the function block type, performs a calculation or decision and then generates one or more output. The output of the function blocks can also be used as an input for another function block. (Cox, 2014)

Below is an example of how a FBD works.

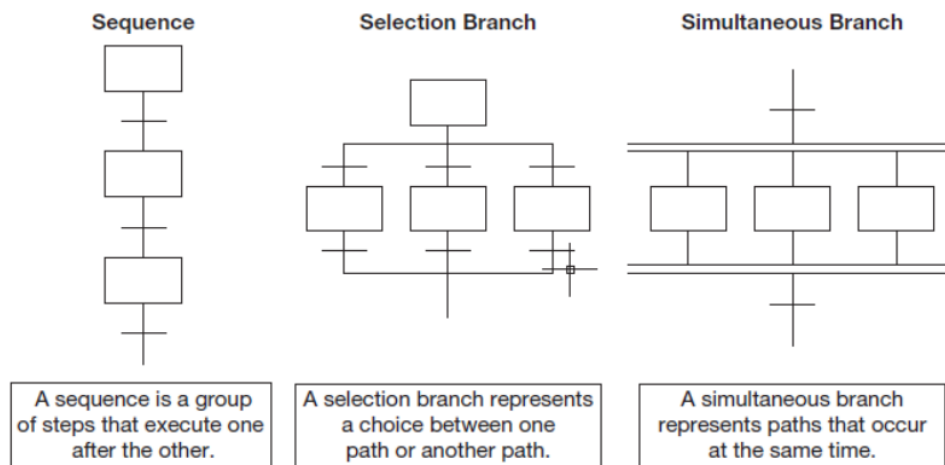


Sequential function chart (SFC)

Another graphical programming language is the SFC. The SFC language is used as a representation tool for scenarios where there are multiple operational states. SFC can break down large and complex processes into smaller and easier to understand information. (Devasia, 2020) [Correctly cited](#)

The SFC is suitable with step by step processes such as chemical processes and washing machines. The graphical language of the SFC helps visualise the dependencies and interdependencies of the required steps much more clearly. (Tiegelkamp, 2010)

Below is an example of SFC structures.



(Tiegelkamp, 2010)

[Correctly cited](#)

The PLC used for the system is the Siemens 1214C with expanded ports in order to support the addition inputs and outputs required in both digital and analogue.

In order to visualise start, stop and reset processes on the system pilot lights show each function and remain illuminated until the next process is activated by pressing normally open pushbuttons that are set to the appropriate sequences. The setpoint of the output is determined with the use of potentiometers. As the varying output voltage is increased or decreased with the turn of the potentiometers this determines the setpoint and allows maximum and minimum voltages to be used as required.

For the tank to fill the valves used are analogue actuator valves and with the signal from the PLC control system and the sensors, the valves are activated or deactivated. Limits will help determine whether the valves require opening or closing. The PID mode setting will release between 0-10V to open or close the valves in ^{pulse?} pulse in order to maintain the filling level or the draining level. When draining a flow meter attached to a paddled wheel will read the speed of the spinning wheel and determine the flow rate within the PLC. The on/off control will allow the valves to half open and close with the supply voltage.

The table below outlines the analogue and digital inputs and outputs.

Input/Output	Tag Address	Tag Name	Data Type	Analogue/Digital
Input	%I0.0	Start	Bool	Digital
Input	%I0.1	Reset	Bool	Digital
Input	%I0.2	Stop	Bool	Digital
Input	%I0.3	Factory IO Running DI	Bool	Digital
Input	%ID30	Level Meter AI Real	Real	Analogue
Input	%ID34	Flow Meter AI Real	Real	Analogue
Output	%Q0.0	Start light	Bool	Digital
Output	%Q0.1	Reset light	Bool	Digital
Output	%Q0.2	Stop light	Bool	Digital
Output	%QD30	Fill Valve AQ Real	Real	Analogue
Output	%QD34	Drain Valve AQ Real	Real	Analogue
Output	%QD38	Setpoint panel meter	DInt	Analogue
Output	%QD42	Process variable panel meter	DInt	Analogue

3. Program Design and Explanation

Two operation screens operate the filling and draining of the tank. These screens are shown in fig 5 and fig 6. The screens are interchangeable between with the PID controls removed on the drain screen as seen in fig 5. The PLC and HMI are connected to each other on the same network but do not have the same IP address if the IP addresses are the same on both HMI and PLC then there is a conflict.

Tag lists can be found from fig 4 to fig 4.4. These tags are used within data blocks from fig 1 to fig 3.7. The function blocks and cyclic interrupts determine and execute the desired functions and commands when operating in each mode.

Manual Mode

In order to fill and empty the tank manually the manual button must be pressed in order to activate. This is verified with a pilot light indicating the correct mode. The manual fill valve can be opened allowing water to fill the tank until the closed button is pressed. The visible graph shows the levels to monitor and close when at the required level. To drain the tank the screen must be changed over to the drain screen, Fig 5, and the manual control pressed the tank can be drained and monitored to the required level.

Semi-automatic mode

When the semi-automatic control level is pressed then there is an option of a selectable on-off and PID control method with a level setpoint. With the on-off control the engineer operating the screen can select a dead-band. When the engineer is using the PID control then the tuning values can be tuned as desired. The filling and draining of the tank is completed with a semi-automatic control of the flow with the engineer able to tune as required.

Automatic

When in automatic mode the tank will automatically fill to the desired fixed level before the flow control begins to run at the designated setpoint rate. When the automatic mode is deactivated or changed the tank will be drained. The HMI screens in fig 5 and fig 6 will display trend graphs in order for the engineer to set the loop tunings.

where is your testing?

5. Meeting the Customer Needs and Team Working

The discussion was held with a small group of 4 with the programs displayed using screen sharing software in Google meet. The group's members were Jonathan Davies, Matthew Perry, Taylor Hoskins and Daniel Oulton. Each member of the group showed the progress of the program while highlighting issues with the software and receiving feedback to make sure that the customer needs have been met as per specification. Issues with the software not operating correctly or issues with the graphs displays were common issues which led to the presenting team members to show the networks and tags within the system and issues, when discovered recommended for change. A further suggestion was made that the auto sequence was created using the ladder logic language with this being easy to fault find, visually easy to follow and components can be added or removed with ease.

6. Conclusions and Recommendations

The design and implementation of the system has been deemed a success with the system fully *no evidence presented* operational when running. Issues have been discovered noted and rectified. From the system design and build its possible to see improvements visually in order to make the interface more accessible. This along with emergency stop being added would be the improvements recommended for the system.

The process has been a strong learning curve in understanding and implementing the customer requirements and demand.

7. Citation List

References

Correct method of listing.

Small Range of Sources

Control Automation, 2021. *LD programming*. [Online]

Available at: <https://control.com/textbook/programmable-logic-controllers/ladder-diagram-ld-programming/>

[Accessed 24 May 2021].

Cox, T. B. & R., 2014. *Technician's Guide to Programmable Controllers*. 6th ed. New York: Delmar Cengage learning.

Devasia, A., 2020. *SFC*. [Online]

Available at: <https://control.com/technical-articles/an-overview-of-sequential-function-chart-sfc-programming/>

[Accessed 24 May 2021].

RTAutomation, 2021. *control IEC*. [Online]

Available at: <https://www.rtautomation.com/technologies/control-iec-61131-3/#:~:text=IEC%2061131%2D3%20is%20the,industry%20by%20over%2050%20countries.&text=IEC%2061131%2D3%20is%20the,industry%20by%20over%2050%20countries.>

[Accessed 06 May 2021].

Tiegelkamp, K.-H. J. ·. M., 2010. *IEC 61131-3: Programming Industrial Automation Systems*. 2nd ed. s.l.:Springer.

There are no titles or comments in your program, no understanding of the program has been demonstrated

8. Appendix

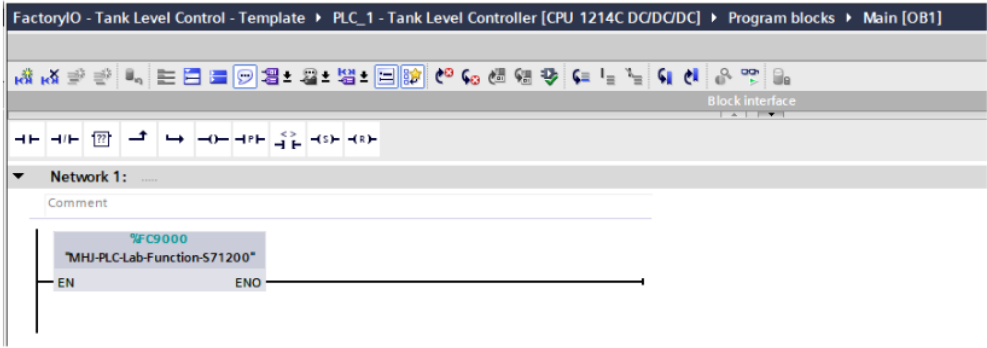


Figure 1

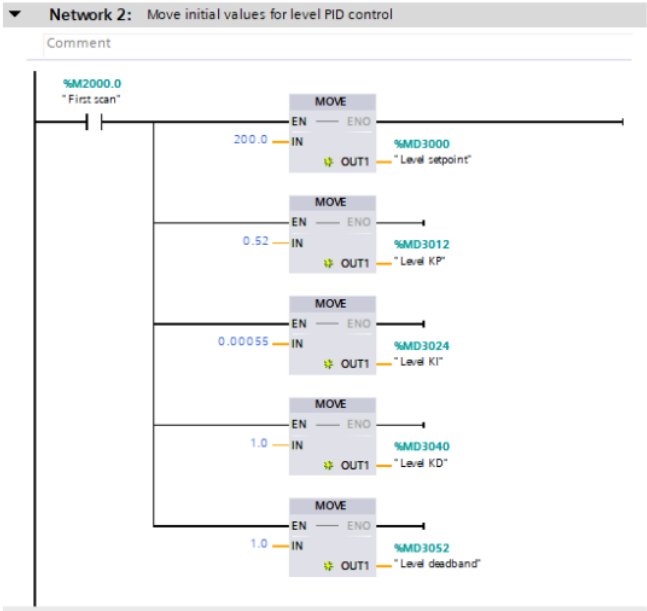


Figure 1. 2

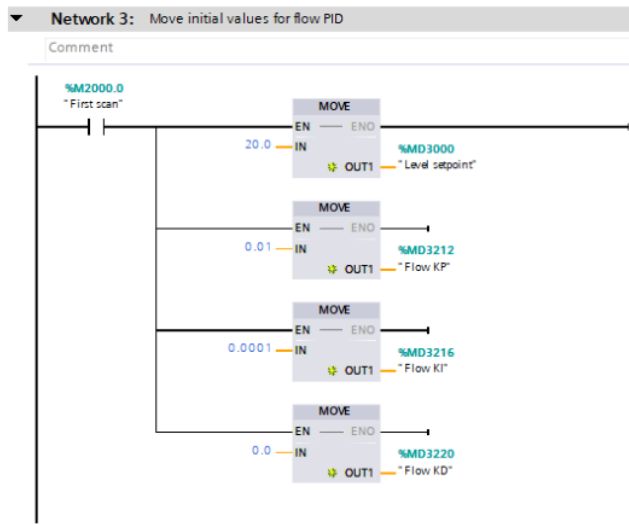


Figure 1.3

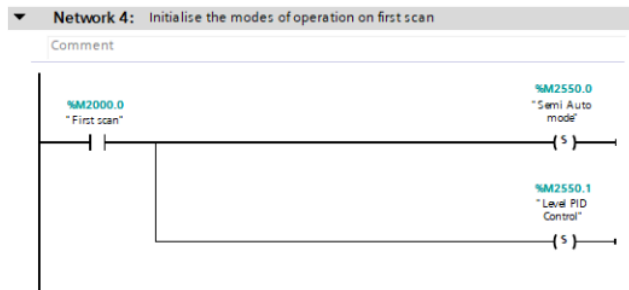


Figure 1.4

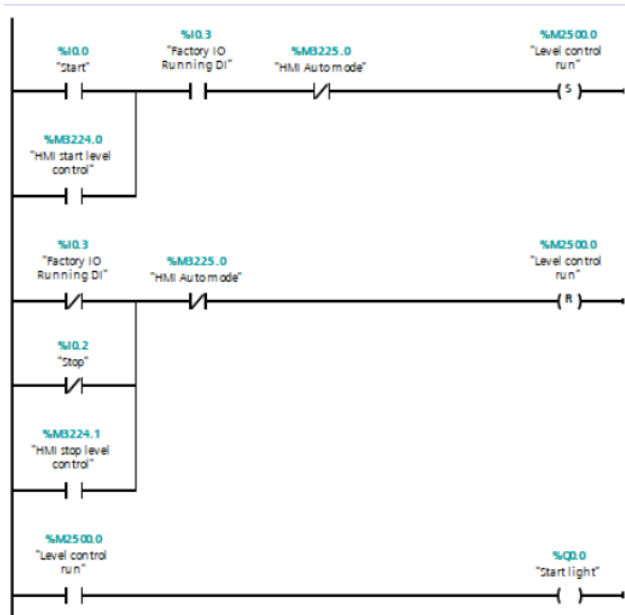


Figure 2

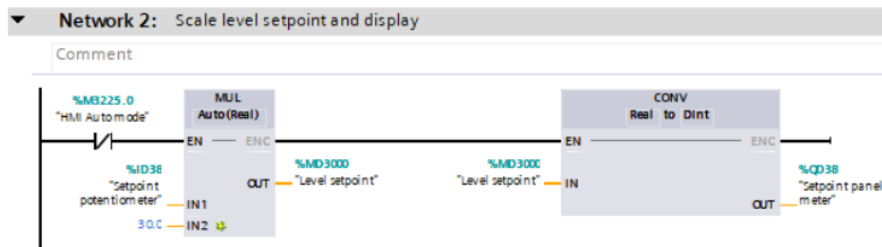


Figure 1.1

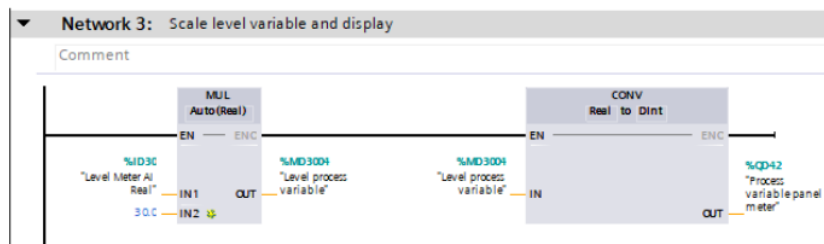


Figure 2.3

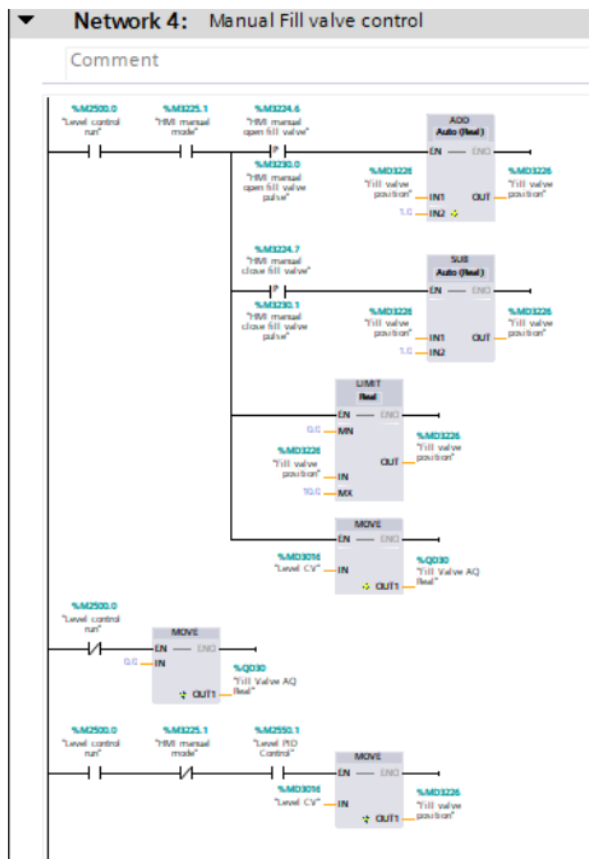


Figure 3.4

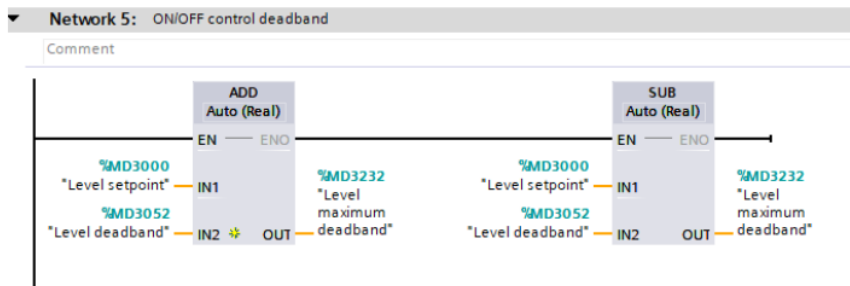


Figure 4.5

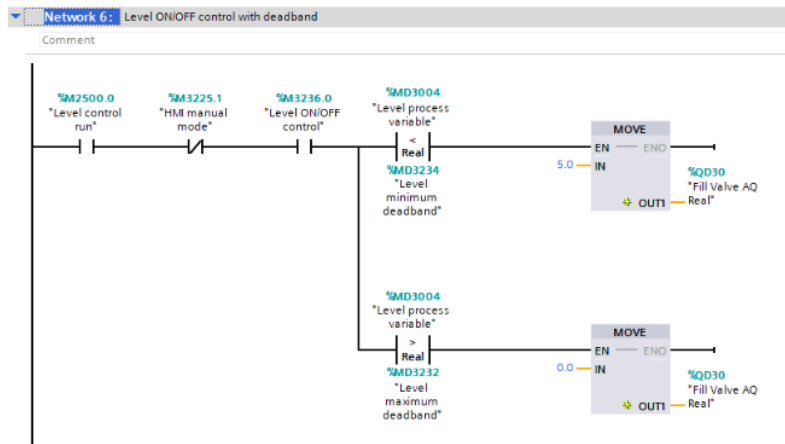


Figure 5.6

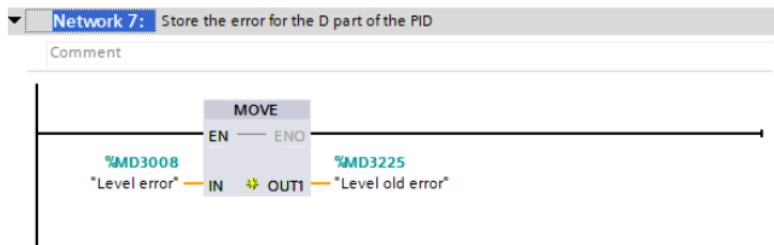


Figure 6.7

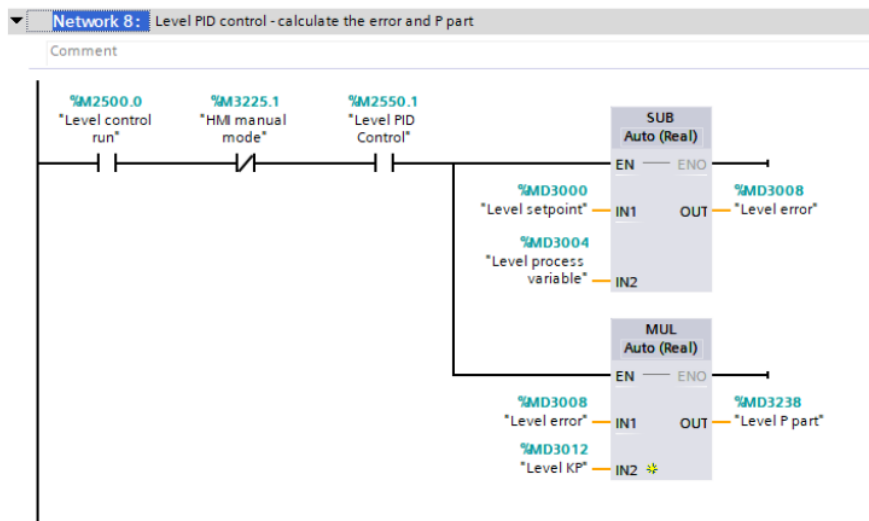


Figure 7.8

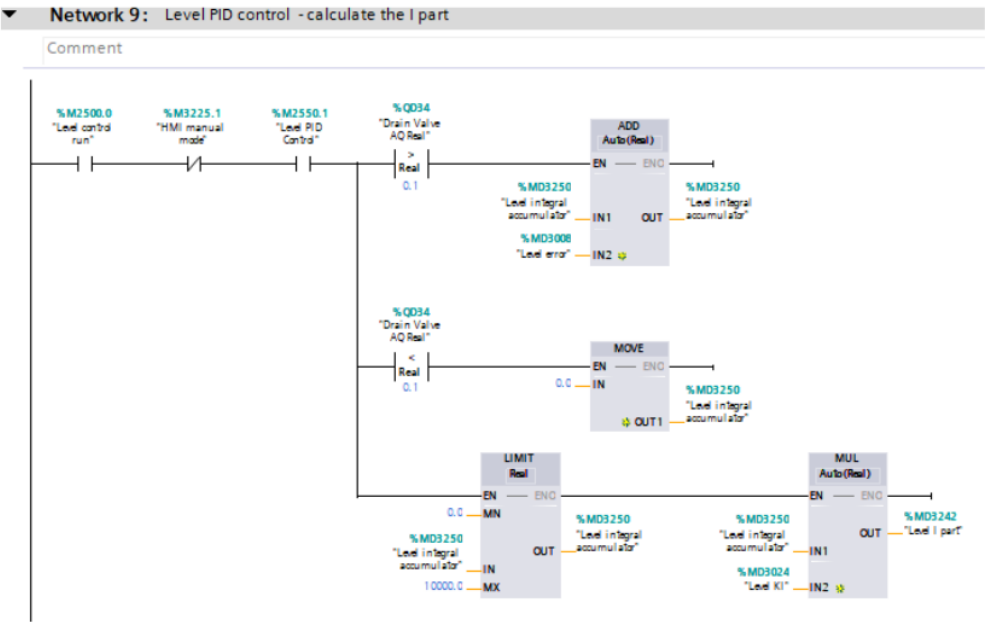


Figure 8.9

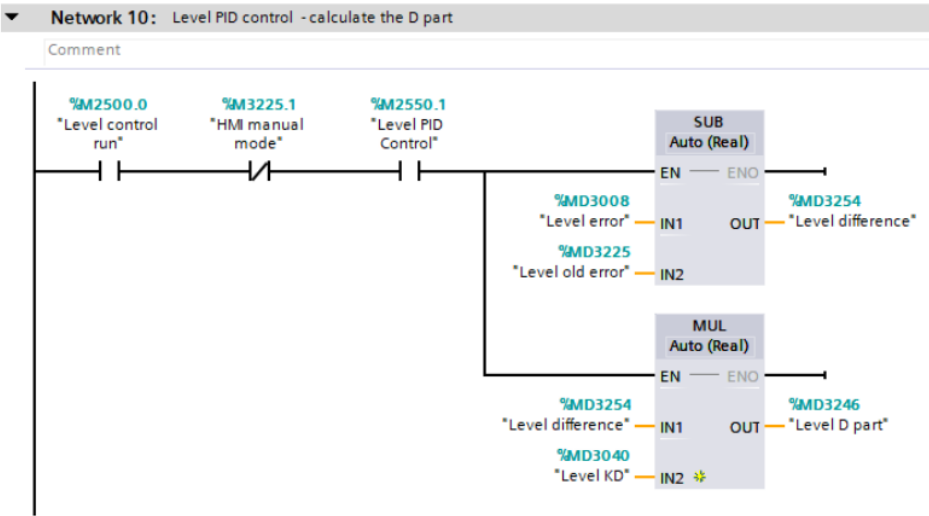


Figure 9.10

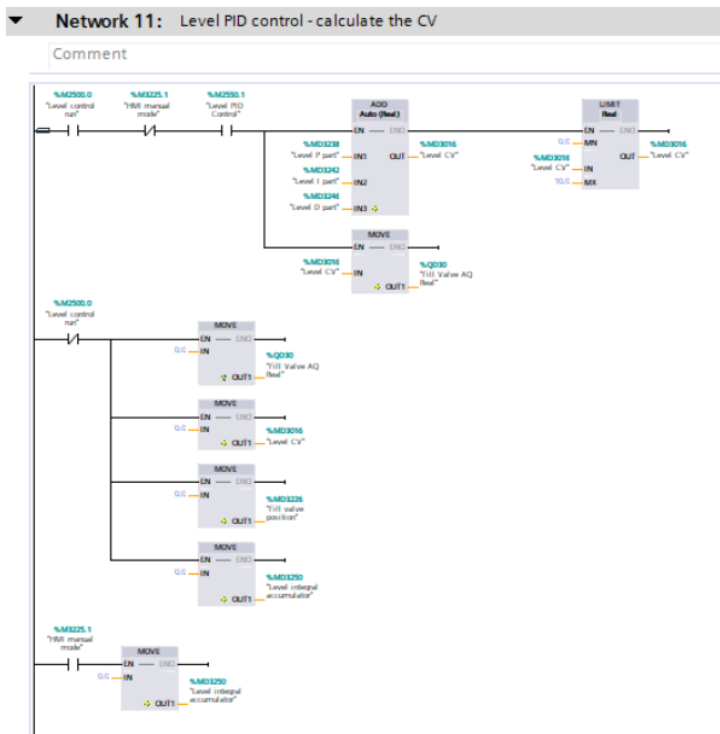


Figure 10.11

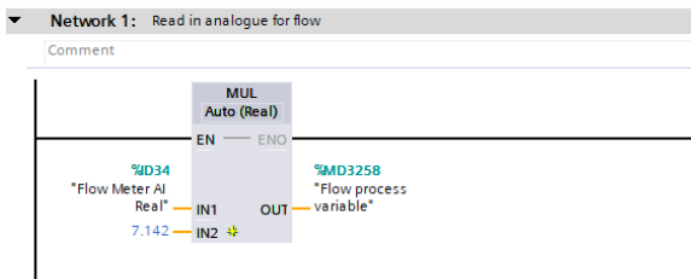


Figure 3

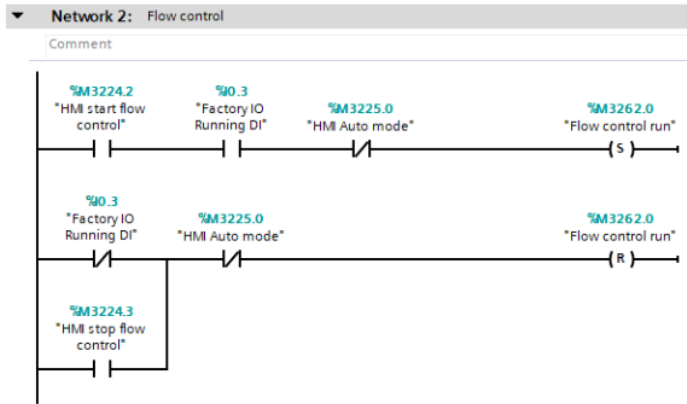


Figure 3.1

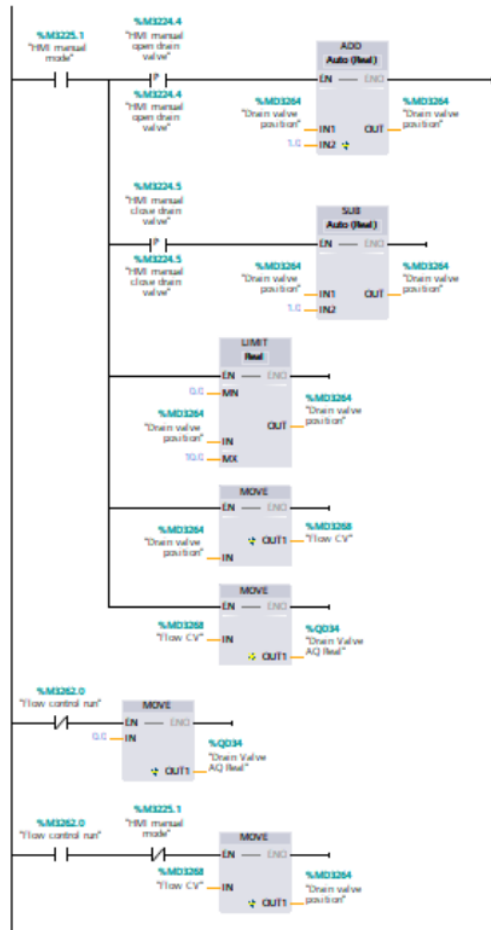


Figure 3.2

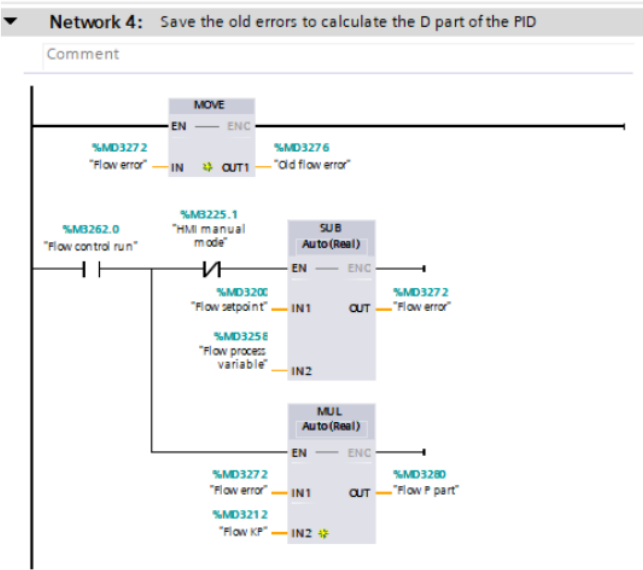


Figure 3.3

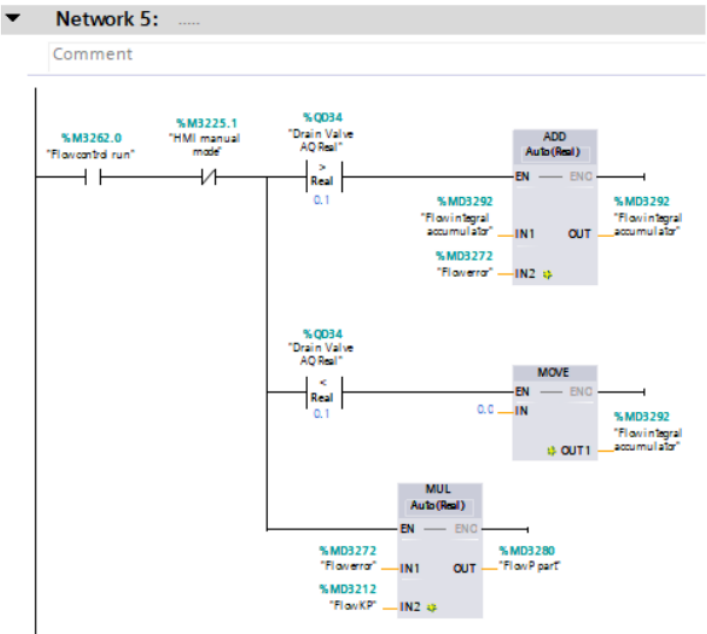


Figure 3.4

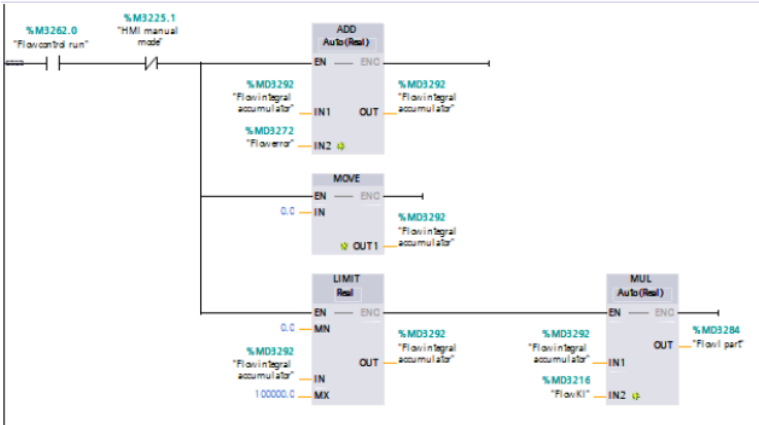


Figure 3.5

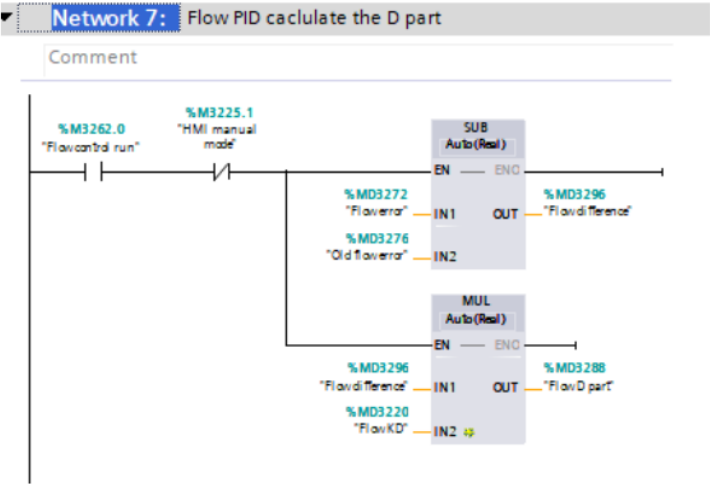


Figure 3.6

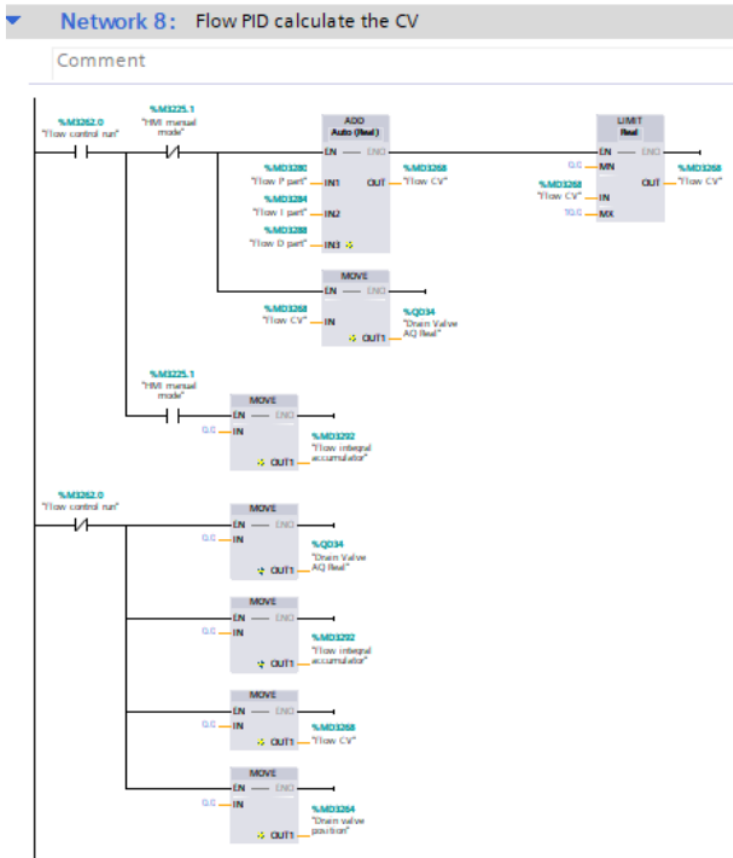


Figure 3.7

PLC tags				
	Name	Tag table	Data type	Address
1	Factory IO Running DI	Factory IO Tag	Bool	%I0.3
2	Level Meter AI Real	Factory IO Tag	Real	%ID30
3	Flow Meter AI Real	Factory IO Tag	Real	%ID34
4	Fill Valve AQ Real	Factory IO Tag	Real	%QD30
5	Drain Valve AQ Real	Factory IO Tag	Real	%QD34
6	Start	Factory IO Tag	Bool	%I0.0
7	Reset	Factory IO Tag	Bool	%I0.1
8	Stop	Factory IO Tag	Bool	%I0.2
9	Setpoint potentiometer	Factory IO Tag	Real	%ID38
10	Start light	Factory IO Tag	Bool	%Q0.0
11	Reset light	Factory IO Tag	Bool	%Q0.1
12	Stop Light	Factory IO Tag	Bool	%Q0.2
13	Setpoint panel meter	Factory IO Tag	DInt	%QD38
14	Process variable panel meter	Factory IO Tag	DInt	%QD42

Figure 4

15	System_byte	System tags	Byte	%MB2000
16	First scan	System tags	Bool	%M2000.0
17	Diagstatus update	System tags	Bool	%M2000.1
18	Always true	System tags	Bool	%M2000.2
19	Always false	System tags	Bool	%M2000.3
20	Clock_byte	System tags	Byte	%MB2001
21	Clock_10Hz	System tags	Bool	%M2001.0
22	Clock_5Hz	System tags	Bool	%M2001.1
23	Clock_2.5Hz	System tags	Bool	%M2001.2
24	Clock_2Hz	System tags	Bool	%M2001.3
25	Clock_1.25Hz	System tags	Bool	%M2001.4
26	Clock_1Hz	System tags	Bool	%M2001.5
27	Clock_0.625	System tags	Bool	%M2001.6
28	Clock_0.5	System tags	Bool	%M2001.7

Figure 4.1

29		Level setpoint	My Tags	Real	%MD3000
30		Level process variable	My Tags	Real	%MD3004
31		Level control run	My Tags	Bool	%M2500.0
32		Level propotional control	My Tags	Bool	%M2500.1
33		Level error	My Tags	Real	%MD3008
34		Level KP	My Tags	Real	%MD3012
35		Level CV	My Tags	Real	%MD3016
36		Level KI	My Tags	Real	%MD3024
37		Level KD	My Tags	Real	%MD3040
38		Level deadband	My Tags	Real	%MD3052
39		Flow KP	My Tags	Real	%MD3212
40		Flow setpoint	My Tags	Real	%MD3200
41		Flow KI	My Tags	Real	%MD3216
42		Flow KD	My Tags	Real	%MD3220
43		Semi Auto mode	Factory IO Tag	Bool	%M2550.0
44		Level PID Control	Factory IO Tag	Bool	%M2550.1
45		HMI start level control	My Tags	Bool	%M3224.0
46		HMI stop level control	My Tags	Bool	%M3224.1
47		HMI start flow control	My Tags	Bool	%M3224.2
48		HMI stop flow control	My Tags	Bool	%M3224.3
49		HMI manual open drain valve	My Tags	Bool	%M3224.4
50		HMI manual close drain valve	My Tags	Bool	%M3224.5
51		HMI manual open fill valve	My Tags	Bool	%M3224.6
52		HMI manual close fill valve	My Tags	Bool	%M3224.7

Figure 4.2

53		HMI Auto mode	My Tags	Bool	%M3225.0
54		HMI manual mode	My Tags	Bool	%M3225.1
55		HMI semi auto mode	My Tags	Bool	%M3225.2
56		Fill valve position	My Tags	Real	%MD3226
57		HMI manual open fill valve pulse	My Tags	Bool	%M3230.0
58		HMI manual close fill valve pulse	My Tags	Bool	%M3230.1
59		Level maximum deadband	My Tags	Real	%MD3232
60		Level ON/OFF control	My Tags	Bool	%M3236.0
61		Level minimum deadband	My Tags	Real	%MD3234
62		Level old error	My Tags	Real	%MD3225
63		Level P part	My Tags	Real	%MD3238
64		Level I part	My Tags	Real	%MD3242
65		Level D part	My Tags	Real	%MD3246
66		Level integral accumulator	My Tags	Real	%MD3250
67		Level difference	My Tags	Real	%MD3254
68		Flow process variable	My Tags	Real	%MD3258
69		Flow control run	My Tags	Bool	%M3262.0
70		Drain valve position	My Tags	Real	%MD3264
71		Flow CV	My Tags	Real	%MD3268
72		Flow error	My Tags	Real	%MD3272
73		Old flow error	My Tags	Real	%MD3276
74		Flow P part	My Tags	Real	%MD3280
75		Flow I part	My Tags	Real	%MD3284
76		Flow D part	My Tags	Real	%MD3288

Figure 4.3

77		Flow integral accumulator	My Tags	Real	%MD3292
78		Flow difference	My Tags	Real	%MD3296

Figure 4.4

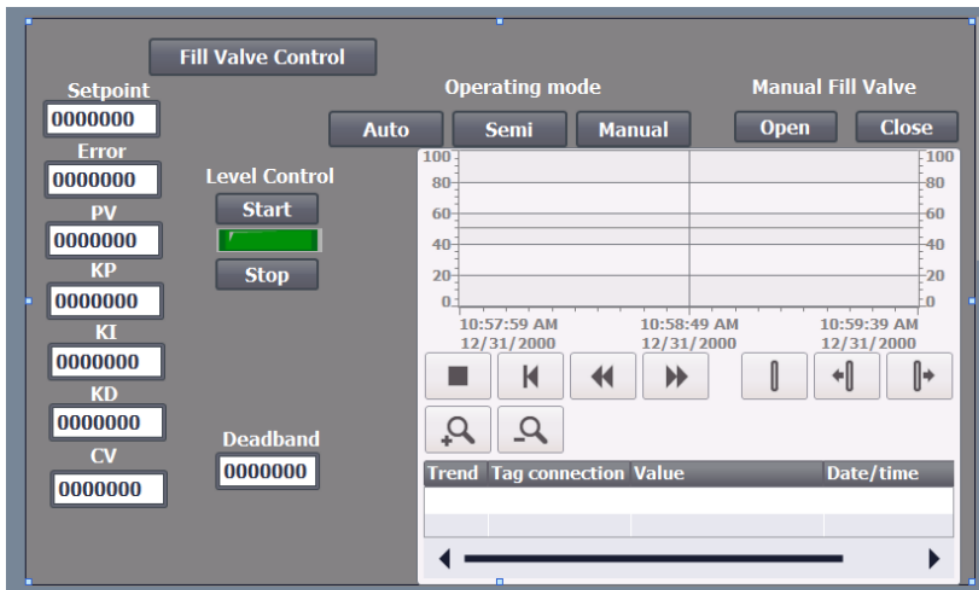


Figure 5

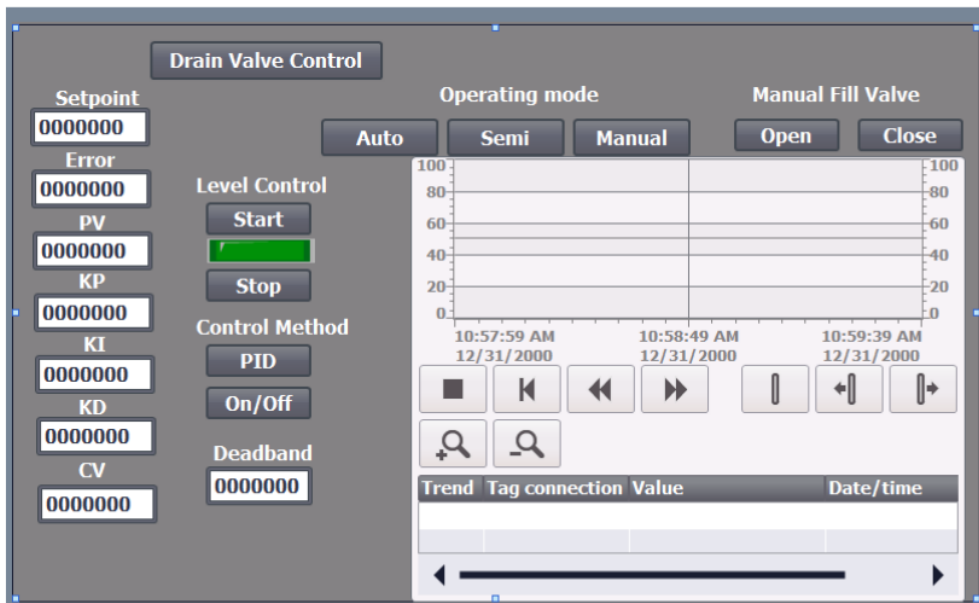


Figure 6

FINAL GRADE

GENERAL COMMENTS

Instructor

40/100

PAGE 1

PAGE 2

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 3

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 4

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 5

PAGE 6

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

Text Comment. new section, new page

PAGE 7

Text Comment. pulse?

PAGE 8

Text Comment. why are they removed?

QM

Awk.

Awkward:

The expression or construction is cumbersome or difficult to read. Consider rewriting.

Text Comment. new section, new page

PAGE 9

Text Comment. where is your testing?

PAGE 10

PAGE 11

Text Comment. no evidence presented



Small Range of Sources

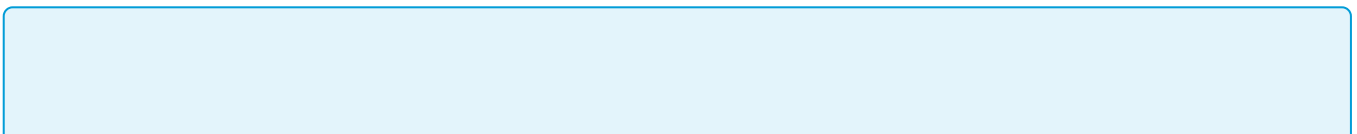
You have shown little evidence of reading, only a few sources listed here



Correct method of listing.

You have listed your sources correctly according to the Harvard method.

Text Comment. There are no titles or comments in your program, no understanding of the program has been demonstrated



1. ENGAGEMENT WITH LITERATURE



Good evidence or reading, sources cited and listed correctly but range is limited. (10/20)

10

2. KNOWLEDGE AND UNDERSTANDING



There is some evidence of understanding. in the initial descriptions and IO listing. However, the program printout contains no comments or even network titles. (9/20)

9

3. COGNITIVE AND INTELLECTUAL



There is little evidence of analysis in the conclusions but no evidence of testing even though there is a claim that the system worked. (6/20)

6

4. PRACTICAL APPLICATION



The program is well written but has no comments. There are no electrical drawings and no evidence of testing. There is some attempt at describing a discussion of user needs but no evidence to support it. (6/20)

6

5. TRANSFERABLE SKILLS



The report is well structured and there is a good research section. Appendix contains program code and some HMI screens. There is evidence of team working. (9/20)

9